ACTIVE PORTFOLIO MANAGEMENT USING COMPUTATIONAL INTELLIGENT TECHNIQUES

by

Dr Nikos Thomaidis

Financial Engineering Group
Dept of Financial Engineering & Management
University of the Aegean, Chios Island, GREECE

Research & Development,
Kepler Asset Management, NY
nthomaid@fme.aegean.gr

Index Tracking

- Portfolio performance is evaluated relatively to the performance of a *benchmark* (stock, bond or commodity index)
- Passive vs *active* (*enhanced indexation*) portfolio management
- *Tracking error*: $r_{TE,t} = r_{B,t} - r_{P,t}$
- *Tracking error standard deviation* (*TESD*):

$$s_{TE} = \sqrt{(1/T)\sum_{t} r_{TE,t}^2}$$

- Probability of *underperforming* the benchmark

$$\Pr ob(r_{TE,t} < 0)$$

Cardinality constraint

- Constraint on the *maximum number of assets* included in the portfoli*o*
- *Why imposing a cardinality constraint*?
  - Benchmarks with high number of constituent assets
  - Avoiding holding large portfolios that accrue management & transaction costs

Cardinality-constraint portfolio optimisation

$$\max_{\mathbf{w}=(w_1,w_2,\ldots,w_N)'\in\Re^N} f(\mathbf{w}) \quad \text{(portfolio objective)}$$

such that

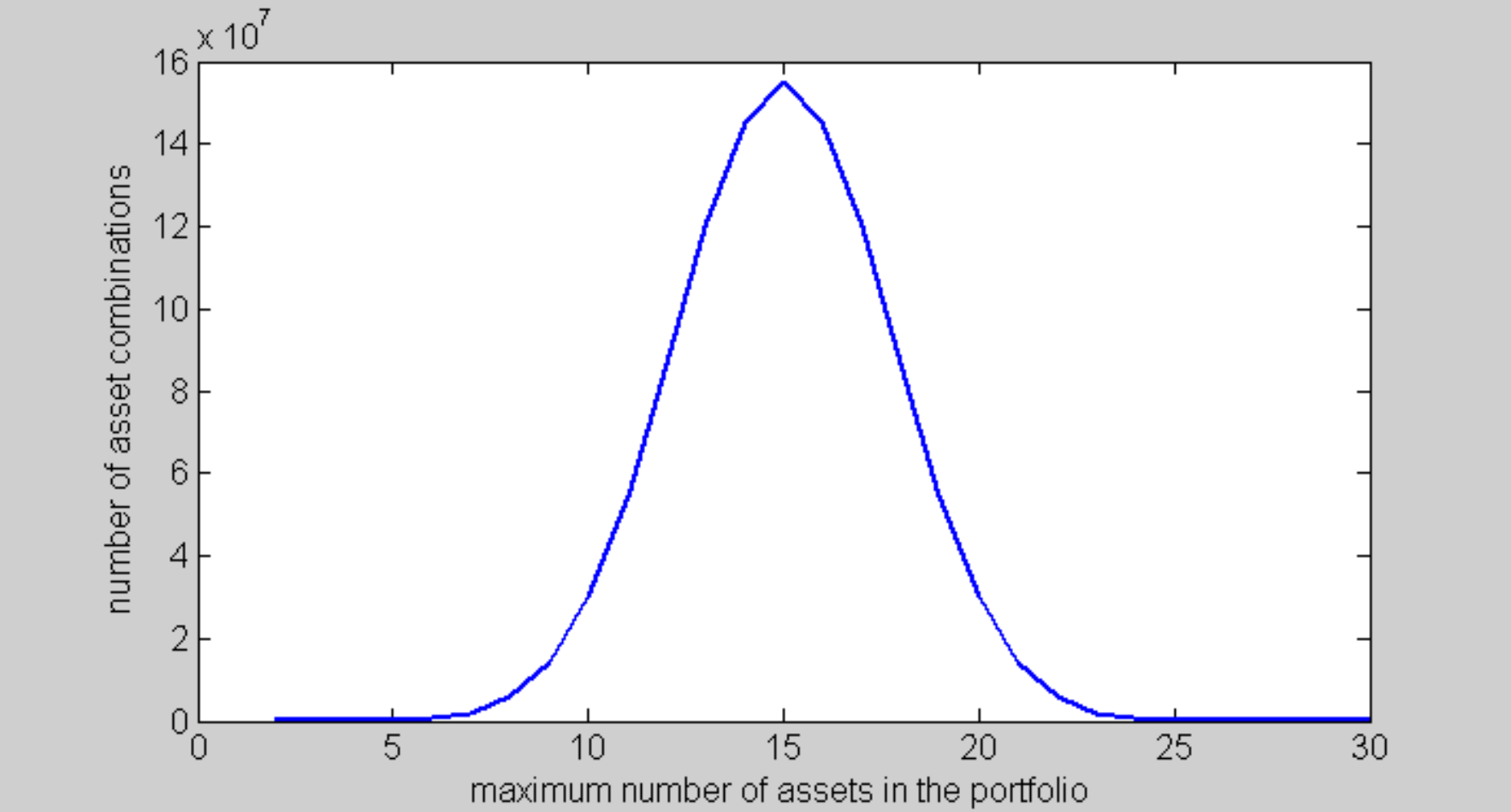
$$\sum_{i=1}^{N} w_i = 1 \quad \text{(full investment constraint)}$$

