J Glob Optim (2009) 44:79–97
DOI 10.1007/s10898-008-9308-8

# The optimal statistical median of a convex set of arrays

**Stefano Benati · Romeo Rizzi**

Received: 26 July 2007 / Accepted: 12 April 2008 / Published online: 6 May 2008


**Abstract** We consider the following problem. A set $r^1, r^2, \ldots, r^K \in \mathbf{R}^T$ of vectors is given. We want to find the convex combination $z = \sum \lambda_j r^j$ such that the statistical median of $z$ is maximum. In the application that we have in mind, $r^j$, $j = 1, \ldots, K$ are the historical return arrays of asset $j$ and $\lambda_j$, $j = 1, \ldots, K$ are the portfolio weights. Maximizing the median on a convex set of arrays is a continuous non-differentiable, non-concave optimization problem and it can be shown that the problem belongs to the APX-hard difficulty class. As a consequence, we are sure that no polynomial time algorithm can ever solve the model, unless P = NP. We propose an implicit enumeration algorithm, in which bounds on the objective function are calculated using continuous geometric properties of the median. Computational results are reported.

**Keywords** Global optimization · Median optimization · Statistical median and quantile optimization · Robust statistics · Branch&bound algorithms

## 1 Introduction

We consider the following problem. A set $r^1, r^2, \ldots, r^K \in \mathbf{R}^T$ of vectors is given. We want to find the convex combination $z = \sum_{j=1}^{K} \lambda_j r^j$, such that the median of $z$ is maximum. In the application that we have in mind, $r^j$, $j = 1, \ldots, K$ are the historical return arrays of asset $j$ and the decision variable $\lambda_j$, $j = 1, \ldots, K$ are the weight of asset $j$ in the portfolio with maximum median.

S. Benati (✉)
Dipartimento di Sociologia e Ricerca sociale, Università di Trento, Via Verdi 6, 38100 Trento, Italy
e-mail: Stefano.Benati@unitn.it

R. Rizzi
Dipartimento di Matematica ed Informatica, Università di Udine, Via delle Scienze 208, 33100 Udine, Italy
e-mail: Romeo.Rizzi@dimi.uniud.it

The problem is motivated from the Markovitz optimal portfolio model. In the standard approach, the decision maker selects the assets in such a way that the portfolio expectation is maximized, under the constraint that risk—e.g. variance—must be kept under a fixed threshold. There have been many papers discussing whether risk measures other than variance should be more appropriate, see for example [2,3,7,9,12,14]. Here we take a different view. We note that, under some assumptions, the sample mean that appears in the mean/variance problem can be replaced by other location estimators, for example by the sample median. If data are slightly not normal, or they are biased by outliers, then it is known from robust statistic theory that the median is a distribution location estimator more efficient than the mean, see [13,8]. Therefore introducing the median in portfolio optimization is an interesting matter, but, given the difficulty of finding the optimal median, in this paper we will limit our interest to the simple unconstrained case, that is risk constraints are not included to the model.

The optimal median model is also connected to more general quantile optimization [4,7]. The median is a distribution $\alpha$-quantile, where $\alpha = 0.5$. Letting $\alpha$ range from 0 to 1, all distribution quantiles can be obtained. In financial application, when $\alpha$ is low, e.g. 0.05 or less, it is called Value-at-Risk and it has been widely used by financial institutions for risk measuring. In [4] it has been shown how to develop and solve optimal Mean/Value-at-Risk portfolio model. It has been shown that commercial softwares as CPlex cannot solve problems with values of $\alpha$ larger than 0.1; the value of 0.5 corresponding to the median is the most difficult optimization problem that we experimented in our previous research. The computational techniques that are developed in this paper can be easily extended to quantile optimization models and to VaR optimization.

The paper is structured as follows. Section 2 contains the problem statement. It is shown that it is a case of non-concave optimization, where the objective function has many non-differentiable local optima, and that the problem can be formulated as a mixed integer non-linear programming model. In this model, there are two groups of variables: the first group contains the original continuous variables, e.g. portfolio weights, while binary variables correspond to hyperplanes, that are to be selected to determine a local problem optimum. Therefore the model has a hybrid nature, being intrinsically both continuous and combinatorial. The problem solution method will combine these properties.

In Sect. 3, we give a formal proof of the problem difficulty and it is shown that the problem is APX-hard. It means that the problem is difficult even in the approximate version, that is when one is satisfied with a solution within a fixed approximation to the optimal solution. More formally, there exists a $\varepsilon > 0$ such that no poly-time algorithm can produce a solution of quality at least the optimum times $(1 - \varepsilon)$ unless P = NP. As a consequence, only branch &bound or other complete enumeration algorithms can find the best solution.

In Sect. 4, we will state some lemmas to determine the upper bounds to the objective function. The first lemma uses a geometric property of the continuous median function to limit the feasible range of continuous variables. This property will be combined with valid inequalities, that can be deduced from the combinatorial structure of the problem. Then we will show how this approach can determine some upper bounds to the optimal median.

In Sect. 5, we describe the exact method, e.g. the branch&bound algorithm, that uses all bounds discussed previously. The computational results on some test problems are reported: the number of array $K$ does not pose a serious computational burden, but, if the dimension $T$, that corresponds to the number of binary variables, is greater than some 40, then computation times can be $>1$ h.

## 2 Quantile optimization and the optimal median problem

Let $r$ be a real $T$-dimensional vector, so that $r \in \mathbf{R}^T$. Let $r_i$ be entry $i$ of vector $r$. Rank $r$ in increasing order and obtain:

$$r_{1:T} \leq r_{2:T} \leq \cdots \leq r_{T:T}$$

so that $r_{i:T}$ is the value on position $i$ over $T$. For example:

$$r_{1:T} = \min\{r_i : i = 1, \ldots, T\};$$

$$r_{T:T} = \max\{r_i : i = 1, \ldots, T\}.$$

Let $\alpha = \frac{1}{T}, \frac{2}{T}, \ldots, \frac{T-1}{T}$, let $q(\alpha) = \alpha T$:

**Definition 1** The $\alpha$-quantile of vector $r$ is the element in the $q(\alpha)$-position: $r_{q(\alpha):T}$.

The terminology comes from order statistics theory, since $r_{q(\alpha):T}$ is a quantile estimator, called the $q(\alpha)$-*order statistics*.

The most important quantile estimator is the median.

**Definition 2** Let $T$ be odd, then the median of $r$, denoted by med$\{r\}$, is $r_{\frac{T+1}{2}:T}$.

A set of vectors $\mathcal{U} = \{r^j \in \mathbf{R}^T : j = 1, \ldots, K\}$ is given. Let $r_{ij}$ be entry $i$ of vector $j$. We want to find the convex combination $z$ of the vectors of $\mathcal{U}$, maximizing the $q(\alpha)$-order statistic of $z$. The Order Statistic Optimization Problem can be formulated as follows (problem **P1**).

$$\max_{\lambda, z} z_{q(\alpha):T} \tag{1}$$

s.t.

$$z_i = \sum_{j=1}^{K} \lambda_j r_{ij} \quad \text{for } i = 1, \ldots, T \tag{2}$$

$$\sum_{j=1}^{K} \lambda_j = 1 \tag{3}$$

$$\lambda_j \geq 0 \quad \text{for } j = 1, \ldots, K \tag{4}$$

where $z_i$ is entry $i$ of vector $z$.

