

Using ℓ^p -norms for Fairness in Combinatorial Optimisation

Tolga Bektas* Adam N. Letchford†

August 2019; Revised March 2020, April 2020

Abstract

The issue of fairness has received attention from researchers in many fields, including combinatorial optimisation. One way to drive the solution toward fairness is to use a modified objective function that involves so-called ℓ^p -norms. If done in a naive way, this approach leads to large and symmetric mixed-integer nonlinear programs (MINLPs), that may be difficult to solve. We show that, for some problems, one can obtain alternative MINLP formulations that are much smaller, do not suffer from symmetry, and have a reasonably tight continuous relaxation. We give encouraging computational results for certain vehicle routing, facility location and network design problems.

Keywords: Fairness, mixed-integer nonlinear programming, vehicle routing, facility location, network design.

1 Introduction

The issue of fairness has received considerable attention from researchers in many fields, including, for example, computer science [28], economics [42, 48, 51], marketing [52], operational research [11, 33], philosophy [49], psychology [29] and recreational mathematics [18]. As one might expect, it has also received attention from the combinatorial optimisation community (see, e.g., [2, 5, 11, 19, 20, 31, 32, 34, 38–41, 43, 44, 46, 47, 53]).

As noted in [20], many combinatorial optimisation problems (COPs) of interest can be modelled as follows. We have a set of *workers* and a set of *tasks*. Each task must be assigned to one worker, and the total cost depends on the allocation of tasks to workers. The issue of fairness then arises immediately, since, if one worker has a significantly higher workload than another, the solution may be perceived to be unfair.

*Department of Operations and Supply Chain Management, University of Liverpool, United Kingdom. Email: t.bektas@liverpool.ac.uk

†Department of Management Science, Lancaster University, United Kingdom. Email: a.n.letchford@lancaster.ac.uk

A variety of approaches have been proposed to avoid unfair solutions, without incurring a significant increase in cost. (We survey these in Subsection 2.2.) In this paper, we focus on an approach suggested in [34], that uses a modified objective function involving so-called ℓ^p -norms. We make the following specific contributions:

- We give an intuitive explanation for why the ℓ^p -norm approach tends to lead to fairer solutions.
- We present a generic formulation for ℓ^p -norm problems as convex *mixed-integer nonlinear programs* (MINLPs).
- For some specific COPs, we present alternative convex MINLP formulations that are much smaller, do not suffer from symmetry, and have a reasonably tight continuous relaxation.
- We give some encouraging computational results for some simple vehicle routing, facility location and network design problems.

The paper has a simple structure. The literature is reviewed in Section 2. The intuitive explanation and generic formulation are presented in Section 3. The specialised formulations are described in Section 4. The computational results are given in Section 5, and some concluding remarks are made in Section 6.

2 Literature Review

We now review the relevant literature. Subsection 2.1 covers fairness in general, and Subsection 2.2 covers fairness in combinatorial optimisation.

2.1 Fairness in general

As mentioned in the introduction, the literature on fairness is vast. For brevity, we mention here seven key works, each of which studies fairness in a quantitative manner. Throughout, we assume that there are n people and, for $i = 1, \dots, n$, we let x_i denote the wealth allocated to the i th person. (The vector $x \in \mathbb{R}_+^n$ goes by various names, such as the *payoff*, *wealth allocation* or *resource allocation* vector.) The seven works in question, in chronological order, are as follows.

1. Gini [22] proposed to measure the unfairness of a given vector $x \in \mathbb{R}_+^n$ by computing the following “index”:

$$\frac{\sum_{1 \leq i < j \leq n} |x_i - x_j|}{(n-1) \sum_{i=1}^n x_i}.$$

It takes the value 1 when the one person has all of the resource and the rest have none, and it takes the value 0 when the resource is shared equally between all people.

2. Nash [42] considered the following simple game. There are two players. A bounded and convex set $C \subset \mathbb{R}_+^2$ of possible payoff vectors is given. The players have to agree to select a single vector $x \in C$. For $i = 1, 2$, the utility function of player i is concave and is denoted by U^i . Nash showed that a vector is both fair and stable if and only if it is the (unique) vector in C that maximises $U(x_1)U(x_2)$.
3. Rawls [49] considered a political problem in which one must simultaneously decide how to produce goods and how to distribute them among the people. He argued that a solution is “fair” if and only if it maximises the wealth of the worst-off person in the society. In our notation, the solution must maximise $\min_{1 \leq i \leq n} \{x_i\}$. (In other words, Rawls was effectively arguing for the use of a “max-min” objective function.)
4. Weymark [51] proposed a different way to measure the wealth of a society. Let $\tilde{x}_1, \dots, \tilde{x}_n$ denote the components of x , but sorted in non-increasing order. Also let $\lambda_1, \dots, \lambda_n$ be non-negative and non-decreasing “weights”. Weymark proposed to use the index

$$\sum_{i=1}^n \lambda_i \tilde{x}_i.$$

For example, if we set all weights to $1/n$, we obtain the mean wealth; and if we set λ_n to 1 and all other weights to 0, we obtain the minimum wealth.

5. Jain *et al.* [28] proposed to measure the fairness of a given vector $x \in \mathbb{R}_+^n$ by computing the following index:

$$\frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2}.$$

It takes the value $1/n$ when the one customer has all of the resource and the rest have none, and it takes the value 1 when the resource is shared equally between all customers.

6. Bertsekas and Gallego [9] considered a problem in which we must determine the wealth allocation vector x , subject to the constraint that x lies in some given bounded and convex subset $C \subset \mathbb{R}_+^n$. They propose to maximise the minimum payoff, then maximise the second smallest, and so on, in lexicographic fashion.

7. Kelly [33] considered the same problem. He calls a vector $x \in C$ “proportionally fair” if, for all other vectors $x' \in C$, we have:

$$\sum_{i=1}^n \frac{x'_i - x_i}{x_i} \leq 0.$$

He then shows that a proportionally fair vector can be found by solving a convex optimisation problem.

We remark that all of these works concentrate on the case in which profits, rather than costs, must be shared. Oddly enough, most of the works on fairness in combinatorial optimisation concentrate on costs rather than profits. This will become apparent in the next subsection.

2.2 Fairness in combinatorial optimisation

There is by now a considerable literature on fairness in combinatorial optimisation. Two application areas that have received particular attention are facility location and vehicle routing; see, e.g., [5, 44, 46] for good surveys on the former and Matl *et al.* [40] for a good survey on the latter. Other application areas that have received attention include, e.g., resource allocation [13, 38], machine scheduling [2], air traffic flow management [10], nurse rostering [39], examination timetabling [41] and flows in telecommunications networks [9, 53].

As noted in [32, 34], many of the COPs addressed in the literature fall into the following general framework. We have a set W of *workers* and a set T of *tasks*. Each task must be assigned to exactly one worker. If worker w is assigned a set $S \subseteq T$ of tasks, then a cost $c_w(S)$ is incurred. (If it is impossible to assign all of the tasks in S to worker w , then $c_w(S)$ is infinity.) We seek an assignment of tasks to workers that minimises the total cost. That is, we wish to solve:

$$\min \left\{ \sum_{w \in W} c_w(S_w) : \bigcup_{w \in W} S_w = T, S_i \cap S_j = \emptyset \ (\{i, j\} \subseteq W) \right\},$$

where S_w is the set of tasks assigned to worker w .

Let us call $c_w(S_w)$ the “workload” of worker w . If, in a given feasible solution, one worker has a significantly higher workload than another, then the solution may be perceived as unfair. Ways to address this issue include the following:

- minimise the maximum workload (e.g., [1, 4, 15, 37]);
- minimise a convex combination of the mean and maximum workloads [26].
- minimise the sum of the k largest workloads [20, 25, 50];

- minimise the maximum workload, then the second highest, and so on, in lexicographic fashion [38];
- minimise a convex combination of the mean workload and the difference between the maximum and minimum workloads [30];
- impose an upper bound on the workloads [36];
- impose both lower and upper bounds on the workloads [8, 53],
- minimise the sum of the squared workloads [31, 39];
- minimise the ℓ^p -norm of the workload vector for some $p > 1$ [13, 34], i.e., the quantity

$$\left(\sum_{w \in W} c_w(S_w)^p \right)^{1/p}.$$

Unfortunately, most of these approaches require significant computational effort, and some of them can lead to solutions that have undesirable properties (see, e.g., [8, 27, 45, 46]). In this paper, we follow the last strategy which, as we will see, is computationally viable and usually leads to “acceptable” solutions.

