RECent
Center for Economic Research

WORKING PAPER SERIES

Exact and Heuristic Approaches for the Index Tracking Problem with UCITS Constraints

Andrea Scozzari, Fabio Tardella, Sandra Paterlini and Thiemo Krink

Working Paper 81

May 2012

www.recent.unimore.it

RECent: c/o Dipartimento di Economia Politica, Viale Berengario 51, I-41100 Modena, ITALY
Phone +39 059 2056856, Fax +39 059 2056947, E-mail: RECent@unimore.it

Exact and Heuristic Approaches for the Index Tracking Problem with UCITS Constraints

Andrea Scozzari[1], Fabio Tardella[2], Sandra Paterlini[3], Thiemo Krink[4,*]

[1]Università degli Studi "Niccolò Cusano" - Telematica, Roma, IT, andrea.scozzari@unisu.it
[2]Sapienza University of Rome, Rome, IT, fabio.tardella@uniroma1.it
[3]Dept. of Economics, RECent & CEFIN, University of Modena and Reggio E., IT, sandra.paterlini@unimore.it
[4]Allianz Investment Management, Allianz Life Insurance Co., Minneapolis, USA, thiemo.krink@allianzlife.com

May 25, 2012

Abstract

Index tracking aims at determining an optimal portfolio that replicates the performance of an index or benchmark by investing in a smaller number of constituents or assets. The tracking portfolio should be cheap to maintain and update, i.e., invest in a smaller number of constituents than the index, have low turnover and low transaction costs, and should avoid large positions in few assets, as required by the European Union Directive UCITS (Undertaking for Collective Investments in Transferable Securities) rules. The UCITS rules make the problem hard to be satisfactorily modeled and solved to optimality: no exact methods but only heuristics have been proposed so far.
The aim of this paper is twofold. First, we present the first Mixed Integer Quadratic Programming (MIQP) formulation for the constrained index tracking problem with the UCITS rules compliance. This allows us to obtain exact solutions for small- and medium-size problems based on real-world datasets. Second, we compare these solutions with the ones provided by the state-of-art heuristic Differential Evolution and Combinatorial Search for Index Tracking (DECS-IT), obtaining information about the heuristic performance and its reliability for the solution of large-size problems that cannot be solved with the exact approach. Empirical results show that DECS-IT is indeed appropriate to tackle the index tracking problem in such cases. Furthermore, we propose a method that combines the good characteristics of the exact and of the heuristic approaches.

Keywords: Index tracking, mixed integer quadratic programming, stochastic search heuristics, differential evolution, cardinality constraints.

1 Introduction

In the last decade, passive management products have increasingly attracted much attention both from small and large investors. The recent crises and the disbelief in the alpha's persistence of asset managers have prompted even further the investors' desires of investing in cheap and transparent financial products, such as Exchange-Traded Funds (ETFs). The idea of implementing quantitative approaches for index tracking (IT) or benchmark replication, not new in

*Disclaimer: The views and opinions expressed in this article are those of the author and do not necessarily reflect the official policy or position of Allianz Investment Management or Allianz Life Insurance.

the financial literature, has then become even more appealing, given that it could allow to build simple, transparent, and low-cost passive investment products. In fact, IT aims at replicating a given index or benchmark by selecting a subset of assets, or constituents, together with their weights so as to best track the performance of the index. Considering only a smaller number of constituents allows to reduce the administrative, monitoring, and transaction costs and avoids holding small and illiquid positions.

The basic IT problem can be formulated as a constrained optimization problem: minimize a given distance between the index and the tracking portfolio using at most K assets out of the n available. From a complexity viewpoint, even this basic version of cardinality constrained IT with a quadratic tracking error measure was shown to be NP-Hard in [30] via a reduction from the Subset Sum problem. Furthermore, choosing only a small number of constituents could result in ineffective passive management and in holding too large amounts of few assets. Thus, in order to take into account practical limitations on the portfolio composition, other constraints should be considered. For instance, to guarantee a cheap rebalancing strategy, a turnover constraint is imposed in order to avoid large changes in asset weights values over time. It is also reasonable to require that the weights of the assets included in the tracking portfolio should be larger than a given lower bound, to reduce monitoring costs due to small and illiquid positions, and smaller than a given upper bound, to avoid holding too large positions. The latter requirement is also stated in official regulations, such as the European Union Directive UCITS (Undertaking for Collective Investments in Transferable Securities) rules, that require, among others, that the sum of all asset weights exceeding 5% must be smaller than 40%.

In this work, we fill a gap in the literature by proposing an exact approach to the IT problem with the hard real-world constraints previously described. More precisely, we propose the first Mixed Integer Quadratic Programming (MIQP) formulation for the index tracking problem subject to the cardinality constraint, the turnover constraint, the buy-in thresholds, and to the additional constraint imposed by the UCITS rules compliance. We note that UCITS rules are often binding for many portfolio managers in the EU, and should be therefore taken into account in real-world practice. However, to our knowledge, so far they have been modeled in the literature as general nonlinear constraints, and have been dealt with only by means of heuristic approaches (see [7, 8, 23]). This implies that the exact solution for the IT problem with UCITS constraints has not yet been obtained, even for markets with few assets, before this study. In fact, using our new MIQP formulation we can solve the index tracking problem for small and medium size markets with up to 225 assets by using a standard solver (CPLEX). In particular, we apply our method to financial data from the Dow Jones 65 (65 assets), the Dax 100 (98 assets), the S&P 100 (98 assets), and the Nikkei 225 (225 assets). For each dataset, we adopt a rolling window scheme to evaluate the in-sample and the out-of-sample tracking performance of the portfolios obtained. We then compare the results with those obtained with the Differential Evolution and Combinatorial Search for Index Tracking (DECS-IT) developed in [23]. We confirm that DECS-IT is indeed an efficient and accurate heuristic approach that can be fruitfully used to tackle IT problems, and is one of the few tools available for large-size problems. Furthermore, we observe that for large-size multi-period problems one could devise a hybrid approach that uses the heuristic method to provide a good solution for the first period in a reasonable time and the exact solver to efficiently find optimal solutions for all subsequent periods.

The index tracking problem has received large attention in the literature. Comprehensive reviews can be found in [2] and, more recently, in [4]. In the last decade, there has been an increasing complexity in index tracking models due to the inclusion of new application oriented constraints in the effort of better adapting the models to real-world practice. Much debate has also been focused on establishing which tracking error measure is more appropriate (see, e.g., [2, 4, 24]).