If $i = \frac{T+1}{2}$, then we obtain the Optimal Median Problem:

$$\max_{\lambda, z} \text{med}\{z\} \tag{5}$$

s.t.

$$z_i = \sum_{j=1}^{K} \lambda_j r_{ij} \quad \text{for } i = 1, \ldots, T \tag{6}$$

$$\sum_{j=1}^{K} \lambda_j = 1 \tag{7}$$

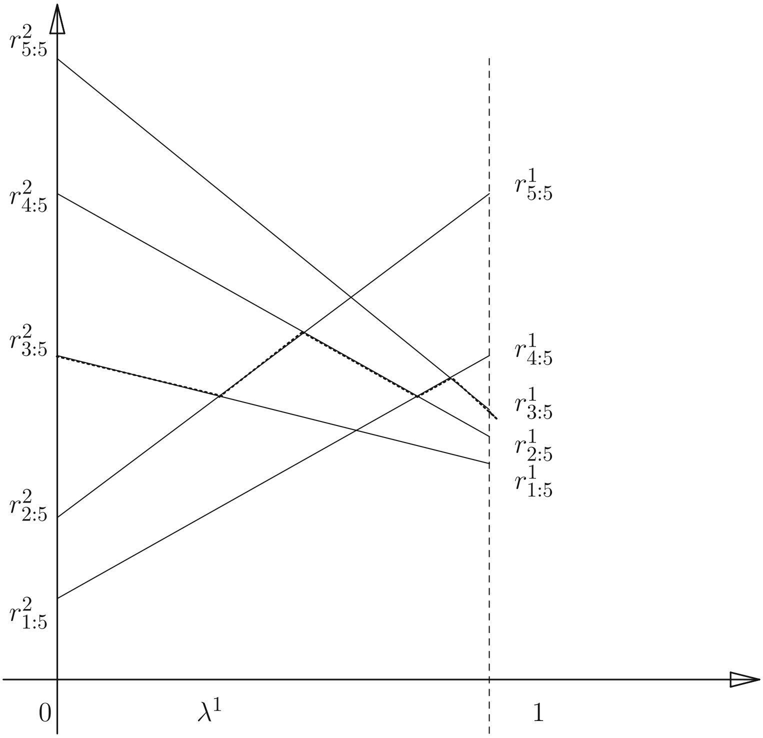


**Fig. 1** The draw of the median of the convex combination of two arrays $r^1$ and $r^2$ ($T = 5$). Note that for $\lambda^1 = 0$, then $r^2_{3:5} = med\{r^2\}$

$$\lambda_j \geq 0 \quad \text{for } j = 1, \ldots, K \tag{8}$$

*Remark 1* The Order Statistics Optimization problem is a case of continuous, non-differentiable and non-convex optimization.

Consider the Optimal Median Problem where $K = 2$ and $T = 5$. Then the median depends on only one variable $\lambda_1$. The draw of $\text{med}\{\lambda_1 r^1 + (1 - \lambda_1) r^2\}$ is reported in Fig. 1.

Problem **P1** can be reformulated as a non-linear mixed integer programming model. Let $r^{Min} = \min\{r_{ij} : i = 1, \ldots, T;\ j = 1, \ldots, K\}$. The problem, later on referred to as problem **P2**, is:

$$\max_{\lambda, y, z^{Med}} z^{Med} \tag{9}$$

s.t.

$$\sum_{j=1}^{K} \lambda_j r_{ij} \geq r^{Min} + (z^{Med} - r^{Min}) y_i \quad \text{for every } i = 1, \ldots, T \tag{10}$$

$$\sum_{i=1}^{T} y_i \geq \frac{T+1}{2} \tag{11}$$

$$\sum_{j=1}^{K} \lambda_j = 1 \tag{12}$$

$$y_i \in \{0, 1\} \quad \text{for every } i = 1, \ldots, T \tag{13}$$

$$\lambda_j \geq 0 \quad \text{for every } j = 1, \ldots, K. \tag{14}$$

Problem variables are:

- $\lambda_j$, $j = 1, \ldots, K$ are the original continuous convex weights;
- $z^{Med}$ is the continuous variable representing the median;
- $y_i$, $i = 1, \ldots, T$ are binary variables representing the inequalities that define $z^{Med}$.

Let $z_i = \sum_{j=1}^{K} \lambda_j r_{ij}$. To define $z^{Med}$, observe that at least $\frac{T+1}{2}$ values of $z_i$ must be greater or equal to $z^{Med}$. To accomplish this, we introduced binary variables $y_i$, $i = 1, \ldots, T$ that count how many times the feasible region is defined through an inequality of the type $z_i \geq z^{Med}$. If $y_i = 0$, then the constraint (10) is always satisfied, but if $y_i = 1$, then $\sum_{j=1}^{K} \lambda_j r_{ij} \geq z^{Med}$. The number of those constraints must be at least $\frac{T+1}{2}$ and, since we are maximizing $z^{Med}$, they do not exceed this number. Therefore, if $(\lambda^*, z^{Med^*}, y^*)$ is the optimal solution to **P2**, then $z^{Med^*}$ is the median of $z_i = 1, \ldots, T$: $z^{Med^*} = \text{med}\{\sum_{j=1}^{K} \lambda_j^* r_{ij}\}$. This shows that **P1** and **P2** are equivalent formulations of the Optimal Median Problem.

Considering quantile optimization problems, problem **P2** can be easily modified to deal with general $q(\alpha)$. For example, when $\alpha$ is low, e.g. 0.05, and dealing with financial data, then we are dealing with Value-at-Risk optimization, [4,7]. However, computational experiments contained in [4] show that the median is the most difficult quantile optimization problem to solve. Therefore it will be the only quantile problem that we will consider.

## 3 Inapproximability results

In this section, we show that Problem **P2** is APX-hard. This negative result holds even if we restrict our attention to instances in which $r^{Min} = 0$, and every entry $r_{i,j}$ is either 0 or 1.

We refer to the Minimum Node Cover Problem restricted to graphs with maximum degree $\Delta \leq 3$ as to the MNC3 Problem. The MNC3 Problem is known [11,1,6] to be APX-hard. This means that there exists an $\varepsilon > 0$ such that no $(1 + \varepsilon)$-approximation algorithm exists for MNC3 unless P = NP. We show the same holds for Problem **P2** by means of a Turing $L$-reduction from MNC3 to Problem **P2**. (Clearly, since Problem **P2** is a *maximization* problem, we have to show that there exists an $\varepsilon > 0$ such that no $(1 - \varepsilon)$-approximation algorithm exists for Problem **P2** unless P = NP.)