Another popular approach is similar to the approach of Weymark [51], mentioned in the previous subsection, but adapted to the minimisation case. Let n be the number of workers. For a given feasible solution, let $\tilde{c}_1, \dots, \tilde{c}_n$ denote the workloads sorted in non-increasing order. Also let $\lambda_1, \dots, \lambda_n$ be non-negative and non-increasing weights. Then we minimise

$$\sum_{w=1}^n \lambda_w \tilde{c}_w.$$

One can check that this approach includes as special cases the first four approaches mentioned above. It is often called the *ordered weighted average* (OWA) approach (e.g., [5, 14, 19, 32, 34, 44]).

We remark that the ℓ^p -norm has some desirable properties (e.g., non-negativity, symmetry, strict monotonicity and strict quasi-convexity), and so does the OWA when all weights are positive; see [5, 32, 34] for details.

3 General Remarks on the ℓ^p -Norm Approach

In this section, we make some general remarks on the ℓ^p -norm approach. In Subsection 3.1, we try to explain why unfair solutions arise in the first place, and why the approach can help. In Subsections 3.2 and 3.3, we present generic MINLP and 0-1 LP formulations for COPs with ℓ^p -norm objective.

From now on, we say that the workers are *identical* if $c_w(S) = c_{w'}(S)$ for all pairs of workers w, w' and all sets $S \subseteq T$. In this case, we write $c(S)$ instead of $c_w(S)$.

3.1 Why the ℓ^p -norm approach can help

In our preliminary experiments, we found that using the traditional “min-sum” objective frequently leads to very unfair solutions. We do not have a full explanation for this phenomenon, but the following proposition provides some insight.

Proposition 1 *Suppose that the workers are identical, and that there exists a vector $v \in \mathbb{Q}_+^{|T|}$, a positive integer s , and a strictly concave increasing function $f : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ such that the following holds for all $S \subseteq T$:*

$$\begin{aligned} c(S) &= f\left(\sum_{t \in S} v_t\right) && (\text{if } |S| \leq s) \\ &= \infty && (\text{otherwise}). \end{aligned}$$

Then, in the optimal min-sum solution to the COP, all workers will be either idle or as busy as possible, with the possible exception of a single worker.

Proof. Consider a feasible solution to the COP. Suppose that there exists a pair of workers w, w' , each of which has been assigned between 1 and $s-1$ tasks. Without loss of generality, suppose that $\sum_{t \in S_w} v_t \geq \sum_{t \in S_{w'}} v_t$. Let t be a task that is currently assigned to worker w' . Since $f(\cdot)$ is increasing and strictly concave, if we take the task t and assign it to worker w instead, the total cost will decrease. Repeating this process, if necessary, leads to a solution with the stated property. \square

Of course, for most COPs arising in practice, the conditions in the proposition are unlikely to hold. On the other hand, in some COPs, it may well be that the functions $c_w(S)$, viewed as set functions, have some property analogous to concavity, such as subadditivity or submodularity.

The above proposition also provides some insight into the potential benefit of the ℓ^p -norm approach. Even if the function $f(\cdot)$ is strictly concave, there may well exist a small rational $p > 1$ such that $f(\cdot)^p$ is strictly convex. (For example, if $f(r) = \sqrt{r}$, then setting p to $2 + \epsilon$ will suffice.) Minimising this strictly convex function in place of the original will lead to solutions where $c(S_w)$ takes a similar value for each worker.

3.2 A generic MINLP formulation

We now define a broad family of COPs, in which, for any worker w and set of tasks S , the cost $c_w(S)$ can be computed by solving an integer linear program (ILP). (We remark that a similar family of COPs was presented in [12], in the context of cooperative game theory.)

Definition 1 *A COP with worker set W and task set T will be called “ILP-representable” if, for any given worker w , there exist*

- positive integers p_w and q_w , both bounded by a polynomial in $|W|$ and $|T|$,
- vectors $c^w \in \mathbb{Z}^{|T|}$ and $d^w \in \mathbb{Z}^q$,
- matrices $A^w \in \mathbb{Z}^{p_w \times |T|}$ and $E^w \in \mathbb{Z}^{p_w \times q_w}$,
- and a right-hand side vector $b^w \in \mathbb{Z}^{p_w}$,

such that, for each set $S \subseteq T$, the cost $c_w(S)$ is equal to the solution value of the following ILP:

$$\text{Min.} \quad c^w \cdot y + d^w \cdot z$$

$$\text{s.t.} \quad A^w y + E^w z \geq b^w$$

$$y_t = 1 \quad (t \in S) \quad (1)$$

$$y_t = 0 \quad (t \in T \setminus S) \quad (2)$$

$$z \in \mathbb{Z}_+^{q_w}.$$

(If this ILP is infeasible for some S , we set $c_w(S)$ to infinity for all w .)

To illustrate this concept, consider the following simple COP. We have a set W of workers and a set T of tasks. Each task must be assigned to one worker. If worker w does task t , it costs c_{tw} and takes q_{tw} units of time. Worker is contracted to work for Q_w time units. However, workers are willing to work overtime, at a cost of d per time unit. The task is to minimise the total cost. This COP is ILP-representable because, for each worker w and set S , the cost $c_w(S)$ is equal to:

$$\text{Min.} \quad \sum_{t \in T} c_{tw} y_t + dz$$

$$\text{s.t.} \quad \sum_{t \in T} q_{tw} y_t \leq Q_w + z$$

$$(1), (2)$$

$$z \in \mathbb{Z}_+.$$

Given an ILP-representable COP and a constant $p > 1$, the problem of finding a solution that minimises the ℓ^p -norm can be formulated as follows. For $w \in W$, let λ_w be a non-negative continuous variable, representing the workload of worker w ; also let \tilde{y}^w and \tilde{z}^w be “copies” of y and z , respectively. Then:

$$\text{Min.} \quad \sum_{w \in W} \lambda_w^p$$

$$\text{s.t.} \quad \sum_{w \in W} \tilde{y}_t^w = 1 \quad (t \in T) \quad (3)$$

$$\lambda_w \geq c^w \cdot \tilde{y}^w + d^w \cdot \tilde{z}^w \quad (w \in W)$$

$$A^w \tilde{y}^w + E^w \tilde{z}^w \geq b^w \quad (w \in W)$$

$$\lambda \in \mathbb{R}_+^{|W|}$$

$$\tilde{y}^w \in \{0, 1\}^{|T|} \quad (w \in W)$$

$$\tilde{z}^w \in \mathbb{Z}_+^{q_w} \quad (w \in W).$$

Note that the objective function is convex and the constraints are all linear. So, in principle, this problem can be tackled using any algorithm for convex MINLP, such as the ones surveyed in [16]. (In fact, if p is rational, then the problem can be converted into a mixed-integer *second-order cone* (SOC) program; see, e.g., Subsection 2.3 of [3].)

3.3 A generic 0-1 LP formulation

Unfortunately, the generic MINLP formulation has two big drawbacks. The first is its size, which grows with the number of workers. The second is the fact that, when workers are identical, it suffers from *symmetry*. (That is, given any feasible solution, one can obtain many other solutions of the same cost by permuting the worker indices.)

As explained in [6], one way to deal with these drawbacks is to apply *Dantzig-Wolfe decomposition* [17]. Specifically, we keep the constraints (3) in the master, and move the other constraints to the pricing subproblem.

For the case of identical workers, the resulting formulation is as follows. Let $\Omega \subseteq 2^T$ be the set of all feasible sets of tasks that can be handled by one worker. For each $S \in \Omega$, let μ_S be a binary variable, taking the value 1 if and only if a worker is assigned the tasks in S . Then we have:

$$\begin{aligned} \text{Min.} \quad & \sum_{S \in \Omega} c(S)^p \mu_S \\ \text{s.t.} \quad & \sum_{S \in \Omega} \mu_S = |W| \\ & \sum_{S \in \Omega: t \in S} \mu_S = 1 \quad (t \in T) \\ & \mu_S \in \{0, 1\} \quad (S \in \Omega). \end{aligned}$$

This is a pure 0-1 LP, but it has an exponentially large number of columns, meaning that one must use branch-and-price [6]. Moreover, the presence of the parameter p in the objective function means that the pricing subproblem is itself a nonlinear (though convex) ILP, which could be hard to solve in some cases.

