Obtaining Approximately Optimal and Diverse Solutions via Dispersion

Jie Gao[1], Mayank Goswami[2(✉)], C. S. Karthik[1], Meng-Tsung Tsai[3], Shih-Yu Tsai[4], and Hao-Tsung Yang[5]

[1] Department of Computer Science, Rutgers University, New Brunswick, USA
[2] Department of Computer Science, Queens College, City University of New York, New York, USA
mayank.goswami@qc.cuny.edu
[3] Institute of Information Science, Academia Sinica, Taipei, Taiwan
[4] Department of Computer Science, Stony Brook University, Stony Brook, USA
[5] Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan

Abstract. There has been a long-standing interest in computing diverse solutions to optimization problems. In 1995 J. Krarup [28] posed the problem of finding k-edge disjoint Hamiltonian Circuits of minimum total weight, called the peripatetic salesman problem (PSP). Since then researchers have investigated the complexity of finding diverse solutions to spanning trees, paths, vertex covers, matchings, and more. Unlike the PSP that has a constraint on the total weight of the solutions, recent work has involved finding diverse solutions that are all optimal.

However, sometimes the space of exact solutions may be too small to achieve sufficient diversity. Motivated by this, we initiate the study of obtaining sufficiently-diverse, yet approximately-optimal solutions to optimization problems. Formally, given an integer k, an approximation factor c, and an instance I of an optimization problem, we aim to obtain a set of k solutions to I that a) are all c approximately-optimal for I and b) maximize the diversity of the k solutions. Finding such solutions, therefore, requires a better understanding of the global landscape of the optimization function.

Given a metric on the space of solutions, and the diversity measure as the sum of pairwise distances between solutions, we first provide a general reduction to an associated budget-constrained optimization (BCO) problem, where one objective function is to optimized subject to a bound on the second objective function. We then prove that bi-approximations to the BCO can be used to give bi-approximations to the diverse approximately optimal solutions problem.

As applications of our result, we present polynomial time approximation algorithms for several problems such as diverse c-approximate *maximum matchings, $s-t$ shortest paths, global min-cut, and minimum weight bases of a matroid.* The last result gives us *diverse c-approximate minimum spanning trees*, advancing a step towards achieving diverse c-approximate TSP tours.

A. Castañeda and F. Rodríguez-Henríquez (Eds.): LATIN 2022, LNCS 13568, pp. 222–239, 2022.
https://doi.org/10.1007/978-3-031-20624-5_14

We also explore the connection to the field of multiobjective optimization and show that the class of problems to which our result applies includes those for which the associated DUALRESTRICT problem defined by Papadimitriou and Yannakakis [35], and recently explored by Herzel et al. [26] can be solved in polynomial time.

Keywords: Diversity · Minimum spanning tree · Maximum matching · Shortest path · Travelling salesman problem · Dispersion problem

1 Introduction

Techniques for optimization problems focus on obtaining optimal solutions to an objective function and have widespread applications ranging from machine learning, operations research, computational biology, networks, to geophysics, economics, and finance. However, in many scenarios, the optimal solution is not only computationally difficult to obtain, but can also render the system built upon its utilization vulnerable to adversarial attacks. Consider a patrolling agent tasked with monitoring n sites in the plane. The most efficient solution (i.e., maximizing the frequency of visiting each of the n sites) would naturally be to patrol along the tour of shortest length[1] (the solution to TSP - the Traveling Salesman Problem). However, an adversary who wants to avoid the patroller can also compute the shortest TSP tour and can design its actions strategically [39]. Similarly, applications utilizing the minimum spanning tree (MST) on a communication network may be affected if an adversary gains knowledge of the network [13]; systems using solutions to a linear program (LP) would be vulnerable if an adversary gains knowledge of the program's function and constraints.

One way to address the vulnerability is to use a set of approximately optimal solutions and randomize among them. However, this may not help much to mitigate the problem, if these approximate solutions are combinatorially too "similar" to the optimal solution. For example, all points in a sufficiently small neighborhood of the optimal solution on the LP polytope will be approximately optimal, but these solutions are not too much different and the adversaries can still effectively carry out their attacks. Similarly one may use another tree instead of the MST, but if the new tree shares many edges with the MST the same vulnerability persists. Thus k-best enumeration algorithms [18,24,30,31,33] developed for a variety of problems fall short in this regard.

One of the oldest known formulations is the Peripatetic Salesman problem (PSP) by Krarup [28], which asks for k-edge disjoint Hamiltonian circuits of minimum total weight in a network. Since then, several researchers have tried to compute diverse solutions for several optimization problems [4,5,16,23]. Most of these works are on graph problems, and diversity usually corresponds to the size

[1] We assume without loss of generality that the optimal TSP is combinatorially unique by a slight perturbation of the distances.

of the symmetric difference of the edge sets in the solutions. Crucially, almost all of the aforementioned work demands either every solution individually be optimal, or the set of solutions in totality (as in the case of the PSP) be optimal. Nevertheless, **the space of optimal solutions may be too small to achieve sufficient diversity**, and it may just be singular (unique solution). In addition, for NP-complete problems finding just one optimal solution is already difficult. While there is some research that takes the route of developing FPT algorithms for this setting [5,17], to us it seems practical to also consider the relaxation to approximately-optimal solutions.

This motivates the problem of finding a set of diverse and *approximately optimal* solutions, which is the problem considered in this article. The number of solutions k and the desired approximation factor $c > 1$ is provided by the user as input. Working in the larger class gives one more hope of finding diverse solutions, yet every solution has a guarantee on its quality.

1.1 Our Contributions

We develop approximation algorithms for finding k solutions to the given optimization problem: for every solution, the quality is bounded by a user-given approximation ratio $c > 1$ to the optimal solution and the diversity of these k solutions is maximized. Given a metric on the space of solutions to the problem, we consider the diversity measure given by the sum (or average) of pairwise distances between the k solutions. Combining ideas from the well-studied problem on dispersion (which we describe next), we reduce the above problem to a budget constrained optimization (BCO) program.

1.2 Dispersion

Generally speaking, if the optimization problem itself is $\mathcal{NP}$-hard, finding diverse solutions for that problem is also $\mathcal{NP}$-hard (see Proposition 1 for more detail). On the other hand, interestingly, even if the original problem is not $\mathcal{NP}$-hard, finding diverse and approximately optimal solutions can still be $\mathcal{NP}$-hard. This is due to the connection of the diversity maximization objective with the general family of problems that consider selecting k elements from the given input set with maximum "dispersion", defined as max-min distance, max-average distance, and so on.