Currently, there is no generally accepted mathematical model for IT (see [4]). For instance, in [4] the authors propose a mixed integer linear programming formulation for the IT problem with transaction costs and a cardinality constraint, and use a standard solver (CPLEX) for providing optimal tracking portfolios. Numerical examples are presented for datasets taken from Beasley's OR Library involving up to 2151 stocks. In [28] a constraint aggregation method for the mathematical formulation introduced in [2, 4] is presented, thus obtaining a mixed integer non linear programming problem that is solved for the Hang Seng 31 stock market. Fang and Wang [11] formulate the Index Tracking problem as a bi-objective programming problem with the aim of optimizing the excess return and the absolute downside deviation from the index return. A fuzzy approach leading to the solution of a linear programming problem is presented and applied to the Shanghai 180 index with 30 stocks.
Other authors consider hybrid approaches for tackling IT problems. Jansen and van Dijk [18] introduce an objective function that takes into account both the tracking error and the (integer) number of assets to be included in the portfolio. Once the number of stocks has been decided, the amount of total budget invested in each asset is found by solving a quadratic programming problem. Some experiments have been reported for financial datasets of up to 250 stocks. A similar strategy is adopted in [30], where the authors combine an evolutionary algorithm with quadratic programming for solving IT problems with respect to some financial datasets from the OR Library [2].
Stochastic programming has also received consensus in the literature. For instance, Yu et al. [34] provide a downside risk approach for the index tracking problem formulated via a Markowitz model. Small examples based on subsets of five assets taken from the Hang Seng stock market are reported. Multistage stochastic programming is another recent field of research for IT problems (see [1] and the references therein). In [1] the authors formulate and solve a multistage tracking error model in a stochastic programming framework, and test their model by dynamically replicating the MSCI Euro index using an increasing number of scenarios, although with very few assets (up to 9). A two-stage stochastic program is also presented in [33] and applied to track a Canadian S&P/TSX index composed by a universe of 1150 stocks.

In general, the exact solution of the IT problem through mathematical programming formulations and methods poses serious challenges in terms of the computational time required, especially for large-size problems. As a consequence, several heuristic procedures have been proposed in the literature. In particular, much attention has been devoted to stochastic search heuristics. These procedures are usually more flexible to deal with the increased complexity of the models due to the introduction of new real-world constraints. For example, [2] proposes to use Genetic Algorithms (see also [17]); [13] and [9] rely on a Threshold Accepting heuristic procedure and, more recently, [23], [26], and [32] suggest to use a Differential Evolution procedure for the index tracking problem with the cardinality constraint. In particular, in [23] the authors test their algorithm on two well-known financial dataset, namely Dow Jones 65 and Nikkei 225. The reader is referred to [15] for a recent and comprehensive description and discussion on several well-known heuristic techniques and their use in tackling actual financial problems.

The paper is organized as follows. Section 2 is the methodological core of this work. Here, we state the IT optimization problem and we provide its first reformulation as a mixed integer quadratic programming problem. Section 3 briefly describes the state-of-art stochastic search heuristic we use for comparison with our exact approach, and Section 4 reports the main empirical findings on real-world datasets and presents a hybrid approach that combines the exact and the heuristic methods.

2 The Index Tracking Problem

2.1 Problem Formulation

We set up the IT problem as a minimization problem with the tracking error volatility as objective function. Formally, we have

$$\min \ \mathtt{f}(\mathbf{w}) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(R_t^P(\mathbf{w}) - R_t^B)^2} \tag{1}$$

subject to

$$\sum_{i=1}^{n} w_i = 1 \tag{2a}$$

$$0 \le w_i \le 1 \qquad i = 1, \dots, n \tag{2b}$$

$$\varepsilon_i \delta(w_i) \le w_i \le \xi_i \delta(w_i) \quad \text{with} \quad \delta(w_i) = \begin{cases} 1 & \mathtt{if}\ w_i > 0 \\ 0 & \mathtt{otherwise} \end{cases} \qquad i = 1, \dots, n \tag{2c}$$

$$L \le \sum_{i=1}^{n} \delta(w_i) \le K \tag{2d}$$

$$\sum_{i=1}^{n} |\overline{w}_i - w_i| \le TO \tag{2e}$$

$$\sum_{i=1}^{n} \lambda(w_i) w_i \le Ub \quad \text{with} \quad \lambda(w_i) = \begin{cases} 1 & \mathtt{if}\ w_i > Lb \\ 0 & \mathtt{otherwise} \end{cases} \qquad i = 1, \dots, n \tag{2f}$$

where

n is the number of available assets composing the index;

$\mathbf{w}$ is the $1 \times n$ vector whose entries w_i are the fractions of a given capital invested in asset i;

B_t is the index or benchmark value at time t, with $t = 1, ..., T$;

$R_t^B = ln(\frac{B_t}{B_{t-1}})$ is the index log-return at time $t = 1, ..., T$;

$R_t^P(\mathbf{w}) = r_{\boldsymbol{t}}^{P}\mathbf{w}$ is the portfolio return at time $t = 1, ..., T$;

r_t^P is the $1 \times n$ vector of the n assets log-returns at $t = 1, ..., T$.

Constraints (2a) and (2b) are the so-called budget and no-short selling constraints, while (2c) and (2d) limit the quantity and the number of assets to be included in the tracking portfolio, where $\varepsilon_i,\ \xi_i \in [0,1]$, with $\varepsilon_i < \xi_i$ for $i = 1,\dots,n$, are the lower and upper bounds for each individual asset weight w_i; L and K are the lower and upper bounds on the number of assets. These two constraints are the well-known Buy-in Threshold and Cardinality constraints, respectively.

Constraint (2e), where $\overline{w}_i$ refers to the fraction of capital invested in asset i in the previous time period, is the so-called Turnover constraint. It sets an upper bound $TO \in (0,1)$ on the total change in the portfolio composition, expressed in asset weights, between two consecutive time periods. It allows to control for transaction costs when updating the tracking portfolio in time. Finally, constraint (2f) models the concentration limits of the UCITS rules, where Lb is the lower bound threshold that characterizes an asset weight to be "very large", and Ub is the allowed maximal percentage of the sum of "very large" asset weights [23].

The above optimization problem is non linear and non convex due to the presence of constraints (2c)-(2f). Such problems are typically quite difficult to solve, particularly when the problem size is large. However, we now show that the index tracking problem can be reformulated as a more structured problem for which exact solution methods are available.

2.2 Problem Reformulation

We present here the first Mixed Integer Quadratic Programming (MIQP) formulation for the Index Tracking problem subject to real-world constraints including the UCITS rule described by constraint (2f).