When workers are not identical, a more complicated formulation is needed, with a separate set of columns for each worker. We omit details for brevity.

4 Compact Formulations for Specific Problems

It turns out that, for certain specific COPs, one can obtain alternative MINLP formulations that are both small and free from symmetry. In Subsections 4.1 to 4.3, we show this for the multiple TSP, the capacitated minimum spanning tree problem, and the k -median problem. Then, in Subsection 4.4, we discuss how to strengthen the continuous relaxations of the formulations.

4.1 The multiple travelling salesman problem

The *multiple travelling salesman problem* or m -TSP is a generalisation of the standard TSP (see, e.g., [7]). We have a complete directed graph G with vertex set $V = \{0, \dots, n\}$ and arc set A . Node 0 represents a depot and the other nodes represent customers. For each arc $(i, j) \in A$, the cost of travel from i to j is c_{ij} . We are also given an integer $m \geq 2$, which represents the number of vehicles. The problem calls for a minimum-cost set of m vehicle routes, each starting and ending at the depot, such that each non-depot node is visited by exactly one vehicle. Here, we view the vehicles as (identical) workers, and the non-depot nodes as the tasks. That is, $|W| = m$ and $|T| = n$.

One can check that, if we formulate the ℓ^p -norm version of the m -TSP using the generic approach described in Subsection 3.2, the resulting MINLP has at least $m(n + |A| + 1)$ variables. Here is a more compact formulation. For all $(i, j) \in A$, let x_{ij} be a binary variable taking the value 1 if and only if a vehicle travels from i to j . Also, for all arcs (i, j) , let z_{ij} be a non-negative continuous variable with the following interpretation: if $x_{ij} = 1$, then z_{ij} represents the distance travelled so far by the vehicle when it arrives at j . Otherwise z_{ij} takes the value 0. Finally, let M be any integer that is guaranteed to be larger than the cost of the most expensive vehicle route. We then have:

$$\begin{aligned} \text{Min.} \quad & \sum_{i=1}^n z_{i0}^p & (4) \\ \text{s.t.} \quad & \sum_{i=1}^n x_{i0} = m & (5) \\ & \sum_{j \neq i} x_{ij} = 1 & (i = 1, \dots, n) & (6) \\ & \sum_{j \neq i} x_{ji} = 1 & (i = 1, \dots, n) & (7) \\ & \sum_{j \neq i} z_{ij} = \sum_{j \neq i} z_{ji} + \sum_{j \neq i} c_{ij} x_{ij} & (i = 1, \dots, n) & (8) \\ & z_{0i} = c_{0i} x_{0i} & (i = 1, \dots, n) & (9) \\ & z_{ij} \leq M x_{ij} & ((i, j) \in A) & (10) \\ & x_{ij} \in \{0, 1\} & ((i, j) \in A) & (11) \\ & z_{ij} \geq 0 & ((i, j) \in A). & (12) \end{aligned}$$

The objective function (4) exploits the fact that, if a vehicle traverses the arc $(i, 0)$, then z_{i0} must equal the total distance travelled by the vehicle. Constraint (5) simply states that m vehicles must be used. Constraints (6) and (7) are the standard degree equations. Constraints (8)-(10) ensure that the z variables take the correct values. The remaining constraints are just binary and non-negativity conditions.

The formulation (4)-(12) has only $O(n^2)$ variables and constraints, and it does not suffer from symmetry. The objective function (4) is convex and the constraints (5)-(10) are linear.

4.2 The capacitated minimum spanning tree problem

The *capacitated minimum spanning tree problem* (CMSTP) arises in telecommunications (see, e.g., [23]). We are given a complete directed graph $G = (V, A)$ with $V = \{0, \dots, n\}$, and arc costs c_{ij} , exactly as in the m -TSP. (However, node 0 is now called the *root*.) We are also given a positive integer Q . The problem calls for a minimum-cost directed spanning tree (sometimes called an arborescence), formed by a union of rooted sub-trees, such that the number of non-root nodes in each sub-tree does not exceed Q .

To fit the CMSTP into our framework, we assume that we are also given an integer $m \geq 2$, and require exactly m rooted sub-trees. We can then view the sub-trees as (identical) workers and the non-root nodes as the tasks. That is, $|W| = m$ and $|T| = n$, as before.

As in the case of the m -TSP, if we use the generic approach in Subsection 3.2, the resulting MINLP will have $O(n^2m)$ variables and constraints. Here is a more compact formulation. Let $V_0 = V \setminus \{0\}$ and $V' = V_0 \cup \{d\}$ where d is a dummy node that is connected all nodes in V_0 with zero cost. For all $(i, j) \in A$, let x_{ij} be a binary variable taking the value 1 if and only if the tree contains an arc from i to j . Also, for all arcs (i, j) , let z_{ij} be a non-negative continuous variable with the following interpretation: if $x_{ij} = 1$, then z_{ij} represents the cost of the part of the sub-tree that enters node j . Otherwise z_{ij} takes the value 0. An auxiliary non-negative variable g_{ij} , defined for each arc $(i, j) \in A$, is used to ‘count’ the number of non-root nodes in each sub-tree using constraints that model the flow of a single-commodity. The variable g_{ij} is bounded by $Q - 1$ if the arc (i, j) is not connected to the root node, and is bounded by Q if connected to the root node. Finally, let M be any integer that is guaranteed to be larger than the cost of the most expensive sub-tree. We then have:

$$\begin{array}{ll}
\text{Min.} & \sum_{i \in V_0} z_{0i}^p \\
\text{s.t.} & \sum_{i \in V_0, i \neq j} x_{ij} = 1 \quad (j \in V_0) \\
& \sum_{i \in V_0} x_{0i} = m \\
& \sum_{i \in V', i \neq j} x_{ji} \geq 1 \quad (j \in V_0) \\
& \sum_{i \in V, i \neq j} g_{ij} - \sum_{i \in V_0, i \neq j} g_{ji} = 1 \quad (j \in V_0) \\
& x_{ij} \leq g_{ij} \leq (Q - 1)x_{ij} \quad (i, j \in V, i \neq j) \\
& x_{i0} \leq g_{i0} \leq Qx_{i0} \quad (i \in V_0) \\
& z_{jd} = 0 \quad (j \in V_0) \\
& \sum_{j \in V, j \neq i} z_{ji} - \sum_{j \in V', j \neq i} z_{ij} = \sum_{j \in V', j \neq i} c_{ji}x_{ji} \quad (i \in V_0) \\
& c_{ij}x_{ij} \leq z_{ij} \leq Mx_{ij} \quad (i, j \in V_0, i \neq j) \\
& z_{0j} \leq Mx_{0j} \quad (j \in V_0) \\
& z_{jd} \leq Mx_{jd} \quad (j \in V_0)
\end{array}$$

$$\begin{aligned}
x_{ij} &\in \{0, 1\} && (i, j \in V \cup \{d\}, i \neq j) \\
z_{ij}, g_{ij} &\geq 0 && (i, j \in V \cup \{d\}, i \neq j).
\end{aligned}$$

For brevity, we do not give a detailed explanation of this formulation. The key point is that it has only $O(n^2)$ variables and constraints, and does not suffer from symmetry.

4.3 The k -median problem

In the k -median problem [35], we have n potential locations for facilities, m clients, and a positive integer $k < n$. For each location i and client j , the cost of serving client j from a facility at location i is c_{ij} . The task is to open k facilities, and assign each client to one open facility, in order to minimise the total cost. As in [1], we view the k facilities as workers, and the clients as tasks. That is, $|W| = k$ and $|T| = m$. Note that, in this case, the workers are not identical.

One can check that, if we use the generic approach in Subsection 3.2, the resulting MINLP will have $O(mnk)$ variables and constraints. Here is a more compact formulation. For $i = 1, \dots, n$, let y_i be a binary variable indicating whether facility i is opened. Also let λ_i be a continuous variable which takes the value zero if y_i is zero, but otherwise represents the total workload of the open facility. Then, for $i = 1, \dots, n$ and $j = 1, \dots, m$, let x_{ij} be a binary variable which indicates whether client j is assigned to facility i . Then:

$$\begin{aligned}
\text{Min.} \quad & \sum_{i=1}^n \lambda_i^p \\
\text{s.t.} \quad & \sum_{i=1}^n y_i = k \\
& \sum_{i=1}^n x_{ij} = 1 \quad (j = 1, \dots, m) \\
& x_{ij} \leq y_i \quad (i = 1, \dots, n; j = 1, \dots, m) \\
& \lambda_i \geq \sum_{j=1}^m c_{ij} x_{ij} \quad (i = 1, \dots, n) \\
& \lambda \in \mathbb{R}_+^n \\
& y \in \{0, 1\}^n \\
& x \in \{0, 1\}^{n \times m}.
\end{aligned}$$

This formulation is more or less self-explanatory. It has only $O(mn)$ variables and constraints, and it does not suffer from symmetry.