The dispersion problem has a long history, with many variants both in the metric setting and the geometric setting [15,29,38]. For example, finding a subset of size k from an input set of n points in a metric space that maximizes the distance between closest pairs or the sum of distances of the k selected points are both $\mathcal{NP}$-hard [1,37]. For the max-sum dispersion problem, the best known approximation factor is 2 for general metrics [7,25], although PTAS are available for Euclidean metrics or more generally, metrics of negative type, even with matroid constraints [10,11].

Dispersion in Exponentially-Sized Space. We make use of the general framework of the 2-approximation algorithm [8,37] to the max-sum k-dispersion problem, a greedy algorithm where the $i+1$th solution is chosen to be the most distant/diverse one from the first i solutions. Notice that in our setting, there is an important additional challenge to understand the space within which the approximate solutions stay. In all of the problems we study, the total number of solutions can be *exponential in the input size*. Thus we need to have a non-trivial way of navigating within this large space and carry furthest insertion without considering all points in the space. This is where our reduction to budget constrained problem comes in.

Self Avoiding Dispersion. Furthermore, even after implicitly defining the $i+1$th furthest point insertion via some optimization problem, one needs to take care that the (farthest, in terms of sum of distances) solution does not turn out to equal one of the previously found i solutions, as this is a requirement for the furthest point insertion algorithm. This is an issue one faces because of the implicit nature of the furthest point procedure in the exponential-sized space of solutions: in the metric k-dispersion problem, it was easy to guarantee distinctness as one only considered the $n-i$ points not yet selected.

1.3 Reduction to Budget Constrained Optimization

Combining with dispersion, we reduce the diversity computational problem to a budget constrained optimization (BCO) problem where the budget is an upper (resp. lower) bound if the quality of solution is described by a minimization (resp. maximization) problem. Intuitively the budget guarantees the quality of the solution, and the objective function maximizes diversity. Recall that the number of solutions k and the approximation factor c is input by the user; a larger c allows for more diversity.

We show how using an (a, b) bi-approximation algorithm for the BCO problem provides a set of $O(a)$-diverse, bc approximately-optimal solutions to the diversity computational problem (the hidden constant is at most 4). This main reduction is described in Theorem 1.

The main challenge in transferring the bi-approximation results because of a technicality that we describe next. Let $\mathcal{S}(c)$ be the space of c approximate solutions. A $(*, b)$ bi-approximation algorithm to the BCO relaxes the budget constraint by a factor b, and hence only promises to return a faraway point in the larger space $\mathcal{S}(b \cdot c)$. Thus bi-approximation of BCO do not simply give a farthest point insertion in the space of solutions, and instead *return a point in a larger space*. Nevertheless, we prove that in most cases, one loses a factor of at most 4 in the approximation factor for the diversity.

Once the reduction to BCOs is complete, for diverse approximate matchings, spanning trees and shortest paths we exploit the special characteristics of the corresponding BCO to solve it optimally ($a = b = 1$). For other problems such as global min-cut, diverse approximate minimum weight spanning trees, and the more general minimum weight bases of a matroid, we utilize known bi-approximations to the BCO to obtain bi-approximations for the diversity problem. For all problems except diverse (unweighted) spanning trees[2], our algorithms are the first polynomial time bi-approximations for these problems.

We also connect to the wide literature on multicriteria optimization and show that our result applies to the entire class of problems for which the associated DUALRESTRICT problem (defined by Papadimitriou and Yannakakis [35], and recently studied by Herzel et al. [26]) has a polynomial time solution. We discuss this in more detail after presenting our reduction.

Layout: The rest of this paper is organized as follows: we survey related work in Sect. 2, and formulate the problem in Sect. 3. In Sect. 4 we mention the connection to dispersion and describe the reduction to the budget constrained optimization problem (Theorem 1). Sections 5, 6, 7 and 8 describe four applications of our technique to various problems such as diverse approximate matchings, global min-cuts, shortest paths, minimum spanning trees, and minimum weight bases of a matroid. We remark that this list is by no means exhaustive, and we leave finding other interesting optimization problems which are amenable to our approach for future research. **Due to space constraints, all proofs can be found in the publicly available full version of this paper at** [19].

2 Related Work

Recently there has been a surge of interest in the tractability of finding diverse solutions for a number of combinatorial optimization problems, such as spanning trees, minimum spanning trees, k-paths, shortest paths, k-matchings, etc. [16,17,21–23]. Most of the existing work focuses on finding diverse optimal solutions. In cases when finding the optimal solution is NP-complete, several works have focused on developing FPT algorithms [5,17]. Nevertheless, as pointed out in [22], it would be more practical to consider finding a set of diverse "short" paths rather than one set of diverse shortest paths. They show that finding a set of approximately shortest paths with the maximum diversity is NP-hard, but leave the question of developing approximation algorithms open, a question that we answer in our paper for several problems. Similarly the problem of finding diverse maximum matchings was proved to be NP-hard in [16]. We remark that the main difference between our result and previous work is that our algorithms can find a diverse set of c-approximate solutions in polynomial time. If the attained diversity is not sufficient for the application, the user can input a larger c, in hopes of increasing it.

Multicriteria Optimization: In this domain, several optimization functions are given on a space of solutions. Clearly, there may not be a single solution that is the best for all objective functions, and researchers have focused on obtaining Pareto-optimal solutions, which are solutions that are non-dominated by other solutions. Put differently, a solution is Pareto-optimal if no other solution

[2] While an exact algorithm for diverse unweighted spanning trees is known [23], we give a faster (by a factor $\Omega(n^{1.5}k^{1.5}/\alpha(n,m))$ where $\alpha(\cdot)$ denotes the inverse of the Ackermann function), 2-approximation here.

can have a better cost for all criteria. Since exact solutions are hard to find, research has focused on finding ϵ Pareto-optimal solutions, which are a $1+\epsilon$ factor approximations of Pareto-optimal solutions. Papadimitriou and Yannakakis [35] showed that under pretty mild conditions, any mutlicriteria optimization problem admits an ϵ Pareto-optimal set of fully polynomial cardinality. In terms of being able to *find* such an ϵ Pareto-optimal set, they show that a (FPTAS) PTAS exists for the problem if and only if an associated GAP problem can be solved in (fully) polynomial time. Very recently, Herzel et al. [26] study the class of problems for which an FPTAS or PTAS exists for finding ϵ Pareto-optimal solutions that are *exact* in one of the criteria. Such problems are a subset of the ones characterized by GAP. Herzel et al. [26] characterize the condition similarly: an FPTAS (PTAS) exists if and only if an associated DUALRESTRICT problem can be solved in (fully) polynomial time. For more details we refer the reader to the survey by Herzel at al. [27].