Assume given an instance of MNC3, that is, a graph $G = (V, E)$ with $\Delta(G) \leq 3$. We are asked to find a minimum cardinality *node cover* of $G$, that is, an $X \subseteq V$ with $|X|$ as small as possible and such that every edge of $G$ has at least one endnode in $X$. We show that, assuming the existence of a $(1 - \varepsilon)$-approximation algorithm for Problem **P2** for any $0 < \varepsilon \leq \frac{1}{2}$, then, for every natural $k$ such that $G$ admits a node cover of size $k$ we can produce in poly-time a node cover of $G$ of size at most $(1 + 6\varepsilon)k$. Running this for all possible values of $k = 1, 2, \ldots, |V|$, we obtain that the existence of a $(1 - \varepsilon)$-approximation algorithm for

Problem **P2** implies the existence of a $(1 + 6\varepsilon)$-approximation algorithm for MNC3, hence the APX-hardness of Problem **P2** follows from the APX-hardness of MNC3.

*The reduction* Given the graph $G = (V, E)$ comprising the input instance of MNC3, and a natural $k \leq |V|$, we build up an associated instance of Problem **P2** as follows. Assume $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_m\}$. We can always assume $k \geq n/2$ by the following well known theorem of Nemhauser and Trotter.

**Theorem 1** (Nemhauser and Trotter [10]) *Given a graph $G = (V, E)$, introduce a new node $v'$ for every node $v \in V$. Let $V' = \{v' : v \in V\}$ and $F = \{uv' : uv \in E\}$. Consider the bipartite graph $H = (V, V'; F)$ on $2|V|$ node and $2|E|$ edges. Let $X$ be a minimum node cover of $H$. Let $Y = \{v : v \in X \text{ .and. } v' \in X\}$ and $Z = \{v : v \in X \text{ .xor. } v' \in X\}$. Then the following three properties hold:*

(i) *if a set $D \subseteq Z$ covers $G[Z]$ then $D \cup Y$ covers $G$;*

(ii) *there exists a minimum cover of $G$ which contains $Y$;*

(iii) *every node cover of $G[Z]$ contains at least $|Z|/2$ nodes.*

Therefore, since a minimum node cover in a bipartite graph can be found in poly-time, then we can assume $k \geq \frac{n}{2}$.

We take $K := n$ and $T := 2m + 4k - 1$. Intuitively, the $K$ columns $R^1, R^2, \dots, R^K$ of matrix $(r_{i,j})$ correspond to the $n$ nodes $v_1, v_2, \dots, v_n$ of $G$. Moreover, in our intentions, the first $m$ rows $R_1, R_2, \dots, R_m$ of matrix $(r_{i,j})$ correspond to the $m$ edges $e_1, e_2, \dots, e_m$ of $G$. Indeed, for $i = 1, 2, \dots, m$, (and for every $j = 1, 2, \dots, n$,) we set $r_{i,j} = 1$ if $v_j \in e_i$ and $r_{i,j} = 0$ otherwise. In practice, the first $m$ rows of matrix $(r_{i,j})$ give the incidence matrix of $G$. The next $2n$ rows of matrix $(r_{i,j})$ come into pairs, with each such a pair associated with a different node of $G$. More precisely, for $i = 1, 2, \dots, n$, (and for every $j = 1, 2, \dots, n$,) we set $r_{m+2i,j} = r_{m+2i-1,j} = 1$ if $i = j$ and $r_{m+2i,j} = r_{m+2i-1,j} = 0$ otherwise. The remaining $T - m - 2n := m - 1 + 4k - 2n$ rows are taken to be all 0. Then, by definition, $r^{Min} = 0$. Right now it suffices to observe that $m - 1 + 4k - 2n \geq 0$ follows from $k \geq n/2$, hence it is possible to perform all the actions prescribed by our reduction.

The reduction is complete: we have by now fully specified an instance of Problem **P2** on the basis of the pair $\langle G, k\rangle$. Clearly, all the actions prescribed in the above reduction can be performed in polynomial time.

*Example 1* Consider the following graph $G = (V, E)$, in which $V = \{1, 2, 3, 4, 5\}$ and $E = \{(1, 2), (1, 3), (2, 3), (2, 5), (3, 4), (4, 5)\}$; $n = 5, m = 6$ and we want to know whether $G$ admits a node cover of cardinality $k = 3$.

The constraints of the form (10) are the following.

The first $m = 6$ are the rows describing the arcs of $G$: the corresponding "node-arc" submatrix is:

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix};$$

then there is the "double-identity" submatrix:

$$N_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix};$$

and the "zero" submatrix:

$$O = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The constraint matrix is:

$$R = \begin{bmatrix} M \\ N_2 \\ O \end{bmatrix}$$

If $y_k = 1$ for some row index $k$, then one of the following types of constraint is active: $\lambda_i + \lambda_j \geq z^{Med}$ for some $(i, j) \in E$; $\lambda_i \geq z^{Med}$ for some $i$.

The following two lemmas introduce our theorem on the complexity of the median problem.

**Lemma 2** *The instance of Problem* **P2** *associated to a graph $G = (V, E)$ with $\Delta(G) \leq 3$ and an integer $k \geq n/2$ by means of the reduction described above admits a feasible solution of value $\frac{1}{k}$ whenever $G$ admits a node cover of size $k$.*

*Proof* Let $X \subseteq V$ be a node cover of $G$ with $|X| = k$. Consider the following choice for the decision variables of the associated instance of Problem **P2**: take $\lambda_j = \frac{1}{k}$ if $v_j \in X$ and $\lambda_j = 0$ if $v_j \notin X$ and take $z^{Med} = \frac{1}{k}$. Furthermore, for all $i \leq m$, take $y_i = 1$, whereas, for all $i > m + 2n$, take $y_i = 0$. Finally, for every $j = 1, 2, \ldots, n$, take $y_{m+2j} = y_{m+2j-1} = 1$ if $v_j \in X$, and $y_{m+2j} = y_{m+2j-1} = 0$ otherwise.

Notice that all constraints of type (10) are satisfied. Indeed, if $i > m+2n$, then constraints (10) are satisfied since $y_i = 0$. Moreover, for every $j = 1, 2, \ldots, n$, the constraints (10) with $i = m + 2j$ and $i = m + 2j - 1$ are satisfied since either $y_i = 0$ or $v_j \in X$ and hence $\lambda_j = \frac{1}{k}$. In the second case remember also that $r_{i,j} = 1$. Finally, all constraints of type (10) with $i \leq m$ are satisfied with $y_i = 1$, since $X$ is a node cover and the first $m$ rows of matrix $(r_{i,j})$ form the incidence matrix of $G$. Notice that the number of variables $y_i$ set to 1 is $m + 2k = \frac{T+1}{2}$. This implies that $\sum y_i = \frac{T+1}{2}$. and that constraint (11) is also satisfied. □

*Example 1* (*continue*) Observe that $X = \{2, 3, 5\}$ is a node cover of cardinality $k = 3$. The corresponding solution is: $\lambda_i = \frac{1}{3}, i = 2, 3, 5, \lambda_i = 0, i = 1, 4; y_j = 1, j = 1, \ldots, 6; j = 9, \ldots, 12; j = 15, 16, y_j = 0$ for all other $j$; $z^{Med} = \frac{1}{3}$.