4.4 Strengthening the compact formulations

A disadvantage with the compact formulations presented above is the presence of “big M ” constraints, such as (10), which lead to fairly weak continuous relaxations. We use the approach in [21, 24] to strengthen the relaxations. Consider, for example, the m -TSP. For $i = 1, \dots, n$, we replace the

term z_{i0}^p in the objective function (4) with a new variable, say t_i , and then add the constraint

$$t_i \geq x_{i0}^{1-p} z_{i0}^p. \quad (13)$$

The (convex) function on the right of (13) is called a *perspective function*. By convention, it takes the value 0 when $x_{i0} = z_{i0} = 0$.

To see why this reformulation leads to a tighter relaxation, note that $x_{i0}^{1-p} > 1$ when $p > 1$ and x_{i0} is fractional. Thus, if any of the x_{i0} variables are fractional, the lower bound will increase.

Provided that p is rational, the constraint (13) can be modelled using SOC constraints (see again [3]). In practice, however, mixed-integer SOC solvers are rather slow. So, we also considered using another approach from [21, 24], which approximates perspective functions with linear inequalities called *perspective cuts*. In our context, this means replacing the nonlinear constraints (13) with a collection of linear inequalities of the form:

$$t_i \geq p\bar{z}^{p-1}z_{i0} + (\bar{z}^p - p\bar{z}^p)x_{i0} \quad (\bar{z} \in (0, M]).$$

One can then solve the approximated problem with a standard MILP solver. As we will see in the next section, just ten perspective cuts for each i is already sufficient to give a pretty good (under-)approximation of the perspective function.

The same approach can be used for the CMSTP. The application to the k -median problem is similar, except that the perspective functions take the form $y_i^{1-p}\lambda_i^p$.

5 Computational Results

In this section, we present computational results for the m -TSP, CMSTP and k -median problem, for varying values of p . All MILP and MINLP formulations were solved to optimality using the mixed-integer LP and SOCP solvers of CPLEX¹ 12.7, using default settings. The experiments were run on a MacBook Pro with a 2.3GHz Intel Core i5 processor and with 8GB memory. A time limit of 3600 seconds was imposed on each instance.

We remark that preliminary experiments, using both CPLEX and SCIP², indicated that the generic formulation presented in Subsection 3.2 and those in 4.1–4.3, as stated in their original form, are of no use at all computationally. In particular, for most instances, not even a feasible solution was found within the time limit. For that reason, we present results only for the formulations presented in Section 4.4.

¹<https://www.ibm.com/products/ilog-cplex-optimization-studio>

²<https://scip.zib.de>

5.1 The multiple traveling salesman problem

We began by taking TSP instances from TSPLIB³, and converting them to m -TSP instances with values $m = 2, 3, 4$. For each instance, we considered three values of p : $p = 1$ (minimise the sum of the route lengths, or “Min-Sum”); $p = 2$ (minimise the sum of squares, or “Min-SoS”), and $p \rightarrow \infty$ (minimise the maximum route length, or “Min-Max”). Note that Min-SoS can be viewed as a compromise between Min-Sum and Min-Max.

For each instance and each value of p , we imposed a time limit of one hour. For instances that CPLEX was unable to solve within the time limit, we computed the “relative optimality gap”, calculated as $(v^I - v^O)/v^I$, where v^I is the value of the best integer solution found and v^O is the lower bound at termination.

Table 1 presents the results. For each instance and each value of m , there are six rows. The line titled “Min-SoS” shows results for $p = 2$ after perspective reformulation has been applied (see Subsection 4.4). The three lines that follow are also for $p = 2$, but produced by MILP formulations that under-approximate the perspective functions with perspective cuts. There, $|B|$ is the number of cuts used for each worker.

The columns of the table show the following: the total length of all routes (column Total), the length of the longest and shortest routes (columns Max and Min, respectively), the ratio between Max and Min, the Gini index [22], and the index proposed in [28], that we denote by JCH (where the latter three are rounded to two decimal places). Note that the Gini index decreases to zero as the solution becomes more fair, whereas the JCH index approaches one. The column Time/Gap shows either the total time required to identify an optimal solution to the corresponding instance, or the relative optimality gap if the time limit was exceeded. The final column (titled CoF) measures the “cost of fairness”, expressed as the percentage increase in the Min-Sum solution values to achieve the solutions identified by the Min-SoS and Min-Max formulations.

The results indicate that Min-Sum solutions can be computed quickly, but tend to be extremely unfair. Our Min-SoS formulation requires significantly more solution time, but it almost always yields much fairer solutions with only a small increase in cost. Using perspective cuts instead usually leads to a substantial saving in computing time, yet with little or no loss of quality. We also observe that the effect of $|B|$ on solution quality and computing time is not predictable. Finally, the Min-Max solutions are even more fair, as one would expect, but this gain comes at the expense of significantly increased solution times and, usually, significantly higher cost.

All things considered, we believe that $p = 2$ represents a good compromise between cost, fairness and computing time. Of course, one could also experiment with other values of p .

³<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>

Instance	m		Total	Max	Min	Ratio	Gini	JCH	Time/Gap	CoF
ftv33	2	Min-Sum	1302	1195	107	11.17	0.84	0.59	0.87	
		Min-SoS	1389	696	693	1	0.00	1	211.90	6.68
		$ B = 10$	1389	696	693	1	0.00	1	148.92	6.68
		$ B = 15$	1389	696	693	1	0.00	1	118.81	6.68
		$ B = 20$	1389	696	693	1	0.00	1	116.47	6.68
		Min-Max	1389	696	693	1.00	0.00	1	777.07	6.68
ftv33	3	Min-Sum	1328	1195	26	45.96	0.88	0.41	0.99	
		Min-SoS	1424	529	402	1.32	0.09	0.99	0.14*	7.23
		$ B = 10$	1424	555	399	1.39	0.11	0.98	631.22	7.23
		$ B = 15$	1424	555	399	1.39	0.11	0.98	302.27	7.23
		$ B = 20$	1424	555	399	1.39	0.11	0.98	186.59	7.23
		Min-Max	1458	498	470	1.06	0.02	1	0.04*	9.79
ftv33	4	Min-Sum	1367	1142	26	43.92	0.82	0.35	0.72	
		Min-SoS	1539	460	253	1.82	0.14	0.96	0.18*	12.58
		$ B = 10$	1539	460	253	1.82	0.14	0.96	0.07*	12.58
		$ B = 15$	1539	460	253	1.82	0.14	0.96	0.07*	12.58
		$ B = 20$	1539	460	253	1.82	0.14	0.96	0.09*	12.58
		Min-Max	1592	411	383	1.07	0.02	1	0.11*	16.46
ftv35	2	Min-Sum	1489	1463	26	56.27	0.97	0.52	1.82	
		Min-SoS	1491	829	662	1.25	0.11	0.97	598.39	0.13
		$ B = 10$	1491	829	662	1.25	0.11	0.99	5.2	0.13
		$ B = 15$	1491	829	662	1.25	0.11	0.99	5.36	0.13
		$ B = 20$	1491	829	662	1.25	0.11	0.99	1.89	0.13
		Min-Max	1542	780	762	1.02	0.01	1	0.01*	3.56
ftv35	3	Min-Sum	1511	1393	26	53.58	0.90	0.39	1.6	
		Min-SoS	1586	611	423	1.44	0.12	0.98	0.09*	4.96
		$ B = 10$	1608	552	514	1.07	0.02	1	808.77	6.42
		$ B = 15$	1586	611	423	1.44	0.12	0.98	214.36	4.96
		$ B = 20$	1586	611	423	1.44	0.12	0.98	362.88	4.96
		Min-Max	1633	550	541	1.02	0.01	1	0.03*	8.07
ftv35	4	Min-Sum	1551	1357	26	52.19	0.86	0.32	1.05	
		Min-SoS	1712	472	377	1.25	0.06	0.99	0.17*	10.38
		$ B = 10$	1712	472	377	1.25	0.06	0.99	0.03*	10.38
	4	$ B = 15$	1712	472	377	1.25	0.06	0.99	0.08*	10.38
		$ B = 20$	1712	472	377	1.25	0.06	0.99	0.07*	10.38
		Min-Max	1746	459	415	1.11	0.03	1	0.24*	12.57
ftv38	2	Min-Sum	1546	1439	107	13.45	0.86	0.57	2.1	
		Min-SoS	1548	829	719	1.15	0.07	0.99	989.04	0.13
		$ B = 10$	1548	829	719	1.15	0.07	0.99	6.62	0.13
		$ B = 15$	1548	829	719	1.15	0.07	0.99	11.98	0.13
		$ B = 20$	1548	829	719	1.15	0.07	0.99	6.14	0.13
		Min-Max	1569	795	774	1.03	0.01	1	433.42	1.49
ftv38	3	Min-Sum	1569	1467	26	56.42	0.92	0.38	1.25	
		Min-SoS	1629	611	487	1.25	0.08	0.99	230.82	3.82
		$ B = 10$	1629	611	487	1.25	0.08	0.99	90.42	3.82
		$ B = 15$	1617	671	472	1.42	0.12	0.97	80.76	3.06
		$ B = 20$	1629	611	487	1.25	0.08	0.99	43.99	3.82
		Min-Max	1669	568	544	1.04	0.01	1	0.05*	6.37
ftv38	4	Min-Sum	1608	1414	26	54.38	0.87	0.32	1.8	
		Min-SoS	1744	514	333	1.54	0.11	0.98	0.16*	8.46
		$ B = 10$	1729	523	333	1.57	0.12	0.97	3285.57	7.52
		$ B = 15$	1746	474	377	1.26	0.06	0.99	0.02*	8.58
		$ B = 20$	1729	523	333	1.57	0.12	0.97	1572.6	7.52
		Min-Max	1797	460	437	1.05	0.02	1	0.11*	11.75

