

# Revisiting $k$ -sum optimization

J. Puerto<sup>1</sup> · A. M. Rodríguez-Chía<sup>2</sup> · A. Tamir<sup>3</sup>

Received: 3 February 2016 / Accepted: 22 November 2016 / Published online: 16 December 2016  
© Springer-Verlag Berlin Heidelberg and Mathematical Optimization Society 2016

**Abstract** In this paper, we address continuous, integer and combinatorial  $k$ -sum optimization problems. We analyze different formulations of this problem that allow to solve it through the minimization of a relatively small number of minisum optimization problems. This approach provides a general tool for solving a variety of  $k$ -sum optimization problems and at the same time, improves the complexity bounds of many ad-hoc algorithms previously reported in the literature for particular versions of this problem. Moreover, the results developed for  $k$ -sum optimization have been extended to the more general case of the convex ordered median problem, improving upon existing solution approaches.

**Mathematics Subject Classification** 90B80 · 90B85 · 90C10 · 90C27 · 90C35

## 1 Introduction

The concept of  $k$ -sum optimization is a natural extension of the standard minisum optimization that looks for a feasible solution of a problem with the property that it minimizes the sum of the  $k$  largest components of an  $n$ -dimensional cost vector. The term  $k$ -sum optimization was coined by Gupta and Punnen in their paper from 1990. Throughout the years, this type of an objective function has attracted the attention of researchers, and one can find a number of results that consider  $k$ -sum optimization

---

✉ J. Puerto  
puerto@us.es

<sup>1</sup> Instituto de Investigación Matemática de la Universidad de Sevilla (IMUS), Universidad de Sevilla, Sevilla, Spain

<sup>2</sup> Facultad de Ciencias, Universidad de Cádiz, Cádiz, Spain

<sup>3</sup> School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel

applied to a variety of feasible domains, ranging from continuous or integer variables to pure combinatorial problems.

The idea of  $k$ -sum (also called  $k$ -centrum in the location analysis literature) optimization can be traced back at least to the late seventies in the papers by Slater [40,41] dealing with network location problems. Later, Punnen and Aneja in 1996 also derived some new results on the optimization of  $k$ -sum objective functions, and Tamir [44], Tamir et al. [46] and Mesa et al. [31] considered  $k$ -centrum location problems on graphs. Kalcsics et al. [24,25] also considered a similar problem applied to different location problems (continuous and discrete) and Ogryczak and Tamir [33] developed a linear time algorithm for the minimization of some family of  $k$ -sum problems. Using a different perspective, looking for robust solutions to combinatorial optimization problems, Bertsimas and Sim [6] and Bertsimas and Weismantel [7] revisited the problem and extended an old result by Punnen and Aneja [38], that was instrumental in their results. Puerto and Tamir [36] analyzed the problem of locating a tree shaped facility using  $k$ -centrum criterion for the strategic and tactical models on a tree network. The case of an  $s-t$  shortest path was studied by Garfinkel et al. [15]. In each of these cases, the authors developed “ad hoc” arguments that help in solving each particular problem. Nevertheless, one observes the lack of a general methodology to be applicable in great generality to solve these problems.

As mentioned, in Bertsimas and Sim [6], the authors provided very nice algorithms for the class of the so called robust combinatorial optimization problems. Nevertheless, for the important class of the  $k$ -sum flow problems they only provided an approximate algorithm. Their approach left open the problem of the existence of a strongly polynomial time algorithm for that class of problems. Hence, our initial motivation was to address the above open problem in Bertsimas and Sim so as to develop an exact strongly polynomial time algorithm. Starting with that analysis we have identified a general theory that is applicable to the class of  $k$ -sum optimization problems regardless of the underlying feasible set, and that with minor modification can be cast to (continuous) polyhedral sets, integer lattices and pure combinatorial problems.

This tool will allow us to obtain new complexity bounds for some already considered  $k$ -sum problems and to develop complexity results for newly addressed problems. Roughly speaking, the methodology presented in this paper consists of solving a  $k$ -sum optimization problem by solving a polynomial number of minisum problems in the same or slightly modified feasible region. Here, we would like to remark that the converse is not true: The minisum knapsack problem (with positive coefficients) is NP-hard. However, the minimax version ( $k = 1$ ) is in P. The contributions of this paper are twofold: (1) to develop a new methodology applicable to the optimization of  $k$ -sum objective functions in great generality, and (2) to obtain new algorithms and complexity results for a number of problems, improving or getting similar bounds, but using the same approach in all cases (See Table 1).

Under a more general paradigm,  $k$ -sum optimization can also be seen as a particular case of the *ordered median objective function* (see [32,34]). Ordered median optimization has been intensively applied in Location Analysis, (Nickel and Puerto [32], Puerto and Rodríguez-Chía [35]) and more recently extended to deal with abstract combinatorial optimization problems [12], and more specialized cases such as paths

or matchings on graphs [13]. Therefore, our goal is also to analyze if this methodology can be extended further to the more general ordered median objective function (OMP).

The paper is organized in seven sections. In Sect. 2, we provide a general reformulation of the  $k$ -sum optimization problem. The continuous, integer and combinatorial versions of this problem are analyzed in Sects. 3, 4 and 5, respectively. Section 6 considers the extension of the previous methodology to the ordered median problem. We conclude the paper with several remarks.

## 2 A reformulation of the $k$ -sum optimization problem

Given a matrix  $A = (a_{ij}), i = 1, \dots, m; j = 1, \dots, n$ , a pair of vectors  $c, d \in \mathbb{R}^n$ , and a vector  $b \in \mathbb{R}^m$ , consider the following  $k$ -sum problem (also called  $k$ -centrum in the literature on location analysis),

$$\min_{x \in X} \left( cx + \max \left\{ \sum_{j \in S_k} d_j x_j : S_k \subseteq \{1, \dots, n\}, |S_k| = k \right\} \right), \tag{1}$$

with  $X = \{x : Ax = b, x \in \mathcal{X}\}$ , where  $\mathcal{X}$  is some domain set for  $x$ , for instance  $\mathbb{R}_+^n, \mathbb{N}^n$  or  $\{0, 1\}^n$ .

*Remark 2.1* Although we have referred to Problem (1) as the  $k$ -sum problem, the above formulation is more general than the original  $k$ -sum optimization problem in Gupta and Punnen [18], Punnen [37] where only the minimization of the  $k$ -largest costs out of  $n$  is considered ( $c = 0$ ). Actually, the so called centdian problem, see Halpern [20] and Handler [21], namely the minimization of a convex combination of the sum and the maximum costs, can also be seen as a particular case of the above problem, where  $k = 1, c' = \alpha c$ , replaces  $c$  in (1),  $d' = (1 - \alpha)c$ , replaces  $d$  in (1), and  $\alpha$  is a scalar in  $(0, 1)$ . Therefore, the complexity bounds provided in this paper for Problem (1) can be applied, among others, to centdian problems.

**Theorem 2.1** *Problem (1) is equivalent to solving the following problem,*

$$Z_X^* = \min_{r \in \mathbb{R}} Z_X(r) = \min_{r \in \mathbb{R}} \left\{ kr + \min_{x \in X} \left\{ cx + \sum_{j=1}^n \max\{d_j x_j - r, 0\} \right\} \right\} \tag{2}$$

*Proof* The inner maximum in Problem (1) can be rewritten as:

$$\begin{aligned} & \max \sum_{j=1}^n d_j x_j v_j \\ & \text{s.t. } \sum_{j=1}^n v_j = k, \\ & \quad v_j \in \{0, 1\}, \quad \forall j = 1, \dots, n. \end{aligned}$$

**Table 1** Summary of the results

Problem	Best known complexity	Our results
$k$ -centrum ( $k$ -sum) minimum cost network flow problem	Polynomial time approximate algorithm, Bertsimas and Sim [6]	Exact Strongly polynomial algorithm
$k$ -centrum path problem on a tree	Unknown	$O(n^2 \log n)$
Continuous tactical $k$ -centrum subtree problem on a tree	$O(n^3 + n^{2.5} I)$ , Puerto and Tamir [36]	$O(n \log n)$
Continuous tactical $k$ -centrum path problem on a tree	Unknown	$O(n(n\alpha(n) \log n)^2)$
Continuous strategic $k$ -centrum subtree problem on a tree	$O(kn^7)$ , Puerto and Tamir [36]	$O(n \log n)$
$k$ -centrum ( $k$ -sum) Chinese postman problem	Unknown	Strongly polynomial
The discrete $p$ -facility $k$ -centrum problem on a tree	$O(pk^2n^2)$ , Kalcsics [23]	$O(pn^4)$
The discrete $p$ -facility $k$ -centrum problem on paths	Unknown	$O(pn^3)$
The discrete tactical $k$ -centrum path problem on a tree	Unknown	$O(n^3 \log n)$
The discrete strategic $k$ -centrum subtree problem on a tree	$O(kn^3)$ , Puerto and Tamir [36]	$O(n^3)$
The $k$ -centrum shortest path problem	$O(n^2 m^2)$ , Garfinkel et al. [15]	$O(m^2 + mn \log n)$
The convex continuous ordered median problem (OMP)	Unknown	Polynomial
The continuous $p$ -facility OMP on fork-free $\lambda = (a, s, a, b, n-s, b)$	$O(pn^9 s^2)$ , Kalcsics et al. [25]	$O(pn^9 \log^2 n)$
The OMP min cost flow problem	Unknown	Strongly polynomial

The above constraint matrix is totally unimodular and thus, the dual of this problem is:

$$\begin{aligned} \min \quad & kr + \sum_{j=1}^n p_j \\ \text{s.t.} \quad & p_j + r \geq d_j x_j, \quad \forall j = 1, \dots, n, \\ & p_j \geq 0, \quad \forall j = 1, \dots, n, \end{aligned}$$

and the result follows. □

### 3 The $k$ -sum optimization problem on polyhedral sets

In this section, we focus on the case where the feasible domain  $X$  is a polyhedron, which can be obtained for instance if  $\mathcal{X} = \mathbb{R}_+^n$ . Let  $X_L := \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$  be the region  $X$  for this particular case (continuous linear case).