3 Diversity Computational Problem (DCP)

First, we define some notations. We use the definition of optimization problems given in [3] with additional formalism as introduced in [20].

Definition 1 (Optimization Problem). *An **optimization problem** Π is characterized by the following quadruple of objects $(I_\Pi, \mathsf{Sol}_\Pi, \Delta_\Pi, \mathsf{goal}_\Pi)$, where:*

- *I_Π is the set of instances of Π. In particular for every $n \in \mathbb{N}$, $I_\Pi(n)$ is the set of instances of Π of input size at most n (bits);*
- *Sol_Π is a function that associates to any input instance $x \in I_\Pi$ the set of feasible solutions of x;*
- *Δ_Π is the measure function[3], defined for pairs (x, y) such that $x \in I_\Pi$ and $y \in \mathsf{Sol}_\Pi(x)$. For every such pair (x, y), $\Delta_\Pi(x, y)$ provides a non-negative integer which is the value of the feasible solution y;*
- *$\mathsf{goal}_\Pi \in \{\min, \max\}$ specifies whether Π is a maximization or minimization problem.*

We would like to identify a subset of our solution space which are (approximately) optimal with respect to our measure function. To this effect, we define a notion of approximately optimal feasible solution.

Definition 2 (Approximately Optimal Feasible Solution). *Let $\Pi(I_\Pi, \mathsf{Sol}_\Pi, \Delta_\Pi, \mathsf{goal}_\Pi)$ be an optimization problem and let $c \geq 1$. For every $x \in I_\Pi$ and $y \in \mathsf{Sol}_\Pi(x)$ we say that y is a **c-approximate optimal solution of** x if for every $y' \in \mathsf{Sol}_\Pi(x)$ we have $\Delta_\Pi(x, y) \cdot c \geq \Delta_\Pi(x, y')$ if $\mathsf{goal}_\Pi = \max$ and $\Delta_\Pi(x, y) \leq \Delta_\Pi(x, y') \cdot c$ if $\mathsf{goal}_\Pi = \min$.*

[3] We define the measure function only for feasible solutions of an instance. Indeed if an algorithm solving the optimization problem outputs a non-feasible solution, then the measure just evaluates to -1 in case of maximization problems and ∞ in case of minimization problems.

Definition 3 (Computational Problem). *Let $\Pi(I_\Pi, \mathsf{Sol}_\Pi, \Delta_\Pi, \mathsf{goal}_\Pi)$ be an optimization problem and let $\lambda : \mathbb{N} \to \mathbb{N}$. The* ***computational problem associated with*** *(Π, λ) is given as input an instance $x \in I_\Pi(n)$ (for some $n \in \mathbb{N}$) and real $c := \lambda(n) \geq 1$ find a c-approximate optimal feasible solution of x.*

Definition 4 (DCP - Diversity Computational Problem). *Let $\Pi(I_\Pi, \mathsf{Sol}_\Pi, \Delta_\Pi, \mathsf{goal}_\Pi)$ be an optimization problem and let $\lambda : \mathbb{N} \to \mathbb{N}$. Let $\sigma_{\Pi,t}$ be a diversity measure that maps every t feasible solutions of an instance of I_Π to a non-negative real number. The* ***diversity computational problem associated with*** *$(\Pi, \sigma_{\Pi,t}, k, \lambda)$ is given as input an instance $x \in I_\Pi(n)$ (for some $n \in \mathbb{N}$), an integer $k := k(n)$, and real $c := \lambda(n) \geq 1$, find k-many c-approximate solutions $y_1, \ldots, y_k$ to x which maximize the value of $\sigma_{\Pi,k}(x, y_1, \ldots, y_k)$.*

Proposition 1. *Let $\Pi(I_\Pi, \mathsf{Sol}_\Pi, \Delta_\Pi, \mathsf{goal}_\Pi)$ be an optimization problem and let $\lambda : \mathbb{N} \to \mathbb{N}$. Let $\sigma_{\Pi,t}$ be a diversity measure that maps every t feasible solutions of an instance of I_Π to a non-negative real number. If the computational problem associated with (Π, λ) is $\mathcal{NP}$-hard, then the diversity computational problem associated with $(\Pi, \sigma_{\Pi,t}, \lambda)$ also is $\mathcal{NP}$-hard.*

Therefore the interesting questions arise when we compute problems associated with (Π, λ) which are in $\mathcal{P}$, or even more when, $(\Pi, \mathbb{1})$ is in $\mathcal{P}$ where $\mathbb{1}$ is the constant function which maps every element of the domain to 1. For the remainder of this paper, we will consider $\lambda(n)$ to be the constant function, and will simply refer to the constant as c.

Finally, we define bicriteria approximations for the diversity computational problem:

Definition 5 ((α, β) Bi-approximation for the Diversity Computational Problem). *Consider the diversity computational problem associated with $(\Pi, \sigma_{\Pi,t}, k, c)$, and a given instance $x \in I_\Pi(n)$ (for some $n \in \mathbb{N}$). An algorithm is called an (α, β) bi-approximation for the diversity computational problem if it outputs k feasible solutions $y_1, \ldots, y_k$ such that a) y_i is a $\beta{\cdot}c$-approximate optimal feasible solution to x for all $1 \leq i \leq k$, and b) for any set $y_1^{'}, \ldots, y_k^{'}$ of k-many c-approximate optimal feasible solutions, $\sigma_{\Pi,k}(y_1, \cdots, y_k) \cdot \alpha \geq \sigma_{\Pi,k}(y_1^{'}, \cdots, y_k^{'})$. Furthermore, such an algorithm is said to run in polynomial time if the running time is polynomial in n and k.*

4 The Reduction: Enter Dispersion and Biobjective Optimization

As stated in the introduction, our problems are related to the classical dispersion problem in a metric space. Here we state the dispersion problem and use dispersion to reduce the problem of finding diverse, approximately optimal solutions to solving an associated budget constrained optimization problem.

4.1 Dispersion Problem

Definition 6 (k-Dispersion, Total Distance). *Given a finite set of points P whose pairwise distances satisfy the triangle inequality and an integer $k \geq 2$, find a set $S \subseteq P$ of cardinality k so that $W(S)$ is maximized, where $W(S)$ is the sum of the pairwise distances between points in S.*

The main previous work on the k-dispersion problem relevant to us is [37], where the problem was named as Maximum-Average Facility Dispersion problem with triangle inequality (MAFD-TI). The problems are equivalent as maximizing the average distance between the points also maximizes the sum of pairwise distances between them and vice-versa.