First, note that constraints (2c) and (2d) are commonly modeled (see, e.g., [19, 31]) by using binary variables $y_i \in \{0,1\}$, with $i = 1,\dots,n$, so that we can re-write them as:

$$\varepsilon_i y_i \le w_i \le \xi_i y_i \qquad i = 1,\dots,n, \tag{3}$$

$$L \le \sum_{i=1}^{n} y_i \le K. \tag{4}$$

The sum of absolute values in the Turnover constraint (2e) can be linearized as in [22] by adding new variables $\gamma_i \ge 0$ bounding $|\overline{w}_i - w_i|$ for $i = 1,\dots,n$, so that constraint (2e) becomes equivalent to the following system of linear constraints:

$$-\gamma_i \le \overline{w}_i - w_i \le \gamma_i \qquad i = 1,\dots,n, \tag{5}$$

$$\sum_{i=1}^{n} \gamma_i \le TO. \tag{6}$$

We now describe the first mixed integer linear reformulation for the UCITS constraint $\sum_{i=1}^{n} \lambda(w_i)w_i \le Ub$ in (2f). To this end, we need to introduce new variables $v_i \in \{0,1\}$, and $u_i \ge 0$, for $i = 1,\dots,n$, such that:

$$u_i = \begin{cases} w_i & \text{if } w_i > Lb \\ 0 & \text{otherwise} \end{cases} \qquad i = 1,\dots,n, \tag{7}$$

and

$$v_i = \begin{cases} 1 & \text{if } w_i > Lb \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, \ldots, n. \tag{8}$$

We can now replace the non-linear UCITS constraint (2f) with the following linear ones:

$$v_i \in \{0, 1\} \qquad i = 1, \ldots, n, \tag{9}$$

$$0 \leq u_i \leq v_i \qquad i = 1, \ldots, n, \tag{10}$$

$$w_i - (1 - v_i) \leq u_i \leq w_i \qquad i = 1, \ldots, n, \tag{11}$$

$$v_i Lb \leq w_i \leq v_i + Lb \qquad i = 1, \ldots, n, \tag{12}$$

$$\sum_{i=1}^{n} u_i \leq Ub. \tag{13}$$

The proposed reformulation is justified by the following new result.

Theorem 1 *Let* $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{R}^n$ *satisfy (2a) and (2b). If* $\mathbf{w}$ *satisfies constraint (2f), then it also satisfies constraints (9)-(13) together with the vectors* $\mathbf{u} = (u_1, \ldots, u_n)$ *and* $\mathbf{v} = (v_1, \ldots, v_n)$ *defined by (7) and (8), respectively. Furthermore, if* $\mathbf{u}, \mathbf{v}$ *and* $\mathbf{w}$ *satisfy (9)-(13), then* $\mathbf{w}$ *satisfies (2f).*

Proof:
Let $\mathbf{w}$ satisfy (2a), (2b), and (2f). Observe that, by definition (7), $u_i = \lambda(w_i)w_i$ so that constraint (2f) trivially implies constraint (13). To prove that constraint (2f) also implies constraints (9)-(12) it is sufficient to check that (9)-(12) hold both in the case $w_i \leq Lb$, where $u_i = 0$ and $v_i = 0$, and in the case $w_i > Lb$, where $u_i = w_i$ and $v_i = 1$.
Conversely, to prove that constraints (9)-(13) imply (2f), we consider the following cases.
i) If $w_i \geq Lb + \epsilon$ (so that $\lambda(w_i) = 1$), then (12) implies $v_i = 1$, so that (11) implies $u_i = w_i = \lambda(w_i)w_i$.
ii) If $w_i < Lb$ (so that $\lambda(w_i) = 0$). Then (12) implies $v_i = 0$, so that (10) implies $u_i = 0 = \lambda(w_i)w_i$.
iii) If $w_i = Lb$ (so that $\lambda(w_i) = 0$), then from (9)-(13) it follows that we have either $u_i = 0$, $v_i = 0$ or $u_i = w_i = Lb$, $v_i = 1$. In both cases we have $u_i \geq \lambda(w_i)w_i$.
Thus, in all cases we have $\sum_{i=1}^{n} \lambda(w_i)w_i \leq \sum_{i=1}^{n} u_i \leq Ub$, and the proof is complete. □

Due to the monotonicity of the square root, the objective function (1) can be squared without changing the set of global minimizers of the IT problem so that (1) can be equivalently rewritten as a quadratic function of the form:

$$\mathbf{w}'Q\mathbf{w} + c'\mathbf{w}. \tag{14}$$

Thus, we can reformulate the Index Tracking problem as a Mixed Integer Quadratic Programming problem that minimizes function (14) subject to the linear constraints (2a),(2b), (3), (4), (5), (6), and (9)-(13). The resulting model has $3n$ continuous variables and $2n$ binary variables.

Obviously, the theoretical complexity status of the MIQP problem does not change as it still falls into the class of NP-Hard problems. However, the above problem can now be tackled by using some standard efficient optimization solvers like CPLEX, which allows to find exact solutions in a reasonable amount of time for small- and medium-size problems as reported in Section 4.

3 A Heuristic Approach

Stochastic search heuristics work by iteratively refining candidate solutions to an optimization problem using mathematical operators, usually inspired by natural and biological processes. They work by promoting the survival of the fittest, which corresponds to the optimal solution at the end of the run. Among them, Simulated Annealing (SA) [21] and Threshold Acceptance [9] work by refining a single solution until convergence, while Genetic Algorithms (GA) [12] and [17], Particle Swarm Optimization (PSO) [20] and Differential Evolution (DE) [32] work by simultaneously evolving a population of candidate solutions. One of their main advantages is their capability of tackling an optimization problem as complex as it is, without requiring any rigid mathematical assumption, such as linearity and continuity, or any additional information, such as the gradient or the Hessian matrix. They have often been criticized because of the need of long runtime, extensive parameter tuning and the lack of theoretical guarantee of converging to the global optimum. However, the current progress in hardware development and the possibility of parallel running have contributed to speed-up their runtime by orders of magnitude, decreasing the gap with alternative methods. Moreover, despite often been considered as a cons, the possibility of using different parameter settings could also provide significant advantages, given that it often allows a better control of the exploitation-exploration trade-off, avoiding premature convergence to local optima. Furthermore, it has been shown that some heuristics, such as Differential Evolution, are rather insensitive to parameter tuning. They are currently becoming standard tools in many application fields, including finance. The reader is referred to [14] for a more comprehensive introduction and discussion of their usage in financial and economic applications.

Index tracking is no exception. As [2] and [4] report in the literature review sections, many studies have shown that different heuristics could allow to effectively tackle such problem: for example, among the many contributions, [2] proposes to use Genetic Algorithms, [13] proposes to use Threshold Accepting, and [26] relies on Differential Evolution for index tracking. More recently, in [23] the authors have provided a new heuristic, Differential Evolution and Combinatorial Search for Index Tracking (DECS-IT), which relies on Diffential Evolution and on an ad hoc combinatorial search operator. DECS-IT is a population-based heuristic based on Differential Evolution, originally introduced by [32], which requires little parameter tuning and has shown better performances in continuous numerical problems and less sensitivity to parameter tuning, when compared to other heuristics, such as genetic algorithms and particle swarm optimization (see, e.g., [29] for clustering problems and [23] for IT).

