Pacific Northwest
NATIONAL LABORATORY

PNNL-31807

Harnessing Condorcet Methods to Improve Decision-making Based on Ranked Data

August 2021

Elias A. Parzen
Christian D. Johnson

U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

**Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov**

**Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <https://www.ntis.gov/about>
Online ordering: http://www.ntis.gov**

PNNL-31807

Harnessing Condorcet Methods to Improve Decision-making Based on Ranked Data

August 2021

Elias A. Parzen
Christian D. Johnson

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

Decision-makers must often choose between multiple alternatives based on their relative merits across a variety of criteria. These multiple-criteria decision-making (MCDM) problems must be approached in an objective, measurable, and transparent fashion to obtain well-supported outcomes. The challenge of using sets of ranked data to identify the optimal choice is not unlike the challenge of using ranked ballots to identify the winner of an election in a preference voting system. As such, the methods of rank aggregation used in elections can be applied to help resolve MCDM problems. Previous research into the design of elections and ballots has yielded many algorithms with well-understood properties. A subset of these, called Condorcet methods, reliably identify the "Condorcet winner," if it exists, giving the result that would defeat any other in a pairwise comparison. That, in addition to several other desirable qualities, makes Condorcet methods like the Schulze method, Ranked Pairs, and Copeland's method effective and scalable means for tackling MCDM problems. These rank aggregation methods have been implemented in JavaScript functions for incorporation into a web-based MCDM decision tool. These algorithms can be applied in a wide variety of contexts (for example, determining the "best" environmental remediation method or selecting a subcontractor) to facilitate effective decision-making. The performance of the JavaScript Condorcet prototype tool was compared across the implemented algorithms and with both a non-Condorcet rank aggregation method and an implementation of the Simple Multi-Attribute Rating Technique (SMART) algorithm.

Acknowledgments

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships (SULI) Program.

This research would not have been possible without the guidance of my mentor, Christian Johnson, who developed the original idea for the project. His knowledge and support were indispensable.

Sadie Montgomery graciously provided her time and expertise with the SMART algorithm and its use in the Decision Information Engagement Tool (DIET) decision support tool.

Acronyms and Abbreviations

CERCLA	Comprehensive Environmental Response, Compensation, and Liability Act
DOE	U.S. Department of Energy
DIET	Decision Information Engagement Tool
MCDM	Multiple-criteria decision-making
MNA	Monitored natural attenuation
SMART	Simple multi-attribute rating technique
SULI	Science Undergraduate Laboratory Internships
TCE	Trichloroethene
UI	User interface
WDTS	Workforce Development for Teachers and Scientists

Contents

Figures

Tables

1.0 Introduction

Multiple-criteria decision-making (MCDM) problems are commonly faced by almost everyone, not just by policymakers, scientists, or politicians. The hallmark of a MCDM problem is having to compare alternatives across many criteria. For example, choosing an environmental remediation method for a particular site based on cost, long and short-term effectiveness, time to complete, and compliance with regulation and standards. While it is possible that one remediation technique will dominate its competitors in each of those criteria, making the best option obvious, it's more likely that different alternatives will excel in different ways. In that case, it may be difficult to identify the optimal choice. MCDM problems are prevalent and span a wide gamut of domains, for example: ranking athletes [Parker, 2018], selecting a subcontractor to hire [Biruk and Jaśkowski, 2016], or choosing a vendor [Sahida et al., 2019].

Approaches for determining the best or overall ranking of alternatives are well studied and include multiple rank aggregation methods. Using rank aggregation to determine an optimal alternative has similarities with assessing ranked voting systems to determine a winning candidate. In each case, an algorithm must use the information contained within many initial rankings to identify the best option. Research into the design and analysis of ranked voting systems, referred to as social choice theory, dates back centuries, with major contributions to the field made by Ramon Llull (1232-1315), Jean-Charles de Borda (1733-1799), and the Marquis de Condorcet (1743-1794). This mature field has yielded several algorithms that possess many desirable characteristics from the standpoint of decision science. However, repurposing ranked voting systems for use with MCDM problems brings along their weaknesses along with their strengths. As articulated by Arrow in his Impossibility Theorem [Arrow, 1950], and further elaborated by Gibbard [1973] and Satterthwaite [1975], it is not possible for an ordinal voting system to meaningfully represent the preferences of a broad set of participants while simultaneously allowing any number of candidates and being resistant to voters casting their ballots in a manipulative way.

