Efficiently Approximating High-Dimensional Pareto Frontiers for Tree-Structured Networks Using Expansion and Compression

Yiwei Bai[1(✉)], Qinru Shi[2], Marc Grimson[1], Alexander Flecker[3], and Carla P. Gomes[1]

[1] Department of Computer Science, Cornell University, Ithaca, USA
{yb263,mg2425}@cornell.edu, gomes@cs.cornell.edu
[2] Center for Applied Mathematics, Cornell University, Ithaca, USA
qs63@cornell.edu
[3] Department of Ecology and Evolutionary Biology, Cornell University, Ithaca, USA
asf3@cornell.edu

Abstract. Real-world decision-making often involves working with many distinct objectives. However, as we consider a larger number of objectives, performance degrades rapidly and many instances become intractable. Our goal is to approximate higher-dimensional Pareto frontiers within a reasonable amount of time. Our work is motivated by a problem in computational sustainability that evaluates the trade-offs between various ecological impacts of hydropower dam proliferation in the Amazon river basin. The current state-of-the-art algorithm finds a good approximation of the Pareto frontier within hours for three-objective problems, but a six-objective problem cannot be solved in a reasonable amount of time. To tackle this problem, we developed two different approaches: an *expansion method*, which assembles Pareto-frontiers optimized with respect to subsets of the original set of criteria, and a *compression method*, which assembles Pareto-frontiers optimized with respect to compressed criteria, which are a weighted sum of multiple original criteria. Our experimental results show that the aggregation of the different methods can reliably provide good approximations of the true Pareto-frontiers in practice. Source code and data are available at https://github.com/gomes-lab/Dam-Portfolio-Selection-Expansion-and-Compression-CPAIOR.

Keywords: Multi-objective Optimization · Approximation DP

1 Introduction

Multi-objective optimization (MOO) is of vital importance in many real-world problems in computational sustainability [7,13,30], which often involve balancing various environmental, economic, and social objectives, as captured e.g., in

Y. Bai and Q. Shi—Equal contribution.

A. A. Cire (Ed.): CPAIOR 2023, LNCS 13884, pp. 1–17, 2023.
https://doi.org/10.1007/978-3-031-33271-5_1

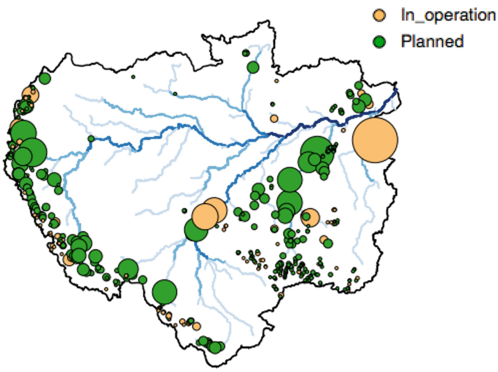

Fig. 1. Amazon hydropower dam portfolio selection problem. Green circles refer to potential dam sites while yellow circles represent already built dams. The sizes of the circles reflect the sizes of the dams in terms of energy output. (Color figure online)

the Sustainable Development Goals [28], that are all crucial to consider when designing solutions to these problems in alignment with human values. Such multi-objective optimization problems often have a large number of competing objectives that must be simultaneously optimized. However, most multi-objective algorithms only work efficiently for 2 or 3 objectives due to the curse of dimensionality [4,17,29]. Thus, finding methods to adapt state-of-the-art MOO algorithms to higher-dimensional problems is a topic of great interest. We propose two effective methods for efficiently approximating higher-dimensional Pareto Frontiers on tree-structured networks, using a state-of-the-art approximation algorithm, which works well in practice on lower-dimensional multi-objective problems.

Our main motivation comes from the real-world problem of strategic planning of hydropower dam expansion [16,32] in the Amazon basin (see Fig. 1), which has a lasting impact on a multitude of ecosystem services provided by the river network such as fish habitat and migration routes, sediment transportation, and fish biodiversity [2,8,34]. Finding optimal portfolios of hydropower dams while balancing the trade-offs between various ecological, social, and economic goals is a good example of a challenging combinatorial multiobjective optimization problem with a relatively large number of objectives. The current state-of-the-art algorithm [14,31] exploits the underlying tree-structure of the river networks and uses a dynamic programming scheme to approximate the Pareto frontier with provable guarantees, within an arbitrary small ϵ factor, and a runtime that is polynomial in the size of the instance and $\frac{1}{\epsilon}$. However, the runtime of this algorithm is still exponential with respect to the number of objectives. For large river networks such as the Amazon basin, while the algorithm is able to solve three or four-objective optimization problems efficiently with a small approximation factor, its performance drops off dramatically once we reach five objectives.

To encompass the complexity of balancing hydropower generation with ecosystem service impacts in the Amazon, higher numbers of objectives need to be considered. Here we address a set of six objectives associated with the proliferation of hydropower dams in the Amazon: **hydropower generation**, the main benefit provided by dams; **River connectivity index**, an indicator of the amount of habitat accessible to migratory fish; **sediment transportation**, the amount of sediment and nutrients transported by the river to the main stem and is essential for flood plain agriculture and fish habitat; **biodiversity impact**, which indicates the overall impact of dams on local biodiversity; **degree of regulation**, which represents the change of river flow regimes caused by dams and has a lasting influence on fish populations; and **greenhouse gases emissions**, which is an estimate of the total amount of greenhouse gases emitted by the construction and operation of dams, such as methane emissions due to the anaerobic decomposition of organic matter from areas flooded by the dams. Failing to consider any one of these six objectives leads to a less comprehensive representation of overall dam impacts. Thus, we aim to approximate the higher-dimensional (e.g., 6 criteria) Pareto frontier with the state-of-the-art algorithm that works efficiently on lower-dimensional (e.g., 3 criteria) problems.