The k-dispersion problem is $\mathcal{NP}$-hard, but one can find a set S whose $W(S)$ is at least a constant factor of the maximum possible in polynomial time by a greedy procedure [37]. We call the greedy procedure *furthest insertion*. It works as follows. Initially, let S be a singleton set that contains an arbitrary point from the given set. While $|S| < k$, add to S a point $x \notin S$ so that $W(S \cup \{x\}) \geq W(S \cup \{y\})$ for any $y \notin S$. Repeat the greedy addition until S has size k. The final S is a desired solution, which is shown to be a 4-approximation in [37]. It is worth noting that the furthest insertion in [37] initializes S as a furthest pair of points in the given set, and the above change does not worsen the approximation factor. In a later paper [8], the greedy algorithm of choosing an arbitrary initial point is shown to be a 2-approximation, which is a tight bound for this algorithm [7].

Lemma 1 (Furthest Insertion in [8,37]). *The k-dispersion problem can be 2-approximated by the furthest insertion algorithm.*

The running time of the furthest insertion algorithm is polynomial in $|S|$ (the size of S), as it performs k iterations, each performing at most $O(k|S|)$ distance computations/lookups. Note that in our case S is the collection of objects of a certain type (matchings, paths, trees, etc.). Hence the size of our metric space is typically exponential in $|V|$ and $|E|$. This adds a new dimension of complexity to the traditional dispersion problems studied.

4.2 Reduction to Budget Constrained Optimization

Recall the definitions of the Diversity Computational Problem (Definition 4) and (a, b) bi-approximations (Definition 5). As the input instance $x \in I_\Pi$ will be clear from context, we drop the dependence on x, and assume a fixed input instance to a computational problem. Thus Sol_Π will denote the set of feasible solutions, and $\Delta_\Pi(y)$ the measure of the feasible solution y.

Diversity and Similarity Measures from Metrics. Let $d : \mathsf{Sol}_\Pi \times \mathsf{Sol}_\Pi \to \mathbb{R}^+$ be a metric on the space of feasible solutions. When such a metric is available, we will consider the diversity function $\sigma_{\Pi,t} : \mathsf{Sol}_\Pi \times \cdots \times \mathsf{Sol}_\Pi \to \mathbb{R}^+$ that assigns the diversity measure $\sum_{i,j} d(y_i, y_j)$ to a t-tuple of feasible solutions $(y_1, \cdots, y_t)$. Also, given such a metric d, define D to be the diameter of Sol_Π under d, i.e.,

$D = \max_{y,y' \in \mathsf{Sol}_\Pi} d(y, y')$. In many cases, we will be interested in the *similarity* measure $s_{\Pi,t}$ defined by $s_{\Pi,t}(y_1, \cdots, y_t) = \sum_{i,j}(D - d(y_i, y_j))$. The examples the reader should keep in mind are graph objects such as spanning trees, matchings, shortest paths, Hamiltonian circuits, etc., such that $d(y, y')$ denotes the Hamming distance, a.k.a. size of the symmetric difference of the edge sets of y and y', and s denotes the size of their intersection.

In the remainder of the paper we consider the above total distance (resp. similarity) diversity measures $\sigma_{\Pi,t}$ arising from the metric d (resp. similarity measure s), and we will parameterize the problem by d (resp. s) instead.

Definition 7 (Budget Constrained Optimization). *Given an instance of a computational problem Π, a constant $c \geq 1$, and a set $\{y_1, \ldots, y_i\}$ of feasible solutions in Sol_Π, define the* ***metric budget constrained optimization problem*** *$BCO(\Pi, (y_1, \ldots, y_i), c, d)$ as follows:*

- *If $\mathsf{goal}_\pi = \min$, define $\Delta^* := \min_{y \in \mathsf{Sol}_\Pi} \Delta_\Pi(y)$. Then $BCO(\Pi, (y_1, \ldots, y_i), c, d)$ is the problem*

$$\begin{aligned} \max \quad & f_d(y) := \sum_{j=1}^{i} d(y, y_j) \\ s.t. \quad & \Delta_\Pi(y) \leq c \cdot \Delta^* \\ & y \in \mathsf{Sol}_\Pi \backslash \{y_1, \ldots, y_i\} \end{aligned} \tag{1}$$

- *If $\mathsf{goal}_\pi = \max$, define $\Delta^* := \max_{y \in \mathsf{Sol}_\Pi} \Delta_\Pi(y)$. Then $BCO(\Pi, (y_1, \ldots, y_i), c, d)$ is the problem*

$$\begin{aligned} \max \quad & f_d(y) := \sum_{j=1}^{i} d(y, y_j) \\ s.t. \quad & \Delta_\Pi(y) \cdot c \geq \Delta^* \\ & y \in \mathsf{Sol}_\Pi \backslash \{y_1, \ldots, y_i\} \end{aligned} \tag{2}$$

- *Given a similarity measure s, define the* ***similarity budget constrained optimization problem*** *$BCO(\Pi, (y_1, \ldots, y_i), c, s)$ with the same constraint set as above (depending on goal_π), but with the objective function changed to $g_s(y) := \min \sum_{j=1}^{i} s(y, y_j)$ instead of $\max \sum_{j=1}^{i} d(y, y_j)$.*

Definition 8 (Bi-approximation to BCO). *An algorithm for an associated BCO is called an (a, b) bi-approximation algorithm if for any $1 \leq i \leq k$, it outputs a solution y such that the following holds.*

- *If $\mathsf{goal}_\Pi = \min$ and the associated BCO is $BCO(\Pi, (y_1, \ldots, y_i), c, d)$, then a) $y \in \mathsf{Sol}_\Pi \backslash \{y_1, \cdots, y_i\}$, b) $\Delta_\Pi(y) \leq b \cdot c \cdot \Delta^*$, and c) for all y' satisfying the constraints of $BCO(\Pi, (y_1, \ldots, y_i), c, d)$, $f_d(y) \cdot a \geq f_d(y')$.*
- *If $\mathsf{goal}_\Pi = \max$ and the associated BCO is $BCO(\Pi, (y_1, \ldots, y_i), c, d)$, then a) $y \in \mathsf{Sol}_\Pi \backslash \{y_1, \cdots, y_i\}$, b) $\Delta_\Pi(y) \cdot b \cdot c \geq \Delta^*$, and c) for all y' satisfying the constraints of $BCO(\Pi, (y_1, \ldots, y_i), c, d)$, $f_d(y) \cdot a \geq f_d(y')$.*

- *If* $\mathsf{goal}_\Pi = \min$ *and the associated BCO is* $BCO(\Pi, (y_1, \ldots, y_i), c, s)$, *then a)* $y \in \mathsf{Sol}_\Pi \backslash \{y_1, \cdots, y_i\}$, *b)* $\Delta_\Pi(y) \le b \cdot c \cdot \Delta^*$, *and c) for all* y' *satisfying the constraints of* $BCO(\Pi, (y_1, \ldots, y_i), c, s)$, $g_s(y) \le g_s(y') \cdot a$.
- *If* $\mathsf{goal}_\Pi = \max$ *and the associated BCO is* $BCO(\Pi, (y_1, \ldots, y_i), c, s)$, *then a)* $y \in \mathsf{Sol}_\Pi \backslash \{y_1, \cdots, y_i\}$, *b)* $\Delta_\Pi(y) \cdot b \cdot c \ge \Delta^*$, *and c) for all* y' *satisfying the constraints of* $BCO(\Pi, (y_1, \ldots, y_i), c, s)$, $g_s(y) \le g_s(y') \cdot a$.