Because no single ranked voting algorithm can be optimal in every way, it is important to have multiple methods to evaluate and understand which traits are most important for a particular context. Previous research into adapting these ranked voting systems to address MCDM problems has indicated promising results from the family of algorithms known as Condorcet methods [Tomczak et al., 2019; Sahida et al., 2019; Biruk and Jaśkowski, 2016]. This report explores three Condorcet algorithms, why they were chosen for implementation into a decision support tool, how they work, and how they are expected to perform relative to each other, as well as relative to two non-Condorcet approaches for MCDM problems.

2.0 Background

This section describes the Condorcet methods and two non-Condorcet methods, providing algorithm background information and calculation approaches.

2.1 Condorcet Algorithm Descriptions

For Condorcet methods, the information from each initial ranking is first used to create pairwise comparisons between each unique pair of alternatives, recording how many ballots ranked one alternative higher than the other. If a Condorcet winner exists, it can be identified at this stage, without any additional computation.

2.1.1 Copeland's Method

The Copeland's method algorithm [Saari and Merlin, 1996] takes the results of the pairwise comparisons, and uses them to generate a results matrix, r. The elements of the results matrix, r_{ij}, can take one of three different values based on the outcome of the pairwise comparison between alternatives i and j:

$r_{ij} = 1$, if *i* defeated *j*
$r_{ij} = ½$ if *i* tied *j*
$r_{ij} = 0$ if *j* defeated *i*

A Copeland score, C_i, is then calculated for each alternative *i* by taking the summation of r_{ij} over *j* (Equation 1).

$$C_i = \sum_j r_{ij} \qquad (1)$$

Should an alternative have a Copeland score of n – 1, then that alternative is the Condorcet winner. If no such winner exists, ranking the alternatives in descending order by Copeland score identifies the first ranked alternative as the winner.

2.1.2 Ranked Pairs

The goal of the Ranked Pairs algorithm [Tideman, 1987] is to represent the results of the pairwise comparisons as an acyclic, directed graph with nodes representing the alternatives and edges representing the pairwise victories. An example graph is shown in Figure 1.

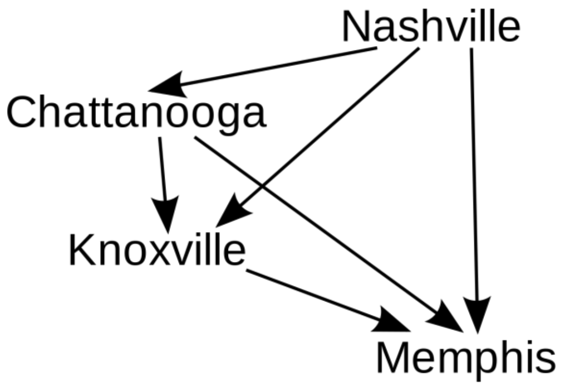


Figure 1. This example directed graph shows a source node, Nashville, that defeats all other alternatives in pairwise comparisons.

The first step in the Ranked Pairs algorithm is to find the "majorities" from the pairwise results. That is, results in which the first alternative defeated the second are identified. These majorities are then sorted by the number of initial rankings preferring the victor. The majorities are then "locked" in the order that they are ranked. To lock in a majority means that the directed graph must reflect the result of the majority. For example, if the majority of alternative A over alternative B is locked in, the graph must contain an edge from node A to node B, and not include an edge from B to A. Once the first two majorities are locked in, no additional majorities can be locked in without first checking to see if they would create a Condorcet cycle. A cycle exists if transitivity of the locked-in results is broken. For example, if the majorities A over B, B over C, and C over A are all locked in, this would imply that A is superior to B, which is superior to C, but that C is superior to A. The existence of such a Condorcet cycle would render the results of the system ambiguous. The Ranked Pairs algorithm avoids this outcome by not locking in any majority that would break transitivity.