*Optimality gap upon termination of the optimisation after 3600 seconds.

Table 1: Computational results for TSPLIB instances with $m = 2, 3, 4$

5.2 The capacitated minimum spanning tree problem

Next, we looked at the CMSTP. We took ten of the ‘tc/te’ benchmark instances⁴ with 41 nodes (including the root), and set the sub-tree capacities to either $Q = 5$ or $Q = 10$. We also considered the uncapacitated case (equivalently, $Q \geq n$.) For each instance, we set the number of workers m to the degree of the root node in the optimal Min-Sum solution. (This is to ensure that the number of sub-trees in the Min-Sum, Min-SoS and Min-Max solutions are the same.) The results for the three values of Q are shown in Tables 2 to 4.

Unfortunately, we found that the Min-Max version of the CMSTP was extremely hard to solve, regardless of the MILP model used. Indeed, CPLEX was unable to obtain meaningful bounds for any instance within an hour. For this reason, we do not present statistics for the Min-Max version in the tables. Another issue is that, for some of the uncapacitated instances, the Min-Sum solution contained only one sub-tree. Since the issue of fairness does not arise when $|W| = 1$, we omit those instances from Table 4.

On the whole, the effect is not as dramatic as in the case of the m -TSP. For some instances (e.g., tc40-1 with $Q = 10$), the Min-Sum and Min-SoS solutions are identical. For some other instances (e.g., te40-2 with $Q = 5$), the solutions are distinct, but differ little in terms of fairness. There are however some instances (e.g., tc40-1 and tc40-2, either uncapacitated or with $Q = 5$) for which the Min-SoS solution is dramatically more fair than the Min-Sum one, regardless of whether one considers the Ratio, the Gini index or the JCH index.

Of course, the increase in fairness comes at the expense of increased computing times. On the other hand, in almost all cases, the CoF is very small. In fact, for some instances, the CoF is zero, meaning that there exist several alternative optimal Min-Sum solutions, some of which are much fairer than others.

5.3 The k -median problem

Next, we considered ten instances of the k -median problem⁵ with $n = 100$ or $n = 200$, and where k ranges from 5 to 67.

The results are shown in Table 5. Note that the ratios are missing for instances pmed05, pmed09 and pmed10. For these instances, the minimum workloads in the Min-Sum and Min-SoS solutions were zero. (This is caused by the existence of a self-assigned client node, i.e., a client is the only node assigned to the facility opened at the same node.)

It is clear that, for all instances, the Min-Sum solutions are very unfair, with ratios ranging from 2 to 75.44. For all apart from three instances, using

⁴Available at <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/capmstinfo.html>

⁵Available at <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html>

Instance		Total	Max	Min	m	Ratio	Gini	JCH	Time/Gap	CoF
tc40-1	Min-Sum	586	86	10	9	8.6	0.17	0.91	7.53	
	Min-SoS	596	70	60	9	1.17	0.03	1	381.85	1.71
	$ B = 10$	596	70	60	9	1.17	0.03	1	18.61	1.71
	$ B = 15$	596	70	60	9	1.17	0.03	1	15.08	1.71
	$ B = 20$	596	70	60	9	1.17	0.03	1	21.39	1.71
tc40-2	Min-Sum	578	85	10	9	8.5	0.18	0.9	3.32	
	Min-SoS	579	78	42	9	1.86	0.12	0.97	75.23	0.17
	$ B = 10$	579	78	42	9	1.86	0.12	0.97	7.92	0.17
	$ B = 15$	579	78	42	9	1.86	0.12	0.97	31.19	0.17
	$ B = 20$	579	78	42	9	1.86	0.12	0.97	5.87	0.17
tc40-3	Min-Sum	577	89	20	9	4.45	0.16	0.92	7.65	
	Min-SoS	577	78	45	9	1.73	0.09	0.98	26.88	0.00
	$ B = 10$	577	78	45	9	1.73	0.09	0.98	5.04	0.00
	$ B = 15$	577	78	45	9	1.73	0.09	0.98	28.52	0.00
	$ B = 20$	577	78	45	9	1.73	0.09	0.98	6.39	0.00
tc40-4	Min-Sum	617	91	30	9	3.03	0.17	0.93	8.78	
	Min-SoS	621	90	56	9	1.61	0.09	0.98	127.12	0.65
	$ B = 10$	621	90	56	9	1.61	0.09	0.98	12.62	0.65
	$ B = 15$	621	90	56	9	1.61	0.09	0.98	16.52	0.65
	$ B = 20$	621	90	56	9	1.61	0.09	0.98	19.31	0.65
tc40-5	Min-Sum	600	87	60	8	1.45	0.07	0.99	10.34	
	Min-SoS	600	87	60	8	1.45	0.07	0.99	639.02	0.00
	$ B = 10$	600	87	60	8	1.45	0.07	0.99	48.81	0.00
	$ B = 15$	600	87	60	8	1.45	0.07	0.99	115.18	0.00
	$ B = 20$	600	87	60	8	1.45	0.07	0.99	136.69	0.00
te40-1	Min-Sum	830	136	66	8	2.06	0.14	0.95	36.88	
	Min-SoS	835	136	79	8	1.72	0.12	0.97	0.03*	0.60
	$ B = 10$	835	136	79	8	1.72	0.12	0.97	1734.08	0.60
	$ B = 15$	835	136	79	8	1.72	0.12	0.97	0.02*	0.60
	$ B = 20$	835	136	79	8	1.72	0.12	0.97	0.01*	0.60
te40-2	Min-Sum	792	141	66	8	2.14	0.14	0.95	14.31	
	Min-SoS	792	141	66	8	2.14	0.14	0.95	0.03*	0.00
	$ B = 10$	793	138	70	8	1.97	0.13	0.96	1399.32	0.13
	$ B = 15$	793	138	70	8	1.97	0.13	0.96	1310.64	0.13
	$ B = 20$	793	138	70	8	1.97	0.13	0.96	1800.41	0.13
te40-3	Min-Sum	797	139	50	8	2.78	0.16	0.94	18.03	
	Min-SoS	801	131	62	8	2.11	0.14	0.95	0.05*	0.50
	$ B = 10$	799	131	62	8	2.11	0.14	0.95	0.03*	0.25
	$ B = 15$	799	131	62	8	2.11	0.14	0.95	0.03*	0.25
	$ B = 20$	799	131	62	8	2.11	0.14	0.95	0.02*	0.25
te40-4	Min-Sum	814	134	58	8	2.31	0.15	0.95	9.77	
	Min-SoS	815	134	58	8	2.31	0.15	0.95	0.04*	0.12
	$ B = 10$	815	134	58	8	2.31	0.15	0.95	0.03*	0.12
	$ B = 15$	815	134	58	8	2.31	0.15	0.95	0.03*	0.12
	$ B = 20$	815	134	58	8	2.31	0.15	0.95	0.03*	0.12
te40-5	Min-Sum	784	128	74	8	1.73	0.12	0.97	29.42	
	Min-SoS	784	128	74	8	1.73	0.12	0.97	0.03*	0.00
	$ B = 10$	784	128	74	8	1.73	0.12	0.97	1447.96	0.00
	$ B = 15$	784	128	74	8	1.73	0.12	0.97	1451.14	0.00
	$ B = 20$	784	128	74	8	1.73	0.12	0.97	1686.49	0.00

*Optimality gap upon termination of the optimisation after 3600 seconds.