High-dimensional real-world data are often shown to dwell on low-dimensional manifolds [15,18]. Similarly, we make the assumption that, for multi-objective optimization problems on river networks, the six-objective Pareto frontier might approximately lie on a lower-dimensional manifold. Current state-of-the-art works are able to solve this type of MOO problem with three or four objectives efficiently and with a guaranteed approximation factor. Given that these solutions are very likely to be on the Pareto frontier for more objectives, we conjecture that the aggregated solutions from Pareto frontiers optimized for all combinations of three or four-element subsets of the six objectives may form a good approximation of various local regions of the six-objective Pareto frontier. This naturally leads to two questions. First, for a specific n-objective optimization problem, can the true Pareto frontier be approximated by the Pareto frontiers defined by combinations of $k < n$ objectives? Complementary to the first question, can we reduce $k' > k$ objectives to k objectives via different linear combinations of criteria and still approximate the true k' dimensional Pareto frontier?

In answering these questions, we provide two major contributions to greatly improve the approximation of the Pareto frontier for 6 objectives for the Amazon river basin: (1) An ***expansion*** method, which computes the Pareto frontier with respect to different combinations of subsets of the original n criteria, aggregating the resulting non-dominated solutions with respect to all original criteria. (2) A ***compression*** method, complementary to the *expansion* method, which computes the Pareto frontier with respect to the original criteria compressed into fewer criteria via linear combinations, aggregating the resulting non-dominated solutions with respect to all original criteria. (3) We show our approaches produce high-quality Pareto frontiers, in a reasonable amount time, and demonstrate their effectiveness for three different sub-basins within the Amazon and for the entire Amazon basin.

Related Work. Our work leverages a state-of-the-art dynamic programming (DP) algorithm for computing the exact or approximation-guaranteed Pareto frontier for tree-structured networks, referred to as tree-DP [10,14,31]. Typically, the size of the Pareto frontier increases dramatically when the number of criteria increases and tree-DP's running time is proportional to it. Moreover, tree-DP considers all the criteria at the same time so they cannot run in parallel, which is not computationally efficient. Our methods approximate the Pareto frontier from subsets of all the criteria and they can naturally be computed in parallel. Parallel DP [9] may also be employed to boost its speed. Moreover, Genetic Algorithms (GA) have been widely applied to approximate Pareto frontiers and solve multiobjective optimization problems. Many well-established multiobjective GA methods have been developed over the past 40 years, including, but not limited to, vector evaluated GA (VEGA) [25], Multi-objective GA (MOGA) [11], Non-dominated Sorting Genetic Algorithm and its iterations (NSGA, NSGA-II, and NAGA-III) [5,6,27], and multiobjective evolutionary algorithm based on decomposition (MOEA/D) [33]. Nevertheless, GA approaches are not competitive with the current state-of-the-art algorithm [14,31], which exploits the underlying tree-structure of the river networks and uses a dynamic programming scheme to be able to approximate the Pareto frontier with provable guarantees with a runtime that is polynomial in the number of nodes in the network. Other methods, for instance, decision diagrams [3], propositional logic [26] and ray-based methods [19–21,23] are also be used for multiobjective optimization problems, but they cannot scale for the dam portfolio selection problem.

2 Problem Formulation

In this paper, we consider a multi-objective optimization problem with n ($n \geq 3$) objective functions $z^1, z^2, \cdots, z^n$, where the values of these functions are determined by a solution π (also referred to as a policy). Without loss of generality, we assume that all these objectives are to be **maximized**. For any solution π, we define the value vector of π to be

$$v(\pi) = (z^1(\pi), \cdots, z^n(\pi)).$$

Pareto Dominance: For two solutions π and π', if $z^i(\pi) \geq z^i(\pi')$ for all $i = 1, 2, \cdots, n$ and $z^i(\pi) > z^i(\pi')$ holds for at least one $i = 1, 2, \cdots, n$, then we say that the solution π dominates the solution π'

Pareto Frontier: If a solution π is not dominated by any other feasible solution, we say that π is a Pareto-optimal solution. The set of all Pareto-optimal solutions is called the Pareto frontier (denoted as P).

ϵ-approximations for multi-objective solutions: for two Pareto frontiers P_1, P_2, we say P_1 is ϵ-approximated by P_2 if and only if for any $\pi_1 \in P_1$, there exists a $\pi_2 \in P_2$, we have $\pi_1 \geq (1 - \epsilon)\pi_2$ for all objectives.

Hydropower Dam Portfolio Selection Problem: Hydropower dams generate hydroelectricity, which accounts for 16.6% of the world's total electricity and 70% of all renewable electricity [1]. However, the construction of a hydropower dam can cause significant adverse environmental impacts, e.g., disruption of fish migration routes, alteration of river flow regimes, and greenhouse gas emissions.

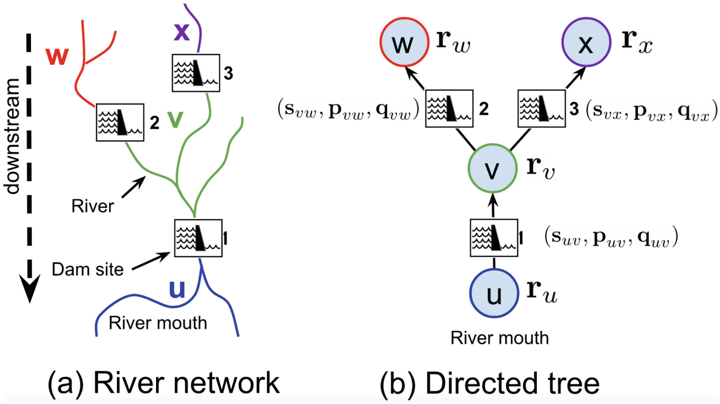

Fig. 2. An example of converting a river network (a) to a tree-structure (b). A node in the tree is a contiguous section of river uninterrupted by dam sites. Edges in the tree are dam sites that connect upstream and downstream segments. The mouth of the river (labelled u in this example) becomes the root of the tree. The tree-DP algorithm leverages this tree structure to be an efficient approximation algorithm.