$$w_{\min} \leq w_i \leq w_{\max} \quad \text{(floor/ceiling constraint)}$$

$$\sum_{i=1}^{N} s(w_i) = K \leq N \quad \text{(cardinality constraint)}$$

$$s(w_i) = \begin{cases} 1, w_i \neq 0 \text{(asset } i \text{ in portfolio)} \\ 0, w_i = 0 \end{cases}$$

Combinatorial explosion with CC portfolios



$N = 30$

Computational issues

- CC significantly increase the computational effort
  - mixed nonlinear-integer programming
  - N=30, K=15, # combinations~=$16x10^7$!!!
  - combinatorial explosion
- CC optimisation problems pose a challenge to traditional optimisation techniques
- Need for more *advanced* and *flexible* optimisation heuristics that can efficiently handle associated complexities
  - *Simulated Annealing*
  - *Genetic Algorithms*
  - *Particle Swarm Optimisation*

Traditional vs Fuzzy Portfolio Selection

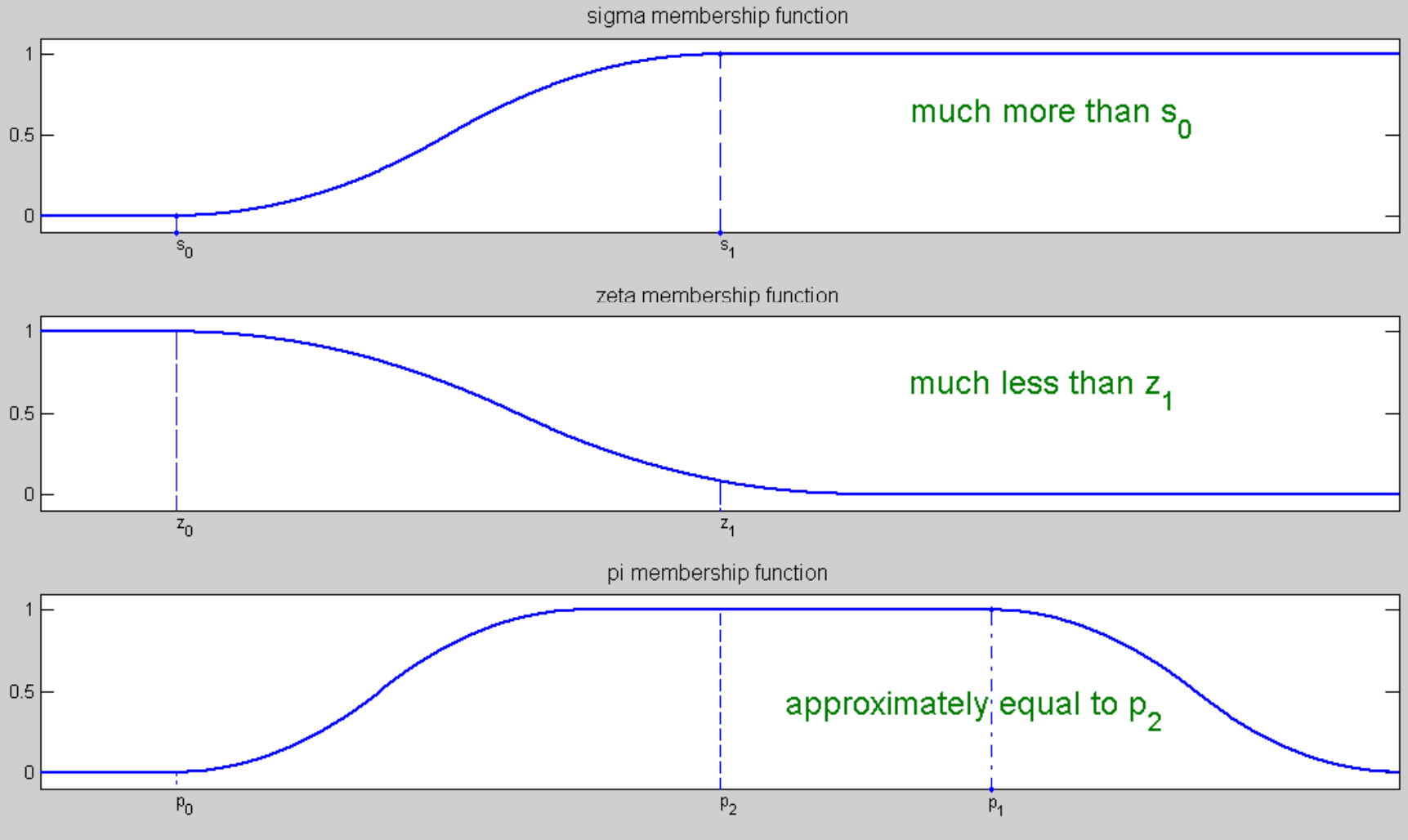
- Much of portfolio selection is about setting

  a) aspiration levels for performance criteria and

  b) constraints on risk measures

- This is often done within a "*hard*" mathematical programming setting (requires *precise* definition of objectives, preference to return and attitude towards risk, often leads to *"edge solutions"*, cannot easily reconcile multiple conflicting objectives)
- In practice, portfolios are structured around *imprecise views* of asset managers on the risk profile of trading positions