Table 2: Computational results for CMSTP instances with $Q = 5$

Instance		Total	Max	Min	m	Ratio	Gini	JCH	Time/Gap	CoF
tc40-1	Min-Sum	498	116	74	5	1.57	0.11	0.98	0.96	
	Min-SoS	498	116	74	5	1.57	0.11	0.98	16.91	0.00
	$ B = 10$	498	116	74	5	1.57	0.11	0.98	1.89	0.00
	$ B = 15$	498	116	74	5	1.57	0.11	0.98	3.85	0.00
	$ B = 20$	498	116	74	5	1.57	0.11	0.98	3.59	0.00
tc40-2	Min-Sum	490	134	116	4	1.16	0.04	1	1.09	
	Min-SoS	490	134	116	4	1.16	0.04	1	16.41	0.00
	$ B = 10$	490	134	116	4	1.16	0.04	1	21.69	0.00
	$ B = 15$	490	134	116	4	1.16	0.04	1	11.03	0.00
	$ B = 20$	490	134	116	4	1.16	0.04	1	2.19	0.00
tc40-3	Min-Sum	500	132	112	4	1.18	0.04	1	0.60	
	Min-SoS	500	132	112	4	1.18	0.04	1	16.59	0.00
	$ B = 10$	500	132	112	4	1.18	0.04	1	41.19	0.00
	$ B = 15$	500	132	112	4	1.18	0.04	1	1.60	0.00
	$ B = 20$	500	132	112	4	1.18	0.04	1	10.38	0.00
tc40-4	Min-Sum	512	116	90	5	1.29	0.07	0.99	1.23	
	Min-SoS	512	116	90	5	1.29	0.07	0.99	10.76	0.00
	$ B = 10$	512	116	90	5	1.29	0.07	0.99	2.60	0.00
	$ B = 15$	512	116	90	5	1.29	0.07	0.99	3.14	0.00
	$ B = 20$	512	116	90	5	1.29	0.07	0.99	2.91	0.00
tc40-5	Min-Sum	504	122	80	5	1.52	0.11	0.97	1.38	
	Min-SoS	504	122	84	5	1.45	0.09	0.98	14.96	0.00
	$ B = 10$	504	122	84	5	1.45	0.09	0.98	1.79	0.00
	$ B = 15$	504	122	84	5	1.45	0.09	0.98	2.01	0.00
	$ B = 20$	504	122	84	5	1.45	0.09	0.98	1.99	0.00
te40-1	Min-Sum	596	172	120	4	1.43	0.1	0.98	84.33	
	Min-SoS	602	164	124	4	1.32	0.07	0.99	0.05*	1.01
	$ B = 10$	601	162	142	4	1.14	0.04	1	869.94	0.84
	$ B = 15$	604	166	138	4	1.2	0.05	1	0.06*	1.34
	$ B = 20$	596	172	120	4	1.43	0.1	0.98	0.05*	0.00
te40-2	Min-Sum	573	154	135	4	1.14	0.04	1	2.17	
	Min-SoS	573	154	135	4	1.14	0.04	1	33.54	0.00
	$ B = 10$	573	154	135	4	1.14	0.04	1	89.77	0.00
	$ B = 15$	573	154	135	4	1.14	0.04	1	72.08	0.00
	$ B = 20$	573	154	135	4	1.14	0.04	1	120.07	0.00
te40-3	Min-Sum	568	160	104	4	1.54	0.1	0.98	149.97	
	Min-SoS	571	147	140	4	1.05	0.01	1	0.02*	0.53
	$ B = 10$	568	160	108	4	1.48	0.1	0.98	0.05*	0.00
	$ B = 15$	570	151	139	4	1.09	0.02	1	0.04*	0.35
	$ B = 20$	568	160	108	4	1.48	0.1	0.98	0.06*	0.00
te40-4	Min-Sum	596	178	112	4	1.59	0.12	0.97	21.57	
	Min-SoS	598	160	140	4	1.14	0.03	1	332.13	0.34
	$ B = 10$	598	160	140	4	1.14	0.03	1	351.31	0.34
	$ B = 15$	598	160	140	4	1.14	0.03	1	191.43	0.34
	$ B = 20$	598	160	140	4	1.14	0.03	1	190.91	0.34
te40-5	Min-Sum	572	158	128	4	1.23	0.06	0.99	16.89	
	Min-SoS	572	152	130	4	1.17	0.04	1	538.32	0.00
	$ B = 10$	572	152	130	4	1.17	0.04	1	167.87	0.00
	$ B = 15$	572	152	130	4	1.17	0.04	1	1309.82	0.00
	$ B = 20$	572	152	130	4	1.17	0.04	1	52.69	0.00

*Optimality gap upon termination of the optimisation after 3600 seconds.

Table 3: Computational results for CMSTP instances with $Q = 10$

Instance		Total	Max	Min	m	Ratio	Gini	JCH	Time/Gap	CoF
tc40-1	Min-Sum	476	436	10	3	43.6	0.89	0.4	0.46	
	Min-SoS	490	166	160	3	1.04	0.01	1	19.52	2.94
	$ B = 10$	490	166	160	3	1.04	0.01	1	1.62	2.94
	$ B = 15$	490	166	160	3	1.04	0.01	1	1.31	2.94
	$ B = 20$	490	166	160	3	1.04	0.01	1	1.58	2.94
tc40-2	Min-Sum	460	346	114	2	3.04	0.5	0.8	0.41	
	Min-SoS	460	264	196	2	1.35	0.15	0.98	5.09	0.00
	$ B = 10$	460	264	196	2	1.35	0.15	0.98	1.30	0.00
	$ B = 15$	460	264	196	2	1.35	0.15	0.98	1.21	0.00
	$ B = 20$	460	264	196	2	1.35	0.15	0.98	1.10	0.00
tc40-4	Min-Sum	480	282	198	2	1.42	0.17	0.97	0.56	
	Min-SoS	480	282	198	2	1.42	0.17	0.97	0.03*	0.00
	$ B = 10$	480	282	198	2	1.42	0.17	0.97	1.27	0.00
	$ B = 15$	480	282	198	2	1.42	0.18	0.97	1.70	0.00
	$ B = 20$	480	282	198	2	1.42	0.18	0.97	1.43	0.00
tc40-5	Min-Sum	478	196	104	3	1.88	0.19	0.94	0.48	
	Min-SoS	490	174	144	3	1.21	0.06	0.99	2.95	2.51
	$ B = 10$	490	174	144	3	1.21	0.06	0.99	1.17	2.51
	$ B = 15$	490	174	144	3	1.21	0.06	0.99	1.18	2.51
	$ B = 20$	490	174	144	3	1.21	0.06	0.99	1.26	2.51
te40-1	Min-Sum	496	380	116	2	3.28	0.53	0.78	0.53	
	Min-SoS	504	270	234	2	1.15	0.07	0.99	10.70	1.61
	$ B = 10$	504	270	234	2	1.15	0.07	0.99	0.77	1.61
	$ B = 15$	504	270	234	2	1.15	0.07	0.99	0.79	1.61
	$ B = 20$	504	270	234	2	1.15	0.07	0.99	0.93	1.61

*Optimality gap upon termination of the optimisation after 3600 seconds.