**Lemma 3** *Assume $\varepsilon \leq \frac{1}{2}$. Assume given a feasible solution $(\lambda, y, z^{Med})$ with $z^{Med} \geq (1-\varepsilon)\frac{1}{k}$ for the instance of Problem* **P2** *associated to $G = (V, E)$ and $k$ as better specified above. Then we can produce a node cover of $G$ of size at most $(1 + 6\varepsilon)k$ in poly-time.*

*Proof* Let $\lambda_1, \lambda_2, \ldots, \lambda_K, z^{Med}$, and $y_1, y_2, \ldots, y_T$ be a feasible solution with $z^{Med} \geq (1 - \varepsilon)\frac{1}{k}$. We assume the value of no variable $\lambda_j$ and $z^{Med}$ can be decreased without making the solution infeasible. It implies $z^{Med} = (1-\varepsilon)\frac{1}{k}$ and that $\max\{\lambda_j | j = 1, \ldots, n\} = (1-\varepsilon)\frac{1}{k}$, since if $\lambda_j > (1 - \varepsilon)\frac{1}{k}$ for some $j$, then also $\lambda_j + \lambda_w > (1 - \varepsilon)\frac{1}{k}$ for all $j, w$ such that $(j, w) \in E$, so that $\lambda_j$ can be decreased while keeping values $y_k$ fixed. Furthermore, we can always assume that either $\lambda_j = (1 - \varepsilon)\frac{1}{k}$ or $\lambda_j = 0$ occurs for every $j = 1, 2, \ldots, K$. Indeed, let $V^* = \{v_j \in V \ : 0 < \lambda_j < (1 - \varepsilon)\frac{1}{k}\}$ and consider a node $v_a \in V^*$. By the above assumption on the minimality of the $\lambda_j$'s in the feasible solution, there must exist an edge $v_a v_b \in E$ such that $\lambda_a + \lambda_b = (1 - \varepsilon)\frac{1}{k}$. Notice that $v_b$ also belongs to $V^*$. We hence propose to modify the given feasible solution by setting $\lambda_a = (1 - \varepsilon)\frac{1}{k}, \lambda_b = 0$. To keep the solution feasible we may be forced to set $y_f = y_g = 0$ for the (at most) two edges $e_f$ and $e_g$ of $E$ incident with $v_b$ and other than $v_a v_b$, but now we can also set $y_{n+2a} = y_{n+2a-1} = 1$ hence maintaining the validity of constraint (11).

Let $V^+ = \{v_j \in V \ : \lambda_j = (1 - \varepsilon)\frac{1}{k}\}$. Notice that, by constraint (12),

$$|V^+| \leq \frac{k}{1 - \varepsilon} = k\frac{1 + 2\varepsilon}{1 + \varepsilon - 2\varepsilon^2} \leq k(1 + 2\varepsilon) \, ,$$

where the last inequality holds insofar $1 + \varepsilon - 2\varepsilon^2 \geq 1$, and hence holds since $\varepsilon \leq \frac{1}{2}$. By constraint (11), we know that the number of $y_i$'s set to 1 is at least $m + 2k$ and all have $i \leq m + 2n$ since for $i > m + 2n$ row $R_i$ is all 0. Since $|V^+| \leq k(1 + 2\varepsilon)$, then at least $m + 2k - 2k(1 + 2\varepsilon)$ of the $y_i$'s set to 1 have $i \leq m$. This means that $|V^+|$ covers at least $m - 4k\varepsilon$ edges of $G$. Therefore, by just picking one node from each one of the edges left uncovered by $V^+$ and adding it to $V^+$, we obtain a node cover of $G$ of size at most $|V^+| + 4k\varepsilon \leq k(1 + 6\varepsilon)$. □

**Theorem 4** *The Maximum Median Problem is APX-hard.*

*Proof* If we know the solution to the Maximum Median Problem, then we know the solution to problem **P2**. But if we know how to solve **P2**, then, from the previous lemmas, we know if a graph admits an Approximate Node Cover. It follows that the Maximum Median Problem is APX-hard. □

## 4 Problem upper bounds

From the problem formulation and the complexity result, the only exact method to solve the problem is implicit complete enumeration. In the following, we will implement a branch & bound scheme, in which the tree is pruned if the upper bound of a node is smaller than the best found feasible solution. The upper bounds are calculated using the continuous properties of the median function and with the help of valid inequalities that can be added to the problem.

### 4.1 Bounds on portfolio variables

The first step is to show how to restrict continuous variables feasible range. From (12) and (14), there is an implicit upper bound $\lambda_j \leq 1$, for every $j$. These bounds can be decreased using the following lemmas.

The following facts are straightforward:

**Lemma 5** *Let* $r \in \mathbf{R}^T$, $\lambda \in \mathbf{R}^+$; *let* $\lambda = [\lambda_i = \lambda; i = 1, \ldots, T]$ *then*

- $med\{\lambda r\} = \lambda med\{r\}$;
- $max\{\lambda r\} = \lambda max\{r\}$;
- $med\{r + \lambda\} = med\{r\} + \lambda$.

**Lemma 6** *Let* $r^1, r^2 \in \mathbf{R}^T$, *then* $med\{r^1 + r^2\} \leq med\{r^1\} + \max\{r^2\}$.

*Proof* Let $\lambda = \max\{r^2\}$ and then $\lambda = [\lambda, \ldots, \lambda]$. Then $\text{med}\{r^1 + r^2\} \leq \text{med}\{r^1 + \lambda\} = \text{med}\{r^1\} + \lambda = \text{med}\{r^1\} + max\{r^2\}$. □

Combining the above lemmas, we obtain the following inequality:

**Lemma 7** *Let* $r^1, r^2 \in \mathbf{R}^T$, *let* $\lambda \in [0, 1]$, *then* $med\{\lambda r^1 + (1-\lambda) r^2\} \leq \lambda med\{r^1\} + (1-\lambda) max\{r^2\}$.