One of the main challenges in index tracking is given by the presence of the cardinality constraint in a large search space. This requires to select the optimal subset of assets (a combinatorial problem) and then to fine-tune the optimal asset weights (a continuous problem). In [23] a position-swapping operator is introduced, which allows to avoid premature convergence of the DECS-IT and to better explore different asset subsets by swapping active weights with zero weights using a probabilistic scheme. After generating and evaluating an initial population, different evolutionary operators are iteratively applied until a termination condition is satisfied. First, the population undergoes mutation and recombination, according to the '$Rand/1/Exp$'

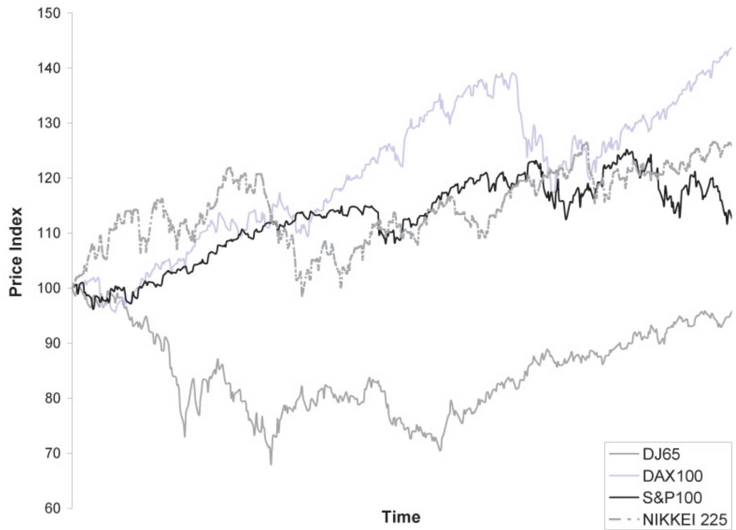

Figure 1: Price Indexes (rebased 100).

scheme [32]. Second, solutions are scaled to sum to one to satisfy the budget constraint. Third, unfeasible solutions are repaired (see [23], Section 3.3, for a description of the constraint handling mechanism). Fourth, the position swapping is applied. Finally, solutions in the current population are evaluated and if they are better than the ones in the previous iteration, they survive to the next iteration. The reader is referred to [23] for a detailed description of DECS-IT, extensive investigation on parameter tuning, and empirical results on the IT problem with real-world data.

4 Empirical Results

4.1 The Data

The empirical analysis has been performed on the daily log-returns of four equity indexes and their constituents.[1] More precisely, we consider the Dow Jones 65 (DJ65 - Period: 12/04/2002-19/12/2003), the German Dax 100 (Dax100 - Period: 17/03/2005-12/12/2006), the Standard & Poor's 100 (S&P100 - Period: 25/05/2006-31/01/2008) and the Nikkei 225 (Nikkei225 - Period: 18/11/2005-27/07/2007) market indexes, using for each of them a sample of 441 observations. As Figure 1 shows, while the Dax100 and the S&P100 show an increase in value over time, the DJ65 first decreases in value and then grows back almost to the original level. The Nikkei225 alternates up and down movements, ending with an overall increase in the time period. Table 1 reports the summary statistics of the indexes daily log-return times series. All of them have mean and median values close to zero, small standard deviations and domain of variation, slight asymmetries and moderate kurtosis.

[1]For the Dax100 and the S&P100 we consider only 98 out of 100 constituents.

Data	n	$\overline{T}$	Mean ($\times 10^{-2}$)	Median ($\times 10^{-2}$)	Std. Dev. ($\times 10^{-2}$)	Skewness	Kurtosis	Min	Max
DJ65	65	440	0.00	0.00	1.30	0.19	4.40	-0.05	0.05
Dax100	98	440	0.08	0.15	0.90	-0.48	4.21	-0.03	0.03
S&P100	98	440	0.02	0.03	0.90	-0.51	5.14	-0.04	0.03
Nikkei225	225	440	0.05	0.02	1.10	-0.26	3.84	-0.04	0.03

Table 1: Descriptive Statistics of the daily log-returns.

4.2 Experimental Set-Up

We solve the optimization problem by considering in all the experiments the following parameter values: $\epsilon_i = 0.01$, $\xi_i = 0.1$, $i = 1, \ldots, n$, $L = 0$, $Lb = 0.05$, $Ub = 0.4$ and $TO = 0.1$.
DECS-IT has been implemented in MATLAB initializing the population randomly at the first iteration and setting the population size equal to 200, the number of iterations to 10000, the scaling factor to 0.3 and the crossover rate to 0.8. We note however that the DECS-IT performance, as shown in Appendix A of [23], is rather insensitive to parameter tuning. The largest cost in terms of runtime depends on the number of function evaluations, which corresponds to the product between the population size and the number of iterations, and on the repairing of unfeasible solutions during constraint handling. On the other hand, increasing the problem size while keeping the same number of iterations, does not require much longer runtime.The reader is referred to Appendix A in [23] for an extensive discussion regarding the parameter tuning.

The Mixed Integer Quadratic Programming problem has been solved using the toolbox TOMLAB/CPLEX 11.2 in a MATLAB 7.11.0 environment on a workstation with Intel Core2 Duo CPU (T7500, 2.2 GHz, 4Gb RAM) under the Windows operating system (MIQP-CPLEX). We also set the value of 10^{-8} as absolute and relative tolerances on the gap between the best integer objective found and the objective of the best node remaining in the branch and bound tree.

4.3 Heuristic versus Exact Method

Our analysis relies on two different sets of experiments. In the first set we aim at comparing DECS-IT and MIQP-CPLEX by considering single-period optimization, $K = 20$ and $K = 40$, and the different problems sizes of the four datasets ($n = 65$ for DJ65, $n = 98$ for Dax100 and S&P100, and $n = 225$ for the Nikkei225 market). In the second set of experiments we use a rolling window mechanism: by using a window size of 200 observations, we solve the IT problem for overlapping windows built by moving forward in time with step size 20 for 12 time periods. This corresponds to monthly rebalancing. We then compare the objective function values of the portfolios obtained by DECS-IT and by MIQP-CPLEX in each time period. Furthermore, we propose a hybrid method using both approaches that works best for the multi-period case.

4.3.1 Single-Period Results

The first set of experiments provides some insights regarding the on-going debate between the use of heuristics versus exact methods in IT problems. Here, we solve the Index Tracking optimization problem only on the first time window, thus excluding the turnover constraint. In

$PRE(\%)$	DJ65 K=20	DJ65 K=40	Dax100 K=20	Dax100 K=40	S&P100 K=20	S&P100 K=40	Nikkei225 K=20	Nikkei225 K=40
Min	3.17	0.22	0.77	4.50	-0.28	5.30	1.14	2.38
Mean	14.76	7.05	11.92	11.69	20.79	16.66	10.77	5.73
Median	14.82	6.29	9.48	10.90	17.34	15.37	9.29	5.41
Max	27.24	22.41	38.94	22.36	91.23	33.19	39.48	11.93
Std. Dev (%)	6.83	5.00	9.85	4.41	18.73	7.19	7.26	2.83
Time MIQP-CPLEX	640	790	3000	3600	>2h	>2h	>2h	>2h
Time DECS-IT (for 10000 it.)	780	730	900	934	900	685	840	1080

Table 2: Descriptive Statistics (in %) of the Percentage Relative Errors of DECS-IT vs. MIQP-CPLEX: Min, Mean, Median, Max and Standard Deviations of PRE_j ($j = 1, \dots, 30$). In the table, ">2h" means that CPLEX has been stopped after two-hours of computing.

other words, we determine the optimal solutions by only using the first 200 observations of our datasets. Since DECS-IT is a stochastic search heuristics, we run the algorithm 30 times, re-starting each time from a randomly chosen initial population. On the contrary, a single run of MIQP-CPLEX is performed.