[2,12]. So the selection of which potential dam sites to build is of vital importance for balancing energy production with ecosystem impacts. The hydropower dam portfolio selection problem is to generate an (approximated) Pareto frontier (the portfolio) of deciding what dams should be built (or selected) from a candidate pool of dam locations proposed by experts with respect to the six important criteria mentioned in the introduction. One solution in the portfolio is a subset of the dam candidate pool to be built.

The Off-the-Shelf Algorithm: Our methods leverages an algorithm that can compute low-dimensional Pareto frontiers efficiently for tree-structured problems. In this paper, we use the state-of-the-art tree dynamic programming (tree-DP) based approximation algorithm [14,31]. It can compute the exact solution given enough time or compute an ϵ approximated solution. The tree-DP algorithm models the entire river system as a tree structure (directed tree). Each dam site represents an edge and two vertices of that edge are the upstream river region and downstream river region respectively, where the river region is a contiguous part of the river, i.e., the streams of that region are connected and not blocked by any potential dam position (see Fig. 2). A bottom-up DP process can be done to compute the Pareto frontier. The running complexity of the DP algorithm is proportional to the number of solutions considered at each node, it can round the value of each criterion to the multiplicative of a small value (the approximation factor) to merge many similar solutions into one solution.

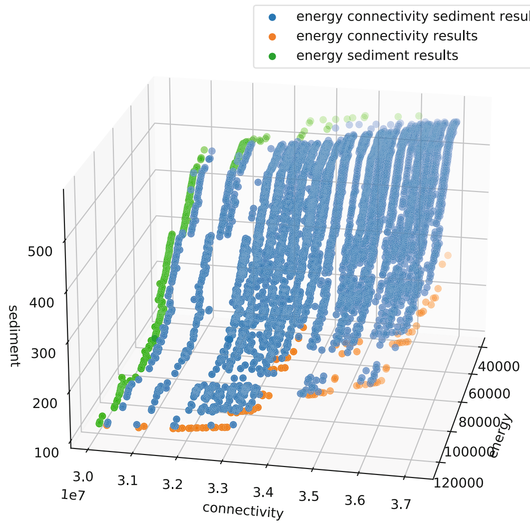

Fig. 3. An example of trying to approximate a three-criteria Pareto frontier with two-criteria optimization results. Each dot represents the values of the three criteria of one solution. For the two criteria solutions, we compute the value of the remaining criterion based on the dams built in that solution. We can observe that the two-criteria solutions only cover the edges of the Pareto optimal sets formed by the three-criteria optimization results.

3 The Expansion Method

We denote the actual n-objective Pareto frontier as P_n and define $V_n = \{v(\pi)|\pi \in P_n\}$. Given a positive integer $2 \leq k < n$, for all $1 \leq i_1 < i_2 < \cdots < i_k \leq n$, i.e., all the possible sized k combinations of n criteria, we compute an ϵ-approximate Pareto frontier $\tilde{P}_{i_1,\cdots,i_k}$ w.r.t. $z^{i_1}, z^{i_2}, \cdots, z^{i_k}$. We define the union for these k-objective Pareto frontiers to be

$$\tilde{P}_k = \bigcup_{1 \leq i_1 < \cdots < i_k \leq n} \tilde{P}_{i_1,\cdots,i_k},$$

and

$$\tilde{V}_k = \{v(\pi)|\pi \in \tilde{P}_k\}.$$

$\tilde{P}_k$ is the output of the ***Expansion method*** (see Fig. 4). In this paper we study the following proposition: for some real-world problems, $\tilde{V}_k$ forms a sufficiently good coverage of V_n with appropriate choices of k and ϵ.

The ***Expansion method*** method might look counter-intuitive at first because when we consider the smallest possible cases, two-criteria optimization solutions usually only cover the edges of a three-criteria Pareto frontier (see Fig. 3). However, in practice, we are able to approximate higher-dimensional Pareto frontiers using lower-dimensional Pareto frontiers. We will show an example after introducing the compression method.

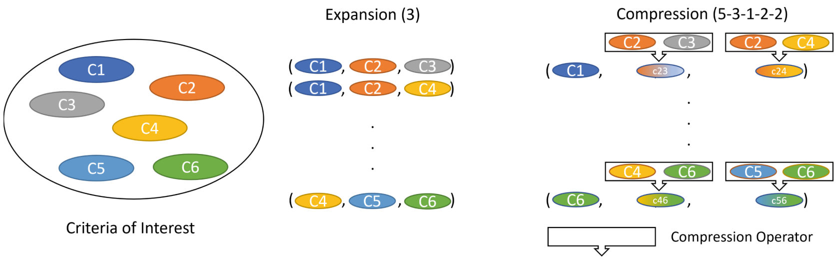

Fig. 4. High-level depiction of the expansion and compression methods. The left circle contains all the criteria we are interested in. Both the expansion and compression methods use the off-the-shelf Pareto-frontier algorithm optimized with respect to the criteria in the parentheses. In the expansion method, we choose all possible combinations of three criteria, from all criteria, and merge their results to generate the final results (evaluated with respect to all criteria). The compression method reduces the number of criteria by compressing the original criteria into fewer criteria. For example, 5-3-1-2-2 denotes that five criteria are reduced into three by keeping the first one as is and compressing the last two pairs of criteria. The compression operator is defined in Eq. 1.

4 The Compression Method

The ***Expansion method*** is likely to miss some solutions since it only optimizes with respect to a subset of the full criteria. We, therefore, propose a ***compression method*** (see Fig. 4) to further complement the Pareto frontier computed by the expansion method. By compressing $k' > k$ criteria into k criteria, the off-the-shelf algorithm can compute the Pareto frontier for k criteria while implicitly considering k' criteria.