Table 4: Computational results for uncapacitated MSTP instances

Instance		Total	Max	Min	Ratio	Gini	JCH	Time/Gap	CoF
pmed01	Min-Sum	5819	2147	241	8.91	0.41	0.75	0.27	
	Min-SoS	5935	1321	1073	1.23	0.05	0.99	119.78	1.99
	$ B = 10$	5932	1321	1073	1.23	0.06	0.99	3.20	1.94
	$ B = 15$	5945	1268	1133	1.12	0.03	1	2.06	2.17
	$ B = 20$	5935	1321	1073	1.23	0.05	0.99	2.81	1.99
pmed02	Min-Sum	4093	1358	18	75.44	0.52	0.54	0.48	
	Min-SoS	4258	532	336	1.58	0.09	0.98	37.26	4.03
	$ B = 10$	4258	532	336	1.58	0.09	0.98	2.16	4.03
	$ B = 15$	4265	532	336	1.58	0.09	0.98	2.87	4.2
	$ B = 20$	4258	532	336	1.58	0.09	0.98	1.60	4.03
pmed03	Min-Sum	4250	736	100	7.36	0.24	0.87	0.45	
	Min-SoS	4345	496	361	1.37	0.08	0.99	100.85	2.24
	$ B = 10$	4345	496	361	1.37	0.08	0.99	4.14	2.24
	$ B = 15$	4345	496	361	1.37	0.08	0.99	2.93	2.24
	$ B = 20$	4345	496	361	1.37	0.08	0.99	3.60	2.24
pmed04	Min-Sum	3034	591	10	59.1	0.48	0.54	0.20	
	Min-SoS	3173	244	83	2.94	0.16	0.93	42.38	4.58
	$ B = 10$	3186	244	83	2.94	0.15	0.94	4.86	5.01
	$ B = 15$	3156	244	83	2.94	0.17	0.92	4.24	4.02
	$ B = 20$	3173	244	83	2.94	0.16	0.93	4.70	4.58
pmed05	Min-Sum	1355	187	0	-	0.69	0.36	0.20	
	Min-SoS	1521	96	0	-	0.36	0.72	34.66	12.25
	$ B = 10$	1532	94	0	-	0.35	0.73	11.40	13.06
	$ B = 15$	1521	96	0	-	0.36	0.72	12.46	12.25
	$ B = 20$	1521	96	0	-	0.36	0.72	13.81	12.25
pmed06	Min-Sum	7824	2263	1129	2	0.16	0.94	4.99	
	Min-SoS	7845	2195	1280	1.71	0.12	0.96	0.05*	0.27
	$ B = 10$	7844	1636	1418	1.15	0.03	1	26.96	0.26
	$ B = 15$	7849	1669	1542	1.08	0.02	1	24.99	0.32
	$ B = 20$	7848	1609	1488	1.08	0.02	1	22.75	0.31
pmed07	Min-Sum	5631	1158	252	4.6	0.27	0.83	1.82	
	Min-SoS	5728	702	488	1.44	0.07	0.99	2212.55	1.72
	$ B = 10$	5734	702	476	1.47	0.06	0.99	17.56	1.83
	$ B = 15$	5728	702	488	1.44	0.07	0.99	16.86	1.72
	$ B = 20$	5728	702	488	1.44	0.07	0.99	14.55	1.72
pmed08	Min-Sum	4445	660	31	21.29	0.41	0.65	1.47	
	Min-SoS	4638	331	130	2.55	0.12	0.96	560.73	4.34
	$ B = 10$	4647	331	130	2.55	0.11	0.96	26.67	4.54
	$ B = 15$	4644	303	130	2.33	0.11	0.96	30.21	4.48
	$ B = 20$	4638	331	130	2.55	0.12	0.96	28.92	4.34
pmed09	Min-Sum	2734	227	0	-	0.48	0.59	1.32	
	Min-SoS	2869	109	0	-	0.18	0.9	3172.35	4.94
	$ B = 10$	2869	109	0	-	0.18	0.9	159.02	4.94
	$ B = 15$	2869	109	0	-	0.18	0.9	147.12	4.94
	$ B = 20$	2869	109	0	-	0.18	0.9	143.02	4.94
pmed10	Min-Sum	1255	106	0	-	0.66	0.37	1.22	
	Min-SoS	1383	45	0	-	0.39	0.69	645.34	10.2
	$ B = 10$	1383	45	0	-	0.39	0.69	88.19	10.2
	$ B = 15$	1383	45	0	-	0.39	0.69	60.70	10.2
	$ B = 20$	1383	45	0	-	0.39	0.69	72.45	10.2

*Optimality gap upon termination of the optimisation after 3600 seconds.

Table 5: Computational results for the ORLIB k -median instances

	p						
	1	1.25	1.5	1.75	2	2.25	2.5
pmed01	8.91	1.51	1.31	1.31	1.23	1.23	1.23
	0.27	3.06	1.60	2.01	3.48	3.37	3.12
pmed02	75.44	3.38	3.34	1.58	1.58	1.58	1.58
	0.48	1.54	3.10	1.66	2.00	3.33	3.08
pmed03	7.36	2.32	2.15	2.07	1.37	1.37	1.37
	0.45	1.96	3.42	3.01	4.09	4.99	4.84
pmed04	59.10	4.04	3.05	3.05	2.94	2.94	2.94
	0.20	1.34	1.98	2.44	4.35	14.93	15.74
pmed05	[0,187]	[0,139]	[0,113]	[0,96]	[0,94]	[0,94]	[0,80]
	0.20	1.27	2.24	6.23	10.78	16.50	17.46
pmed06	2.00	1.31	1.23	1.15	1.15	1.15	1.15
	4.99	58.24	49.60	22.78	26.65	20.03	27.42
pmed07	4.60	2.72	1.78	1.63	1.47	1.47	1.47
	1.82	18.94	25.74	13.73	18.52	24.47	52.90
pmed08	21.29	13.65	3.32	3.18	2.55	2.55	2.33
	1.47	12.35	21.16	25.72	27.14	91.33	277.14
pmed09	[0,227]	[0,131]	[0,125]	[0,109]	[0,109]	[28,107]	[28,106]
	1.32	10.26	71.95	75.77	149.89	153.72	169.81
pmed10	[0,106]	[0,60]	[0,56]	[0,53]	[0,45]	[0,45]	[0,44]
	1.22	8.96	15.58	38.41	89.31	99.54	133.80

Table 6: Sensitivity to the value of p on the k -median instances

the Min-SoS objective leads to a dramatic improvement in fairness, often at little extra cost. Moreover, when perspective cuts are used to approximate the Min-SoS objective, the computing times are very reasonable.

5.4 Sensitivity to p

Finally, we investigate the sensitivity of the fairness ratio to the value of p , using instances of the k -median problem solved using ten perspective cuts per worker.

The results are shown in Table 6 for values of p varying from 1 to 2.5 in increments of 0.25, where the first line shows the fairness ratio for each instance and the second line shows the computing times in seconds. For instances where the minimum workload is zero, we instead provide the range [Min, Max] of workloads in the corresponding solutions.

The results reveal a dramatic increase in fairness as p goes from 1 to 1.5, and only a moderate increase after that. For most instances, the computing time does not increase much as p increases. Exceptions are pmed04 and pmed08–pmed10, for which there is a substantial slowdown for larger values of p . Overall, the Min-SoS objective ($p = 2$) works well in most cases.

6 Discussion

Fairness has received a great deal of attention in many disciplines. To achieve fairness in combinatorial optimisation, we recommend using the ℓ^p -norm approach. We have shown that, for certain network optimisation problems, it is possible to derive compact and convex MINLP formulations of the ℓ^p variants, that do not suffer from symmetry. We have also shown that outer-approximating the MINLP, using a relatively small number of perspective cuts, often enables one to obtain results of acceptable quality using a standard MILP solver.

There are several possibilities for future research. One is to derive compact formulations for the ℓ^p version of other important combinatorial optimisation problems. Another is to develop guidelines on the suitable value for p for a given problem or a given instance. A third is to design effective heuristics, perhaps based on local search, for finding solutions to large-scale instances that give an acceptable trade-off between cost and fairness. Finally, one should consider how to achieve fairness when the objective is to maximise profit rather than to minimise cost.

Acknowledgements

We thank an anonymous reviewer for very useful comments, that have led to a significant improvement in the paper.