Remark: Minimization and maximization are essentially equivalent (by changing the sign), and so optimally solving one solves the other. The reason why we continue to treat them separately is because obtaining *an approximation* to minimizing total similarity $g_s(y) := \sum_{j=1}^{i} s(y, y_i)$ is not equivalent to an approximation to maximizing total distance $f_d(y) := \sum_{j=1}^{i} d(y, y_i)$– in fact, these functions are the "opposite" of each other, as $f_d(y) = Di - g_s(y)$.

We are now ready to state our main theorem.

Theorem 1 (Reduction of DCP to BCO). *Consider an input* (Π, k, d, c) *to the diversity computational problem (DCP).*

- *For metric BCO,*
  1. *An* $(a, 1)$ *bi-approximation to* $BCO(\Pi, (y_1, \ldots, y_i), c, d)$ *can be used to give a* $(2a, 1)$ *bi-approximation to the DCP, and*
  2. *An* (a, b) *bi-approximation to* $BCO(\Pi, (y_1, \ldots, y_i), c, d)$ *can be used to give a* $(4a, b)$ *bi-approximation to the DCP.*
- *For similarity BCO,*
  3. *A* $(1, 1)$ *bi-approximation to* $BCO(\Pi, (y_1, \ldots, y_i), c, s)$ *can be used to give a* $(2, 1)$ *bi-approximation to the DCP,*
  4. *A* $(1, b)$ *bi-approximation to* $BCO(\Pi, (y_1, \ldots, y_i), c, s)$ *can be used to give* $(4, b)$ *bi-approximation to the DCP,*
  5. *A* $(1 + \epsilon, 1)$ *bi-approximation to* $BCO(\Pi, (y_1, \ldots, y_i), c, s)$ *can be used to give* $(4, 1)$ *bi-approximation to the DCP, under the condition that the average pairwise distance in the optimal solution to the DCP is at least* $D\frac{4\epsilon}{1+2\epsilon}$.

In all of the above, the overhead for obtaining a bi-approximation for the DCP, given a bi-approximation for BCO problem, is $O(k)$.

A few remarks are in order:

- The above theorem provides a recipe for solving the diversity computational problem for any given optimization problem. As long as *either* the metric or the similarity budget constrained optimization problems can be solved or approximated in polynomial time, one has an analogous result for the DCP.
- In the remainder of this paper we will see several applications that follow from the above 5 "types" of bi-approximations available. These include DCP for Maximum Matching and Global Min-Cut (Type 1), DCP for shortest path (Type 3), DCP for minimum weight bases of a matroid, minimum spanning trees (Types 4 and 5).

– Whenever either a or b (or both) is set to be $1+\epsilon$, we call a bi-approximation for the BCO problem an FPTAS if the running time is polynomial in $1/\epsilon$ in addition to being polynomial in d and k. Otherwise we call it a PTAS.

Relation to Multicriteria Optimization: Observe that for similarity BCOs, we need either a or b to be 1. This class of biobjective problems that have a PTAS that is exact in one of the criteria is a special case of the multicriteria problems that have a PTAS that is exact in one of the criteria. Herzel et al. [26] showed that this class is exactly the class of problems for which the DUALRESTRICT version of the problem, posed by Diakonikolas and Yannakakis [14]), can be solved in polynomial time. These are also the class of problems having a polynomial-time computable approximate ϵ-Pareto set that is exact in one objective. This equivalence means that our theorem is applicable to this entire class of problems.

4.3 Relaxed BCOs and Self-avoidance

Before we delve into our applications, we describe another challenge in directly applying results from multicriteria optimization literature. For a BCO, the second constraint demands that $y \in \mathsf{Sol}_{\Pi} \backslash \{y_1, \cdots, y_i\}$. Intuitively y is the farthest point to the set of already discovered solutions $\{y_1, \cdots, y_i\}$, and because it is defined implicitly, without the second constraint y may equal one of the y_j $(1 \leq j \leq i)$. Consider an alternate BCO, which we call BCO^r where the constraint is relaxed to $y \in \mathsf{Sol}_{\Pi}$. For many graph problems, solving BCO^r combined with the approach by Lawler [30] gives a solution to the original BCO. This is extremely useful because most of the literature on multicriteria optimization concerns optimization of the relaxed type of problems BCO^r, and one can borrow results derived before without worrying about the second constraint. We remark that for other problems, k-best enumeration algorithms (see [18,24,30,31,33] for examples) may be useful to switch from the BCO to its relaxed version. *Thus any algorithm for BCO^r can be used, modulo the self-avoiding constraint (to be handled using Lawler's approach), to give a polynomial time algorithm for the Diversity Computational Problem with the same guarantees as in Theorem* 1. We provide examples of the approach by Lawler in subsequent sections where we consider specific problems.

5 Application 1: Diverse Spanning Trees

In this section, we discuss the diverse spanning trees problem, which is the diversity computational problem for spanning trees with Hamming distance function as the diversity measure. Let $G = (V, E)$ be an n-node m-edge undirected graph. The problem aims to output a set S of k spanning trees $T_1, \cdots, T_k$ of G such that the sum of the pairwise distances $\sum_{i,j \in S} d(T_i, T_j)$ is maximized, where d is the Hamming distance between the edge sets of the trees. While this problem actually has an exact algorithm running in time $O((kn)^{2.5}\ m)$ [23], we get a faster approximation algorithm.

Theorem 2. *Given an n-node m-edge undirected graph $G = (V, E)$, there exists an $O(knm \cdot \alpha(n, m))$-time algorithm, where $\alpha(\cdot)$ is the inverse of the Ackermann function, that generates k spanning trees $T_1, \cdots, T_k$, such that the sum of all pairwise Hamming distances is at least half of an optimal set of k diverse spanning trees.*

We prove the above theorem by developing an exact $(1, 1)$ polynomial time subroutine for the associated BCO problem. The proof can be found in the full version.