  - "Given current market conditions, we would appreciate an annual return of not *much more* than 5% in excess of the benchmark"
  - "The probability of downfall should not *significantly* exceed 20%"
- Fuzzy optimization theory offers a very convenient framework for accommodating such "vague" information

Fuzzy goals and constraints



Empirical study

- Benchmark: *Dow Jones Industrial Average*
- Portfolio: 30 member stocks of the DJIA (as of 14/11/2008)
- Data: Daily quotes of the DJIA and its constituents from 21/01/2004 to 13/01/2005 (500 observations)
- 250 were used for *estimation* and 250 for *out-of-sample evaluation*
- Portfolio Constraints: w_{min}=0.05, w_{max}=0.8 (no short selling)
- Algorithms' parameters:
  - Population size: 100, number of generations: 200
  - GA: real-encoding scheme (solutions are real vectors), p_c=0.8, p_m=0.01
  - PSO: $w_{min=}$ 1.0000e-003, w_{max}= 2, c_1= 2, c_2=2
  - SA: γ= (1e-03)^(1/200)
  - 500 independent runs
- Two more heuristics for detecting good asset combinations:
  - *MC-search* (generate 2000 random asset combinations and compute optimal weight, pick the best one)
  - *Fundamental stock-picking heuristic* (pick stocks based on an optimal combination of Size and Stock Beta)

How to deal with the cardinality constraint?