Formally, as defined before, P_n refers to the actual $n-$objective Pareto frontier and $V_n = \{v(\pi)|\pi \in P_n\}$. The compression configuration can be defined as $(k', k, a_1, a_2, \ldots, a_k$ (Fig. 4) where $0 < k < k' \leq n$ and $\sum_{i=1}^{k} a_i = k'$. The idea of this configuration is to compress k' criteria into k criteria and a_i $(i = 1, \cdots, k)$ describe what criteria should be merged. The compression operator is defined as follows: for all the possible sized k' combinations of n criteria: $0 < i_1, i_2, \cdots, i_{k'} \leq n$, the compressed criteria evaluation function c_i' can be computed as:

$$c_i' = \sum_{j=sum[i-1]+1}^{sum[i]} w_j * z^j \quad (1)$$

where $a_0 = 0$, $sum[i] = \sum_{j=1}^{i} a_j, sum[0] = 0$ and w_j is the scalar weight. Note, if any two criteria i_p, i_q are compressed into one criterion, then $p \neq q$. The scales of different criteria vary substantially so the selection of the weights w_i is vital to the performance. A straightforward selection strategy is to normalize the criteria into the same scale: denote z_j^{max} as the max j-th criterion value among all the dams/rivers and w_j can be set as $1 - \frac{z_j^{max}}{\sum_{i=1}^{n} z_i^{max}}$. This normalized strategy treats each criterion as having the same importance. Then, for all the possible $0 < i_1, i_2, \cdots, i_{k'} \le n$, i.e., k' combinations of n criteria, we compute the Pareto frontier $\tilde{P}_{1',2',\ldots,k'}$. We define the union of these k-objective Pareto frontiers to be $\tilde{P}_{k'}$ and it is the output of our compression method (see Fig. 4 for more details). In general, k' is the number of actual criteria we consider while k refers to the number of compressed criteria considered by the algorithm. The a_i $(i = 1, \ldots, k)$ specifies how we compress the k' criteria into k criteria.

These two methods share a common idea, i.e., approximate the high-dimensional Pareto frontier using many low-dimensional Pareto frontiers. The difference is that the compression method implicitly considers more criteria by compressing multiple criteria into fewer criteria.

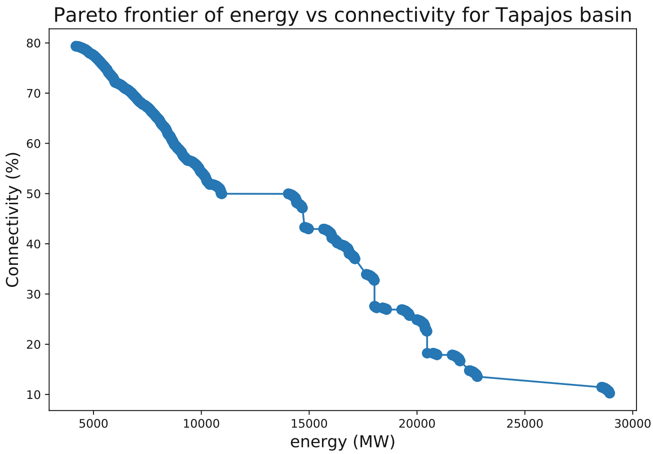

Fig. 5. Exact non-Convex Pareto frontier of energy-connectivity for the Tapajós basin.

5 Experiments

5.1 Experimental Setup

Our study focuses on the Amazon basin, where more than 350 large hydropower dams have been proposed. To show the generalizability of our methods and provide scalability insights, we also considered three sub-basins of the Amazon basin: Marañón, Tapajós and the West Amazon. We compute the Pareto frontier with respect to the six important criteria introduced in the introduction: hydropower generation, river connectivity index, sediment transportation, biodiversity impact, the degree of river regulation, and greenhouse gases emissions. For our underlying off-the-shelf algorithm and baseline, we use the state-of-the-art tree DP algorithm that computes the exact or approximate Pareto frontier, adopting the original papers' recommended configurations [14,31]. Our baseline is to directly consider all six criteria with the minimal approximation factor the runtime constraints allow.

5.2 Evaluation Method

To compare the optimization results of the various methods, we need a metric that can evaluate both the optimality and the coverage of the approximate Pareto frontiers. Note that the exact Pareto frontier we are trying to approximate can be non-convex (see Fig. 5). So, not only do we care about the overall shape of the Pareto frontier, but also the evaluation of the individual solutions. Therefore, we propose an evaluation method that divides the solution space into ϵ hypercubes following [24]'s approach.

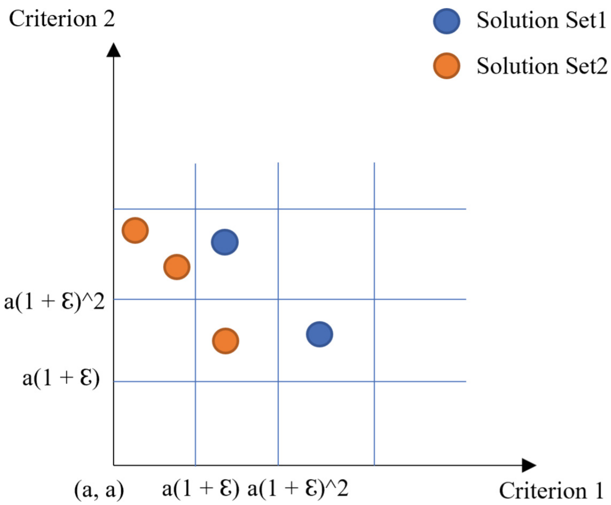

Fig. 6. We use two criteria as an example. The solution space is divided into several hypercubes. The upper bound of each cube is $1+\epsilon$ of its lower bound. The lower bound a is the minimum value of its criterion. The number of hypercubes one solution covers is a good metric. Consider two solutions sets 1 and 2. Set1 covers two hypercubes, while set2 covers two hypercubes. Note that these numbers are computed when we consider each solution set individually. When we compare them, we need to compute the new Pareto frontier after merging their solutions.