The result of checking and locking in majorities is an acyclic directed graph. The winner, or winners in the event of a tie, can be readily identified because their respective nodes on the graph will be source nodes, having only edges leading away from, but not to, themselves. The Ranked Pairs algorithm is not designed to produce a full ranking. However, by repeatedly running the algorithm and removing the previous winner each time will yield an approximate ranking.

2.1.3 Schulze Method

Like the Ranked Pairs algorithm, the Schulze method [Schulze, 2021] seeks to represent the pairwise comparisons of the alternatives in a directed graph. Unlike Ranked Pairs, no attempt is made to keep the graph acyclic in the Schulze method. Instead, each pairwise result is reflected by an edge between the two relevant nodes with a weighting, d, equal to the number of initial rankings preferring the source node over the destination node (Equation 2).

$$d_{V,W} = \text{the number of ballots or rankings placing alternative V above alternative W} \qquad (2)$$

Once these weightings have been established, the strongest path between each pair of alternatives is determined. A path from alternative V to alternative W is a sequence of alternatives, C(1), C(2), …. , C(n), where the following properties hold:

1. C(1) = V and C(n) = W
2. $\forall$ i = 1, 2, …, (n – 1) : $d_{[Ci,\ Ci+1]} > d_{[Ci+1,\ Ci]}$

The strength of a single path from alternative V to alternative W is the minimum value of d along that path. The strength of the strongest path from V to W is denoted as p[V, W].

The Schulze method judges alternative V to be superior to alternative W if p[V, W] > p[W, V]. These Schulze relations are inherently transitive [Schulze, 2011], so Condorcet cycles are not a concern. For an alternative X out of the set of alternatives, S, to be chosen as the winner by the Schulze method, alternative X must have the following property (Equation 3):

$$\forall\ X, Y \in S\text{: } p[X, Y] > p[Y, X] \qquad (3)$$

The Schulze method does not return a full ranking of alternatives by default, but instead selects a single winner, or a set of tied winners. An approximation of a full ranking may be obtained by repeatedly running the algorithm, removing the previous winner each time.

2.2 Other Algorithms

Two additional approaches to resolving MCDM problems were chosen as points of comparison for the Condorcet methods.

2.2.1 Borda's Method

Borda's method [Emerson, 2013] is another ranked voting system, albeit one that does not meet the Condorcet or majority criteria. This algorithm has been used to address MCDM rank aggregation problems in the past. Given the shared lineage with the Condorcet methods, Borda's method provides a natural point of comparison.

For each alternative A in a ranking, r, let B_{Ar} = the number of alternatives that A outranks. The Borda Count, BC, for alternative A is then calculated as the sum of B_{Ar} , as shown in Equation 4):

$$BC[A] = \sum_{r} B_{Ar} \qquad (4)$$

Borda's method selects the alternative with the highest Borda Count as the winner. A full ranking list is obtained by sorting the alternatives by Borda Count.

2.2.2 SMART Algorithm

The SMART algorithm [Patel et al., 2017] is another approach to MCDM problems that is grounded in multi-attribute utility theory, rather than social choice theory, making it an interesting complement to Borda’s method for comparisons to the Condorcet methods. The SMART algorithm functions by assigning user-selected weights to the criteria. Weights can be assigned to groups of related criteria, as well as to each individual criterion. Each alternative’s performance in each criterion is then converted to a standardized utility score. The SMART algorithm then calculates a weighted sum of these utility scores for each alternative (Equation 5).

$$U_j = \sum_k w_k u_{jk} \qquad (5)$$

Where U_j represents the final rating of alternative j, w_k is the normalized weighting of criterion k, and u_{jk} is the utility score of alternative j for criterion k.

The winner is the alternative with the maximum weighted sum. The list of alternatives sorted by their weighted sums forms a full ranking.

3.0 Approach

To create the decision support tool prototype, it was necessary to select algorithms to focus on, implement them in JavaScript, and identify other approaches for comparing the outputs.

3.1 Algorithm Selection

In the context of a decision support tool, because the “voters” are merely tables of ranked data, they cannot “vote” in ways that don’t reflect their true preferences for the purpose of sabotaging one alternative or helping another. Thus, susceptibility to manipulation was not taken into account when selecting algorithms. Freed from that requirement, prioritization of algorithms for implementation was based on their performance across four primary criteria.