References

- [1] S. Ahmadian, B. Behsaz, Z. Friggstad, A. Jorati, M.R. Salavatipour & C. Swamy (2018) Approximation algorithms for minimum-load k -facility location. *ACM Trans. Alg.*, 14(2), art. 16.
- [2] M. Ajtai, J. Aspnes, M. Naor, Y. Rabani, L.J. Schulman & O. Waarts (1998) Fairness in scheduling. *J. Algorithms*, 29, 306–357.
- [3] F. Alizadeh & D. Goldfarb (2003) Second-order cone programming. *Math. Program.*, 95, 3–51.
- [4] D. Applegate, W. Cook, S. Dash & A. Rohe (2002) Solution of a min-max vehicle routing problem. *INFORMS J. Comput.*, 14, 132–143.
- [5] M. Barbati & C. Piccolo (2016) Equality measures properties for location problems. *Optim. Lett.*, 10, 90–920.
- [6] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh & P.H. Vance (1998) Branch-and-price: column generation for solving huge integer programs. *Oper. Res.*, 46, 316–329.

- [7] T. Bektaş (2006) The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34, 209–219.
- [8] T. Bektaş & J. Lysgaard (2015) Optimal vehicle routing with lower and upper bounds on route durations. *Networks*, 65, 166–179.
- [9] D. Bertsekas & R. Gallager (1992) *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall.
- [10] D. Bertsimas & S. Gupta (2015) Fairness and collaboration in network air traffic flow management: an optimization approach. *Transp. Sci.*, 50, 57–76.
- [11] D. Bertsimas, V.F. Farias & N. Trichakis (2011) The price of fairness. *Oper. Res.*, 59, 17–31.
- [12] A. Caprara & A.N. Letchford (2010) New techniques for cost sharing in combinatorial optimization games. *Math. Program.*, 124, 93–118.
- [13] A.K. Chandra & C.K. Wong (1975) Worst-case analysis of a placement algorithm related to storage allocation. *SIAM J. Comput.*, 4, 249–264.
- [14] A. Chassein & M. Goerigk (2015) Alternative formulations for the ordered weighted averaging objective. *Infor. Proc. Lett.*, 115, 604–608.
- [15] T.C.E. Cheng & C.C.S. Sin (1990) A state-of-the-art review of parallel-machine scheduling research. *Eur. J. Oper. Res.*, 47, 271–292.
- [16] C. D’Ambrosio & A. Lodi (2013) Mixed integer nonlinear programming tools: an updated practical overview. *Ann. Oper. Res.*, 204, 301–320.
- [17] G.B. Dantzig & P. Wolfe (1960) Decomposition principle for linear programs. *Oper. Res.*, 8, 101–111.
- [18] L.E. Dubins & E.H. Spanier (1961) How to cut a cake fairly. *Amer. Math. Monthly*, 68, 1–17.
- [19] E. Fernández, M.A. Pozo & J. Puerto (2014). Ordered weighted average combinatorial optimization: formulations and their properties. *Discr. Appl. Math.*, 169, 97–118.
- [20] C. Filippi, W. Ogryczak & M.G. Speranza (2017) Beyond average and minimax in MILP. *Technical report*, Department of Economics and Management, University of Brescia.
- [21] A. Frangioni & C. Gentile (2006) Perspective cuts for a class of convex 0-1 mixed-integer programs. *Math. Program.*, 106, 225–236.
- [22] C. Gini (1912) Variabilità e mutabilità. *Research report*, Department of Law, University of Cagliari.

- [23] L. Gouveia (1993) A comparison of directed formulations for the capacitated minimal spanning tree problem. *Telecom. Systems*, 1, 51–76.
- [24] O. Günlük & J. Linderoth (2011) Perspective reformulation and applications. In: J. Lee & S. Leyffer (eds.) *Mixed Integer Nonlinear Programming*, pp. 61–89. Berlin: Springer.
- [25] S.K. Gupta & A.P. Punnen (1990) k -sum optimization problems. *Oper. Res. Lett.*, 9, 121–126.
- [26] J. Halpern (1978) Finding minimal center-median convex combinations (cent-dian) of a graph. *Mgmt. Sci.*, 24, 535–544.
- [27] P. Hansen, M. Labbé & J.F. Thisse (1991) From the median to the generalized center. *RAIRO Oper. Res.*, 25, 73–86.
- [28] R.K. Jain, D.-M.W. Chiu & W.R. Hawe (1984) A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *Tech. Report DEC-TR-301*, Digital Equipment Corporation.
- [29] O. Janssen (2000) Job demands, perceptions of effort-reward fairness and innovative work behaviour. *J. Occup. Organ. Psychol.*, 73, 287–302.
- [30] N. Jozefowicz, F. Semet, & E.-G. Talbi (2009) An evolutionary algorithm for the vehicle routing problem with route balancing. *Eur. Oper. Res.*, 195, 761–769.
- [31] Ö. Karsu & M. Azizoglu (2019) An exact algorithm for the minimum squared load assignment problem. *Comput. Oper. Res.*, 106, 76–90.
- [32] Ö. Karsu & A. Morton (2015) Inequity averse optimization in operational research. *Eur. J. Oper. Res.*, 245, 343–359.
- [33] F. Kelly (1997) Charging and rate control for elastic traffic. *Trans. Emerg. Telecom. Technol.*, 8, 33–37.
- [34] M.M. Kostreva, W. Ogryczak & A. Wierzbicki (2004) Equitable aggregations and multiple criteria analysis. *Eur. J. Oper. Res.*, 158, 362–377.
- [35] M. Labbé & F. Louveaux (1997) Location problems. In M. DellAmico, F. Maffoli & S. Martello (eds.) *Annotated Bibliographies in Combinatorial Optimization*, pp. 261–281. Chichester: Wiley.
- [36] G. Laporte, M. Desrochers & Y. Nobert (1984) Two exact algorithms for the distance-constrained vehicle routing problem. *Networks*, 14, 161–172.

- [37] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan & D.B. Shmoys (1993) Sequencing and scheduling: algorithms and complexity. In S.C. Graves *et al.* (eds.) *Logistics of Production and Inventory*, pp. 445–524. Amsterdam: North Holland.
- [38] H. Luss (1999) On equitable resource allocation problems: a lexicographic minimax approach. *Oper. Res.*, 47, 361–378.
- [39] S. Martin, D. Ouelhadj, P. Smet, E. Özcan & G. Vanden Berghe (2013) Cooperative search for fair nurse rosters. *Exp. Sys. Appl.*, 40, 6674–6683.
- [40] P. Matl, R.F. Hartl & T. Vidal (2018) Workload equity in vehicle routing problems: a survey and analysis. *Transp. Sci.*, 52, 239–260.
- [41] A. Muklason, A.J. Parkes, E. Özcan, B. McCollum & P. McMullan (2017) Fairness in examination timetabling: student preferences and extended formulations. *Appl. Soft Comput.*, 55, 302–318.
- [42] J.F. Nash Jr (1950) The bargaining problem. *Econometrica*, 18, 155–162.
- [43] V.H. Nguyen & P. Weng (2017) An efficient primal-dual algorithm for fair combinatorial optimization problems. In X. Gao, H. Du & M. Han (eds.) *Proc. COCOA 2017*, pp. 324–339. Cham, Switzerland: Springer.
- [44] S. Nickel & J. Puerto (2005) *Location Theory: A Unified Approach*. Berlin: Springer.
- [45] W. Ogryczak (1997) On cent-dians of general networks. *Location Sci.*, 5, 15–28.
- [46] W. Ogryczak (2009) Inequality measures and equitable locations. *Ann. Oper. Res.*, 167, 61–86.
- [47] J. Puerto, A.M. Rodriguez-Chia & A. Tamir (2017) Revisiting k -sum optimization. *Math. Program.*, 165, 579–604.
- [48] M. Rabin (1993) Incorporating fairness into game theory and economics. *Amer. Econ. Rev.*, 83, 1281–1302.
- [49] J. Rawls (1971) *A Theory of Justice*. Cambridge, MA: Harvard University Press.
- [50] P.J. Slater (1978) Centers to centroids in graphs. *J. Graph Th.*, 2, 209–222.
- [51] J.A. Weymark (1981) Generalized Gini inequality indices. *Math. Soc. Sci.*, 1, 409–430.

- [52] L. Xia, K.B. Monroe & J.L. Cox (2004) The price is unfair! A conceptual framework of price fairness perception. *J. Market.*, 68, 1–15.
- [53] M. Zukerman, M. Mammadov, L. Tan, I. Ouveysi & L.L. Andrew (2008) To be fair or efficient or a bit of both. *Comput. Oper. Res.*, 35, 3787–3806.