**Theorem 3.1** 1.  $Z_{X_L}(r)$  is a piecewise linear convex function.  
 2. Suppose that there is a combinatorial algorithm (i.e., performs only additions, subtractions, multiplication by a scalar and comparisons) of  $O(T(n, m))$  complexity to compute  $Z_{X_L}(r)$  for any given  $r$ . Then,  $Z_{X_L}^*$  can be computed in  $O((T(n, m))^2)$  time. Moreover, if  $T(n, m) = O(n)$  and satisfies the bounded fan-in/fan-out property (see [10]) then  $Z_{X_L}^*$  can be computed in  $O(n \log n)$  time.

*Proof* First, we have

$$\begin{aligned} Z_{X_L}(r) = kr + \min_{p,x} \quad & \left\{ cx + \sum_{j=1}^n p_j \right\} \\ \text{s.t.} \quad & p_j - d_j x_j \geq -r, \quad j = 1, \dots, n, \\ & \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m, \\ & x_j, p_j \geq 0, \quad j = 1, \dots, n. \end{aligned}$$

Taking the dual of the inner minimum problem, we have that

$$\begin{aligned} Z_{X_L}(r) := kr + \max_{\alpha, \beta} \quad & \left\{ -r \sum_{j=1}^n \alpha_j + \sum_{i=1}^m b_i \beta_i \right\} \\ \text{s.t.} \quad & \alpha_j \geq 1, \quad j = 1, \dots, n, \\ & -\alpha_j d_j + \sum_{i=1}^m a_{ij} \beta_i \geq c_j, \quad j = 1, \dots, n. \end{aligned}$$

Therefore,  $Z_{X_L}(r)$  is the sum of a linear function plus the maximum of a finite number of linear functions which is a piecewise linear convex function. The second

statement for general  $T(n, m)$  follows from the parametric approach, see Megiddo [29], obtaining a complexity of  $O((T(n, m))^2)$ . Next, if its comparisons can be performed independently in  $s$  phases with  $C_i, i = 1, \dots, s$ , comparisons each one we can apply the multiparametric search in Cohen and Megiddo [9] to obtain a complexity of  $O(T(n, m) \sum_{i=1}^s \lceil \log C_i \rceil)$ . We observe that the application of this result to a combinatorial algorithm  $T(n, m)$  with complexity  $O(n)$  will result in an overall complexity of  $O(n \log^2 n)$  for computing  $Z_{X_L}^*$ . If in addition,  $T(n, m)$  satisfies the bounded fan-in/fan-out property then  $Z_{X_L}^*$  can be computed in  $O(n \log n)$  time, Cole [10]. □

*Remark 3.1* We observe that there exists an earlier proof of the polynomiality for computing  $Z_{X_L}^*$ , by Punnen [37], using an efficient separation oracle for a linear program with an exponential number of constraints.

### 3.1 Applications in the continuous case

A particular important instance of the above linear model is the robust minimum cost network flow problem considered in Bertsimas and Sim [6], Bertsimas and Weismantel [7]. In these publications the authors provide an efficient algorithm to approximate the optimal solution of that flow problem within any desirable accuracy level  $\epsilon$ , but fail to develop an exact strongly polynomial algorithm. As we will show in the following, our approach gives an exact algorithm with strongly polynomial complexity.

**Theorem 3.2** *The  $k$ -sum minimum cost network flow problem can be solved in  $O((m \log n)^2(m + n \log n)^2)$  time, where  $n$  and  $m$  are, respectively, the number of nodes and edges in the underlying network.*

*Proof* Observe that the evaluation of  $Z_{X_L}(r)$  for the minimum cost network flow problem reduces to solving a minimum cost flow problem with piecewise linear convex cost functions with at most one breakpoint per function. Indeed,

$$Z_{X_L}(r) = kr + \min_{x \in X} cx + \sum_{i=1}^n \sum_{j=1}^n \max\{d_{ij}x_{ij} - r, 0\}.$$

Observe that  $X$  is the flow polytope associated with the original graph  $G$ . To solve this flow problem, we can duplicate each arc  $(i, j)$  of the original graph assigning capacity  $r/d_{ij}$  and zero cost to the first duplicate arc and no capacity and cost  $d_{ij}$  to the second one. The resulting flow problem has linear costs on the arcs. Hence, the complexity of evaluating  $Z_{X_L}(r)$  is  $O((m \log n)(m + n \log n))$  for a given value of  $r$ , Ahuja et al. [1].

This implies that we can apply Theorem 3.1.ii to the problem and thus, the overall complexity of solving the  $k$ -sum minimum cost flow problem is  $O((m \log n)^2(m + n \log n)^2)$ . □

We can improve upon the complexity bound at the end of the proof, by using the parallelization techniques in Megiddo [30], Section 6.

3.1.1 The  $k$ -centrum path problem on a tree

Another model where we can apply the same machinery is the location of  $k$ -centrum paths on a tree, i.e., locating a path on a tree network that minimizes the sum of the  $k$ -largest distances from the nodes of the tree to the path. We are given a rooted tree  $T = (V, E)$ , with root  $v_0$  and  $|V| = n$ . Let  $\mathcal{A}(T)$  denote the continuum set of points on the edges of  $T$ . In this framework we call  $Y$  a subtree if it is a closed and connected subset of  $\mathcal{A}(T)$ . Each edge  $e \in E$  has a positive length  $\ell_e$  and it is assumed to be rectifiable. The edge lengths induce a distance function on  $\mathcal{A}(T)$ . For any pair of points  $x, y \in \mathcal{A}(T)$ , we let  $d(x, y)$  denote the length of the unique simple path in  $\mathcal{A}(T)$  connecting  $x$  and  $y$ . Let  $w_i$  be a nonnegative weight associated with  $v_i \in V$ . A path containing  $v_0$  which minimizes the weighted sum of distances can be found in  $O(n)$  time, Averbakh and Berman [4].

**Theorem 3.3** *The  $k$ -centrum path problem on a tree can be solved in  $O(n^2 \log n)$  time.*

*Proof* Consider the following valid formulation for the path containing node  $v_0$ , the root, which minimizes the weighted sum of distances to the path. Define the variable  $x_q \in [0, 1]$  for any  $e_q \in E$  for  $q = 1, \dots, n - 1$ , let  $P[v_q, v_0]$  be the unique path from node  $v_q$  to the root node  $v_0$  and  $ES(e_q)$  be the set of descendant edges of  $e_q$ , i.e., given  $v$ , an endnode of  $\bar{e} \in E$ , we say that  $\bar{e} \in ES(e_q)$  iff  $e_q \in P[v, v_0]$ .

$$\begin{aligned} \min \quad & \sum_{q=1}^{n-1} w_q \sum_{j:e_j \in P[v_q, v_0]} \ell_j (1 - x_j) \\ \text{s.t.} \quad & \sum_{q:e_q \in ES(e_i)} x_q \leq x_i, \quad \forall i = 1, \dots, n - 1, \\ & 0 \leq x_j \leq 1, \quad \forall j = 1, \dots, n - 1. \end{aligned}$$

We observe that the above linear formulation has an integer solution since the constraints are flow conservation constraints and therefore the matrix is totally unimodular. This formulation allows us to apply Theorem 3.1 to this model for the minimization of the sum of the  $k$ -largest weighted distances. Indeed, by the results in Averbakh and Berman [4], the evaluation of  $Z_{X_L}(r)$  has  $O(n)$  complexity, then by Theorem 3.1 the corresponding  $k$ -centrum problem is solvable in  $O(n \log n)$  time provided that the path contains  $v_0$ . Repeating the same argument and rooting the tree at the different nodes, the overall complexity for solving the problem is  $O(n^2 \log n)$ .  $\square$

*Remark 3.2* In the next sections we will distinguish between the discrete and continuous versions of the problem of locating a subtree or a path on a tree. In the continuous case it is assumed that the end points of the subtree or path to be located are not necessarily nodes of the original tree, while in the discrete case these end points should be nodes. Observe that a solution of the above continuous  $k$ -centrum path problem is discrete. Therefore the discrete version of the  $k$ -centrum path problem is also solvable in  $O(n^2 \log n)$ .

### 3.1.2 The continuous tactical $k$ -centrum subtree/path problem on a tree

The tactical (minisum) median subtree (path) problem consists of locating a subtree (path),  $Y \subseteq \mathcal{A}(T)$  under the size constraint, i.e.,  $L(Y) \leq L$ . The demand points (nodes) are allocated to the closest server, minimizing the weighted sum of the distances of the nodes to their respective closest point on the facility. The continuous version of this problem can be formulated as follows:

$$\begin{aligned} \min_{Y \subseteq \mathcal{A}(T)} \quad & \sum_{i=1}^n w_i d(v_i, Y) \\ \text{s.t.} \quad & L(Y) \leq L, \end{aligned}$$

where  $d(v_i, Y)$  is the distance from node  $v_i$  to its closest point of  $Y$ . The case of locating a median subtree is solvable in  $O(n)$  time, see Tamir [43] and the median path in  $O(n\alpha(n) \log n)$  time, see Alstrup et al. [2]; where  $\alpha(n)$  is the inverse of the Ackermann function (see [39]).

In this section we consider the continuous tactical  $k$ -centrum subtree and path problems. To the best of our knowledge, the time complexity for the  $k$ -centrum version of locating a subtree using the continuous tactical model is still  $O(n^3 + n^{2.5}I)$ , given in Puerto and Tamir [36]; where  $I$  is the total number of bits needed to represent the input. That algorithm is based on submodular optimization. Our approach will give us an improved strongly polynomial complexity bound.