- The first selection criterion was the Condorcet criterion. To meet this requirement, the algorithm must always identify and select as the winner an alternative that outranks each of its competitors on a majority of the initial rankings.
- The second selection criterion was the majority criterion. The majority criterion is satisfied if the algorithm always selects an alternative that is given the highest ranking by a majority of the ballots as the winner, should such an alternative exist.
- The third selection criterion was monotonicity. The monotonicity criterion requires that ranking an alternative higher cannot harm its chance of winning and that ranking an alternative lower cannot improve its chance of winning.
- The final selection criterion was that the algorithm run in polynomial time, which facilitates scalability for problems with more dimensions of ranked data.

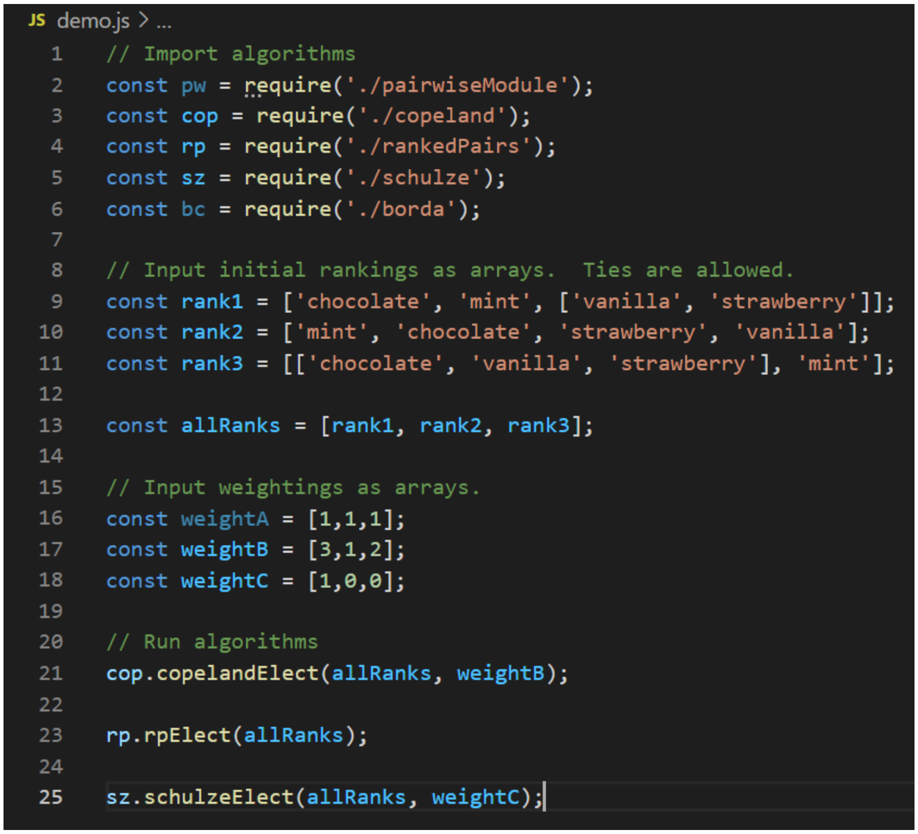
The goal of the first two criteria is to ensure that algorithms selected for implementation would reliably choose a clearly optimal alternative, if it exists. Monotonicity is deemed important because it guarantees an intuitive and meaningful relationship between the ranking of an alternative in each initial ranking and the final outcome. Polynomial time algorithms are preferred for the purpose of scalability to larger problems. Guided by these criteria, three Condorcet methods were prioritized for implementation: Copeland’s Method, Ranked Pairs, and the Schulze Method.

3.2 Algorithm Implementation and Application

The three selected rank aggregation algorithms, as well as Borda’s method were implemented in JavaScript for the purpose of inclusion in a web-based decision tool. Visual Studio Code was used as the text editor and IDE to code the algorithms and run the code in a Node.js environment. Because each algorithm relies on pairwise comparisons and extracting the names of the alternatives from the rankings, these functions were placed in a common utility file called pairwiseModule.js.