To compare the performance of DECS-IT and MIQP-CPLEX, we consider the Percentage Relative Error (PRE) between the optimal solution found by MIQP-CPLEX ($MIQPopt$) and the optimal solutions of DECS-IT ($bestDECS_j$, $j = 1, \dots, 30$) in the single-period optimization problem. The Percentage Relative Error in the j-th run (PRE_j) is defined as:

$$PRE_j = \frac{bestDECS_j - MIQPopt}{MIQPopt}\%. \tag{15}$$

Table 2 reports the minimum, mean, median, maximum, and standard deviation values of PRE for $K = 20$ and $K = 40$ for the four datasets. We notice a great variability on the maximum PRE that ranges from 11.93% to 91.23% but, on the other hand, the mean and especially the median values are small, pointing out that the heuristic procedure is able on average to obtain solutions very close to the optimal (or nearly optimal) ones found by CPLEX. Furthermore, the minimum values are very small, always less than 5.30%, thus giving further evidence of the good performance of the DECS-IT heuristic. Actually, in one case (i.e., S&P100 for $K = 20$) the PRE is slightly negative, showing that the best solution found by the DECS-IT heuristic is better than the best solution found by CPLEX within two hours of computing time. It should be observed, however, that running the heuristic 30 times requires around 27000 seconds (i.e., 900s per 30 runs) in this case, which is much longer than the time allocated to CPLEX. We also observe that standard deviations are relatively large (more than 8%) only in two of the most challenging cases, reaching at most the value of 18.73%. The last two rows of Table 2 report the time spent for finding an optimal solution using MIQP-CPLEX and DECS-IT (average time in 30 runs). As expected, when the problem size n increases, the time requested by MIQP-CPLEX to find an optimal solution increases considerably. Hence, we decided to stop the runs after two-hours of computing (7200 seconds) for the S&P100 and the Nikkei225 datasets thus obtaining a good (possibly optimal) solution but without a guarantee for optimality. On the other hand, increasing the problem size has a smaller effect on the DECS-IT average time. The good performance of DECS-IT is also remarked in Table 3 where we report some statistics

Statistics ($\times 10^{-2}$)	DJ65 K=20	DJ65 K=40	Dax100 K=20	Dax100 K=40	S&P100 K=20	S&P100 K=40	Nikkei225 K=20	Nikkei225 K=40
Max	6.30	2.20	1.20	0.90	8.90	4.20	5.80	2.80
Mean	0.50	0.30	0.06	0.17	0.90	0.76	0.27	0.25
Std. Dev	1.20	0.40	0.21	0.33	1.90	0.99	1.00	0.59
Abs. Differences	0.33	0.18	0.51	0.16	0.94	0.74	0.61	0.56

Table 3: Descriptive Statistics of the differences and the sum of the absolute differences in the assets weights composition between the portfolios found by MIQP-CPLEX and the best portfolios found by DECS-IT.

about the (absolute) differences between the weights of the portfolios found by CPLEX and the best portfolios found by DECS-IT. The maximum, mean and the standard deviation values in the first three rows of the table account for this comparison. The minimum of such differences is always equal to zero. The last row also reports the sum of the absolute differences of the portfolios weights. Table 3 points out that the gap found in the objective function values are indeed mostly due to the small differences in the assets weights composition.

4.3.2 Multi-Period Results and a Hybrid Approach

In the second set of experiments, we compare the tracking errors of the portfolios found in all time windows by MIQP-CPLEX and by DECS-IT. It is important to notice that some caution is needed in order to provide a fair comparison between the values of the solutions obtained by MIQP-CPLEX and by DECS-IT in a given time window. Indeed, we recall that the turnover constraint depends on the values $\overline{w}_i$, $i = 1, \dots, n$, of the fractions of the capital invested in each asset in the previous window. Hence, such values determine the shape of the feasible region for all windows from 2 to 12. Recall that the values $\overline{w}_i$ are the solutions of the optimization problem in the previous window, and hence could be different if computed by MIQP-CPLEX or by DECS-IT. Thus, to correctly compare the quality of the solutions found by the two methods for all windows, the idea is to first implement DECS-IT for finding the solutions (portfolios) in all the 12 time periods, and then use the weights of the portfolios obtained in the corresponding turnover constraint when implementing MIQP-CPLEX. This guarantees that, in each window, the structure of the turnover constraint is always the same as the one considered in the heuristic approach, thus not affecting the feasible region. More precisely, in order to perform more experiments, among the 30 runs of the DECS-IT heuristic we choose the five sets of 12 solutions that give the best values of the objective function in the first window, and, as mentioned above, we use these solutions in the turnover constraints of the MIQP model to compute the optimal solution for the subsequent window with CPLEX.

In Table 4, for each dataset, for K equal to 20 or 40, and for each time period, we report the average Percentage Relative Errors computed with respect to the five sets of portfolios used, along with the cumulate times required by MIQP-CPLEX to solve all problems for windows 2-12. We observe that now the average relative error is considerably smaller and no more than 10% in all cases. Furthermore, in this case CPLEX was able to solve to optimality all 11 problems in about one tenth of the time needed to solve to optimality (when possible) the single problem regarding the first window, which does not include the turnover constraints. These results suggest that the DECS-IT heuristic could be combined with the exact MIQP-CPLEX approach in a more efficient and accurate hybrid method: in the first window one obtains

Time Period	DJ65 K=20	DJ65 K=40	Dax100 K=20	Dax100 K=40	S&P100 K=20	S&P100 K=40	Nikkei225 K=20	Nikkei225 K=40
2	2.03	1.18	1.89	3.01	0.49	1.73	0.75	0.51
3	0.48	1.63	2.49	3.58	0.46	2.68	1.00	0.40
4	1.08	1.78	0.57	3.97	3.81	4.95	1.44	1.27
5	1.52	2.24	0.25	3.01	0.97	1.88	1.38	1.06
6	1.09	1.60	3.41	1.91	2.20	5.27	2.60	1.87
7	2.49	4.62	3.79	1.20	5.78	7.67	2.66	5.02
8	1.88	4.58	3.30	1.56	2.76	4.29	3.37	5.78
9	2.00	4.28	9.11	2.50	3.29	7.02	1.58	4.55
10	2.98	7.35	1.79	3.43	4.39	4.86	1.87	8.73
11	2.44	4.29	8.91	1.39	4.51	9.25	2.89	5.28
12	4.17	3.86	2.33	2.43	3.51	5.55	3.23	7.51
MIQP-CPLEX Time Sec.	45	100	160	880	200	900	550	4000

Table 4: Mean Percentage Relative Errors (in %) of DECS-IT vs. MIQP-CPLEX.

with the heuristic procedure nearly optimal solutions in a reasonable time. Such solutions are then used as reference points in the turnover constraint of the MIQP model that is solved to optimality for the second window with CPLEX in a reasonable time even for medium/large problems. Then, for windows 3 to 12 the exact optimal solutions are still found with CPLEX using as reference points in the turnover constraint the exact solutions of the previous window. This approach combines the good features of both methods for a fast and accurate solution for the IT problem.