**Theorem 3.4** 1. *The continuous tactical  $k$ -centrum subtree problem on a tree can be solved in  $O(n \log n)$  time.*  
 2. *The continuous tactical  $k$ -centrum path problem on a tree can be solved in  $O(n(n\alpha(n) \log n)^2)$  time.*

*Proof* By Puerto and Tamir [36] the continuous tactical  $k$ -centrum subtree problem on a tree satisfies a nestedness property, i.e., an optimal subtree contains an optimal solution of the single facility  $k$ -centrum problem. Using the LP representation of this problem, see Puerto and Tamir [36], and the fact that  $Z_{X_L}(r)$  can be computed for this problem in  $O(n)$  time for any value  $r$ , see Tamir [43], we can apply Theorem 3.1 obtaining an overall complexity of  $O(n \log n)$  time.

For the case of locating a  $k$ -centrum path on a tree, the function  $Z_{X_L}(r)$  can be evaluated in  $O(n\alpha(n) \log n)$  time for any value  $r$ , see Alstrup et al. [2]. Next, we can apply Theorem 3.1 on the LP formulation of the median path problem on a tree given in Theorem 3.3 (we have to add the size constraint to this formulation). Thus, the complexity of solving the  $k$ -centrum problem is  $O((n\alpha(n) \log n)^2)$  provided that we fix a node that must belong to the optimal path. Rooting the path on the different nodes of the tree, the overall complexity is  $O(n(n\alpha(n) \log n)^2)$  time.  $\square$

### 3.1.3 The continuous strategic $k$ -centrum subtree problem on a tree

The strategic median subtree problem consists of locating a subtree,  $Y \subseteq \mathcal{A}(T)$  but instead of including an explicit bound on the length of this extensive facility, the length



is a variable and the objective function of the problem consists of minimizing the sum of the transportation cost and the length of the subtree. In this section we consider the continuous versions of this problem. The problem can be formulated as follows:

$$\min_{Y \subseteq \mathcal{A}(T)} \sum_{i=1}^n w_i d(v_i, Y) + \delta L(Y),$$

with  $\delta \in \mathbb{R}$  (in the case  $\delta < 0$  the optimal solution would be the complete tree). To the best of our knowledge, the time complexity for solving the  $k$ -centrum version of locating a subtree using the continuous strategic model is  $O(kn^7)$ , given in Puerto and Tamir [36]. In the following we improve this complexity using our approach.

**Theorem 3.5** *The continuous strategic  $k$ -centrum subtree problem on a tree is solvable in  $O(n \log n)$  time.*

*Proof* Based on the results valid for the problem of locating a strategic median subtree on a tree, the function  $Z_{X_L}(r)$  for the continuous strategic  $k$ -centrum subtree can be evaluated in  $O(n)$  time for any value  $r$ , see Kim et al. [26]. By Puerto and Tamir [36] the  $k$ -centrum strategic subtree problem on a tree also satisfies a nestedness property, i.e., an optimal subtree contains the optimal  $k$ -centrum point. Using the LP representation of this problem in Puerto and Tamir [36], we can apply Theorem 3.1, obtaining an overall complexity of  $O(n \log n)$  time.

### 3.1.4 The single facility $k$ -centrum problem

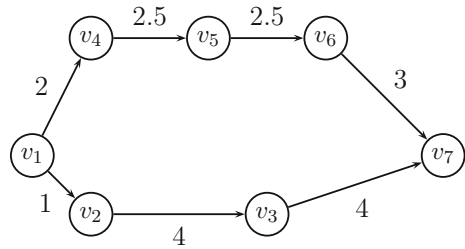
Finally, we wish to conclude Sect. 3 mentioning another family of problems where this methodology can be successfully applied. In the paper by Kalcsics et al. [24], the authors provide complexity results for solving the single point facility  $k$ -centrum problem. Using ad hoc arguments in each framework space, they obtained complexity bounds of  $O(mn \log n)$  for undirected general networks and a linear bound for the continuous case of locating a single facility  $k$ -centrum. Our approach can also be applied to these two problems obtaining slightly worse complexity bounds, but with the advantage of using a general tool based on the same methodology developed in Theorem 3.1.

## 4 The $k$ -sum integer problem

In this section we address the  $k$ -sum optimization problem over a subset of the lattice of nonnegative integer points in  $\mathbb{R}^n$ , i.e.,  $\mathcal{X} \subset \mathbb{Z}_+^n$ . Let  $X_I = \{x \in \mathbb{R}^n : Ax = b, x_j \in \{0, 1, 2, \dots\}, j = 1, \dots, n\}$  be the region  $X$  for this particular case (integer case). The next example demonstrates that unlike the linear case, even for the binary case, the function  $Z_{X_I}(r)$  is not generally convex when  $k = 1$ , and is not generally unimodal even when  $k = 3$ .

*Example 4.1* Consider the directed shortest path problem from  $v_1$  to  $v_7$ , defined on a graph  $G = (V, E)$  with 7 nodes,  $V = \{v_1, v_2, \dots, v_7\}$ . The graph consists of two directed paths from  $v_1$  to  $v_7$ . The first path consists of the 3 edges

Fig. 1 Graph of Example 4.1



$(v_1, v_2)$ ,  $(v_2, v_3)$ ,  $(v_3, v_7)$  of lengths 1, 4 and 4, respectively. The second path consists of the 4 edges  $(v_1, v_4)$ ,  $(v_4, v_5)$ ,  $(v_5, v_6)$ ,  $(v_6, v_7)$  of lengths 2, 2.5, 2.5 and 3, respectively, see Fig. 1. Set the vector  $c = 0$ , and for each edge  $e$  let  $d_e$  be its length. We show that the function  $Z_{X_I}(r)$  is not unimodal.

Since there are only two feasible paths from  $v_1$  to  $v_7$  it is easy to compute and see that for  $k = 1$ ,  $Z_{X_I}(0) = 9$ ,  $Z_{X_I}(1) = 7$ ,  $Z_{X_I}(2) = 4$ ,  $Z_{X_I}(2.5) = 3$ ,  $Z_{X_I}(3) = 3$  and  $Z_{X_I}(4) = 4$ . Although, as we will later see, the function  $Z_{X_I}(r)$  between two consecutive breakpoints is piecewise concave, in this example for  $k = 1$  and  $k = 3$ , this function is linear. Hence, the sequence of slopes between consecutive breakpoints is  $-2, -3, -2, 0$  and  $1$ . This shows that  $Z_{X_I}(r)$  is not convex on the sequence of breakpoints.

Next, consider the above shortest path model with  $k = 3$ . For this case we get  $Z_{X_I}(0) = 9$ ,  $Z_{X_I}(1) = 9$ ,  $Z_{X_I}(2) = 8$ ,  $Z_{X_I}(2.5) = 8$ ,  $Z_{X_I}(3) = 9$  and  $Z_{X_I}(4) = 12$ . The sequence of slopes between consecutive breakpoints is  $0, -1, 0, 2$  and  $4$ . It is easy to see that each  $r \in [0, 1)$  is a local minimum point of  $Z_{X_I}(r)$ , which is not a global minimum point. Hence, for  $k = 3$ ,  $Z_{X_I}(r)$  is not even unimodal.

### 4.1 The $k$ -sum integer optimization with polynomially bounded variables and $d \geq 0$

In this section, we prove that if the components of the cost vector  $d$  are nonnegative and all the integer variables are bounded by  $M = M(n, m)$ , where  $M(n, m)$  is a polynomial in  $m, n$ , (recall that  $A = (a_{ij})$  with  $i = 1, \dots, m, j = 1, \dots, n$ ), then the  $k$ -sum integer optimization problem is polynomially solvable assuming that the matrix  $A$  is totally unimodular.

Using the above notation we have

$$Z_{X_I}(r) = kr + \min_{x \in X_I} \left( cx + \sum_{j=1}^n \max\{d_j x_j - r, 0\} \right). \tag{3}$$

We decompose the interval  $[0, M \max_{j=1, \dots, n} \{d_j\}]$  (assume without loss of generality that  $M$  is integer) into consecutive intervals induced by the set of points  $\{pd_j\}$ ,  $p \in \{0, \dots, M\}$ , and  $j = 1, \dots, n$ . Consider an arbitrary interval of the above collection,  $\mathcal{I} = [pd_s, qd_t]$  with  $p, q \in \{0, \dots, M\}$ , and  $s, t \in \{1, \dots, n\}$ . For each  $j = 1, \dots, n$ , let  $h_j$  be the unique nonnegative integer such that the interval  $[h_j d_j, (h_j + 1)d_j]$

contains  $\mathcal{I}$ . Then, for each  $r \in \mathcal{I}$ , the function  $f_j(\mathcal{I}, x_j) = \max\{d_j x_j - r, 0\}$ , defined over the nonnegative integers, satisfies  $f_j(\mathcal{I}, x_j) = 0$  when  $x_j \leq h_j$ , and  $f_j(\mathcal{I}, x_j) = d_j x_j - r$  when  $x_j \geq h_j + 1$ .

From the above we note that the function  $g(\mathcal{I}, r) = kr + \min_{x \in X_I} cx + \sum_{j=1}^n \max\{d_j x_j - r, 0\}$  is concave (minimum of a finite number of linear functions) for  $r \in \mathcal{I}$ . Specifically,

$$g(\mathcal{I}, r) = kr + \min_{x \in X_I} cx + \sum_{\substack{j=1 \\ x_j > h_j}}^n (d_j x_j - r). \tag{4}$$

Hence, without loss of generality, we may conclude that the minimum of  $Z_{X_I}(r)$  in the interval  $\mathcal{I}$  is achieved in its extreme points, i.e., in  $\{pd_s, qd_t\}$ ,  $p, q \in \{0, \dots, M\}$ .