Data input for the algorithms is specified in the form of arrays, with each array representing one criterion ranking. Each ranking must contain every alternative, although ties are allowed. For each criterion ranking array, the alternatives are listed in order from best alternative first to the worst alternative last. In the event of a tie ranking, an array containing the tied alternatives is used. The ordering of the alternatives within the inner tie array does not matter. Once the rankings have been entered, the user creates a new array containing each of the individual criterion rankings. Weightings are also entered as an array. The first element in the array should be the weighting of the first criterion, the second element should be the weighting of the second criterion, and so on.

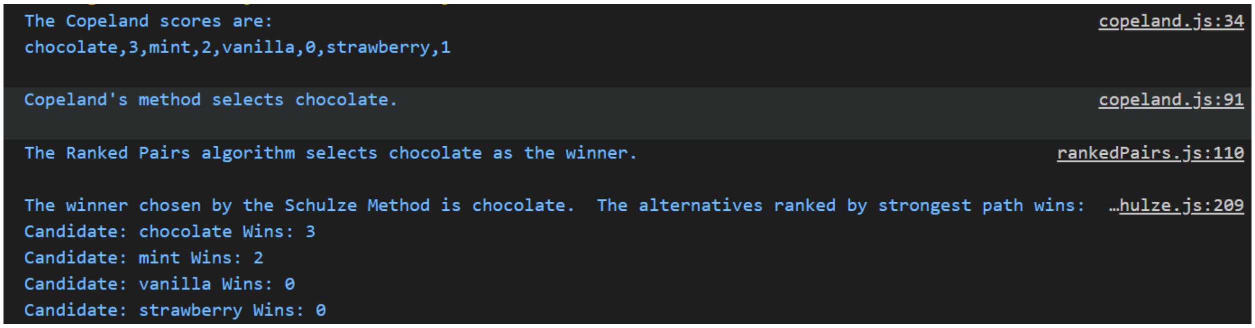
To run the tool, the user first imports the desired algorithms. Copeland's method and Ranked Pairs are dependent on pairwiseModule.js, while the Schulze method and Borda's method are not. The user then calls the function for the method of their choice, while passing in the array of rankings and the array of weights, in that order. If the array of weights is not provided, the algorithms will use a default weighting of 1 for each ranking. The function names are copelandElect(rankingArray, weightArray), rpElect(rankingArray, weightArray), and schulzeElect(rankingArray, weightArray). The imports, arrays, and function calls can be seen in Figure 2.



```
// Import algorithms
const pw = require('./pairwiseModule');
const cop = require('./copeland');
const rp = require('./rankedPairs');
const sz = require('./schulze');
const bc = require('./borda');

// Input initial rankings as arrays.  Ties are allowed.
const rank1 = ['chocolate', 'mint', ['vanilla', 'strawberry']];
const rank2 = ['mint', 'chocolate', 'strawberry', 'vanilla'];
const rank3 = [['chocolate', 'vanilla', 'strawberry'], 'mint'];

const allRanks = [rank1, rank2, rank3];

// Input weightings as arrays.
const weightA = [1,1,1];
const weightB = [3,1,2];
const weightC = [1,0,0];

// Run algorithms
cop.copelandElect(allRanks, weightB);

rp.rpElect(allRanks);

sz.schulzeElect(allRanks, weightC);
```

Figure 2. Code listing to demonstrate running the Condorcet method algorithms. The import statements appear at the top of the image, followed by the rankings, weights, and function calls.

The algorithm output is displayed in the console. Copeland's method returns a ranking based on Copeland scores, as well as the winner. Ranked Pairs returns only the winner. The Schulze method returns the winner, as well as the number of strongest-path victories each alternative earned. The output from the code above is shown in Figure 3.



```
The Copeland scores are:                                                                         copeland.js:34
chocolate,3,mint,2,vanilla,0,strawberry,1

Copeland's method selects chocolate.                                                             copeland.js:91

The Ranked Pairs algorithm selects chocolate as the winner.                                   rankedPairs.js:110

The winner chosen by the Schulze Method is chocolate.  The alternatives ranked by strongest path wins:  …hulze.js:209
Candidate: chocolate Wins: 3
Candidate: mint Wins: 2
Candidate: vanilla Wins: 0
Candidate: strawberry Wins: 0
```

Figure 3. The output from each of the three Condorcet methods.