More specifically, for a n-objective optimization problem where, without loss of generality, every objective is to be maximized and the objective values are strictly non-negative, we define the solution space to be a n-dimensional space where each axis represents the value of one objective. We also make the assumption that the minimum possible value of each objective is non-negative. For a given error bound $\epsilon > 0$, we divide the solution space into hypercubes where the upper bound is $1+\epsilon$ of the lower bound on each axis, with the smallest value of the lower bounds being the minimum possible value of the corresponding objective. Similar to the definition of Pareto-dominance, for two different hypercubes, if for each axis, the upper bound of the first hypercube is greater than or equal

Table 1. Non-dominated hypercubes occupied by the different methods. Note that the occupied non-dominated hypercubes are computed by merging and comparing tree-DP solutions, Expansion solutions, and Compression solutions. The approximation factors are shown in Table 2. The number in the parentheses is the number of criteria that are considered by the tree-DP algorithm. The epsilon is used in the hypercube computation. The Compression method further improves the performance. Expansion+Compression is a good approximation for all the basins, outperforming the baseline tree-DP, which provides a theoretical approximation guarantee.

Basin	epsilon	Tree DP (6)	Expansion-3 (3)	Expansion-4 (4)	Compression-3 (2)	Compression-4 (3)	Compression-5 (3)	Expansion + Compression
Marañón	0.01	12070	9	2	1344	14884	1620	17425
Marañón	0.05	35	1	0	747	771	2	816
Tapajós	0.01	0	681	382	1435	14878	0	17371
Tapajós	0.05	0	44	22	187	1057	0	1277
West Amazon	0.01	0	66	149	306	20851	0	21371
West Amazon	0.05	0	6	11	27	1160	0	1191
Amazon	0.01	0	6778	9	1623	1243	2397	12044
Amazon	0.05	0	485	1	216	85	75	847

to the upper bound of the second hypercube, we say that the first hypercube dominates the second hypercube. More details can be found in the Fig. 6.

To compare two or more sets of solutions, we first identify all of the ϵ-hypercubes that are occupied by at least one solution in any of the solution sets. We then find the set of occupied hypercubes that are not dominated by any other occupied hypercube. Finally, we compute for each set of solutions the number of non-dominated hypercubes they occupy. If one set of solutions covers more non-dominated ϵ-hypercubes than the other set, we say that the first solution set has better ϵ-coverage than the second solution set. Notice that since the set of occupied hypercubes will change depending on the sets of solutions compared and the number of solutions in each set, the number of non-dominated hypercubes covered by one set of solutions may change depending on the solution sets compared to, so the number of non-dominated hypercubes covered cannot be used as a universal metric of the quality of approximate Pareto frontiers. However, when comparing fixed sets of solutions, the metric provides a good comparison of the accuracy and coverage of the solution sets.

The visualized Pareto frontiers computed by different solutions can be more straightforward for comparison. However, it is difficult to visualize Pareto frontiers of dimensions higher than three. For the sake of clear visualization and easier comparison, we use the Uniform Manifold Approximation and Projection (UMAP) [22] method to project a high-dimensional Pareto-frontier onto a two-dimensional plane while preserving the general proximity relationships between the values of solutions. We merge the solutions generated by the tree-DP and our methods, then only save the non-dominated solutions, and finally, use UMAP to visualize these solutions. We have developed a website for visualizing the Pareto frontier.

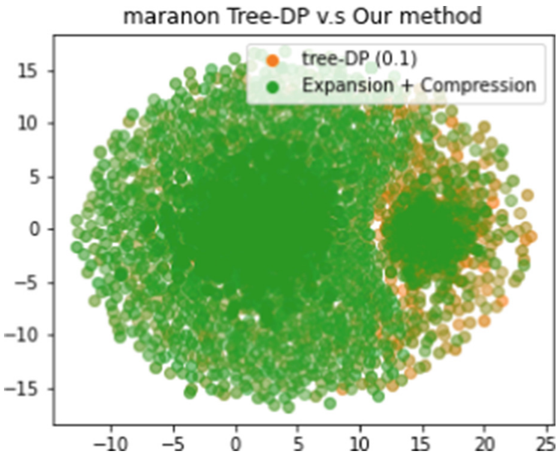

Fig. 7. UMAP results of the baseline approximate six-criteria Pareto frontier of the Marañón basin ($\epsilon = 0.1$) and the Expansion + Compression approximation results. We can observe that our method covers most solutions of the tree-DP algorithm. Note that the solutions fed into the UMAP results are all non-dominated solutions.

5.3 Experimental Results

We first show how well each of the ***Expansion methods*** and ***Compression methods*** can approximate the six criteria Pareto frontier for the full Amazon Basin and three sub-basins. Since the Pareto frontier may be non-convex (see Fig. 5), our metric is the number of non-dominated hypercubes occupied by the solutions computed by a given method. For methods' solution comparison, from both fine-grain and coarse perspectives, we used two hypercube error bounds: $\epsilon = 0.01$ and $\epsilon = 0.05$.

For all the experiments, we always include hydropower generation as a single criterion, otherwise, the optimal solution would be the trivial solution of building no dams as hydropower generation is the only criterion positively correlated with construction. Due to the scale of the problem, the state-of-the-art tree-DP algorithm can only compute or approximate the Pareto frontier in a reasonable amount of time for $k = 3, 4$ criteria. Thus, for both the expan-

Table 2. Approximation factors of the baseline, the expansion method, and the compression method. The number in the parentheses is the number of criteria considered by the tree-DP algorithm. Factors vary and are set so that every experiment is under the 80 hour limit, except for the baseline.