**Theorem 4.1** *Consider the  $k$ -sum integer optimization problem with objective function  $Z_{X_I}(r)$  defined in (3), and assume that the matrix  $A$  is totally unimodular. Suppose further that all integer variables are bounded from above by some polynomial  $M(n, m)$ . Then,  $Z_{X_I}^*$ , the optimal solution of this problem, can be computed in strongly polynomial time.*

*Proof* By the above discussion  $Z_{X_I}^*$  can be computed by evaluating  $Z_{X_I}(r)$  for  $O(nM(n, m))$  values of the parameter  $r$  (the possible extreme points of the intervals defining the partition of  $[0, M \max_{j=1, \dots, n} \{d_j\}]$ ). Specifically, for a fixed value of  $r$ , we need to solve the following problem:

$$\begin{aligned} \min \quad & cx + \sum_{j=1}^n \max\{d_j x_j - r, 0\}, \\ \text{s.t.} \quad & x \in X_I. \end{aligned}$$

The problem above can be solved in strongly polynomial time by substituting  $x_j = u_j + v_j + z_j$ ,  $j = 1, \dots, n$ , and solving the respective integer program, defined by a totally unimodular system,

$$\begin{aligned} \min \quad & c(u + v + z) + \sum_{j=1}^n (d_j(\lceil r/d_j \rceil - r/d_j)v_j + d_j z_j), \\ \text{s.t.} \quad & A(u + v + z) = b, \\ & u_j \in \{0, 1, \dots, \lfloor r/d_j \rfloor\}, \quad j = 1, \dots, n, \\ & v_j \in \{0, 1\}, \quad j = 1, \dots, n, \\ & z_j \in \{0, 1, 2, \dots\}, \quad j = 1, \dots, n. \end{aligned}$$

Since  $A$  is totally unimodular this problem is an LP with  $\{0, \pm 1\}$ -matrix and therefore, by Tardos [47], it is solvable by a strongly polynomial algorithm. □

This result gives a positive answer to a conjecture in Punnen [37], since it proves that  $k$ -sum optimization problem is polynomially solvable assuming that the constraint matrix is totally unimodular and the variables are bounded.

As an application of the above result we have the following property for the Chinese Postman Problem, see Granot et al. [16] where the  $k$ -centrum Chinese Postman Delivery Problem is analyzed from a cooperative perspective.

**Proposition 4.1** *The  $k$ -sum Chinese Postman Problem defined on undirected connected graphs or on strongly connected directed graphs is solvable in strongly polynomial time.*

*Proof* The Chinese Postman Problem, when defined on a strongly connected directed graph, is a flow problem where the integer flow variables are bounded by the number of edges of the graph. Actually, looking only at basic solutions of the respective circulation problem, we can conclude that a sharper upper bound is  $|E| - |V| + 1$ , where  $|E|$  and  $|V|$  are the number of edges and nodes of the underlying graph  $G = (V, E)$ . When the above postman problem is defined on an undirected graph each variable can take on the values 1 and 2. Applying Theorem 4.1 the result follows.  $\square$

*Remark 4.1* Similar arguments as the one above apply also to the rural postman problem (recall that the rural postman problem is an extension of the Chinese Postman Delivery Problem, in which a subset of the edges from the graph are required to be traversed at a minimal cost), which is thus also solvable in strongly polynomial time.

## 5 The $k$ -sum combinatorial optimization problem

Here we consider the  $k$ -sum combinatorial optimization problem, i.e.,  $\mathcal{X} = \{0, 1\}^n$ . Therefore, given a finite set of elements  $E$ , where each  $e \in E$  is associated with a pair of real weights  $(c_e, d_e)$ , and  $X_C$  is a collection of subsets of  $E$ ; the minisum problem is to find a subset  $S \in X_C$  of minimum total weight,  $c(S) + d(S) = \sum_{e \in S} (c_e + d_e)$ . The  $k$ -sum optimization problem with respect to the  $d$  weights is to find a subset  $S \subseteq X_C$  minimizing the sum of  $c(S)$  and the sum of the  $k$ -largest elements in the set  $\{d_e : e \in S\}$ .

We use the following convention to handle the definition of the set of the  $k$  largest elements when the cardinality of a subset is smaller than  $k$ . Each subset  $x \subseteq E$  is viewed as a  $\{0, 1\}$  vector with  $E$  components. Thus, for  $x = (x_1, \dots, x_{|E|}) \in \{0, 1\}^{|E|}$ , the sum of the largest  $k$  elements with respect to the weights  $\{d_e\}$ , is the sum of the largest  $k$  elements in the set  $\{d_e x_e : e \in E\}$ .

The  $k$ -sum version of those combinatorial optimization problems which are efficiently solvable for linear functions (assignment, shortest paths, matching, matroid, et cetera) can also be solved in an efficient way when  $d \geq 0$ . Indeed, as a result of Bertsimas and Sim [6], Bertsimas and Weismantel [7], Punnen and Aneja [38], it follows that solving the  $k$ -sum problem on the respective combinatorial model can be done by solving  $O(t)$  linear optimization problems where  $t$  is the number of different cost coefficients of the elements (e.g., edges of a graph, nodes of a graph, elements in the ground set, etc.)

The supposition that  $d_e \geq 0$ , for each  $e \in E$ , which is made in the papers Bertsimas and Sim [6], Bertsimas and Weismantel [7], Punnen and Aneja [38] is used extensively in the proofs. For example, in the proof of Theorem 3 in Bertsimas and Sim [6], based on this nonnegativity supposition, they can relax the formulation and introduce the constraint that at most  $k$  elements are selected, i.e.,  $\sum_{e \in E} u_e \leq k$  and consequently the associated dual variable is nonnegative. With this relaxed constraint their problem becomes equivalent to the one discussed in Punnen and Aneja [38]. (In Punnen and Aneja [38], they discuss only the case when  $c = 0$ .)

In what follows we analyze the case of arbitrary  $\{d_e\}_{e \in E}$ . Attempting to follow Bertsimas and Sim [6], Punnen and Aneja [38], for the general case we need to impose the constraint  $\sum_{e \in E} u_e = k$ , then the dual variable associated with this constraint is unrestricted in sign. The proof of Theorem 3 in Bertsimas and Sim [6] can be modified to this case as well, obtaining the following result for general  $\{d_e\}$ :

**Theorem 5.1** *Suppose that for any real  $r$  the minisum problem with respect to the weights  $(c_e, \max\{0, d_e - r\})$ ,  $e \in E$ , is solvable in  $T(m)$  time, where  $m = |E|$ . Then, the  $k$ -sum optimization problem with respect to the  $d$  weights can be solved in  $O(m'T(m) + T'(m))$  time, where  $m'$  is the number of distinct elements in the set  $\{d_e : e \in E\}$ , and  $T'(m)$  is the time to solve the original minisum problem with respect to the weights  $(c_e, d_e)$ ,  $e \in E$ . Moreover, if  $c$  is not required to be nonnegative  $T(m) = T(m')$ .*

For example, if  $c = 0$ , and we consider the shortest path problem on a directed graph with nonnegative cycles with respect to the  $\{d_e\}$  weights, then the  $k$ -sum optimization model can be solved by solving the original minisum problem and in addition  $O(m')$  shortest path problems with nonnegative weights. A detailed analysis of this problem is presented in Sect. 5.1.4.

### 5.1 Applications in the combinatorial case

#### 5.1.1 The discrete $p$ -facility $k$ -centrum location problem on trees and paths

Let  $G = (V, E)$  be an undirected path or tree network with node set  $V = \{v_1, \dots, v_n\}$  and edge set  $E = \{e_2, \dots, e_n\}$ .

To formulate the  $p$ -facility median problem we will use the following sets of binary variables:

$$\begin{aligned}
 y_j &= \begin{cases} 1, & \text{if a center is located at node } v_j, \\ 0, & \text{otherwise,} \end{cases} & j = 1, \dots, n, \\
 x_{ij} &= \begin{cases} 1, & \text{if the node } v_i \text{ is assigned to center } j, \\ 0, & \text{otherwise.} \end{cases} & i, j = 1, \dots, n.
 \end{aligned}$$

The model is formulated as:

$$\begin{aligned}
 \min & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\
 \text{s.t. } & x_{ij} \leq y_j, \quad \forall i, j = 1, \dots, n,
 \end{aligned} \tag{5}$$

$$\sum_{j=1}^n y_j = p, \tag{6}$$

$$\sum_{j=1}^n x_{ij} = 1, \quad \forall i = 1, \dots, n, \tag{7}$$

$$x_{ij}, y_j \in \{0, 1\}, \quad \forall i, j = 1, \dots, n. \tag{8}$$

Let us denote by  $X_{med(p)}$  the lattice points defined by (5)–(8).