4.4 Performance analysis

In this section, we aim to provide some insight regarding the in-sample and out-of-sample financial performance of the optimal portfolios obtained by solving the IT problem with MIQP-CPLEX and with DECS-IT. First, we solve the MIQP problem in each of the twelve 200 observations time window (in-sample analysis), and we hold the optimal portfolio found for the following 20 periods (out-of-sample analysis). This is the same rolling window scheme used in [23]. The out-of-sample performance of MIQP-CPLEX is evaluated first by comparing it with the out-of-sample performance statistics of DECS-IT provided in [23] for DJ65 and Nikkei225, where the optimal portfolio is held unchanged for the 20 observations consecutive to the in-sample period. Then, we further investigate on the relations between the in- and out-of-sample errors of MIQP-CPLEX and of DECS-IT finding that the out-of sample errors are, as expected, larger than the in-sample errors. Furthermore, the out-of-sample performances of the in-sample optimal portfolios found by MIQP-CPLEX are good but not always better than the out-of-sample performances of the 30 portfolios found by DECS-IT.

In tables 5 and 6 we compare for DJ65 and Nikkei225 the performances of the optimal portfolios found with MIQP-CPLEX applied to our exact model to those of the portfolios found with the DECS-IT heuristic in [23]. Similar results have been obtained with MIQP-CPLEX for the Dax100 and for the S&P100 indexes. Details are available upon request. We point out that, due to the problem dimension, for the Nikkei225 dataset we stopped MIQP-CPLEX after two-hours of computing. However, for both datasets, we observe that the in-sample tracking volatility values are lower than the corresponding ones presented in [23].

The in-sample annualized Tracking Error volatility and the Excess Return correspond to the average of the annualized in-sample tracking error volatilities and to the annualized in-sample excess returns, respectively, for the 12 time periods considered in our experiments (see also [23]). The out-of-sample beta relates to the regression of the portfolio returns during the holding period against the returns from the index [4]. The closer to 1, the better the tracking performance of a portfolio. Indeed, a portfolio that perfectly tracks a market index has beta exactly equal to 1. The reader is referred to [23] for a detailed description of the reported statistics.

For all dataset, we set all the parameters values as in Section 4.2, except considering $K = \{20, 25, 30, 35, 40, 45, 50\}$ for DJ65 and $K = \{20, 30, 40, 50, 60, 70, 80, 85\}$ for Nikkei225 (see also [23]).

From a tracking performance viewpoint we notice that, as expected, the in-sample tracking errors decrease when increasing the parameter K, with a steeper slope for smaller values of K that eventually flattens for larger values. Clearly there is no guarantee of such behavior for the out-of-sample tracking error. However, this decrease has been observed in all our experiments except for a slight inversion with $K = 35$ and $K = 40$ for the S&P100 index. In spite of this, from a practical viewpoint increasing the number of assets in the tracking portfolio leads to a more expensive investment in terms of transaction and managing costs. We also note that no monotonic behavior seems to show up for the in-sample or out-of-sample excess returns, and for the beta and correlation with respect to the index. However, the latter two parameters turned out to be close to 1 for all values of K thus confirming the excellent tracking properties of the optimized portfolios.

Table 7 provides further evidence of the fact that in-sample optimization does not necessarily guarantee best out-of-sample performance. Indeed, referring only to the first 200 observations, we compare the in-sample and out-of-sample tracking errors of the optimal portfolios found with MIQP-CPLEX with those of the portfolios found with the 30 replications of the DECS-IT heuristic. We observe that there is not a systematic relationship between the in-sample and the out-of-sample tracking performance of the 30 portfolios found. Furthermore, when the in-sample tracking error of the optimal portfolio is the smallest, its corresponding out-of-sample performance is the best one just in one case (Nikkei225), although it is almost in all cases among the best ones. This should partially justify the search for the optimal in-sample solution with MIQP-CPLEX when possible. However, since there is not a complete dominance relation between the out-of-sample volatility values, as well as for the beta and correlation statistics, and since DECS-IT is able to find in reasonable time near-optimal portfolios for the first 200 observations time window (i.e., without turnover constraints), it could be fruitfully used in combination with MIQP-CPLEX, which is faster and exact, in the remaining 11 time windows to obtain in a reasonable time near-optimal IT portfolios that have a good out-of-sample performance.

Acknowledgements

The authors would like to thank two anonymous referees for their helpful advices. Sandra Paterlini conducted part of this research while visiting the School of Mathematics, University of Minnesota, USA. Financial support from the Fondazione Cassa di Risparmio di Modena is gratefully acknowledged.

References

[1] Barro, D. and Canestrelli, E. (2004). Tracking error: a multistage portfolio model. Annals of Operations Research, 165, 47–66.

[2] Beasley, J.E. and Meade, N. and Chang, T.-J. (2009). An evolutionary heuristic for the index tracking problem. European Journal of Operational Research, 148, 621–643.

[3] Bianchi, D. and Gargano, A. (2011). High-dimensional Index Tracking with Cointegrated assets using an hybrid Genetic Algorithm. Manuscript available at: http://ssrn.com/abstract=1785908.

[4] Canagkoz, N.A. and Beasley, J.E. (2008). Mixed-integer programming approaches for index tracking and enhanced indexation. European Journal of Operational Research, 196, 384–399.

[5] Chang, T.-J. and Meade, N. and Beasley, J.E. and Sharaiha, Y.M. (2000). Heuristics for cardinality constrained portfolio optimisation. Computers & Operations Research, 27, 1271–1302.

[6] Coleman, T., and Li, Y. and Henniger, J. (2006). Minimizing Tracking Error While Restricting the Number of Assets. Journal of Risk, 8, 33–56.

[7] Derigs, U. and Nickel, N.H. (2003). Meta-heuristic based decision support for portfolio optimization with a case study on tracking error minimization in passive portfolio management. OR Spectrum, 25, 345–378.

[8] Derigs, U. and Nickel, N.H. (2004). On a Local-Search Heuristic for a Class of Tracking Error Minimization Problems in Portfolio Management. Annals of Operations Research, 131, 45–77.

[9] Dueck, G. and Scheuer, T. (1990). Threshold accepting: a general purpose algorithm appearing superior to simulated annealing. J. Comput. Physics, 90, 161–175.