*Proof* $\text{med}\{\lambda r^1 + (1-\lambda) r^2\} \leq \text{med}\{\lambda r^1\} + \max\{(1-\lambda) r^2\} \leq \lambda \text{med}\{r^1\} + (1-\lambda) \max\{r^2\}$. □

We can extend the last lemma to the case of $K$ vectors:

**Lemma 8** *Let* $r^1, \ldots, r^K \in \mathbf{R}^T$, $\lambda_1, \ldots, \lambda_K \in \mathbf{R}^+ \cup \{0\}$, *such that* $\sum_{j=1}^K \lambda_j = 1$, *then:*

$$med\left\{\sum_{j=1}^K \lambda_j r^j\right\} \leq \lambda_q med\{r^q\} + \sum_{j \neq q} \lambda_j max\{r^j\}. \tag{15}$$

Suppose that a heuristic feasible median $z^h$ is calculated. For example $z^h = \max_j \{\text{med}\{r^j\}\}$. The following theorem show how to decrease the bounds on $\lambda_j$ variables.

**Theorem 9** *Suppose that for some* $j$, *we have* $z^h \geq med\{r^j\}$. *Then:*

$$\lambda_j \leq \frac{\max\{\max\{r^i\} : i \neq j\} - z^h}{\max\{\max\{r^i\} : i \neq j\} - med\{r^j\}}.$$

*Proof* The rule comes from the valid inequality (15):

$$z^h \leq \text{med}\{r^j\}\lambda_j + \sum_{i \neq j} \max\{r^i\}\lambda_i.$$

Then,

$$z^h \leq \text{med}\{r^j\}\lambda_j + \max\{\max\{r^i\} : i \neq j\}(1 - \lambda_j)$$

from which the lemma follows. □

We can see that if $z^h = \text{med}\{r^j\}$, then the lemma provides $\lambda_j \leq 1$. That is, we obtain the same bound on variable $\lambda_j$ as the one in the problem formulation. But if $z^h > \text{med}\{r^j\}$, a condition usually met in practice, then the inequality implies $\lambda_j < 1$. In the following, we will refer to:

$$b_j = \frac{\max\{\max\{r^i\} : i \neq j\} - z^h}{\max\{\max\{r^i\} : i \neq j\} - \mathrm{med}\{r^j\}}$$

as the new upper bound on $\lambda_j$.

A simple corollary shows how some continuous variables can be fixed to 0:

**Corollary 10** *Suppose that for some $q$ we have $z^h > \max\{r^q\}$, then $\lambda^*_q = 0$ in any optimal solution.*

*Proof* Assume the contrary and let $\lambda^*_j$, $j = 1, \ldots, K$ be the optimal value, such that $\lambda^*_q > 0$. Consider the solution $\lambda'_j = \frac{\lambda^*_j}{1-\lambda^*_q}$ for $j \neq q$, $\lambda'_q = 0$.

Notice now that for any $t$ such that $\sum_{j=1}^K \lambda^*_j r_{tj} \geq r^{Med^*}$ we have:

$$\sum_{j=1}^K \lambda'_j r_{tj} = \sum_{j=1; j\neq q}^K \frac{\lambda^*_j}{1-\lambda^*_q} r_{tj} = \frac{1}{1-\lambda^*_q} \sum_{j=1; j\neq q}^K \lambda^*_j r_{tj}$$
$$\geq \frac{r^{Med^*} - \lambda^*_q \max\{r^q\}}{1-\lambda^*_q} > r^{Med^*} \frac{1-\lambda^*_q}{1-\lambda^*_q},$$

contradicting the optimality of $r^{Med^*}$. □

### 4.2 Bound from surrogate constraints

Now we combine the new bounds on continuous variables with valid inequalities that are deduced from the mixed integer formulation. Basically, the bound states that the median cannot be greater than the mean of the best half of the highest numbers. For every column $r^j$, with $j = 1, 2, \ldots, K$, let $\sigma_j : \{1, 2, \ldots, T\} \mapsto \{1, 2, \ldots, T\}$ be any permutation in $S_{|T|}$ such that $r_{\sigma_j(1),j} \geq r_{\sigma_j(2),j} \geq \ldots \geq r_{\sigma_j(T),j}$ and let $\beta_j := \sum_{i=1}^{\frac{T+1}{2}} r_{\sigma_j(i),j}$ denote the sum of the $\frac{T+1}{2}$ biggest entries in column $r^j$. We obtain the following valid inequality:

**Lemma 11** $z^{Med} \leq \frac{2}{T+1}\left(\sum_{j=1}^K \beta_j \lambda_j\right)$.

*Proof* Relabel the index of the inequalities so that the optimal $y^*_i$ selects inequalities $1, \ldots, \frac{T+1}{2}$ of the form:

$$\sum_{j=1}^K \lambda_j r_{ij} \geq z^{Med} \quad \text{for } i = 1, \ldots, \tfrac{T+1}{2}. \tag{16}$$

Summing all inequalities we obtain the valid inequality:

$$\sum_{i=1}^{\frac{T+1}{2}} \sum_{j=1}^K \lambda_j r_{ij} \geq \frac{T+1}{2} z^{Med}. \tag{17}$$

Notice that $\beta_j \geq \sum_{i=1}^{\frac{T+1}{2}} r_{ij}$ for every $j$. Therefore,

$$\frac{2}{T+1}\sum_{j=1}^{K}\beta_j\lambda_j \geq \frac{2}{T+1}\sum_{j=1}^{K}\lambda_j\left(\sum_{i=1}^{\frac{T+1}{2}} r_{ij}\right)$$
$$= \frac{2}{T+1}\sum_{i=1}^{\frac{T+1}{2}}\sum_{j=1}^{K}\lambda_j r_{ij} \geq z^{Med}.$$

□

From the previous lemma, we can calculate the following bound. Let $b_j \in \mathbf{R}$ be the upper bound of continuous variable $\lambda_j$, for example as calculated from Theorem 9 and Corollary 10. Let $z^{Med^*}$ be the optimal median value, then

**Theorem 12** $z^{Med^*} \leq \max\left\{\frac{2}{T+1}\left(\sum_{j=1}^{K}\beta_j\lambda_j\right) \mid \sum_{j=1}^{K}\lambda_j = 1; 0 \leq \lambda_j \leq b_j\ \forall j\right\}$.

Note that the bound can be calculated in a greedy fashion. First compute $\beta_j$ for every $j$, then calculate $\beta_{j_1} = \max\{\beta_j | j = 1, \ldots, K\}$, then $\lambda_{j_1} = b_{j_1}$. The next step is to calculate $\beta_{j_2} = \max\{\beta_j | j = 1, \ldots, K; j \neq j_1\}$ and $\lambda_{j_2} = \min\{b_{j_2}, 1 - b_{j_1}\}$, and so on till we reach $\sum_{j=1}^{K}\lambda_j = 1$.

### 4.3 Bound from dominating inequalities

Now we combine the bounds $b_j$ for all $j$ with a simplified problem version. We replace every convex combination with a constant term, so we are left with a problem with only integer variables that can be solved in a greedy fashion.