Basin	Baseline (6)	Expansion-3 (3)	Expansion-4 (4)	Compression-3 (2)	Compression-4 (3)	Compression-5 (3)
Marañón	0.1	0	0	0	0	0
Tapajós	1.0	0.1	0.4	0.3	0.3	0.3
West Amazon	1.0	0.2	0.2	0.2	0.2	0.2
Amazon	1.25	0.5	2.0	0.5	0.5	0.5

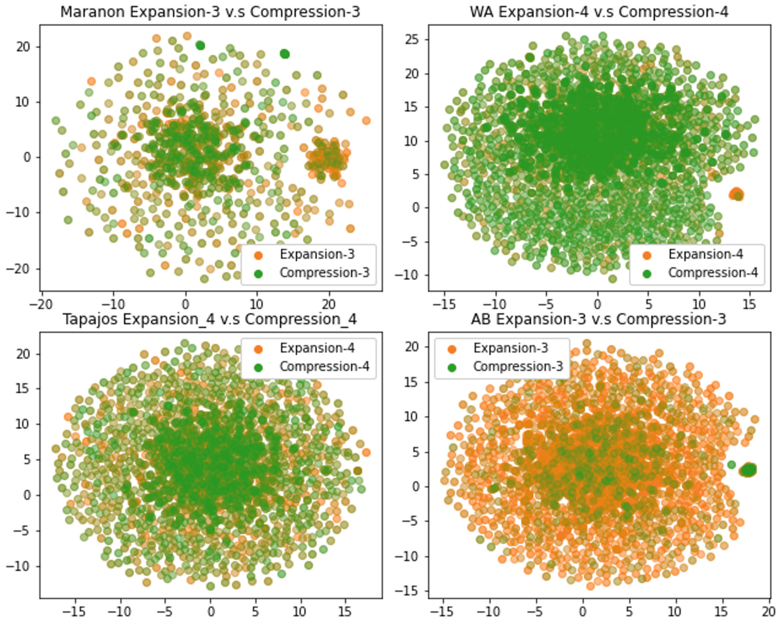

Fig. 8. UMAP results of four basins' Expansion and Compression solutions. We merge their sets of solutions and use all the non-dominated solutions to compute these UMAP results. The number in the names of the method refers to the actual criteria (k' of the Compression method) considered. We can observe that for all the basins, two methods capture different perspectives of the problem and this leads to very different solutions.

sion and compression methods, we consider combinations of 3 and 4 criteria. For the ***Expansion method*** we refer to the experiment that computes all the possible 3 criteria combinations ***Expansion-3*** and 4 criteria combinations as ***Expansion-4***, and they both consider $C^5_{2/3} = 10$ combinations (here we choose from 5 criteria instead of 6 since we always include energy as one criterion). For the compression method, we describe the configuration in the following format, assuming the criteria are sequentially numbered: $(k', k, a_1, \ldots, a_k)$, which denotes that we compress k' criteria into k criteria using the scheme $a_1, \ldots, a_k$. Each a_i denotes how many of the original criteria in the sequence are compressed to produce the final criterion i. The formal definition of compressing these a_i criteria can check Eq. 1. We consider three situations for the compression method: (1) $(3, 2, 1, 2)$ (denoted as ***Compression-3***): since the first target criterion must be the single (uncompressed) energy criterion, we have a total of $C^5_2 = 10$ combinations; (2) $(4, 3, 1, 2, 1)$ (denoted as ***Compression-4***): where we have a total of $C^5_2 \times C^5_1 = 50$ combinations; and (3) $(5, 3, 1, 2, 2)$ (denoted as ***Compression-5***): where we have a total of $C_2^{\frac{4*5}{2}} = 45$ combinations. To reduce the computational overhead, we assign each (compressed) criterion the same importance factor when reducing multiple criteria into one. We tune the

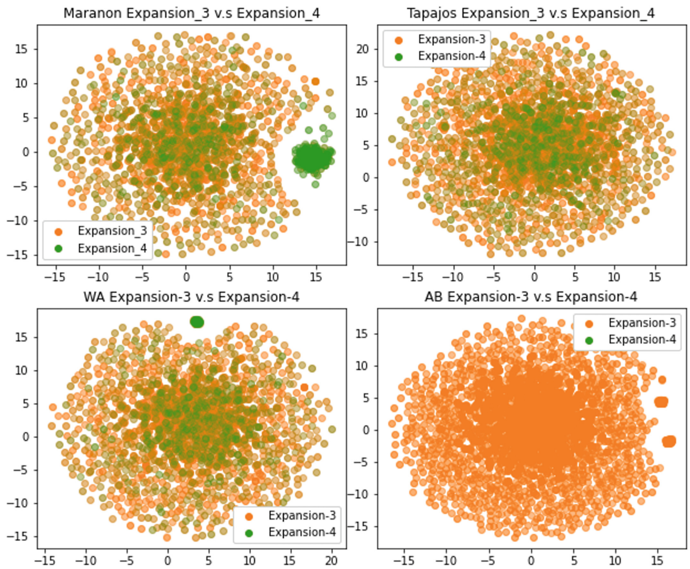

Fig. 9. UMAP results for the four basins' Expansion-3 and Expansion-4 solutions. We merge their sets of solutions and use all the non-dominated solutions to compute these UMAP results. For all the basins, except for the entire Amazon, the Expansion methods compute very different solutions, when considering different numbers of criteria. For the entire Amazon basin, due to the large approximation factor (2.0) used by the Expansion-4 method, it can only find a few solutions.

approximation factor of the tree-DP algorithm to ensure a single experiment (e.g. a combination of energy-connectivity-GHG (greenhouse gas emissions) using the expansion method) is finished within an 80 hours time budget running on a computation node that has 24 Intel(R) Xeon(R) CPU X5690 @ 3.47GHz. Note we do not set a time limit for the baseline method. The baseline runtime for all four basins is greater than 10 days.