[10] Dueck, G. and Winker, P. (1992). New concepts and algorithms for portfolio choice. Applied Stochastic Models and Data Analysis, 8, 159–178.

[11] Fang, Y. and Wang, S.-Y. (2005). A fuzzy index tracking portfolio selection model. Lecture Notes in Computer Science, 3516, 554–561.

[12] Fogel, L.J. and Owens, A.J. and Walsh, M.J. Artificial intelligence through simulated evolution, New York, John Wiley, 1966.

[13] Gilli, M. and Kellezi, E. The Threshold Accepting Heuristic for Index Tracking, Financial Engineering, E-Commerce, and Supply Chain, Kluwer Applied Optimization Series, 2002.

[14] Gilli, M., Maringer, D., Schumann, E., 2011. Numerical Methods and Optimization in Finance. Academic Press.

[15] Gilli, M. and Schumann E. Heuristic optimisation in financial modelling. Annals of Operations Research, DOI 10.1007/s10479-011-0862-y.

[16] Gilli, M., Winker, P., 2009. Heuristic optimization methods in econometrics. In: Belsley, D., Kontoghiorghes, E. (Eds.), Handbook of Computational Econometrics. Wiley, Chichester, pp. 81–119.

[17] Holland, J.H. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Harbor, 1975.

[18] Jansen, R. and van Dijk, R. (2002). Optimal Benchmark Tracking with Small Portfolios. The Journal of Portfolio Management, 28, 33–39.

[19] Jobst, N.J. and Horniman, M.D. and Lucas, C.A. and Mitra, G. (2001). Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. Quantitative Finance, 1, 1–13.

[20] Kennedy, J. and Eberhart, R.C. (1995). Particle swarm optimisation. In: Proc. of the 1995 IEEE International Conference on Neural Networks, IEEE Press, Piscataway, NJ, 4, 1942-1948.

[21] Kirkpatrick, S. and Gelatt, C.D. and Vecchi, M.P. (1983). Optimization by Simulated Annealing. Science, 220, 671-680.

[22] Konno, H. and Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. Management Science, 37, 519–31.

[23] Krink, T. and Mittnik, S. and Paterlini, S. (2009). Differential Evolution and Combinatorial Search for Constrained Index Tracking. Annals of Operations Research, 172, 153–176.

[24] Lobo, M.T. and Fazel, M. and Boyd, S. (2007). Portfolio optimization with linear and fixed transaction costs. Annals of Operations Research, 152, 341–365.

[25] Maringer, D. and Kellerer, H. (2003). Optimization of cardinality constrained portfolios with a hybrid local search algorithm. OR Spectrum, 25, 481–495.

[26] Maringer, D. and Oyewumi, O. (2007). Index tracking with constrained portfolios. Intelligent Systems in Accounting, Finance and Management, 15, 57–71.

[27] Michalewicz, Z. and Fogel, D.B. How to solve it: modern heuristics, Springer, 2004.

[28] Okay, N. and Akman, U. (2003). Index tracking with constraint aggregation. Applied Economics Letters, 10, 913-916.

[29] Paterlini, S. and Krink, T. (2006). Differential evolution and particle swarm optimisation in partitional clustering. Computational Statistics and Data Analysis, 50, 1220-1247.

[30] Ruiz-Torrubiano, R. and Suárez, A. (2009). A hybrid optimization approach to index tracking. Annals of Operations Research, 166, 57–71.

[31] Speranza, M.G. (1996). A heuristic algorithm for a portfolio optimization model applied to the Milan stock market. Computers and Operations Research, 23, 433–441.

[32] Storn, R. and Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, 11, 341–359.

[33] Stoyan, A.S.J. and Kwon, B.R.H. (2007). A two-stage stochastic mixed-integer programming approach to the index tracking problem. Thechnical report Department of Mechanical and Industrial Engineering. University of Toronto, Canada.

[34] Yu, L. and Zhang, S. and Zhou, X.Y. (2006). A downside risk analysis based on financial index tracking models, in: A.N. Shiryaev, M.R. Grossinho, P.E. Oliveira, M.L. Esquivel (Eds.), Stochastic Finance, Springer, 213–236.

K	Tracking Error in-sample (%)	Tracking Error out-of-sample (%)	Excess Return in-sample (%)	Excess Return out-of-sample (%)	Beta out-of-sample	Correlation out-of-sample (%)
20	0.19	3.35	0.20	0.07	0.95	99.70
	0.21	*3.64*	*0.23*	*-6.80*	*0.72*	*99.17*
25	0.16	2.75	1.43	-2.60	0.90	99.83
	0.17	*2.97*	*1.94*	*3.87*	*1.10*	*99.81*
30	0.13	2.55	1.33	0.66	0.96	99.82
	0.14	*2.52*	*2.21*	*-0.39*	*0.94*	*99.80*
35	0.12	2.29	0.72	-0.28	0.94	99.87
	0.13	*2.38*	*1.50*	*-0.78*	*0.94*	*99.82*
40	0.10	2.10	1.20	0.83	1.00	99.95
	0.12	*2.18*	*1.79*	*-0.23*	*0.95*	*99.86*
45	0.09	1.90	1.20	1.00	1.02	99.94
	0.10	*1.89*	*1.83*	*0.08*	*0.98*	*99.94*
50	0.09	1.85	1.18	0.90	1.01	99.96
	0.09	*1.70*	*2.08*	*1.20*	*1.04*	*99.94*

Table 5: In-sample and out-of-sample statistics for DJ65. Italic values refer to the corresponding statistics provided in [23].

K	Tracking Error in-sample (%)	Tracking Error out-of-sample (%)	Excess Return in-sample (%)	Excess Return out-of-sample (%)	Beta out-of-sample	Correlation out-of-sample (%)
20	0.20	3.41	1.28	2.53	1.10	99.07
	0.21	*3.53*	*2.19*	*1.48*	*1.10*	*99.27*
30	0.14	2.55	0.39	-3.50	0.82	97.73
	0.15	*2.81*	*0.71*	*-0.74*	*0.97*	*98.75*
40	0.11	2.09	-1.47	-3.00	0.80	98.67
	0.12	*2.06*	*0.14*	*-2.43*	*0.86*	*99.57*
50	0.10	1.71	0.1	-1.47	0.88	99.00
	0.11	*1.80*	*-0.38*	*-2.01*	*0.87*	*99.14*
60	0.10	1.54	-0.22	-1.27	0.86	99.18
	0.10	*1.68*	*0.16*	*-2.52*	*0.79*	*98.29*
70	0.09	1.49	-0.38	-2.58	0.81	98.50
	0.10	*1.40*	*0.77*	*-2.40*	*0.89*	*99.40*
80	0.09	1.41	-0.38	-2.78	0.79	98.42
	0.10	*1.46*	*0.27*	*-4.55*	*0.70*	*97.46*
85	0.09	1.47	-0.39	-1.35	0.87	99.20
	0.10	*1.43*	*-0.14*	*-2.40*	*0.80*	*98.56*

Table 6: In-sample and out-of-sample statistics for Nikkei 225. Italic values refer to the corresponding statistics provided in [23].