Let $q_i = \max\left\{\sum_{j=1}^{K} r_{ij}\lambda_j \mid \sum_{j=1}^{K}\lambda_j = 1; 0 \leq \lambda_j \leq b_j\ \forall j\right\}$. Note that $q_i$ can be calculated with the same procedure of theorem 12. Then:

**Lemma 13** *If an optimal solution is such that $z^{Med^*} \leq \sum_{j=1}^{K} r_{ij}\lambda_j$ for some $i$, then $z^{Med^*} \leq q_i$.*

Now we can use inequalities $z^{Med} \leq q_i$ to obtain an approximation of problem **P2**.

**Theorem 14** *Consider the following problem:*

$$z^d = \max z^{Med} \tag{18}$$

$$q_i \geq r^{Min} + (z^{Med} - r^{Min})y_i \text{ for every } i \tag{19}$$

$$\sum_{i=1}^{T} y_i = \frac{T+1}{2} \tag{20}$$

$$y_i \in \{0, 1\} \tag{21}$$

*then $z^{Med^*} \leq z^d$.*

Note that $z^d$ can be calculated easily: set $q_i, i = 1, \ldots, T$ in decreasing order and take $z^d = q_{\frac{T+1}{2},T}$.

## 5 Branch&bound algorithm

To solve the problem, we propose a branch&bound algorithm that enumerates implicitly all binary solutions $y_i, i = 1, \ldots, T$. At every tree node, the partial solution is composed of variable sets $T^1 = \{i|y_i = 1\}$ and $T^0 = \{i|y_i = 0\}$. Let $z^{Med}(T^0, T^1)$ be the node best median value, when $y_i$ are constrained according to set $T^0$ and $T^1$.

### 5.1 Adding valid inequalities

First, we note that the formulation of P2 can be strengthened adding valid inequalities that are redundant to define the optimal solution, but that are useful to evaluate a node of the branch&bound tree. Let $r^{Max} = \max\{r_{ij}|i = 1, \ldots, T;\ j = 1, \ldots, K\}$, those inequalities are of the form:

$$\sum_{j=1}^{K} \lambda_j r_{ij} \leq r^{Max} + (z^{Med} - r^{Max})(1 - y_i) \quad \text{for every } i = 1, \ldots, T\ . \tag{22}$$

The equations reduce to $\sum_{j=1}^{K} \lambda_j r_{ij} \leq z^{Med}$ if $y_i = 0$, and $\sum_{j=1}^{K} \lambda_j r_{ij} \leq r^{Max}$ if $y_i = 1$. Consider the following problem:

$$z^l(T^0, T^1) = \max_{\lambda, y, z^l} z^l \tag{23}$$

s.t.

$$\sum_{j=1}^{K} \lambda_j r_{ij} \geq z^l \quad \text{for every } i \in T^1 \tag{24}$$

$$\sum_{j=1}^{K} \lambda_j r_{ij} \leq z^l \quad \text{for every } i \in T^0 \tag{25}$$

$$\sum_{j=1}^{K} \lambda_j \leq 1 \tag{26}$$

$$\lambda_j \geq 0 \quad \text{for every } j = 1, \ldots, K. \tag{27}$$

Note that it is a node relaxation of problem **P2**. Variables $y_i$ such that $i \in \{T^1 \cup T^0\}$ fix inequalities (24) and (25). Conversely, free $y_i$ variables and their corresponding inequalities are cancelled. Clearly, $z^l(T^0, T^1)$ is an upper bound to $z^{Med}(T^0, T^1)$ in node $(T^0, T^1)$ and it can be calculated using linear programming. If $z^h \geq z^l(T^0, T^1)$, then the node can be pruned.

### 5.2 Node upper bounds

The result of theorem 12 can be extended to the analysis of branch&bound nodes. Let $s_j = \sum_{i \in T^1} r_{ij}$, for $j = 1, \ldots, K$. Let $c_j$, $j = 1, \ldots, K$, be the sum of the greatest $\frac{T+1}{2} - |T^1|$

elements $r_{ij}$ of column $j$, such that $i \notin (T^0 \cup T^1)$. Let $\beta_j(T^1, T^0) = s_j + c_j$ and be:

$$z^c(T^0, T^1) = \max \left\{ \frac{2}{T+1} \left( \sum_{j=1}^{K} \beta_j(T^0, T^1)\lambda_j \right) \mid 0 \le \lambda_j \le b_j \ \forall j \right\}. \tag{28}$$

Then $z^c(T^0, T^1)$ is an upper bound of $z^{Med}(T^0, T^1)$.

The problem stated in theorem 14 can be reformulated to compute another upper bound of node $(T^0, T^1)$. The problem is:

$$z^d(T^0, T^1) = \max z^{Med} \tag{29}$$

$$q_i \ge r^{Min} + (z^{Med} - r^{Min})y_i \quad \text{for every } i \tag{30}$$

$$\sum_{i=1}^{T} y_i = \frac{T+1}{2} \tag{31}$$

$$y_i = 1 \quad if \quad i \in T^1; \tag{32}$$

$$y_i = 0 \quad if \quad i \in T^0; \tag{33}$$

$$y_i \in \{0, 1\} \quad if \quad i \in T - (T^1 \cup T^0). \tag{34}$$

then $z^{Med}(T^0, T^1) \le z^d(T^0, T^1)$.

Therefore, in every node of the branch&bound tree, three upper bounds can be calculated. In our computational experience, the strongest bound is $z^l$, if many variables are constrained, and $z^c$, if the problem is less constrained and we are at a low level of the branch&bound tree. But in some cases we observed that also $z^d$ can be the best bound, therefore all three bounds are useful to prune the tree.

### 5.3 Inequality variables fixing

Fixing binary variables to 0 or 1 are strategies that can reduce the computational time. Therefore at every node the following rules are implemented.

If we know an upper bound $z^u(T^0, T^1)$ to the optimal value $z^{Med}(T^0, T^1)$, then the binary variables can be fixed to 0 or 1, if the following condition occurs.

*Rule 1* Suppose that for some $i \notin \{T^0, T^1\}$, $r_i^{Min} = \min\{r_{ij}\lambda_j | \sum_{j=1}^{K} \lambda_j = 1; 0 \le \lambda_j \le b_j\} \ge z^u$, then $y_i = 1$ in any optimal solution for $z^{Med}(T^0, T^1)$.

In the root node, that is when $T^0$ and $T^1$ are empty, the rule follows from $\sum_{j=1}^{K} r_{ij}\lambda_j \ge \min\{r_{ij} : j = 1, \ldots, K\} \ge z^u \ge z^{Med*}$.

Suppose we know an heuristic feasible solution $z^h$, then:

*Rule 2* Suppose that for some $i \notin \{T^0, T^1\}$, $r_i^{Max} = \max\{r_{ij}\lambda_j | \sum_{j=1}^{K} \lambda_j = 1; 0 \le \lambda_j \le b_j\} \le z^h$, then $y_i = 0$ in any optimal solution.

The rule follows from $\sum_{j=1}^{K} r_{ij}\lambda_j \le z^h \le z^{Med*}$.