- We introduce proper transformations on the solution space
  - Mixed nonlinear-integer formulation $\Rightarrow$ unconstraint programming with continuous variables
- *Simultaneously* look for optimal combinations of assets *and* weights (see e.g. Maringer and Oyewumi (2007), Thomaidis et al. (2009)).
- Transformations generally lead to rugged optimisation landscapes
  - Many local optima, "flat" regions, discontinuities
- *Intelligent optimisation heuristics* are used to solve this problem

Fuzzy enhanced indexation

- Objective 1: Obtain *some* return in addition to the benchmark while keeping the total risk of the portfolio *approximately equal* to market's risk
- Objective 2: *Restrict* the probability of under-performing the benchmark while keeping the TESD *small*

Problem formulations

- Objective 1:

$$\max s(m_{TE};1\%,30\%)\cdot p(s_P;99\%s_B,101\%s_B)$$

- Objective 2:

$$\max z(s_{TE};1\%,5\%)\cdot z(P^{-};10\%,40\%)$$

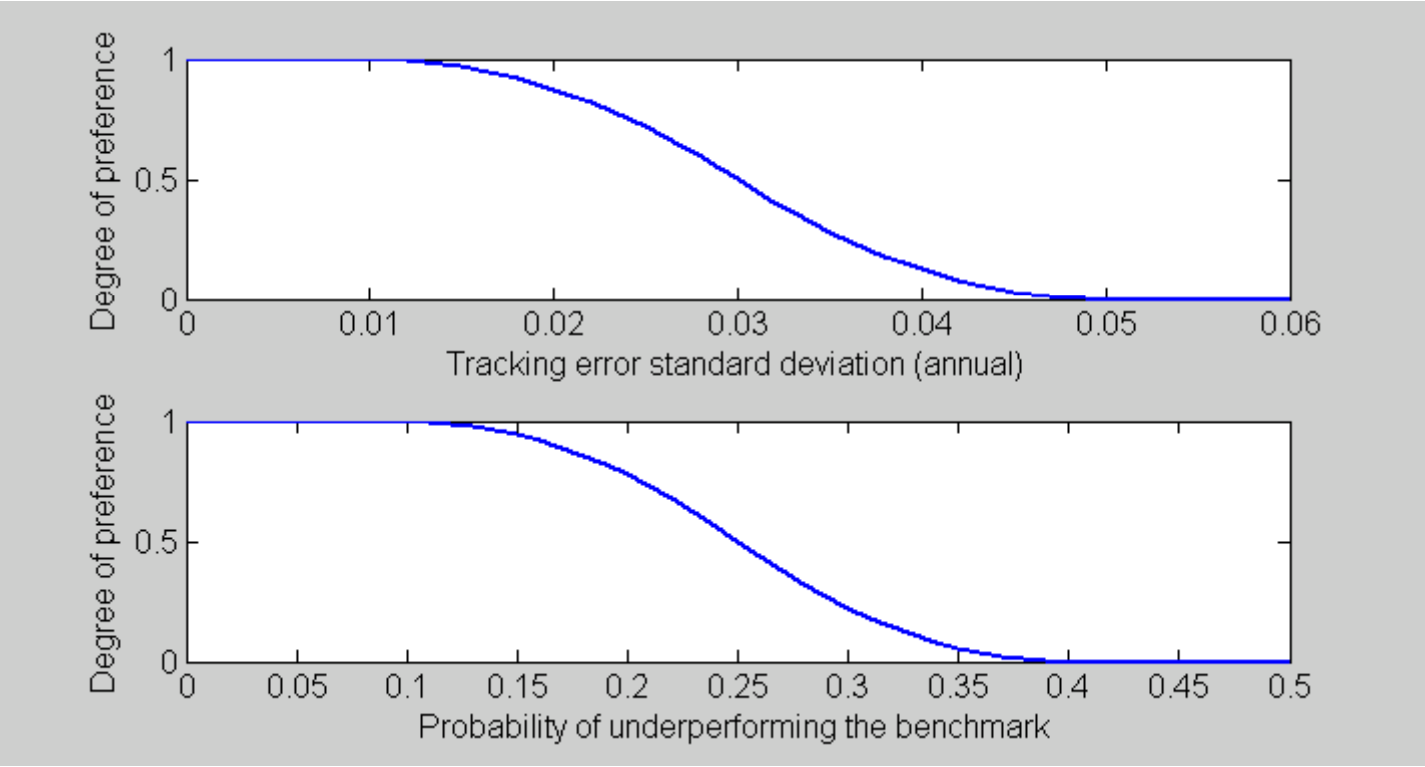
m_{TE} :Mean Tracking error (Excess return)