**Path case:**

For the case of a path the complexity will depend on the time needed to solve the minimum problem with the pseudo metric induced by the values  $\max\{d(v_i, v_j) - r, 0\}$ . Hassin and Tamir [22] solve the  $p$ -median on a path in  $O(pn)$  time. In fact, they consider the following general model:

$$\begin{aligned} \min \quad & \sum_{i \in S} c_i + \sum_{i=1}^n f_i(d(v_i, S)) \\ \text{s.t.} \quad & S \subseteq N \\ & |S| \leq p, \end{aligned}$$

where  $f_i(d(v_i, S))$  is a real monotone nondecreasing function of  $d(v_i, S)$ . In our case, assuming that  $v_1 < \dots < v_n$ , and  $1 \leq j \leq k \leq n$ , we have that

$$\begin{aligned} w(j, k) &= \sum_{t=j}^k f_t(d(v_t, v_j)) = \sum_{t=j}^k w_t \max\{v_t - v_j - r, 0\}, \\ \bar{w}(j, k) &= \sum_{t=j}^k f_t(d(v_t, v_k)) = \sum_{t=j}^k w_t \max\{v_k - v_t - r, 0\}. \end{aligned}$$

These values describe the recursive approach defined by Hassin and Tamir [22] through dynamic programming. In what follows we show how these values can be computed in a preprocessing phase in  $O(n)$  time.

$$\begin{aligned} w(j, k) &= \begin{cases} \sum_{t: v_t > v_j + r}^k w_t(v_t - v_j - r), & \text{if } v_j + r < v_k, \\ 0, & \text{otherwise,} \end{cases} \\ \bar{w}(j, k) &= \begin{cases} \sum_{t: v_t \leq v_k - r}^k w_t(v_k - v_t - r), & \text{if } v_k - r > v_j, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Hence, we have that

$$w(j, k) = \begin{cases} AV_1(k) - AV_2(j) - (A_1(k) - A_2(j))(v_t + r), & \text{if } v_j + r < v_k \\ 0, & \text{otherwise,} \end{cases}$$

$$\bar{w}(j, k) = \begin{cases} (A_3(k) - A_1(j - 1))(v_k - r) - (AV_3(k) - AV_1(j - 1)), & \text{if } v_k - r > v_j \\ 0, & \text{otherwise,} \end{cases}$$

where

$$AV_1(k) = \sum_{t=1}^k w_t v_t, \quad A_1(k) = \sum_{t=1}^k w_t$$

$$AV_2(j) = \sum_{t:v_t < v_j+r} w_t v_t, \quad A_2(j) = \sum_{t:v_t < v_j+r} w_t$$

$$AV_3(k) = \sum_{t:v_t < v_k-r} w_t v_t, \quad A_3(k) = \sum_{t:v_t \leq v_k-r} w_t.$$

In a preprocessing phase these six coefficient sequences can be computed in  $O(n)$  time. Summarizing we have the following complexity result.

**Theorem 5.2** *The discrete  $p$ -facility  $k$ -centerum problem on a path is solvable in  $O(pn^3)$  time.*

*Proof* Sorting the different costs values  $(c_{ij})$  for  $i, j = 1, \dots, n$ , in increasing order, we get the ordered cost sequence:

$$c_{(1)} := 0 < c_{(2)} < \dots < c_{(\#C)} := \max_{1 \leq i, j \leq n} \{c_{ij}\}.$$

However, we can still apply the  $k$ -sum result since it requires to solve  $O(\#C)$  problems of the form:

$$\min \sum_{i=1}^n \sum_{j=1}^n \max\{c_{ij} - c_{(\ell)}, 0\} x_{ij}$$

(9)

*s.t.*  $x \in X_{med(p)}$

The above set of coefficients, i.e.,  $\tilde{c}_{ij}^\ell = \max\{c_{ij} - c_{(\ell)}, 0\}$  corresponds to the pseudometric induced with respect to a neighbor of radius  $c_{(\ell)}$ . As we have proved above, the algorithm in Hassin and Tamir [22] also applies to Problem (9) with a complexity of  $O(pn)$ . Therefore, by Theorem 5.1, since we have at most  $O(n^2)$  different values for  $c_{ij}$ , the  $k$ -centerum  $p$ -facility on a path is solvable in  $O(pn^3)$ . □

**Tree case:**

The problem of locating a median subtree is solvable in polynomial time provided that the  $c_{ij}$  are distances induced by the metric of shortest paths on a tree (it is NP-hard for a general linear objective function.) Actually, by the algorithm in Tamir [42] this problem can be solved in  $O(pn^2)$  time. Therefore, following similar arguments to the ones in Theorem 5.2, we have the following result.

**Theorem 5.3** *The discrete  $p$ -facility  $k$ -centrum on a tree is solvable in  $O(pn^4)$ .*

This complexity improves upon the  $O(\min(k, p)kpn^5)$  bound in Tamir [44] and equals the complexity reported in Kalcsics [23], attained using ad hoc arguments, whereas we use once again the general methodology that follows from our approach.

### 5.1.2 The discrete tactical $k$ -centrum subtree/path problem on a tree

Here we analyze the discrete version of the tactical model described in Sect. 3.1.2, i.e., the leaves of the extensive facility (path or tree) are requested to be located at the nodes of the tree.

The discrete tactical median subtree problem is NP-hard, see Hakimi et al. [19], whereas the case of locating a discrete median path is solvable in  $O(n \log n)$  time, see Alstrup et al. [2]. Applying Theorem 5.1, the  $k$ -centrum version of this model can also be solved in  $O(n^3 \log n)$  time.

### 5.1.3 The discrete strategic $k$ -centrum subtree problem on a tree

Observe that for the location of a strategic median subtree, the continuous and the discrete versions of the problem are equivalent, see Kim et al. [26]. To the best of our knowledge, the best complexity for the  $k$ -centrum version of locating a subtree using the strategic model is  $O(kn^3)$  given in Puerto and Tamir [36]. Using Theorem 5.1 we improved the complexity stated above to  $O(n^3)$  time.

### 5.1.4 $k$ -sum $s - t$ -shortest path problem

Let us assume that the directed underlying graph  $G$  does not have edges of negative length or more generally that it does not contain directed cycles of negative length. The  $k$ -sum shortest path problem can be solved in  $O(n^2m^2)$  time provided that any simple  $s - t$ -path contains at least  $k$  arcs, otherwise this problem is NP-hard, see Garfinkel et al. [15]. We improve the complexity by using Theorem 5.1.

**Theorem 5.4** *The complexity of the  $k$ -sum  $s - t$ -shortest path problem is  $O(m^2 + mn \log n)$  time.*

*Proof* The time to solve the single-source  $s - t$ -shortest path problem on a network with  $n$  nodes and  $m$  edges, having nonnegative lengths, is  $O(m + n \log n)$ , see Ahuja et al. [1] (observe that we can assume that all arc lengths are nonnegative because the number of arcs whose lengths appear in the objective function of the  $k$ -sum  $s - t$ -shortest path problem is  $k$  and then, we can add an arbitrary constant to every arc length without changing the optimal solution, see Garfinkel et al. [15]). Therefore, since the number of different  $d$  values is bounded by  $m$ , then by Theorem 5.1 the  $k$ -sum shortest path problem can be solved in  $O(m^2 + mn \log n)$  time.  $\square$

*Remark 5.1* The  $k$ -sum maximum weight matching problem with positive and negative weights is solvable in  $O(nm(m \log \log \log_{m/n} n + n \log n))$  time, applying Theorem 5.1 and the algorithm by Gabow et al. [14] for the maximum weight matching problem (recall that a matching is a set of edges without common vertices.)



### 5.1.5 $p$ -centered partition problems

Another application of Theorem 5.1 is to solve the  $k$ -centrum  $p$ -centered partition problem with flat costs on a tree. Following Apollonio et al. [3], recall that flat costs mean that they are general and independent of the distance metric induced by the shortest path distances in the graph  $G$ . Let  $G = (V, E)$  be a connected undirected graph with  $|V| = n$ . Assume that  $V$  is partitioned into two subsets,  $S$  and  $U$ , such that  $S \subset V$ , with  $|S| = p$ .  $S$  is the set of centers, and  $U = V \setminus S$  is the set of units that must be assigned to some centers in  $S$ . There is a cost function  $c : U \times S \rightarrow \mathbb{R}$  which associates a cost  $c_{is}$  with each pair  $(i, s)$ ,  $i \in U, s \in S$ .

A *connected  $p$ -partition* of  $G$  is a partition of the vertex set  $V$  into  $p$  non empty subsets (components) such that each component induces a connected subgraph of  $G$ . A connected  $p$ -partition is a  *$p$ -centered partition* if each component contains exactly one center (see [3]).

As in Apollonio et al. [3], we assume that the costs are flat, i.e., independent of the distance metric induced by the graph  $G$ . In Apollonio et al. [3] it is proved that finding a  $p$ -centered partition with minimum total cost is NP-hard in general graphs but polynomially solvable on a tree. Moreover, the minimum problem on a tree is solved by a dynamic programming algorithm in  $O(np)$  time.

**Theorem 5.5** *The  $k$ -centrum  $p$ -centered partition problem on a tree is solvable in  $O(n^2 p^2)$  time.*

*Proof* We observe that the minimum problem is solvable in  $O(np)$  time with respect to the costs  $(c_{is}, d_{is})$  for all  $i \in U, s \in S$  and also with respect to  $(c_{is}, \max\{d_{is} - r, 0\})$  for all  $i \in U, s \in S$ . Thus, by a direct application of Theorem 5.1 we obtain that the complexity for solving the  $k$ -centrum problem is  $O(n^2 p^2)$ . □

## 6 Extension to the ordered median function

As we will later see, the  $k$ -sum function is a particular case of the ordered median function. Therefore, a natural question is whether the methodology developed in the previous sections can be extended to this more general problem. Recall that for a given  $x \in \mathbb{R}^n$  and vectors  $\lambda, d \in \mathbb{R}^n$ , the value of the ordered median function is

$$\sum_{j=1}^n \lambda_j (dx)_{(j)}, \tag{10}$$

where

$$(dx)_{(1)} := d_{\sigma_1} x_{\sigma_1} \geq \dots \geq d_{\sigma_n} x_{\sigma_n} := (dx)_{(n)},$$

and  $\sigma$  is a permutation of  $\{1, \dots, n\}$ .