DJ65 ($\times 10^{-3}$)				Dax100 ($\times 10^{-3}$)				S&P100 ($\times 10^{-3}$)				Nikkei225 ($\times 10^{-3}$)			
in-sample		out-of-sample		in-sample		out-of-sample		in-sample		out-of-sample		in-sample		out-of-sample	
CPLEX	**1.72**	13	1.72	CPLEX	**0.75**	24	1.05	CPLEX	**0.81**	18	0.76	CPLEX	**1.74**	CPLEX	**1.42**
26	1.78	29	1.85	16	0.76	1	1.09	20	0.81	5	0.78	2	1.76	5	1.48
12	1.79	19	2.03	22	0.76	30	1.09	9	0.87	24	0.86	15	1.77	11	1.48
23	1.79	1	2.36	11	0.76	22	1.10	18	0.88	29	0.89	25	1.77	18	1.48
20	1.80	15	2.38	1	0.77	11	1.15	5	0.88	8	0.91	1	1.80	25	1.49
16	1.81	8	2.48	30	0.77	28	1.15	24	0.91	26	0.95	4	1.84	17	1.52
7	1.84	22	2.58	26	0.77	12	1.15	16	0.91	9	1.02	10	1.84	19	1.56
17	1.88	10	2.58	9	0.78	19	1.16	21	0.91	13	1.03	19	1.85	23	1.58
30	1.91	CPLEX	**2.59**	8	0.78	8	1.17	15	0.91	22	1.06	7	1.86	2	1.58
2	1.93	25	2.64	4	0.79	20	1.17	8	0.92	20	1.07	13	1.86	1	1.59
3	1.94	7	2.65	13	0.79	CPLEX	**1.21**	29	0.92	30	1.10	20	1.87	4	1.70
29	1.95	27	2.67	7	0.79	5	1.25	26	0.93	23	1.11	23	1.87	10	1.70
24	1.95	21	2.67	28	0.79	13	1.29	23	0.94	11	1.16	5	1.90	8	1.70
10	1.96	28	2.67	14	0.80	21	1.30	22	0.95	16	1.26	11	1.90	14	1.70
19	1.96	17	2.68	3	0.81	16	1.31	10	0.95	2	1.27	18	1.90	20	1.71
4	1.97	4	2.76	19	0.82	18	1.33	14	0.95	CPLEX	**1.31**	28	1.91	16	1.79
9	1.98	9	2.77	23	0.83	17	1.33	6	0.96	25	1.31	17	1.91	29	1.81
15	1.99	23	2.91	15	0.83	14	1.34	17	0.96	4	1.31	22	1.93	6	1.82
1	1.99	14	2.97	10	0.84	25	1.40	13	0.96	19	1.32	24	1.94	12	1.82
11	2.00	26	2.98	5	0.85	26	1.41	7	0.96	28	1.33	8	1.94	27	1.86
6	2.02	5	3.04	27	0.85	10	1.42	11	0.97	10	1.38	14	1.94	22	1.93
22	2.06	3	3.09	25	0.85	3	1.48	27	0.98	14	1.39	30	1.95	15	1.95
25	2.07	6	3.11	20	0.86	6	1.50	12	0.98	12	1.42	29	1.97	24	1.97
14	2.08	18	3.15	24	0.89	7	1.50	4	0.98	27	1.44	16	1.99	7	1.99
18	2.08	16	3.20	17	0.89	9	1.50	19	1.00	17	1.51	21	1.99	13	1.99
13	2.08	2	3.28	12	0.90	27	1.55	25	1.01	21	1.55	3	2.02	30	2.01
5	2.10	11	3.37	21	0.93	4	1.55	30	1.02	3	1.58	9	2.02	3	2.19
8	2.10	12	3.61	2	0.96	23	1.56	1	1.04	6	1.60	6	2.03	9	2.19
28	2.13	30	3.62	29	0.96	15	1.56	28	1.07	15	1.60	12	2.03	28	2.40
21	2.19	20	3.65	6	0.97	2	1.79	3	1.46	7	1.63	27	2.10	21	2.44
27	2.19	24	3.82	18	1.04	29	1.79	2	1.56	1	1.65	26	2.43	26	2.55

Table 7: Ordered in-sample and out-of-sample tracking errors of the optimal solutions found by CPLEX (in boldface) and of the approximate solutions found with 30 replications of the DECS-IT heuristic for $K = 20$ with a holding (out-of-sample) period of one month. Odd-numbered columns contain the replication numbers of the DECS-IT heuristic that correspond to the in- or out-of-sample errors to their right.

RECent Working Papers Series

The 10 most RECent releases are:

No. 81 EXACT AND HEURISTIC APPROACHES FOR THE INDEX TRACKING PROBLEM WITH UCITS CONSTRAINTS (2012)
A. Scozzari, F. Tardella, S. Paterlini and T. Krink

No. 80 IMMIGRANT STUDENTS AND EDUCATIONAL SYSTEMS. CROSS-COUNTRY EVIDENCE FROM PISA 2006 (2012)
M. Murat, D. Ferrari and P. Frederic

No. 79 THE REGIONAL DISTRIBUTION OF PUBLIC EMPLOYMENT: THEORY AND EVIDENCE (2012)
S. G. Kessing and C. Strozzi

No. 78 ADAPTIVE MINIMAX ESTIMATION OVER SPARSE lq-HULLS (2012)
Z. Wang, S. Paterlini, F. Gao and Y. Yang

No. 77 CONVEX RADIANT COSTARSHAPED SETS AND THE LEAST SUBLINEAR GAUGE (2012)
A. Zaffaroni

No. 76 JOB SEPARATIONS, JOB LOSS AND INFORMALITY IN THE RUSSIAN LABOR MARKET (2012)
H. Lehmann, T. Razzolini and A. Zaiceva

No. 75 THE EVOLUTION OF THE RACIAL GAP IN EDUCATION AND THE LEGACY OF SLAVERY (2011)
G. Bertocchi and A. Dimico

No. 74 DO IMMIGRANT STUDENTS SUCCEED? EVIDENCE FROM ITALY AND FRANCE BASED ON PISA 2006 (2011)
M. Murat

No. 73 SELF-COMMITMENT-INSTITUTIONS AND COOPERATION IN OVERLAPPING GENERATIONS GAMES (2011)
F. Lancia and A. Russo

No. 72 LONG-RUN WELFARE UNDER EXTERNALITIES IN CONSUMPTION, LEISURE, AND PRODUCTION: A CASE FOR HAPPY DEGROWTH VS. UNHAPPY GROWTH (2011)
E. Bilancini and S. D'Alessandro

The full list of available working papers, together with their electronic versions, can be found on the RECent website: http://www.recent.unimore.it/workingpapers.asp