s_{TE} :Tracking error standard deviation

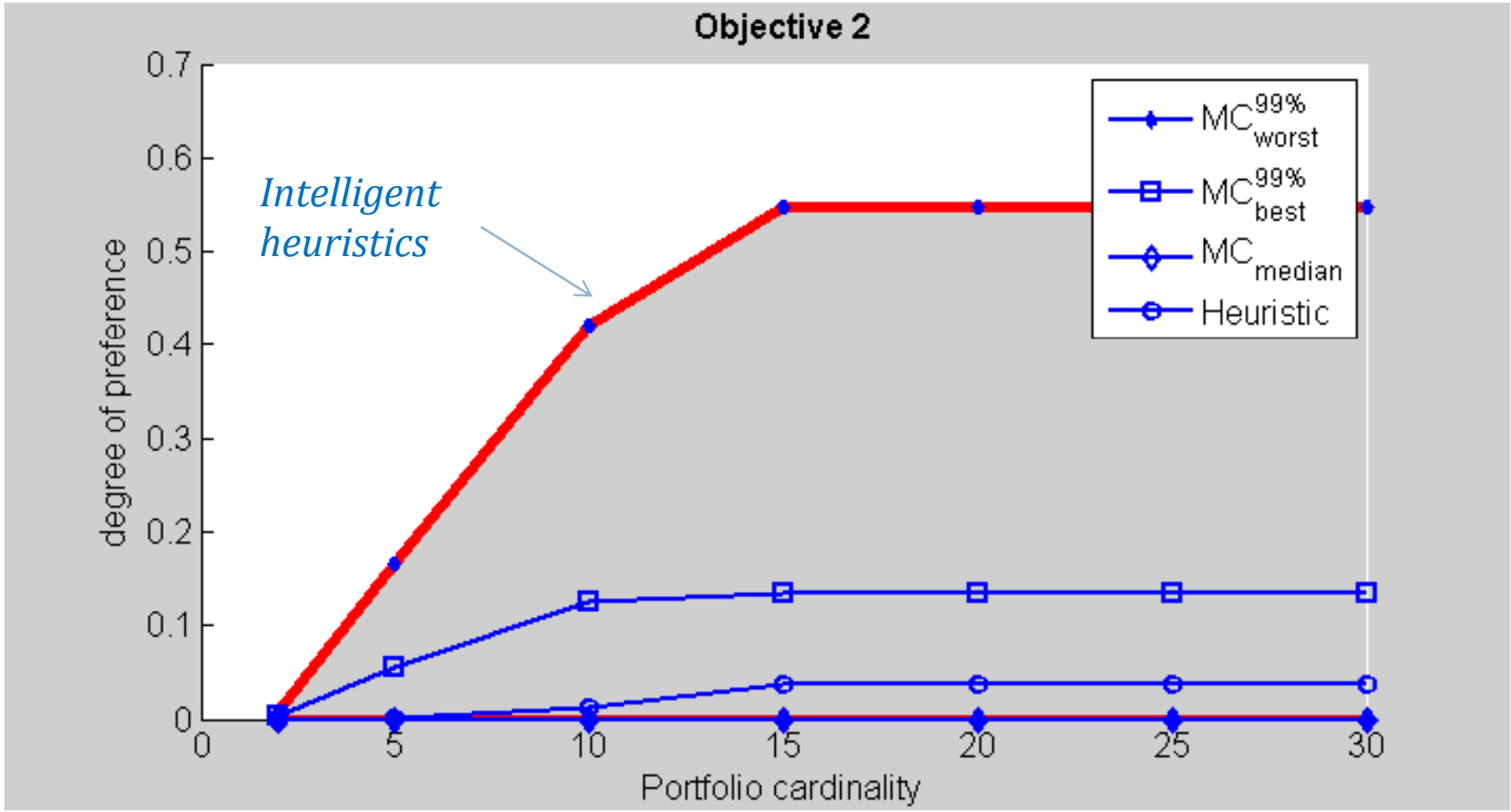
s_P, s_B :Portfolio (benchmark) risk (standard deviation)