Different choices of  $\lambda$  yield different types of objective functions. To see that the ordered median objective function generalizes well-known objective functions, note that for  $\lambda = (1, 1, \dots, 1)$  the ordered objective function is equivalent to the median

objective; for  $\lambda = (1, 0, \dots, 0, 0)$  to the center problem; for  $\lambda = (1, \alpha, \dots, \alpha)$  with  $0 \leq \alpha \leq 1$  to the  $\alpha$ -centdian problem, a convex combination of the median and the center objective functions; and for  $\lambda = (1, \dots, 1, 0, \dots, 0)$ , where the first  $k$  entries are set to one and the last  $n - k$  entries are zero, is equivalent to the  $k$ -centrum problem. Other objectives may also be of practical interest. One example is to take  $\lambda = (0, \dots, 0, 1, \dots, 1, 0, \dots, 0)$ , where the first  $k_1$  and last  $k_2$  entries are zero. This means to average the “middle” part, the so-called  $(k_1 + k_2)$ -trimmed mean, which is a robust measure in statistics. Another example is to take  $\lambda = (1, \dots, 1, 0, \dots, 0, 1, \dots, 1)$ , where the first  $k_1$  and the last  $k_2$  entries are set to one: this leads to the problem of minimizing the sum of the  $k_1$  largest costs and the  $k_2$  smallest costs; the corresponding ordered objective function searches for minimizing the average of those costs that are very large and rather small. Clearly, classical objective functions can easily be cast under this common pattern. Moreover, new meaningful objective functions are easily derived, as shown above.

As mentioned at the beginning of this section, a natural question is whether the methodology developed in the previous sections can be extended to the ordered median optimization problem or, at least, to its convex version, namely, when  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ . The extension of this theory to those problems would imply the polynomial complexity for solving them whenever the corresponding minisum problems are also polynomially solvable. Scanning the specialized literature one can find some partial answers. For instance, it is known that the bottleneck and lexicographic bottleneck models, see Burkard and Rendl [8], Della Croce et al. [11], are polynomially solvable. Moreover, minimizing the balance or range (max-min) is also efficiently solvable by enumerating all the possible combinations of the maximum and the minimum, and then for each such combination a feasibility problem is solved, see Martello et al. [28].

### 6.1 Convex case $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$

Assuming a convex ordered median vector, i.e.,  $\lambda_1 \geq \dots \geq \lambda_n \geq 0$ , the result in Bertsimas and Sim [6], Bertsimas and Weismantel [7], Punnen and Aneja [38] can be almost directly extended to provide polynomial algorithms whenever the number of distinct components of  $\lambda$  is constant. Observe that this assumption is satisfied by several well-known problems in facility location, namely minisum (median), maximum (bottleneck or center), centdian,  $k$ -sum ( $k$ -centrum), et cetera. Although in general the number of different lambda values could be  $n$ , in most of the cases mentioned above, the representation of the problems given by the convex ordered median objective function has at most two different lambda values (minisum, maximum, centdian or  $k$ -centrum). Before we give the formulation of the problem under study, we recall a reformulation of expression (10) for the convex case (see [32] for further details):

$$\sum_{j=1}^n \lambda_j d_{\bar{\sigma}_j} = \max_{\sigma \in \mathcal{P}(1, \dots, n)} \sum_{j=1}^n \lambda_j d_{\sigma_j}$$

where  $\mathcal{P}(1, \dots, n)$  is the set of permutations of  $\{1, \dots, n\}$  and  $\bar{\sigma} \in \mathcal{P}(1, \dots, n)$ , such that,  $d_{\bar{\sigma}_1} x_{\bar{\sigma}_1} \geq \dots \geq d_{\bar{\sigma}_n} x_{\bar{\sigma}_n}$ .

Applying this transformation, the formulation of the problem under study is:

$$\min_{x \in X} \left\{ cx + \max_{\sigma \in \mathcal{P}(1, \dots, n)} \left\{ \sum_{j=1}^n \lambda_j d_{\sigma_j} x_{\sigma_j} \right\} \right\}, \tag{11}$$

The above formulation can be rewritten in an equivalent way, using  $\lambda_{n+1} := 0$ ,

$$\min_{x \in X} \left\{ cx + \max_{\sigma \in \mathcal{P}(1, \dots, n)} \left\{ \sum_{k=1}^n (\lambda_k - \lambda_{k+1}) \sum_{j=1}^k d_{\sigma_j} x_{\sigma_j} \right\} \right\}. \tag{12}$$

This problem can be formulated as a function of the sum of the  $k$ -largest distances:

$$\min_{x \in X} \left\{ cx + \sum_{k=1}^n (\lambda_k - \lambda_{k+1}) \max \left\{ \sum_{j \in S_k} d_j x_j : S_k \subseteq \{1, \dots, n\}, |S_k| = k \right\} \right\}.$$

Therefore, the problem above can be reformulated as:

$$\begin{aligned} Z_X(r_1, \dots, r_n) := \min \quad & cx + \sum_{k=1}^n (\lambda_k - \lambda_{k+1}) \left( kt_k + \sum_{j=1}^n p_{jk} \right) \\ \text{s.t.} \quad & p_{jk} \geq d_j x_j - r_k, \quad j, k = 1, \dots, n, \\ & p_{jk} \geq 0, \quad j, k = 1, \dots, n, \\ & x \in X. \end{aligned}$$

Again, this problem can be reformulated as:

$$\min_{x \in X, (r_1, \dots, r_n) \in \mathbb{R}^n} cx + \sum_{k=1}^n (\lambda_k - \lambda_{k+1}) \left( kt_k + \sum_{j=1}^n \max\{0, d_j x_j - r_k\} \right). \tag{13}$$

Observe that the main difference of this formulation with respect to the one given by (2) for the  $k$ -centrum problem is that the former depends on an  $n$ -dimensional vector  $r = (r_1, \dots, r_n)$  and the latter depends on a scalar  $r$ . If we have  $k_0$  different values of the vector  $\lambda$ , then formulation (13) would depend only on a  $k_0$ -dimensional vector.

**Theorem 6.1** *If the number of different values among the components of the vector  $\lambda = (\lambda_1, \dots, \lambda_n)$  is constant, say  $k_0$ , we have that*

1. *The discrete convex ordered median problem can be solved in  $O(n^{k_0} T_C(n, m))$  time, where  $T_C(n, m)$  is the combinatorial complexity of evaluating  $Z_{X_C}(r_1, \dots, r_{k_0})$  for any vector  $(r_1, \dots, r_{k_0}) \in \mathbb{R}^{k_0}$ .*
2. *Let  $T_L(n, m)$  be the combinatorial complexity of evaluating  $Z_{X_L}(r_1, \dots, r_{k_0})$  and  $C(n, m)$  be the number of comparisons of the algorithm for evaluating*

$Z_{X_L}(r_1, \dots, r_{k_0})$ . Suppose that  $C(n, m)$  can be divided into  $s$  phases, where  $C_i(n, m)$  independent comparisons are performed in each phase  $i = 1, \dots, s$ , and let  $\gamma(k_0) = 3^{O(k_0^2)}$ .

The continuous convex ordered median problem can be solved in time bounded above by

$$O\left(k_0^3 \gamma(k_0) T_L(n, m) \left(\sum_{i=1}^s \lceil \log C_i(n, m) \rceil\right)^{k_0}\right).$$

*Proof* The complexity in (1) of Theorem 6.1, follows from a straightforward application of Theorem 5.1, by solving  $n^{k_0}$  sum problems on  $X_C$ .

The complexity of (2) is obtained by applying the multi-parametric approach by Cohen and Megiddo [9] on  $X_L$ . □

A first application of this result is to the multifacility ordered median location problem with two different lambda values  $a > b$  on fork-free graphs. (See Baïou et al. [5] for a definition of fork-free graphs.) The problem can be formulated as:

$$\min_{x \in X_{med(p)}, (r_1, \dots, r_{k_0}) \in \mathbb{R}^{k_0}} cx + \sum_{k: \lambda_k \neq \lambda_{k+1}} (a - b)(kt_k + \sum_{j=1}^n \max\{0, d_j x_j - r_k\}), \tag{14}$$

where  $X_{med(p)}$ , defined by (5)–(8), is the  $p$ -median polytope on the augmented graph that includes as new nodes the elements of an FDS for this problem (recall that an FDS, finite dominating set, is a finite cardinality set containing a solution of the problem), which is known to have a cardinality of  $O(n^4)$ , Kalcsics et al. [25]. It is well-known that the  $p$ -median polytope, defined on a general graph, is not integer. Nevertheless, it is integer on the class of fork-free graphs (which includes among others, paths and stars), see e.g. Baïou et al. [5].

The above representation allows us to apply the second part of Theorem 6.1. In this case, the complexity of solving the problem for a given set of parameters, applying the algorithm in Tamir et al. [45], is  $pn^8$  (using that the cardinality of an FDS for this problem is  $O(n^4)$ ). Each application of that algorithm can be decomposed into  $n$  phases (one per demand point) and it requires  $O(n^4)$  comparisons. Therefore, applying Theorem 6.1 the overall complexity for solving the problem is  $O(pn^{10}(\log n)^2)$  time.

**Proposition 6.1** *The  $p$ -facility convex ordered median problem with only two distinct lambda values  $a > b \geq 0$  on fork-free graphs can be solved in  $O(pn^{10}(\log n)^2)$  time.*

Another application of Theorem 6.1 is to provide a strong polynomiality result for the convex ordered minimum cost flow problem with a constant number  $k_0$  of different lambda values. Indeed, using the representation given in (13), for each choice of  $(r_1, \dots, r_{k_0})$  the convex ordered minimum cost flow problem can be reduced to the solution of a minimum cost flow problem where the cost on each arc is a piecewise linear convex function of  $k_0$  pieces. This problem can be solved in strongly polynomial

time, Ahuja et al. [1], and therefore by an application of Theorem 6.1 we get the following result.

**Proposition 6.2** *The convex ordered minimum cost flow problem with lambda vector having a constant number  $k_0$  of different values can be solved in strongly polynomial time.*

We consider now the case where the number of different values of the vector  $\lambda$  is not fixed, i.e., it is part of the input. In this situation, the above analysis is not applicable but still we can give some polynomiality result.