3.3 Algorithm Testing and Comparison

To validate the output of the three algorithms of the Condorcet method decision tool and judge its performance, results were compared against two other approaches to MCDM problems: Borda count and the Simple Multi-Attribute Rating Technique (SMART) algorithm. The SMART algorithm is implemented in the Decision Information Engagement Tool (DIET) tool [Tate and Montgomery, 2021] developed at the Pacific Northwest National Laboratory. DIET is a fully developed decision support tool with an aesthetically pleasing user interface (UI) that intuitively guides the use through data input, weighting, and output interpretation with detailed instructions and visualizations.

4.0 Results and Discussion

The three Condorcet methods, as well as Borda's method, were successfully implemented in JavaScript. Unit testing confirmed that the algorithms functioned as intended with varying numbers of alternatives and criteria, and that the user-specified weightings were correctly influencing the pairwise results.

Two datasets representing real-world MCDM problems were acquired for analysis by the decision tool. Both datasets used ratings, rather than rankings, which had to be converted to be compatible with the tool. Conversion was carried out by assigning all alternatives with the highest rating to rank 1, the alternatives with the second highest rating to rank 2, and so on. This process necessarily lost some information contained by the ratings, because ratings that were not used for any alternative received cannot be reflected in the rankings. For example, if one alternative received the best rating, another the second-best rating, and a third the tenth-best rating, their converted rankings would be 1, 2, and 3. This conversion would eliminate the information that two of the alternatives were of a similar quality, while the last was far inferior.

4.1 Selecting Data Portal Software

The first example dataset pertains to a study that evaluated potential data portal software for use at the U.S. Department of Energy Hanford Site to catalog the wide variety of site environmental data. Specific criteria were developed regarding data portal software functionality and integration into the computing/network environment at Hanford. The evaluators assigned ratings to six different potential open data portal software packages across nine different criteria categories (Table 1). The ratings were converted to rankings in the manner described above. For example, for the criteria category "Retrieve," ArcGIS and OpendataSoft were assigned rank 1, CKAN, DKAN, and Socrata were assigned rank 2, and Junar was assigned rank 3.

Table 1. Data portal software rankings

Software Criteria Category Description	ArcGIS	CKAN	DKAN	Junar	OpendataSoft	Socrata
Catalog	5	5	5	2	5	5
Find	5	4	4	4	3	4
Retrieve	5	4	4	3	5	4
Large Dataset Storage	5	5	5	1	2	5
Hanford Site Software Approval	5	4	4	3	3	4
Single Sign On	5	5	5	1	5	5
Role-based Access	5	5	5	2	3	5
Linking/Federation	4	4	4	4	4	4
Deployment	5	3	3	1	1	3

This data portal software dataset, when considered with equal criteria weightings, contains a Condorcet and majority winner, ArcGIS. Having a known optimal candidate provided an opportunity to validate the output of the Condorcet methods, which, by design, should all converge on ArcGIS as the winner. This was indeed the case for all three algorithms, as well as Borda's method and DIET.

4.2 Determining the Best Remedial Alternative

The second dataset pertains a site where release of trichloroethene (TCE) from deteriorating drums in a landfill resulted in a groundwater contaminant plume (Figure 4). This TCE plume migrated downgradient in the upper aquifer at the site, then through a hole in the underlying confining layer into the lower aquifer. Three environmental remediation alternatives were proposed to address the TCE groundwater plume and protect offsite receptors (wells) that are screened in the lower aquifer. One option was to apply pump-and-treat remediation in the upper aquifer to cut off the groundwater plume prior to entering the lower aquifer, which was referred to as Pump-and-Treat of the Upper Aquifer (P&T Upper). A second option was to apply pump-and-treat in the lower aquifer to capture the distal end of the contaminant plume before it could move offsite, which was referred to as Pump-and-Treat of the Lower Aquifer (P&T Lower). The third option was to apply Monitored Natural Attenuation to let natural processes (biodegradation, abiotic reactions, dispersion, etc.) control the TCE plume.