The main results are summarized in Table 1 and the approximation factors of all methods are shown in Table 2. We compute the non-dominated hypercubes for each method, then combine these hypercubes and remove all dominated ones to form the hypercube Pareto frontier. Since the number of solutions can be quite large and there are many solutions that are quite similar to each other, we sort the solutions with respect to the number of dams they build and sample $3,000$ solutions uniformly from each sub-experiment. For the baseline method, we either select all of its solutions or uniformly sample $1,000,000$ solutions. For each method, we then count how many grids of its solution set belong to the hypercube Pareto frontier. Table 1 shows that even for the smallest sub-basin, the Marañón, which the tree-DP algorithm can finish with an $\epsilon = 0.1$ approximation factor, our method can get a better approximation (17425 v.s. 12070). In Fig. 7, we also show visually how our method can cover almost every solution of the six criteria Pareto frontier computed by the Tree-DP algorithm (approximation factor $\epsilon = 0.1$) for the Marañón sub-basin using UMAP. For the other larger basins, where the tree-DP algorithm alone is only able to scale with a very loose approximation factor, *the solution sets computed by our Expansion and Compression methods entirely dominate those of the tree-DP algorithm.*

We also compare the sets of solutions generated by our two methods to study how different they are. The results are summarized in Fig. 8, where we compare the UMAP results after removing all the dominated solutions. For all the basins, except for the full Amazon basin, our two methods generate very different solutions. The Compression methods produce the largest number of non-dominated solutions (see Table 1). In terms of the full Amazon basin, interestingly the Expansion method outperforms the Compression method, even though the Compression method further improves the Expansion method. Understanding the trade-offs of the two approaches is a future research question. In any case, the combination Expansion+Compression clearly outperforms the tree-DP algorithm baseline, as the approximation used by the tree-DP algorithm for six criteria has to be quite loose since the number of solutions is enormously large when directly considering all six criteria. In contrast, the Expansion and Compression methods use a much smaller approximation factor as the number of target criteria is small (2 or 3), efficiently handled by the tree-DP algorithm, which makes up for only optimizing with respect to subsets of the original criteria.

5.4 Ablation Study

We conducted experiments to study how the number of criteria affects the solution sets computed by the expansion method and the compression method. We merge the solution sets of all expansion methods and all compression methods separately, and then remove all the dominated solutions from them. We then run UMAP to project their non-dominated solutions to a 2-dimensional space to analyze the relative distances between the solutions. The results are shown in Fig. 9 (Expansion) and Fig. 10 (Compression). For all the basins, except for the full Amazon basin, Expansion-3 and Expansion-4 cover very different areas. For the entire Amazon basin, Expansion-3 dominates Expansion-4 since it is able to use a much smaller approximation factor (0.5 v.s. 2.0). For the Compression method, in general, solutions from Compression-4 and Compression-5 dominate Compression-3 solutions since the first two methods actually consider one more criterion Moreover, except for the full Amazon basin, Compression-4 and Compression-5 methods cover diverse areas. These results show that considering different numbers of criteria can provide different solution perspectives.

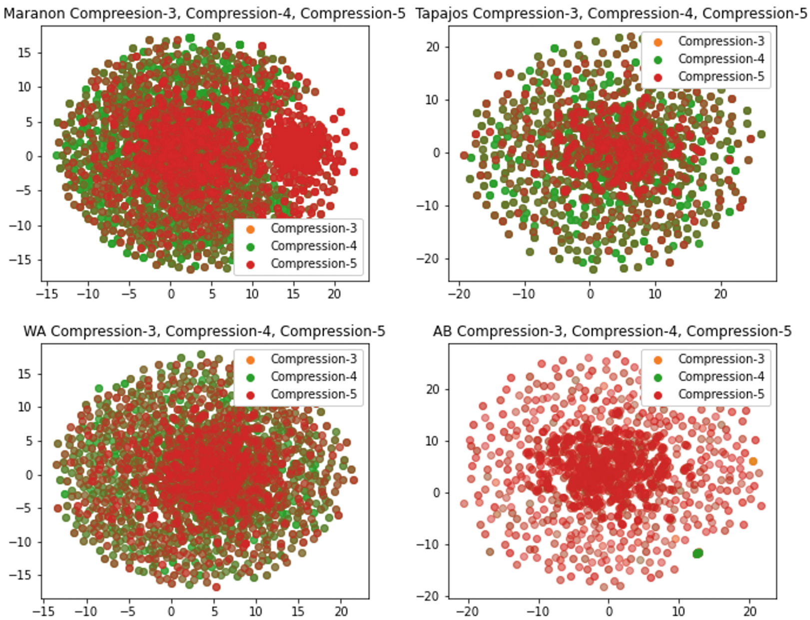

Fig. 10. UMAP results of four basins' Compression-3, Compression-4 and Compression-5 solutions. We merge their sets of solutions and use all the non-dominated solutions to compute these UMAP results. For all the basins, solution sets from the Compression-4 and Compression-5 experiments in general dominate the Compression-3 solutions since the first two methods actually consider one more criterion. Moreover, for all the basins except for the full Amazon basin, the solution sets of Compression-4 and Compression-5 cover very different solutions.

6 Conclusion

We propose the Expansion method to efficiently approximate an n (high)-dimension Pareto frontier by computing all the possible k (low)-dimension Pareto frontiers and merging their solutions together. Moreover, we also introduce a Compression method that compresses multiple criteria into fewer criteria, allowing the algorithm to consider more criteria implicitly, further improving the Expansion method. The combination of the Expansion and Compression methods provides a good Pareto frontier approximation for three Amazon sub-basins and the full Amazon basin for six criteria, in practice outperforming the baseline tree-DP approach, which provides a theoretical approximation guarantee. Understanding the trade-offs between the Expansion and Compression approaches is an interesting topic for further research. We hope this work inspires other approaches for efficiently approximating high-dimensional Pareto frontiers.