**Proposition 6.3** *The continuous convex ordered median problem on the polytope  $X$  can be solved in polynomial time.*

*Proof* We observe that the inner maximum problem in (11) can be rewritten as:

$$\max_{\sigma \in \mathcal{P}(1, \dots, n)} \sum_{j=1}^n \lambda_j d_{\sigma_j} x_{\sigma_j} = \max \left\{ \sum_{i=1}^n \sum_{j=1}^n \lambda_j d_i x_i p_{ij} : \sum_{i=1}^n p_{ij} = 1, \forall j; \sum_{j=1}^n p_{ij} = 1, \forall i \right\}.$$

Hence, dualizing the maximum problem in the right-hand-side of the above equation, (11) is equivalent to:

$$\begin{aligned} \min \quad & c'x + \sum_{i=1}^n u_i + \sum_{j=1}^n v_j \\ \text{s.t.} \quad & u_i + v_j \geq \lambda_j d_i x_i, \quad \forall i, j = 1, \dots, n, \\ & x \in X. \end{aligned}$$

The above is a linear programming problem that can be solved in polynomial time, and thus the result follows. (It is still unknown whether a strongly polynomial algorithm exists.) □

### 6.2 The general case

The ordered median problem in its general case does not assume any sorting in the lambda weights. The formulation of the problem is

$$\min_{x \in X} \left\{ cx + \left\{ \sum_{j=1}^n \lambda_j d_{\sigma_j} x_{\sigma_j} : d_{\sigma_1} x_{\sigma_1} \geq \dots \geq d_{\sigma_n} x_{\sigma_n} \right\} \right\}. \tag{15}$$

However the transformation (12) used in the convex case, based on the representation of the  $k$ -sum problem, does not apply once some negative coefficient appears.

Let  $S_k(dx) = \sum_{j=1}^k d_{\sigma_j} x_{\sigma_j} : d_{\sigma_1} x_{\sigma_1} \geq \dots \geq d_{\sigma_n} x_{\sigma_n}, k = 1, \dots, n$ . Next, the general case can be formulated as:

$$\min_{x \in X} \left\{ cx + \sum_{k=1}^n (\lambda_k - \lambda_{k+1}) S_k(dx) \right\}. \tag{16}$$

This problem can be rewritten as:

$$\min_{x \in X} \left\{ cx + \sum_{k: \lambda_k - \lambda_{k+1} > 0} (\lambda_k - \lambda_{k+1}) S_k(dx) + \sum_{k: \lambda_k - \lambda_{k+1} < 0} (\lambda_k - \lambda_{k+1}) S_k(dx) \right\}.$$

Therefore, using again the transformation developed in Sect. 6.1 for the  $k$ -sum terms with positive coefficient, this problem can be formulated as:

$$\begin{aligned} Z_X(r_1, \dots, r_n) &:= \min cx + \sum_{k: \lambda_k - \lambda_{k+1} > 0} (\lambda_k - \lambda_{k+1}) \left( kt_k + \sum_{j=1}^n p_{jk} \right) \\ &+ \sum_{k: \lambda_k - \lambda_{k+1} < 0} (\lambda_k - \lambda_{k+1}) S_k(dx) \\ \text{s.t. } &p_{jk} \geq d_j x_j - r_k, \quad j, k = 1, \dots, n, \lambda_k - \lambda_{k+1} > 0, \\ &p_{jk} \geq 0, \quad j, k = 1, \dots, n, \lambda_k - \lambda_{k+1} > 0, \\ &x \in X. \end{aligned}$$

To proceed further we need a representation of the problem that minimizes the negation of the sum of the  $k$ -largest  $d$  cost coefficients, namely

$$\begin{aligned} \min_{x \in X} -S_k(dx) &= \min_{x \in X} \min \sum_{j=1}^n -d_j x_j y_j \\ &\text{s.t. } \sum_{j=1}^k y_j = k, \\ &0 \leq y_j \leq 1, \quad \forall j = 1, \dots, n, \\ &= \min_{x \in X} \max \left( kr_k + \sum_{j=1}^n p_j \right) \\ &\text{s.t. } r_k + p_j \leq -d_j x_j, \quad \forall j = 1, \dots, n, \\ &p_j \leq 0, \quad \forall j = 1, \dots, n. \end{aligned}$$

Plugging this formulation into the expression of  $Z_X(r_1, \dots, r_n)$  above, results in:

$$\begin{aligned} \min_{x \in X, (r_1, \dots, r_n) \in \mathbb{R}^n} &\left\{ cx + \sum_{k: \lambda_k - \lambda_{k+1} > 0} (\lambda_k - \lambda_{k+1}) \left( kr_k + \sum_{j=1}^n \max\{0, d_j x_j - r_k\} \right) \right. \\ &+ \left. \sum_{k: \lambda_{k+1} - \lambda_k < 0} (\lambda_{k+1} - \lambda_k) \max \left( kr_k + \sum_{j=1}^n p_{jk} \right) \right\} \\ \text{s.t. } &r_k + p_{jk} \leq -d_j x_j, \quad j, k = 1, \dots, n, \lambda_k - \lambda_{k+1} < 0, \\ &p_{jk} \leq 0, \quad j, k = 1, \dots, n, \lambda_k - \lambda_{k+1} < 0. \end{aligned} \tag{17}$$

Observe that the main difference of this formulation, compared with the one given for the convex ordered median problem (13) is that another valid expression for the terms with negative coefficients is necessary. In spite of that, the evaluation of  $Z_X(r_1, \dots, r_n)$  given by (17) is performed by solving a multiparametric program, but in this case  $Z_X$  is not convex, even in the continuous case. Therefore, results similar to that of Theorem 6.1 may also hold for the general case provided that an efficient oracle is available for the evaluation of  $Z_X$ .

### 6.2.1 Minimizing the middle range problem

One application of the above result is the minimization of the middle range problem. This problem can be stated as finding the solution that minimizes the sum of the  $n - k_1 - k_2$  central sorted values of the vector  $(d_j x_j)_{j=1, \dots, n}$ . It can be cast for the non-convex case of the ordered median objective with  $\lambda = (0, \overset{k_1}{\cdot}, 0, 1, \dots, 1, 0, \overset{k_2}{\cdot}, 0)$ . Let  $k_3 = n - k_2$ . For a given  $x \in \{0, 1\}^n$ , let  $\sum_{i=1}^n \lambda_i d_j x_j = S_{k_3}(dx) - S_{k_1}(dx)$ . The optimization problem is then defined by:

$$\begin{aligned} &\min S_{k_3}(dx) - S_{k_1}(dx) \\ &s.t. \ x \in X, \end{aligned}$$

or equivalently

$$\min_{x \in X} \min_{y_j + t \geq d_j x_j; y_j, t \geq 0 \forall j} k_3 t + \sum_{i=1}^n y_j - \max_{\sum_{j=1}^n v_j = k_1, 0 \leq v_j \leq x_j \leq 1, \forall j} \sum_{j=1}^n v_j d_j.$$

It can be rewritten as:

$$\begin{aligned} &\min k_3 t + \sum_{i=1}^n y_j - \sum_{j=1}^n v_j d_j \\ &s.t. \ x \in X, \\ &\quad y_j \geq d_j x_j - t, \quad \forall j = 1, \dots, n, \\ &\quad \sum_{j=1}^n v_j = k_1, \\ &\quad v_j \leq x_j, \quad \forall j = 1, \dots, n, \\ &\quad y_j, t, v_j \geq 0, \quad \forall j = 1, \dots, n. \end{aligned}$$

Let us assume w.l.o.g. that  $d_1 \geq \dots \geq d_n$ . Following the discussion in the previous section, we observe that for any  $t \in [d_\ell, d_{\ell-1}]$ , we have the following equivalent formulation of the above problem:

$$\begin{aligned}
 & k_3 d_\ell + \min \sum_{j=1}^{\ell-1} (d_j(1 - v_j) - d_\ell)x_j \\
 & \text{s.t.} \quad x \in X, \\
 & \quad \sum_{j=1}^n v_j = k_1, \\
 & \quad 0 \leq v_j \leq x_j, \quad j = 1, \dots, n.
 \end{aligned} \tag{18}$$

Therefore, solving the middle range problem is equivalent to solving the above problem for the different intervals  $[d_\ell, d_{\ell-1}]$  for all  $\ell = 2, \dots, n$ . The question is how to solve the above problem. For each  $d_\ell$  one must consider the  $\binom{\ell-1}{k_1}$  different forms of fixing the  $v$ -variables and for each one of them solve the resulting linear problem (18) with those variables already fixed. Therefore the overall complexity is  $O(T(n)(\#D)^{k_1})$ , where  $T(n)$  is the complexity of solving the corresponding minimum problems that appear in (18) and  $\#D$  is the number of distinct values for  $\{d_j\}$ . Clearly, this approach is in general non polynomial. If we assume that the number  $k_1$  of upper trimmed components is fixed then the above approach is also polynomial.

## 7 Concluding remarks

An interesting case that falls within the framework considered in this paper and deserves a bit of attention is the case where  $X$  is a matroid. Now, the  $k$ -sum problem looks for a base that minimizes the sum of the  $k$ -largest costs. For a matroidal system any ordered median function (not only the  $k$ -sum function) with non negative  $\lambda$ -weights is optimized by the base that optimizes the minimum problem. This result was already observed by Fernández et al. [12], and follows from the isotonicity of the ordered median function (see, for instance Theorem 6.1, page 276 in [27]). It applies, among many others, to the problem of minimizing the middle range problem on a matroid, see Sect. 6.2.1. In addition, one can also observe that this result is still applicable when  $X$  is an extended Polymatroid, see Theorem 10.1.6. in Grötschel et al. [17].