P^{-} : Probability of underperforming the benchmark

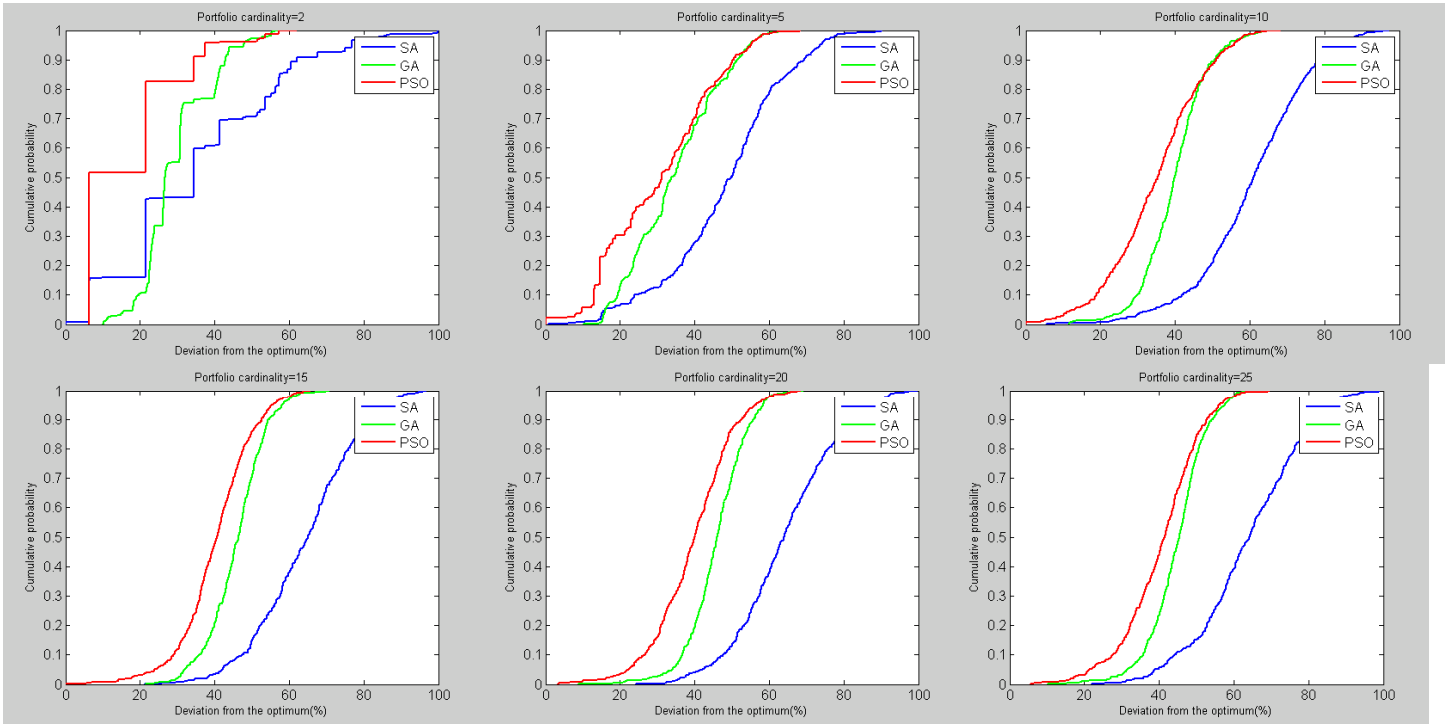
Obj2: Fuzzy goal attainment