Acknowledgments. We thank the reviewers for all the constructive feedback. This research is supported in part by grants from the National Science Foundation, Air Force Office of Scientific Research, and Cornell Atkinson Center for Sustainability.

References

1. Hydroelectricity. https://en.wikipedia.org/wiki/Hydroelectricity. Accessed 26 Jan 2022
2. Almeida, R.M., et al.: Reducing greenhouse gas emissions of amazon hydropower with strategic dam planning. Nat. Commun. **10**(1), 1–9 (2019)
3. Bergman, D., Cire, A.A.: Multiobjective optimization by decision diagrams. In: Rueher, M. (ed.) CP 2016. LNCS, vol. 9892, pp. 86–95. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44953-1_6
4. Brockhoff, D., Zitzler, E.: Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization. In: Runarsson, T.P., Beyer, H.-G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 533–542. Springer, Heidelberg (2006). https://doi.org/10.1007/11844297_54
5. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2013)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
7. Ehrgott, M., Gandibleux, X.: A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spectrum **22**(4), 425–460 (2000)
8. Finer, M., Jenkins, C.N.: Proliferation of hydroelectric dams in the Andean Amazon and implications for Andes-Amazon connectivity. PLOS ONE **7**(4), 1–9 (2012). https://doi.org/10.1371/journal.pone.0035126
9. Fioretto, F., Pontelli, E., Yeoh, W., Dechter, R.: Accelerating exact and approximate inference for (distributed) discrete optimization with GPUs. Constraints **23**, 1–43 (2018)
10. Flecker, A.S., et al.: Reducing adverse impacts of amazon hydropower expansion. Science **375**(6582), 753–760 (2022)
11. Fonseca, C.M., Fleming, P.J., et al.: Genetic algorithms for multiobjective optimization: formulation discussion and generalization. In: ICGA, vol. 93, pp. 416–423 (1993)
12. Forsberg, B.R., et al.: The potential impact of new Andean dams on amazon fluvial ecosystems. Plos One **12**(8), 1–35 (2017). https://doi.org/10.1371/journal.pone.0182254
13. Gomes, C., et al.: Computational sustainability: computing for a better world and a sustainable future. Commun. ACM **62**(9), 56–65 (2019)
14. Gomes-Selman, J.M., Shi, Q., Xue, Y., García-Villacorta, R., Flecker, A.S., Gomes, C.P.: Boosting efficiency for computing the Pareto frontier on tree structured networks. In: van Hoeve, W.-J. (ed.) CPAIOR 2018. LNCS, vol. 10848, pp. 263–279. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93031-2_19
15. Huang, D., Yi, Z., Pu, X.: Manifold-based learning and synthesis. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **39**(3), 592–606 (2009). https://doi.org/10.1109/TSMCB.2008.2007499
16. Kareiva, P.M.: Dam choices: analyses for multiple needs. Proc. Natl. Acad. Sci. **109**(15), 5553–5554 (2012)
17. Khare, V., Yao, X., Deb, K.: Performance scaling of multi-objective evolutionary algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Thiele, L., Deb, K. (eds.) EMO 2003. LNCS, vol. 2632, pp. 376–390. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36970-8_27

18. Li, B., Li, J., Tang, K., Yao, X.: Many-objective evolutionary algorithms: a survey. ACM Comput. Surv. **48**(1) (2015). https://doi.org/10.1145/2792984
19. Lin, X., Zhen, H.L., Li, Z., Zhang, Q., Kwong, S.: Pareto multi-task learning (2019). https://doi.org/10.48550/ARXIV.1912.12854
20. Ma, P., Du, T., Matusik, W.: Efficient continuous pareto exploration in multi-task learning (2020). https://doi.org/10.48550/ARXIV.2006.16434
21. Mahapatra, D., Rajan, V.: Exact pareto optimal search for multi-task learning: touring the pareto front (2021). https://doi.org/10.48550/ARXIV.2108.00597
22. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
23. Nowak, D., Küfer, K.H.: A ray tracing technique for the navigation on a non-convex pareto front (2020). https://doi.org/10.48550/ARXIV.2001.03634
24. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. In: Proceedings 41st Annual Symposium on Foundations of Computer Science, pp. 86–92. IEEE (2000)
25. Schaffer, J.D.: Some experiments in machine learning using vector evaluated genetic algorithms (1985). https://www.osti.gov/biblio/5673304
26. Soh, T., Banbara, M., Tamura, N., Le Berre, D.: Solving multiobjective discrete optimization problems with propositional minimal model generation. In: Beck, J.C. (ed.) CP 2017. LNCS, vol. 10416, pp. 596–614. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66158-2_38
27. Srinivas, N., Deb, K.: Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol. Comput. **2**(3), 221–248 (1994)
28. United Nations General Assembly: Transforming our world: the 2030 agenda for sustainable development (2015). https://sdgs.un.org/2030agenda
29. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 742–756. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70928-2_56
30. Wiecek, M.M., Ehrgott, M., Fadel, G., Figueira, J.R.: Multiple criteria decision making for engineering (2008)
31. Wu, X., et al.: Efficiently approximating the pareto frontier: hydropower dam placement in the amazon basin. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
32. Zarfl, C., Lumsdon, A.E., Berlekamp, J., Tydecks, L., Tockner, K.: A global boom in hydropower dam construction. Aquat. Sci. **77**(1), 161–170 (2015)
33. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
34. Ziv, G., Baran, E., Nam, S., Rodríguez-Iturbe, I., Levin, S.A.: Trading-off fish biodiversity, food security, and hydropower in the Mekong river basin. Proc. Natl. Acad. Sci. **109**(15), 5609–5614 (2012). https://doi.org/10.1073/pnas.1201423109. https://www.pnas.org/content/109/15/5609