### 5.4 Branch&bound scheme

The main subroutine of the branch&bound code that has been implemented is the following recursive subroutine. The main input of the subroutine are sets $T^1$ and $T^0$, that are the $y_i$ variables fixed to 0 or to 1, and $z^h$, that is the best median found. At the beginning $z^h$ is calculated using the heuristic algorithm described in [5], then it is updated if some terminal node is a better solution. At the end of the algorithm, $z^h$ is optimal.

```
Recursive Subroutine Evaluate-Node(T^1, T^0)

Terminal Node: If {T^1, T^0} = T Then
                  Call  Compute z^Med
                  If z^Med > z^h Then
                     Call Update Best Solution
                     End If
                     Return
End If

Examine the partial solution:
Call Compute z^U
      (Comment: z^U is the upper bound corresponding to {T^1, T^0})

Pruning the node:
If z^U ≤ z^h Then
   Return
End If

Variable fixing:
Call Fix Inequalities
      (Comment: According to z^U and z^h, fix as many y_i variables
      as possible)

Branch:
Call Select Variable To Branch
( Comment: Select i ∈ T − (T^1 ∪ T^0) and branch the tree)
Let T_l^1 ← T^1 ∪ {i}; T_l^0 ← T^0
     Call Evaluate-Node(T_l^1, T_l^0)
Let T_r^1 ← T^1; T_r^0 ← T^0 ∪ {i}
     Call Evaluate-Node(T_r^1, T_r^0);
Return

End Subroutine
```

In “Terminal Node”, a terminal node is reached, the corresponding value $z^{Med}$ is calculated using problem P2, that does not contain free integer variables anymore and is a standard linear problem. In “Examine the partial solution”, all bounds that we described are calculated and the least is retained, to prune the tree, in “Pruning the node”. In “Variable fixing”, the procedure helps to reduce the size of the tree. In “Branch”, branching is done by fixing to 0 or 1 the variable $y_q$ that most likely is equal to 1 in the optimal solution. More formally, let $b_i = \max\{r_{ij} : j = 1, \ldots, K\}$ for $i \in T - |T^1 - T^0|$; then $q$ is branched if $b_q = \max_i b_i$.

### 5.5 Computational results

The algorithm has been coded in CompaQ Visual Fortran, using Minos 5.5 Linear Solver. Then it has been tested on two problem classes:

- *Random problems*: all data are generated by (0,1) uniform random numbers.
- *Stock exchange problems*: all data are daily closing prices of assets of the Milan stock exchange (year 2005).

Tests were run on a Pentium III 850 Mhz PC.

Table 1 reports our findings. Let T be the number of the integer variables and we let T vary from 15 to 45. Let K be the number of the continuous variables and we let vary 5 to 50. Then we explore how computational times increase. Consider random problems: the most difficult are solved in less that 1 s for T = 15, in <20 s for T = 25, in <14 min for T = 35, but in almost 4 h for T = 45.

Consider stock exchange problems: the most difficult were solved in <1 s if T = 15, in <44 s for T = 25, in <81 min for T = 35, but the most difficult problems of 45 data were not solved after 8 h of computations, so that the computation has been halted.

Therefore we expect that, if T is <40, the optimal median problem should be solved in an acceptable computational time. But if T is larger than 40, the problem can be untractable. In a related paper, [5], a fast heuristic is implemented. In Table 1, column labelled “Opt H?” is reported whether the heuristic found the optimal solution and can be seen that at least for half problems the heuristic is effective. The heuristic computation times always much <1 s, therefore it is a viable alternative method for calculating the best median. More results are contained in [5].

Moreover, we see that random problems are easier than stock exchange problems. If we compare computational times of problems with the same dimension, we see that times are almost always bigger, in some cases 10 times bigger. We were a bit surprised, since we imagined that stock exchange data should have had a natural order that helped the algorithm, for example fixing $y_i$ variables to 0 and 1. But it was not the case. One possible explanation of this difficulty is the solution structure of the problem. In Table 2, for any test problem, we report the value of $\max\{\lambda_i | i = 1, \ldots, K\}$ of the optimal solution. As can be seen, the values are different from random to real data. Random problems show a maximum $\lambda$ that is very high, such as 0.8, suggesting that the optimal solution is close to the vertices of $\lambda$ convex bounds. For Stock Exchange problems the converse is true. Maximum $\lambda$ is around to 0.35, suggesting that the optimal solution is somewhere “in the middle” of the $\lambda$ convex bounds.

Another explanation of the difficulty of real data problems is the effectiveness of the heuristic procedure. If we exclude problems with T = 45, we discovered that the heuristic algorithm found the exact solution 12 times over 15 for random problems, but only 6 times over 15 for stock exchange problems. Therefore a point of future research is to improve that heuristic.

In Table 1, we see that also the dimension $K$ affects the computational times. By and large, when K = 50 the number of nodes are 4 times the nodes for K = 5. One possible reason is that, if the dimension is higher, then there are more local optima that are close to the best one, and the branch&bound tree must explore all of them. In any case, the heuristic solution is a good approximation and is always obtained in much <1 s.

**Table 1** Computational results

| Data type | T | K | Nodes | Times | Opt H? |
|---|---|---|---|---|---|
| Random | 15 | 5 | 85 | 0.05 | y |
| | | 10 | 609 | 0.33 | n |
| | | 20 | 101 | 0.11 | y |
| | | 30 | 915 | 0.66 | y |
| | | 50 | 101 | 0.16 | y |
| | 25 | 5 | 3988 | 2.2 | n |
| | | 10 | 3639 | 2.25 | y |
| | | 20 | 3417 | 2.69 | y |
| | | 30 | 8519 | 7.14 | y |
| | | 50 | 17981 | 19.33 | y |
| | 35 | 5 | 6205 | 3.73 | n |
| | | 10 | 43587 | 29.77 | y |
| | | 20 | 775285 | 840.58 | y |
| | | 30 | 81489 | 92.88 | y |
| | | 50 | 264875 | 403.71 | y |
| | 45 | 5 | 13005 | 8.52 | n |
| | | 10 | 2173151 | 2336.7 | n |
| | | 20 | 6211469 | 12727.23 | n |
| | | 30 | 317755 | 489.06 | y |
| | | 50 | 1933559 | 5450.81 | y |
| Stock exchange | 15 | 5 | 491 | 0.49 | y |
| | | 10 | 313 | 0.28 | n |
| | | 20 | 737 | 0.49 | y |
| | | 30 | 887 | 0.61 | n |
| | | 50 | 695 | 0.66 | y |
| | 25 | 5 | 4405 | 2.48 | y |
| | | 10 | 5701 | 3.51 | n |
| | | 20 | 20927 | 16.81 | y |
| | | 30 | 15423 | 15.54 | y |
| | | 50 | 36427 | 44.16 | n |
| | 35 | 5 | 20317 | 12.14 | n |
| | | 10 | 244059 | 177.41 | n |
| | | 20 | 713315 | 745.12 | n |
| | | 30 | 1378295 | 2045.76 | n |
| | | 50 | 2381505 | 4866.23 | n |
| | 45 | 5 | 176199 | 127.37 | y |
| | | 10 | 3750001 | 3960.18 | y |