The case where some  $\lambda$ -weights could be negative is not so clear and needs a deeper analysis. In the following we report some partial answers. We can solve the corresponding ordered median problem on matroidal systems with a constant number of coefficient values using separators and matroid intersection algorithms. We illustrate this approach with the problem of minimizing the sum of the  $k$ -largest minus the  $t$  smallest elements. In general, we do not know what is the complexity of this problem. Nevertheless, the latter model can be efficiently solved for matroidal systems using separators. Suppose that  $d_1 \geq d_2 \geq \dots \geq d_m$  are the values of the coefficients of the ground set  $E$ . We solve  $O(m^2)$  subproblems for each pair  $i < j$ . Consider the three subsets:  $\mathbb{E}_1 = \{e_1, \dots, e_i\}$ ,  $\mathbb{E}_2 = \{e_{i+1}, \dots, e_{i+j}\}$  and  $\mathbb{E}_3 = \{e_{i+j+1}, \dots, e_m\}$ .

With each  $e_k \in E_1$  associate a coefficient  $d_k$ , with each  $e_k \in E_2$  associate the coefficient 0, and with each  $e_k \in E_3$  associate a coefficient  $-d_k$ .



Now, for each fixed  $i < j$ , we use matroid intersection to find an optimal base of cardinality  $\bar{n}$ , w.r.t. these weights which contain at most  $k$  elements from  $E_1$ , at most  $t$  elements from  $E_3$ , and at most  $\bar{n} - k - t$  from  $E_2$ .

The above analysis extends further to any ordered median problem with a constant number of distinct coefficient values. Since it translates into solving a polynomial number of problems with a fixed number of separators.

**Acknowledgements** This research has been partially supported by Spanish Ministry of Education and Science/FEDER Grants Numbers MTM2013-46962-C02-(01-02), MTM2016-74983-C2-(1,2)-R, Junta de Andalucía Grant Number FQM 05849 and Fundación Séneca, Grant Number 08716/PI/08.

## References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows. Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs (1993)
2. Alstrup, S., Lauridsen, P., Sommerlund, P., Thorup, M.: Finding cores of limited length. In: Dehne, F., Rau-Chaplin, A., Sack, J.-R., Tamassia, R. (eds.) Algorithms and Data Structures, Lecture Notes in Computer Science n 1272, pp. 45–54. Springer, Heidelberg (1997)
3. Apollonio, N., Lari, I., Ricca, F., Simeone, B., Puerto, J.: Polynomial algorithms for partitioning a tree into single-center subtrees to minimize flat service costs. *Networks* **51**(1), 78–89 (2008)
4. Averbakh, I., Berman, O.: Algorithms for path medi-centers of a tree. *Comput. Oper. Res.* **26**(14), 1395–1409 (1999)
5. Baïou, M., Barahona, F., Correa, J.: On the  $p$ -median polytope of fork-free graphs. *Electron. Notes Discrete Math.* **36**, 143–149 (2010)
6. Bertsimas, D., Sim, M.: Robust discrete optimization and network flows. *Math. Program.* **98**(1–3, Ser. B), 49–71 (2003)
7. Bertsimas, D., Weismantel, R.: Optimization Over Integers. Dynamic Ideas, Belmont (2005)
8. Burkard, R.E., Rendl, F.: Lexicographic bottleneck problems. *Oper. Res. Lett.* **10**(5), 303–308 (1991)
9. Cohen, E., Megiddo, N.: Maximizing concave functions in fixed dimension. In: Pardalos, P.M. (ed.) Complexity in Numerical Optimization, pp. 74–87. World Scientific Publishing, River Edge (1993)
10. Cole, R.: Slowing down sorting networks to obtain faster sorting algorithms. *J. Assoc. Comput. Mach.* **34**(1), 200–208 (1987)
11. Della Croce, F., Paschos, V.T., Tsoukias, A.: An improved general procedure for lexicographic bottleneck problems. *Oper. Res. Lett.* **24**(4), 187–194 (1999)
12. Fernández, E., Puerto, J., Rodríguez-Chía, A.M.: On discrete optimization with ordering. *Ann. Oper. Res.* **207**(1), 83–96 (2013)
13. Fernández, E., Pozo, M.A., Puerto, J.: Ordered weighted average combinatorial optimization: formulations and their properties. *Discrete Appl. Math.* **169**, 97–118 (2014)
14. Gabow, H.N., Galil, Z., Spencer, T.H.: Efficient implementation of graph algorithms using contraction. *J. Assoc. Comput. Mach.* **36**(3), 540–572 (1989)
15. Garfinkel, R., Fernández, E., Lowe, T.J.: The  $k$ -centrum shortest path problem. *Top* **14**(2), 279–292 (2006)
16. Granot, D., Granot, F., Ravichandran, H.: The  $k$ -centrum Chinese postman delivery problem and a related cost allocation game. *Discrete Appl. Math.* **179**, 100–108 (2014)
17. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, volume 2 of Algorithms and Combinatorics: Study and Research Texts. Springer, Berlin (1988)
18. Gupta, S.K., Punnen, A.P.:  $k$ -sum optimization problems. *Oper. Res. Lett.* **9**(2), 121–126 (1990)
19. Hakimi, S., Schmeichel, E., Labbé, M.: On locating path- or tree-shaped facilities on networks. *Networks* **23**(6), 543–555 (1993)
20. Halpern, J.: Finding minimal center-median convex combination (cent-dian) of a graph. *Manag. Sci.* **24**, 535–544 (1978)
21. Handler, G.Y.: Medi-centers of a tree. *Transp. Sci.* **19**(3), 246–260 (1985)
22. Hassin, R., Tamir, A.: Improved complexity bounds for location problems on the real line. *Oper. Res. Lett.* **10**(7), 395–402 (1991)

23. Kalcsics, J.: Improved complexity results for several multifacility location problems on trees. *Ann. Oper. Res.* **191**, 23–36 (2011)
24. Kalcsics, J., Nickel, S., Puerto, J., Tamir, A.: Algorithmic results for ordered median problems. *Oper. Res. Lett.* **30**(3), 149–158 (2002)
25. Kalcsics, J., Nickel, S., Puerto, J.: Multifacility ordered median problems on networks: a further analysis. *Networks* **41**(1), 1–12 (2003)
26. Kim, T.U., Lowe, T.J., Tamir, A., Ward, J.E.: On the location of a tree-shaped facility. *Networks* **28**(3), 167–175 (1996)
27. Lawler, E.: *Combinatorial Optimization: Networks and Matroids*. Saunders College Publishing, Fort Worth (1976)
28. Martello, S., Pulleyblank, W.R., Toth, P., de Werra, D.: Balanced optimization problems. *Oper. Res. Lett.* **3**(5), 275–278 (1984)
29. Megiddo, N.: Combinatorial optimization with rational objective functions. *Math. Oper. Res.* **4**, 414–424 (1979)
30. Megiddo, N.: Applying parallel computation algorithms in the design of serial algorithms. *J. Assoc. Comput. Mach.* **30**(4), 852–865 (1983)
31. Mesa, J.A., Puerto, J., Tamir, A.: Improved algorithm for several network location problems with equality measures. *Discrete Appl. Math.* **130**, 437–448 (2003)
32. Nickel, S., Puerto, J.: *Location theory. A unified approach*. Springer, Berlin (2005)
33. Ogryczak, W., Tamir, A.: Minimizing the sum of the  $k$  largest functions in linear time. *Inf. Process. Lett.* **85**(3), 117–122 (2003)
34. Puerto, J., Fernández, F.R.: Geometrical properties of the symmetrical single facility location problem. *J. Nonlinear Convex Anal.* **1**(3), 321–342 (2000)
35. Puerto, J., Rodríguez-Chía, A. M.: Ordered median location problems. In: Gilbert, L., Stefan, N., Saldanha da Gama, F. (eds.) *Location Science*, pp. 249–288. Springer, New York (2015)
36. Puerto, J., Tamir, A.: Locating tree-shaped facilities using the ordered median objective. *Math. Program.* **102**(2 (A)), 313–338 (2005)
37. Punnen, A.P.:  $k$ -sum linear programming. *J. Oper. Res. Soc.* **43**(4), 359–263 (1992)
38. Punnen, A.P., Aneja, Y.P.: On  $k$ -sum optimization. *Oper. Res. Lett.* **18**(5), 233–236 (1996)
39. Sharir, M., Agarwal, P.K.: *Davenport–Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, Cambridge (1995)
40. Slater, P.J.: Centers to centroids in graphs. *J. Graph Theory* **2**(3), 209–222 (1978)
41. Slater, P.J.: The  $k$ -nucleus of a graph. *Networks* **11**(3), 233–242 (1981)
42. Tamir, A.: An  $O(pn^2)$  algorithm for the  $p$ -median and related problems on tree graphs. *Oper. Res. Lett.* **19**(2), 59–64 (1996)
43. Tamir, A.: Fully polynomial approximation schemes for locating a tree-shaped facility: a generalization of the knapsack problem. *Discrete Appl. Math.* **87**(1–3), 229–243 (1998)
44. Tamir, A.: The  $k$ -centrum multi-facility location problem. *Discrete Appl. Math.* **109**(3), 293–307 (2001)
45. Tamir, A., Pérez-Brito, D., Moreno-Pérez, J.A.: A polynomial algorithm for the  $p$ -centdian problem on a tree. *Networks* **32**(4), 255–262 (1998)
46. Tamir, A., Puerto, J., Pérez-Brito, D.: The centdian subtree of a tree network. *Discrete Appl. Math.* **118**(3), 263–278 (2002)
47. Tardos, É.: A strongly polynomial minimum cost circulation algorithm. *Combinatorica* **5**(3), 247–255 (1985)
48. Zemel, E.: An  $O(n)$  algorithm for the linear multiple choice knapsack problem and related problems. *Inf. Process. Lett.* **18**(3), 123–128 (1984)