6 Application 2: Diverse Approximate Shortest Paths

Given a graph $G = (V, E)$, non-negative edge weights $w(e)$, two vertices s and t, and a factor $c > 1$, the diversity computational problem asks to output k many st paths, such that the weight of each path is within a factor c of the weight of the shortest st path, and subject to this constraint, the total pairwise distance between the paths is maximized. Here the distance between two paths is again the Hamming distance, or size of symmetric difference of their edge sets.

In [22], it is shown that finding k shortest paths with the maximum diversity (i.e. the average Hamming distance between solutions) can be solved in polynomial time, but finding k "short" paths with the maximum diversity is NP-hard. In contrast, in what follows, we will show that finding k "short" paths with constant approximate diversity is polynomial-time solvable.

We will show that the associated budget constrained optimization problem for this is of Type 3 in Theorem 1. In other words, we will show that the BCO can be solved exactly. This will result in a $(2, 1)$ approximation algorithm for the diversity computational problem.

Hence, we need an algorithm that implements: given a set S of c-approximate shortest st-paths, find a c-approximate shortest st-path $P \notin S$ so that $W(S \cup \{P\})$ is maximum among all $W(S \cup \{P'\})$ for c-approximate shortest st-path $P' \notin S$. Here, $W(S')$ is the sum of all pairwise Hamming distances between two elements in S'. This is a special case of the bicriteria shortest paths, for which there is an algorithm in [34]. In our case, **one of the two weight functions is an integral function with range bounded in** $[0, k]$. Hence, it can be solved more efficiently than the solution in [34], which can be summarized as following.

Lemma 2 (Exact solution to the relaxed BCO^r problem). *Given a real $c \geq 1$ and a directed simple graph $G = (V \cup \{s, t\}, E)$ associated with two weight functions on edges $\omega : E \to \mathbb{R}^+$ and $f : E \to \{0, 1, \ldots, r\}$, there is an $O(r|V|^3)$-time algorithm to output an st-path P^* so that $\sum_{e \in E(P^*)} f(e)$ is minimized while retaining $\sum_{e \in E(P^*)} \omega(e) \leq c \sum_{e \in E(P)} \omega(e)$ for all st-paths P.*

Self-avoiding Constraint. We now turn to solving the associated (non-relaxed) *BCO* problem, by generalizing the above lemma to Corollary 1. Thus Corollary 1 will help us avoid the situation that a furthest insertion returns a path that is already picked by some previous furthest insertion.

Corollary 1 (Exact solution to the BCO **problem).** *Given a real* $c \geq 1$*, a directed simple graph* $G = (V \cup \{s,t\}, E)$ *associated with two weight functions on edges* $\omega : E \to \mathbb{R}^+$*,* $f : E \to \{0, 1, \ldots, r\}$*, and two disjoint subsets of edges* $E_{in}, E_{ex} \subseteq E$ *so that all edges in* E_{in} *together form a directed simple path* P_{prefix} *starting from node* s*, there exists an* $O(r|V|^3)$*-time algorithm to output an* c*-approximate shortest* st*-path* P^* *under* ω *so that* $\sum_{e \in E(P^*)} f(e)$ *is minimum among all the* c*-approximate shortest* st*-paths* P *that contain* P_{prefix} *as a prefix and contain no edges from* E_{ex}*, if such an* c*-approximate shortest* st*-path exists.*

We are ready to state our main result for the diverse c-approximate shortest st-paths.

Theorem 3 ($(2,1)$ **Bi-approximation to the Diversity Problem on Shortest Paths).** *For any directed simple graph* $G = (V \cup \{s,t\}, E)$*, given a constant* $c > 1$ *and an integer* $k \in \mathbb{N}$*, there exists an* $O(k^3|V|^4)$*-time algorithm that, if* G *contains at least* k *distinct* c*-approximate shortest* st*-paths, computes a set* S *of* k *distinct* c*-approximate shortest* st*-paths so that the sum of all pairwise Hamming distances between two paths in* S *is at least one half of the maximum possible; otherwise, reports "Non-existent".*

7 Application 3: Diverse Approximate Maximum Matchings, and Global Min-Cut

Consider the diversity computational problem for computing k many c-approximate maximum matchings for undirected graphs. In [16], the authors present an algorithm, among others, to find a pair of *maximum* matchings for *bipartite* graphs whose Hamming distance is maximized. In contrast, our result can be used to find $k \geq 2$ *approximate maximum* matchings for *any graph* whose diversity (i.e. the average Hamming distance) approximates the largest possible by a factor of 2.

We show that this problem can be restated into the budgeted matching problem [6]. As noted in [6], though the budgeted matching is in general $\mathcal{NP}$-hard, if both the weight and cost functions are integral and have a range bounded by a polynomial in $|V|$, then it can be solved in polynomial time with a good probability by a reduction to the exact perfect matching problem [9,32]. The exact running time for such a case is not stated explicitly in [6]. We combine the algorithm in [6] and the approach by Lawler [30] to prove:

Theorem 4. *There exists a* $O(k^4|V|^7 \log^3 k|V|)$ *time,* $(2,1)$ *bi-approximation to the diversity computational problem for* c*-approximate maximum matchings, with failure probability* $1/|V|^{\Omega(1)}$*.*

DCP for Global Min-Cuts: Next, consider the diversity computational problem for computing k many c-approximate global min-cuts: given a graph G and a positive weight function w on its edges, a c-approximate min-cut is a cut C whose cut-edge set $E(C)$ satisfies $\sum_{e \in E(C)} w(e) \leq c \sum_{e \in E(C')} w(e)$ for any other cut C'. Given i cuts, we define the (integral) cost of an edge as the number of cuts in which it appears as a cut edge. Consider the BCO with cost minimization in the objective function (as the cost of a cut is now inversely proportional to its sum of distances from the found cuts) and constraint with upper bound (the weight of the cut should be at most c times that of a global min weight cut). In [2] the authors provide a polynomial-time algorithm for this problem, implying that the BCO can be solved exactly in polynomial time. This gives us a $(2, 1)$ bi-approximation to the diversity computational problem for c-approximate global minimum cuts. We remark that one may be able to exploit integrality of our cost function to obtain a faster algorithm than that in [2].

8 Application 4: Diverse Minimum Weight Matroid Bases and Minimum Spanning Trees

One of the original ways to attack the peripatetic salesman problem (Krarup [28]) was to study the k edge-disjoint spanning trees problem [12]. Note that the existence of such trees is not guaranteed, and one can use our results in Sect. 5 to maximize diversity of the k trees found.