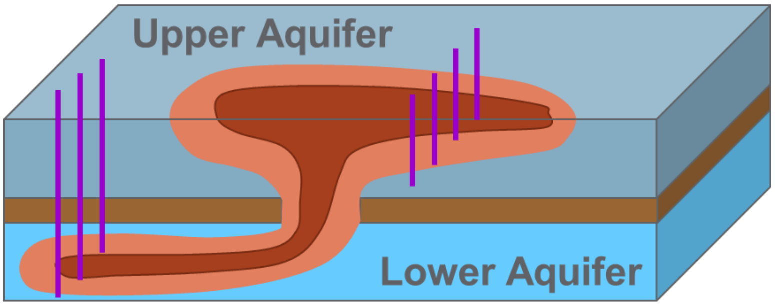


Figure 4. Trichloroethene groundwater plume context for the remedial alternative assessment.

The three remedial alternatives were assessed in a feasibility study and were assigned ratings (Table 2) across seven of the nine CERCLA [1980] criteria (with the final two criteria to be evaluated later). Each alternative was given a rating for each criterion, with 3 being the best rating and 1 the worst. The ratings were converted to rankings in the same manner as in the previous example. Again, a Condorcet and majority winner exists in the unweighted case, this time in the form of Pump-and-Treat of the Lower Aquifer (P&T Lower). All three Condorcet methods again identified the P&T Lower option as the optimal choice, as did Borda's method and DIET.

Table 2. Environmental remediation approaches

Remedial Alternative Criteria	MNA	P&T Lower	P&T Upper
Long-Term Effectiveness and Permanence	1	3	2
Reduction of Toxicity, Mobility, or Volume	1	3	2
Short-Term Effectiveness	2	2	2
Implementability	3	2	1
Cost (Total Non-Discounted value in dollars)	18,586,000	26,194,000	19,897,000
Overall Protection of Human Health and the Environment	1	3	2
Compliance with ARARs	1	3	2

4.3 Weighting Considerations

In these two scenarios, there was no difference in the output of the Condorcet methods, Borda's Method, or DIET. While this does suggest that the prototype is making valid recommendations, these results do not help to differentiate between the three approaches or reveal in which contexts each one is most effective.

The unit testing and MCDM datasets did make clear two important differences in performance between Copeland's method and the two directed graph algorithms. Copeland's method naturally produces a full ranking of the alternatives by Copeland score, while the other two are designed to return just a single winning alternative. Copeland's method is also more prone to ties. These two distinctions mean that Copeland's method might be better suited to situations in which ties do not matter and a full overall ranking would be useful, for example, if the user were trying to create a short list of the top-performing alternatives. When the user requires a single recommendation, Ranked Pairs or Schulze method would be preferred, because ties, while not impossible, would be less common.

While these examples were quite straightforward as MCDM problems, the second dataset can provide a demonstration of how the user can employ the weighting functionality. Weightings can not only communicate priorities to the algorithms, but also explore hypothetical situations which may arise in the future. While the winner of the unweighted scenario is obvious, the user might wonder under what circumstances an alternative like Monitored Natural Attenuation (MNA) might be appropriate. Because the advantage of that method is cost, the user can adjust the weighting of that criteria until the algorithms select MNA as the preferred alternative. The user of the prototype would discover that when cost is at least three times as important as the other criteria, MNA becomes the optimal approach to remediation.

5.0 Conclusions and Future Work

MCDM problems are a common class of decision challenge that scientists and policymakers from many fields commonly encounter. The Condorcet decision tool prototype developed in this work is an attempt at approaching this type of problem in a systematic, efficient manner. The tool can aid the user in quickly identifying an optimal alternative from ranked data with dimensions larger than a human could easily navigate or when an impartial recommendation on a contentious decision is needed. The prototype's weighting capability allows the user to communicate their own priorities, as well as to explore hypothetical scenarios to plan for situations that may arise in the future. While these adapted Condorcet methods are powerful decision-making tools, it is critical for the user to remember that the quality of the prototype's recommendations is only as good as the data and weightings allow it to be, and that ultimately, it is the user who must interpret the recommendation and justify selecting an alternative.