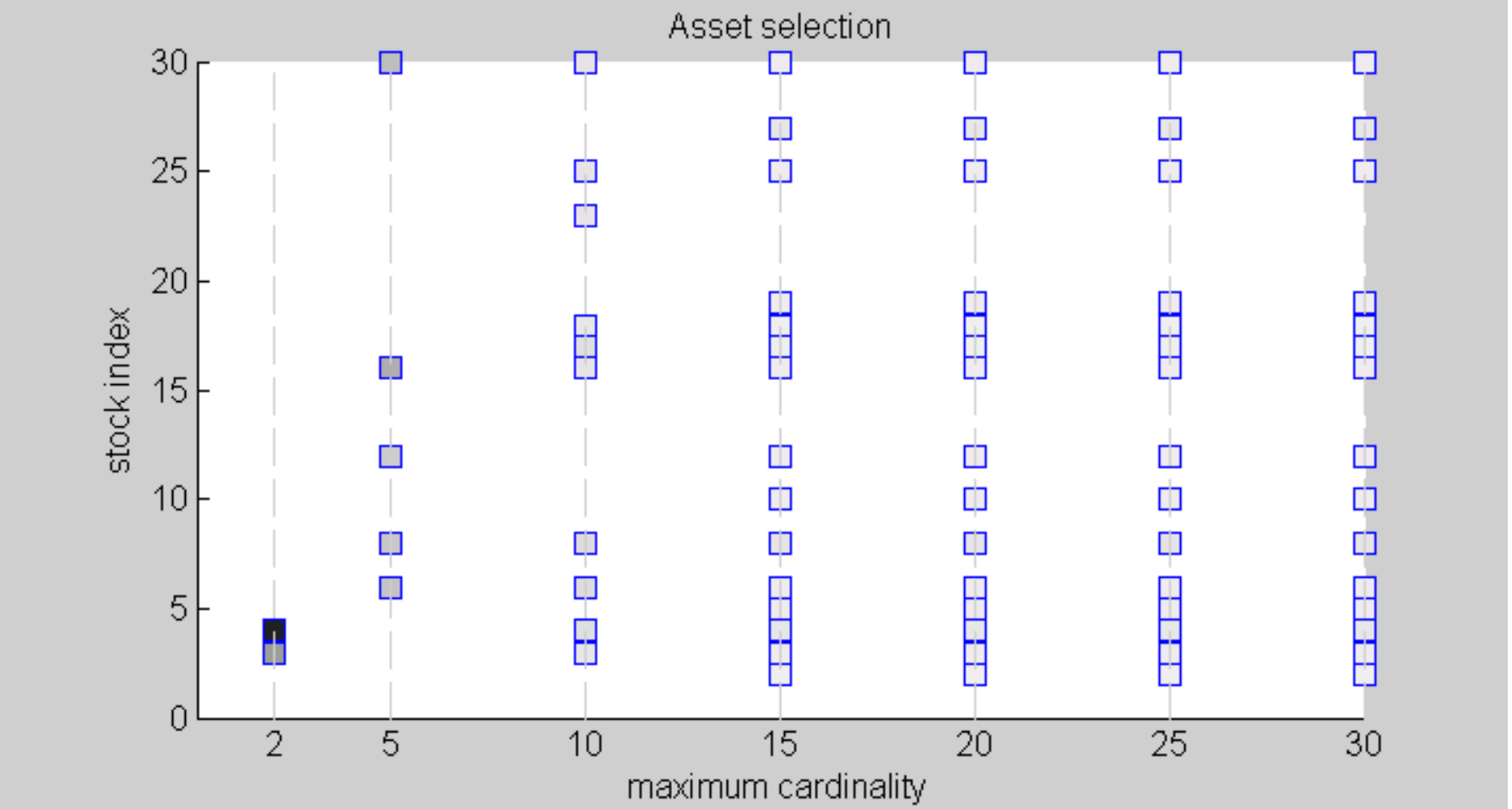
Obj2: Degree of goal attainment vs portfolio cardinality