*Note*: "Nodes" are the nodes of the branch&bound tree. Times are expressed in seconds, "Opt H" says whether the first heuristic solution is optimal

**Table 2** The maximum value of $\lambda$ for different test problems

| Data type | K | T = 15 | T = 25 | T = 35 |
|---|---|---|---|---|
| Random | 5 | 0.84 | 0.7 | 0.7 |
| | 10 | 0.69 | 0.81 | 0.76 |
| | 20 | 0.95 | 0.94 | 0.5 |
| | 30 | 0.79 | 0.78 | 0.8 |
| | 50 | 0.69 | 0.76 | 0.86 |
| Stock exchange | 5 | 0.7 | 0.57 | 0.43 |
| | 10 | 0.34 | 0.35 | 0.29 |
| | 20 | 0.38 | 0.38 | 0.27 |
| | 30 | 0.36 | 0.31 | 0.29 |
| | 50 | 0.24 | 0.43 | 0.25 |

As far as financial applications are concerned, we discovered that optimal median portfolio are well diversified. For those problems, $\lambda$ values are interpreted as portfolio weights. In Table 3, some data about optimal $\lambda$ are reported for problem type "Stock exchange". Those data are:

- $l_1$: the greatest portfolio weight.
- $l_2$: the second greatest portfolio weight.
- $l_{min}$: the smallest portfolio weight.
- $n.asset$: the number of assets that are included in the portfolio.

It can be seen that $l_1$ is almost always at a value of 0.35, that is, only one third of wealth is invested in the most important asset; $l_2$ is always an important investment, since it is always around 0.2 and in some cases, even $l_{min}$ is not negligible. Moreover, the number of asset that are selected by the optimal median model are almost always 5 or 6. Those findings are strikingly different to the solution of the Optimal Mean Problem, that is finding the convex combination of arrays such that the sample mean is maximum: the solution would be trivially $\lambda_j = 1.0$ if $r^j$ is the vector with maximum mean.

But when we started this research, we were moved by the hypothesis that the sample median could replace the sample mean in the Markovitz mean-variance portfolio optimization. After our tests, we see that the optimal median portfolio is completely different to the optimal expectation portfolio, and that, using the median, portfolio diversification can be obtained without using any risk constraint. Of course, those results must be extended to the case that a robust variance estimator is introduced to the model, so that the two models can be better compared. However, we have the impressions that it is a promising direction of research.

## 6 Future research

In this article, we introduced and discussed the optimal median problem, motivated by the fact that, under some assumptions, the sample median is a distribution location estimator that can be used as an alternative to the sample mean. From a mathematical point of view, we showed that the model is rich enough to find an interesting connection with binary optimization and we developed a branch&bound algorithm to find the optimal solution. But, from a financial point of view, the model is very simple, since it does not contain any risk measure. Therefore,

**Table 3** Portfolio weights and number of assetts of optimal solutions to Stock Exchange problems

| T | K | l1 | l2 | lmin | n.assett |
|---|---|---|---|---|---|
| 15 | 5 | 0.7 | 0.17 | 0.13 | 3 |
| | 10 | 0.34 | 0.24 | 0.11 | 5 |
| | 20 | 0.38 | 0.26 | 0.03 | 5 |
| | 30 | 0.36 | 0.24 | 0.01 | 6 |
| | 50 | 0.24 | 0.18 | 0.11 | 6 |
| 25 | 5 | 0.57 | 0.43 | 0.43 | 2 |
| | 10 | 0.35 | 0.3 | 0.12 | 4 |
| | 20 | 0.38 | 0.23 | 0.02 | 7 |
| | 30 | 0.31 | 0.24 | 0.05 | 6 |
| | 50 | 0.43 | 0.31 | 0.01 | 5 |
| 35 | 5 | 0.43 | 0.4 | 0.01 | 5 |
| | 10 | 0.29 | 0.23 | 0.01 | 6 |
| | 20 | 0.27 | 0.21 | 0.03 | 6 |
| | 30 | 0.29 | 0.17 | 0.03 | 7 |
| | 50 | 0.25 | 0.24 | 0.01 | 12 |
| 45 | 5 | 0.81 | 0.17 | 0.02 | 3 |
| | 10 | 0.72 | 0.15 | 0.02 | 5 |

*Note*: l1 is the maximum portfolio weight, l2 the second biggest, lmin the minimum weight; n.assett is the number of non-zero optimal variables

the model should be extended to include risk modelling constraints. It means that the next research step must be to develop bounds and algorithms for an optimal median model with side constraints. Of course, the results that are obtained in this simplified case will be useful.

## References

1. Alimonti, P., Kann, V.: Hardness of approximating problems on cubic graphs. Proceedings of 3rd Italian Conference on Algorithms and Complexity. Lecture Notes in Computer Science, vol. 1203 (Springer-Verlag, 1997), pp. 288–298
2. Artzner, P., Delbaen, F., Eber, J.M., Heath, D.: Coherent measures of risk. Math. Finance **9**, 203–228 (1999)
3. Benati, S.: The optimal portfolio problem with coherent risk measure constraints. Eur. J. Oper. Res. **150**, 572–584 (2003)
4. Benati, S., Rizzi, R.: A mixed integer linear programming formulation of the optimal mean/Value-at-Risk portfolio problem. Eur. J Oper. Res. **176**, 423–434 (2007)
5. Benati, S., Rizzi, R.: A fast heuristic for finding portfolio with maximum median. Working paper, University of Trento (2006)
6. Berman, P., Karpinski, M.: On some tighter inapproximability results. Proceedings of the 26th International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 1644 (Springer-Verlag, Berlin, 2000), pp. 200–209
7. Gaivoronski, A.A., Pflug, G.: Value at risk in portfolio optimization: properties and computational approach. J. Risk **7**, 1–31 (2004)
8. Huber, P.J.: Robust Statistics. Wiley, New York (1981)
9. Konno, H.H., Yamazaki, H.: Mean-absolute deviation portfolio optimization model and its application to Tokyo stock market. Manag. Sci. **37**, 519–531 (1991)

10. Nemhauser, G.L., Trotter, L.E. Jr.: Vertex packings: structural properties and algorithms. Math. Programm **8**, 232–248 (1975)
11. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. Syst. Sci. **43**, 425–440 (1991)
12. Rockafellar, R.T., Uryasev, S.: Optimization of conditional Value-at-Risk. J. Risk **2**, 21–41 (2000)
13. Tukey, J.W.: A survey of sampling from contaminated distributions. In: Olkin, I. (ed.) Contributions to Probability and Statistics, pp. 448–485. Stanford University Press, Stanford (1960)
14. Young, M.R.: A minimax portfolio selection rule with linear programming solution. Manag. Sci. **44**, 673–683 (1998)