However, for an application to the TSP problem, cost conditions must be taken into account. Here we study the diverse computational problem (DCP) on minimum spanning trees: Given a weighted undirected graph $G = (V, E)$ with nonnegative weights $w(e)$, $c > 1$ and a $k \in \mathbb{N}$, return k spanning trees of G such that each spanning tree is a c-approximate minimum spanning tree, and subject to this, the diversity of the k trees is maximized. Here again the diversity of a set of trees is the sum of pairwise distances between them, and the distance between two trees is the size of their symmetric difference.

Our results in this section generalize to the problem of finding k diverse bases of a matroid such that every basis in the solution set is a c approximate minimum-weight basis. The DCP on MSTs is a special case of this problem. However, in order to not introduce extra notation and definitions here, we will describe our method for minimum spanning trees. We will then briefly sketch how to extend the algorithm to the general matroid case.

Starting with $T_1 = MST(G)$ (a minimum spanning tree on G, computable in polynomial time), assume we have obtained i trees $T_1, \cdots, T_i$, all of which are c-approximate minimum spanning trees. Assign to each edge a length $\ell(e)$ which equals $|\{j : 1 \leq j \leq i, e \in T_j\}|$.

Lemma 3. *Given $T_1, \cdots, T_i$, finding T_{i+1} that maximizes $\sum_{j=1}^{i} d(T, T_j)$ is equivalent to finding T that minimizes $\sum_{e \in T} \ell(e)$.*

Proof. An explicit calculation reveals that $\sum_{e \in T} \ell(e) = (n-1)i - \sum_{j=1}^{i} d(T, T_j)$.

Consider now the associated *similarity budget constrained optimization problem*

$$\begin{aligned} \min \quad & \sum_{e \in T} \ell(e) \\ \text{s.t.} \quad & w(T) \leq c \cdot w(MST(G)) \\ & T \in \mathsf{Sol}_{\Pi} \backslash \{T_1, \ldots, T_i\} \end{aligned} \tag{3}$$

Here Sol_{Π} is just the set of spanning trees on G. We will handle the self-avoiding constraints in a similar fashion as in Sect. 5. For the moment consider the relaxed BCO^r where the last constraint is simply $T \in \mathsf{Sol}_{\Pi}$. This is a budget constrained MST with two weights. This problem has been considered by Ravi and Goemans [36], who termed it the CMST problem. They provide a $(1, 2)$ bi-approximation that runs in near-linear time, and a $(1, 1+\epsilon)$ bi-approximation that runs in polynomial time[4]. Also, they show that the $(1, 1+\epsilon)$ bi-approximation can be used as a subroutine to compute a $(1+\epsilon, 1)$ bi-approximation in pseudopolynomial time.

Applying their results and observing that we are in cases 4 and 5 of Theorem 1, we get

Theorem 5 (DCP for Mininum Spanning Trees). *There exists a*

- *polynomial (in n, m and k) time algorithm that outputs a* $(4, 2)$ *bi-approximation to the DCP problem for MSTs.*
- *polynomial (in n, m and k) and exponential in* $1/\epsilon$ *time algorithm that outputs a* $(4, 1+\epsilon)$ *bi-approximation to the DCP problem for MSTs.*
- *pseudopolynomial time algorithm that outputs a* $(4, 1)$ *bi-approximation to the DCP problem for MSTs, as long as the average distance between the trees in the optimal solution to the k DCP on c-approximate minimum spanning trees does not exceed* $\frac{4\epsilon(n-1)}{1+2\epsilon}$.

Extension to Matroids: It is stated in the paper by Ravi and Goemans [36] that the same result holds if one replaces the set of spanning trees by the bases of any matroid. It is straightforward to show that the analog of Lemma 3 hold in the matroid setting too. With a bit of work, one can also generalize the approach of Lawler [30] to avoid self-intersection (the bases found so far), and thus all the techniques generalize to the matroid setting. In all of this, we assume an independence oracle for the matroid, as is standard. In [17], it is shown that, given integers k, d, finding k *perfect* matchings so that every pair of the found matchings have Hamming distance at least d is NP-hard. This hardness result also applies to finding weighted diverse bases and weighted diverse common independent sets.

Acknowledgements. Mayank Goswami would like to acknowledge support from the National Science Foundation grants CRII-1755791 and CCF-1910873. Jie Gao would like to acknowledge support OAC-1939459, CCF-2118953 and CCF-2208663. Karthik C. S. would like to acknowledge support from Rutgers University's Research Council

[4] The latter is a PTAS, not an FPTAS.

Individual Fulcrum Award (#AWD00010234) and a grant from the Simons Foundation, Grant Number 825876, Awardee Thu D. Nguyen. Meng-Tsung Tsai would like to acknowledge support from the Ministry of Science and Technology of Taiwan grant 109-2221-E-001-025-MY3.

References

1. Abbar, S., Amer-Yahia, S., Indyk, P., Mahabadi, S., Varadarajan, K.R.: Diverse near neighbor problem. In: Symposium on Computational Geometry (SoCG), pp. 207–214. ACM (2013)
2. Armon, A., Zwick, U.: Multicriteria global minimum cuts. Algorithmica **46**(1), 15–26 (2006)
3. Ausiello, G., Marchetti-Spaccamela, A., Crescenzi, P., Gambosi, G., Protasi, M., Kann, V.: Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties. Springer, Heidelberg (1999). https://doi.org/10.5555/554706
4. Baste, J., et al.: Diversity of solutions: an exploration through the lens of fixed-parameter tractability theory. Artif. Intell. **303**, 103644 (2022)
5. Baste, J., Jaffke, L., Masařík, T., Philip, G., Rote, G.: FPT algorithms for diverse collections of hitting sets. Algorithms **12**(12), 254 (2019)
6. Berger, A., Bonifaci, V., Grandoni, F., Schäfer, G.: Budgeted matching and budgeted matroid intersection via the gasoline puzzle. Math. Program. **128**(1–2), 355–372 (2011)
7. Birnbaum, B.E., Goldman, K.J.: An improved analysis for a greedy remote-clique algorithm using factor-revealing LPs. Algorithmica **55**(1), 42–59 (2009)
8. Borodin, A., Jain, A., Lee, H.C., Ye, Y.: Max-sum diversification, monotone submodular functions, and dynamic updates. ACM Trans. Algorithms **13**(3), 41:1–41:25 (2017)
9. Camerini, P.M., Galbiati, G., Maffioli, F.: Random pseudo-polynomial algorithms for exact matroid problems. J. Algorithms **13**(2), 258–273 (1992)
10. Cevallos, A., Eisenbrand, F., Zenklusen, R.: Max-sum diversity via convex programming. In: Fekete, S.P., Lubiw, A. (eds.) 32nd International Symposium on Computational Geometry, SoCG 2016, 14–18 June 2016, Boston, MA, USA. LIPIcs, vol. 51, pp. 26:1–26:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
11. Cevallos, A., Eisenbrand, F., Zenklusen, R.: Local search for max-sum diversification. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 130–142. SIAM (2017)
12. Clausen, J., Hansen, L.A.: Finding k edge-disjoint spanning trees of minimum total weight in a network: an application of matroid theory. In: Rayward-Smith, V.J. (ed.) Combinatorial Optimization II. Mathematical Programming Studies, vol. 13, pp. 88–101. Springer, Heidelberg (1980). https://doi.org/10.1007/BFb0120910
13. Commander, C.W., Pardalos, P.M., Ryabchenko, V., Uryasev, S., Zrazhevsky, G.: The wireless network jamming problem. J. Comb. Optim. **14**(4), 481–498 (2007)
14. Diakonikolas, I., Yannakakis, M.: Small approximate pareto sets for biobjective shortest paths and other problems. SIAM J. Comput. **39**(4), 1340–1371 (2010)
15. Erkut, E.: The discrete P-dispersion problem. Eur. J. Oper. Res. **46**(1), 48–60 (1990). https://doi.org/10.1016/0377-2217(90)90297-O, https://www.sciencedirect.com/science/article/pii/037722179090297O