Several items could be addressed in future work. In its current state, the prototype lacks a user interface, limiting its user-friendliness. To make the tool suitable for general use, the first task would be to create a graphical UI, allowing the user to easily enter rankings and weights without directly manipulating JavaScript arrays. This UI would ideally also handle input validation and provide a visualization of the results. For the user's convenience, functionality for converting ratings or other numerical data into rankings could be implemented. It would also be useful to integrate would be allowing incomplete ratings/rankings. This could be useful in cases where one or more criteria are omitted from a ranking. Another potential addition to the tool would be inclusion of score voting algorithms, which could better capture performance differences between rank-adjacent alternatives. In addition, DIET presents a standard to aspire to in terms of allowing for the precise, intuitive weighting of criteria on both a group and individual level.

6.0 References

Arrow, K.J. 1950. “A Difficulty in the Concept of Social Welfare.” *J. Political Economy*, 58(4):328-46.

Biruk, S., and P. Jaśkowski. 2016. “Selection of Subcontractors Using Ordinal Ranking Methods Based on Condorcet Approach.” *Budownictwo i Architektura*, 15(4):33-40. https://doi.org/10.24358/BUD-ARCH_16_154_04.

CERCLA. 1980. *Comprehensive Environmental Response, Compensation, and Liability Act*. Public Law 96-150, as amended, 94 Stat. 2767, 42 USC 9601 et seq.

Emerson, P. 2013. “The Original Borda Count and Partial Voting.” *Social Choice and Welfare*, 40(2):353-58. https://doi.org/10.1007/s00355-011-0603-9.

Gibbard, A. 1973. “Manipulation of Voting Schemes: A General Result.” *Econometrica*, 41(4):587-601. https://doi.org/10.2307/1914083.

Parker, K. 2018. “Rank and Score Aggregation Methods in Competitive Climbing.” *SIAM Undergraduate Research Online*, 11. https://doi.org/10.1137/17S01594X.

Patel, M., B. Bhatt, and M. Vashi. 2017. “SMART-Multi-Criteria Decision-Making Technique for Use in Planning Activities.” In: Proceedings of New Horizons in Civil Engineering (NHCE-2017), Surat, Gujarat, India. https://bvbhatt.com/wp-content/uploads/2018/04/NHCE-2017_SMART-Multi-criteria-decision-making-technique-for-use-in-planning-activities.pdf.

Saari, D.G., and V.R. Merlin. 1996. “The Copeland Method.” *Economic Theory*, 8(1):51-76. https://doi.org/10.1007/BF01212012.

Sahida, A.P., B. Surarso, and R. Gernowo. 2019. “The Combination of the MOORA Method and the Copeland Score Method as a Group Decision Support System (GDSS) Vendor Selection.” *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. https://doi.org/10.1109/ISRITI48646.2019.9034579.

Satterthwaite, M.A. 1975. “Strategy-Proofness and Arrow’s Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions.” *J. Economic Theory*, 10(2):187-217. https://doi.org/10.1016/0022-0531(75)90050-2.

Schulze, M. 2011. “A New Monotonic, Clone-Independent, Reversal Symmetric, and Condorcet-Consistent Single-Winner Election Method.” *Social Choice and Welfare*, 36(2):267-303. https://doi.org/10.1007/s00355-010-0475-4.

Schulze, M. 2021. “The Schulze Method of Voting.” *ArXiv*:1804.02973v9 [cs.GT]. http://arxiv.org/abs/1804.02973.

Tate, L.C., and S.A. Montgomery. 2021. *Decision Information Engagement Tool (DIET) User Guide*. PNNL-31425, Pacific Northwest National Laboratory, Richland, WA.

Tideman, T.N. 1987. "Independence of Clones as a Criterion for Voting Rules." *Social Choice and Welfare*, 4(3):185-206. https://doi.org/10.1007/BF00433944.

Tomczak, M., S. Biruk, and P. Jaskowski. 2019. "Selection of Construction Products Suppliers According to the Condorcet Criterion." *IOP Conference Series: Materials Science and Engineering*, 471:112068. https://doi.org/10.1088/1757-899X/471/11/112068.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

www.pnnl.gov