16. Fomin, F.V., Golovach, P.A., Jaffke, L., Philip, G., Sagunov, D.: Diverse pairs of matchings. In: 31st International Symposium on Algorithms and Computation (ISAAC). LIPIcs, vol. 181, pp. 26:1–26:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)
17. Fomin, F.V., Golovach, P.A., Panolan, F., Philip, G., Saurabh, S.: Diverse collections in matroids and graphs. In: Bläser, M., Monmege, B. (eds.) 38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, 16–19 March 2021, Saarbrücken, Germany (Virtual Conference). LIPIcs, vol. 187, pp. 31:1–31:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021)
18. Gabow, H.N.: Two algorithms for generating weighted spanning trees in order. SIAM J. Comput. **6**(1), 139–150 (1977)
19. Gao, J., Goswami, M., Karthik, C.S., Tsai, M.T., Tsai, S.Y., Yang, H.T.: Obtaining approximately optimal and diverse solutions via dispersion (2022). https://doi.org/10.48550/ARXIV.2202.10028, https://arxiv.org/abs/2202.10028
20. Goldenberg, E., Karthik C. S.: Hardness amplification of optimization problems. In: Vidick, T. (ed.) 11th Innovations in Theoretical Computer Science Conference, ITCS 2020, 12–14 January 2020, Seattle, Washington, USA. LIPIcs, vol. 151, pp. 1:1–1:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPIcs.ITCS.2020.1
21. Hanaka, T., Kiyomi, M., Kobayashi, Y., Kobayashi, Y., Kurita, K., Otachi, Y.: A framework to design approximation algorithms for finding diverse solutions in combinatorial problems. CoRR abs/2201.08940 (2022)
22. Hanaka, T., Kobayashi, Y., Kurita, K., Lee, S.W., Otachi, Y.: Computing diverse shortest paths efficiently: a theoretical and experimental study. CoRR abs/2112.05403 (2021)
23. Hanaka, T., Kobayashi, Y., Kurita, K., Otachi, Y.: Finding diverse trees, paths, and more. In: Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI), pp. 3778–3786. AAAI Press (2021)
24. Hara, S., Maehara, T.: Enumerate lasso solutions for feature selection. In: Singh, S.P., Markovitch, S. (eds.) Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI), pp. 1985–1991. AAAI Press (2017)
25. Hassin, R., Rubinstein, S., Tamir, A.: Approximation algorithms for maximum dispersion. Oper. Res. Lett. **21**(3), 133–137 (1997)
26. Herzel, A., Bazgan, C., Ruzika, S., Thielen, C., Vanderpooten, D.: One-exact approximate pareto sets. J. Global Optim. **80**(1), 87–115 (2021)
27. Herzel, A., Ruzika, S., Thielen, C.: Approximation methods for multiobjective optimization problems: a survey. INFORMS J. Comput. **33**(4), 1284–1299 (2021)
28. Krarup, J.: The peripatetic salesman and some related unsolved problems. In: Roy, B. (ed.) Combinatorial Programming: Methods and Applications. NATO Advanced Study Institutes Series, vol. 19, pp. 173–178. Springer, Dordrecht (1995). https://doi.org/10.1007/978-94-011-7557-9_8
29. Kuby, M.: Programming models for facility dispersion: the P-dispersion and maxisum dispersion problems. Geogr. Anal. **19**(4), 315–329 (1987). https://doi.org/10.1111/j.1538-4632.1987.tb00133.x
30. Lawler, E.L.: A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. Manage. Sci. **18**(7), 401–405 (1972)
31. Lindgren, E.M., Dimakis, A.G., Klivans, A.: Exact map inference by avoiding fractional vertices. In: International Conference on Machine Learning, pp. 2120–2129. PMLR (2017)

32. Mulmuley, K., Vazirani, U.V., Vazirani, V.V.: Matching is as easy as matrix inversion. Comb. **7**(1), 105–113 (1987)
33. Murty, K.G.: An algorithm for ranking all the assignments in order of increasing cost. Oper. Res. **16**(3), 682–687 (1968)
34. Namorado Climaco, J.C., Queirós Vieira Martins, E.: A bicriterion shortest path algorithm. Eur. J. Oper. Res. **11**(4), 399–404 (1982)
35. 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)
36. Ravi, R., Goemans, M.X.: The constrained minimum spanning tree problem. In: Karlsson, R., Lingas, A. (eds.) SWAT 1996. LNCS, vol. 1097, pp. 66–75. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61422-2_121
37. Ravi, S.S., Rosenkrantz, D.J., Tayi, G.K.: Heuristic and special case algorithms for dispersion problems. Oper. Res. **42**(2), 299–310 (1994)
38. Wang, D., Kuo, Y.S.: A study on two geometric location problems. Inf. Process. Lett. **28**(6), 281–286 (1988) https://doi.org/10.1016/0020-0190(88)90174-3, https://www.sciencedirect.com/science/article/pii/0020019088901743
39. Yang, H.T., Tsai, S.Y., Liu, K.S., Lin, S., Gao, J.: Patrol scheduling against adversaries with varying attack durations. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1179–1188 (2019)