Obj2: Stochastic convergence properties of intelligent heuristics



Obj2: Optimal capital allocation vs portfolio cardinality



Obj2: Performance of optimal portfolios

Method	Degree of satisfaction			σ_{TE}	Prob (TE<0)	Average return	Standard deviation of returns	Sharpe ratio	Cumulative return
	Overall	Obj1	Obj2						
Evolutionary strategies	39.75	71.66	48.07	4.20	30.82	13.60	11.11	94.33	14.63
	1.28	65.44	1.59	4.61	46.98	8.58	11.09	49.96	8.99
Heuristic	2.37	46.41	3.90	6.17	44.99	11.07	9.92	79.82	11.76
	0.01	55.11	0.43	5.33	49.37	2.54	9.96	-4.73	2.66
Monte Carlo	11.00	14.74	70.64	4.22	39.41	9.02	11.17	52.68	9.56
	0.27	67.37	0.35	4.42	49.03	4.52	11.2	14.65	4.67
Equally weighted	0.02	97.57	0.02	1.99	49.60	2.48	10.76	-4.82	2.51
	0.95	98.01	0.97	1.90	47.21	4.28	10.70	12.70	4.37

Discussion

- Passively holding the index portfolio is an *inefficient* trading strategy
- Enhanced indexation is feasible
  - One can benefit from careful asset selection and capital allocation
  - Optimisation increases portfolio performance in- & out-of-sample
- Intelligent heuristics $\Rightarrow$ superior means of solving CC portfolio selection problems
  - Outperform MC or simple expert rules of thumb
  - Introduce *randomness into the search process* => avoiding premature convergence and moving towards global optima
  - Stochastic elements may lead to a *large diversity of reported solutions* (especially in complex landscapes)
  - Different approach: Many independent runs are necessary before a near-optimum solution is reached with high confidence
- We provide an analysis of the *stochastic convergence properties* of three popular intelligent heuristics: simulated annealing, genetic algorithms and particle swarm optimisation

Further research

- *Alternative portfolio formulations*
  - Different definitions of reward and risk
- *Optimal parameter setting that would boost algorithmic performance*
- *Comparison with other optimisation techniques*
  - including simple rules of thumb
- *Extending our analysis to other markets*
  - Benchmark indexes with more member stocks (S&P 500, Russell 3000, etc)

Further research

- *Time-series analysis*
  - Neural network – GARCH models
  - Conditional density prediction
- *Multifactor stock pricing models*
  - Forecasts for the mean and correlation structure of stock returns
- *Statistical arbitrage*
  - Detecting mispricings between stocks
  - Cointegration techniques to detect mean-reverting synthetic portfolios
  - Time-series models to predict corrections of mispricings

NN-GARCH models

Dynamic models that *jointly* parametrise the mean *and* the variance of the conditional distribution

$$y_t = m(x_t; \delta) + \epsilon_t \tag{1a}$$

$$\epsilon_t | x_t \sim D(0, h_t(x_t; \delta)) \tag{1b}$$

- δ is a vector of parameters
- $m(x_t; \delta)$ is the *conditional mean* or *expectational* model of y_t given x_t
- $h_t(x_t; \delta)$ is the conditional variance model $\rightarrow$ also a function of the information available to the researcher up to time t
- $D(.)$ is the *distributional model* (normal or student)

Model (1) is in fact a model for the conditional density of y_t given x_t

$$y_t | x_t \sim D(m(x_t; \delta), h_t(x_t; \delta))$$

Simulated annealing

Algorithm 1 Pseudo-code for simulated annealing

1: **initialisation:** generate an initial solution $\mathbf{x}^0$ in the feasible region, set values for the maximum search range R, the temperature parameter T and the population size D.
2: **set** the current solution $\mathbf{x}^c := \mathbf{x}^0$ and compute its fitness $f(\mathbf{x}^c)$.
3: **while** stopping criteria are not met **do**
4: Randomly generate $\mathbf{x}_j$, $j = 1, ..., D$ *feasible* solutions in the neighbourhood $N(\mathbf{x}^c, R)$ of the current solution $\mathbf{x}^c$ and compute their fitness $f(\mathbf{x}_j)$, $j = 1, ..., D$.
5: Find the subset $\mathcal{J} \subset \{1, 2, ..., D\}$ of solutions satisfying either of the two conditions:
(1) $\Delta f_j := f(\mathbf{x}_j) - f(\mathbf{x}^c) < 0$ or
(2) $p_j := \exp(-\Delta f_j / T) > u_j$, where u_j is a uniformly generated random number.
6: **set** $\mathbf{x}^c := \mathbf{x}_j^*$, where $\mathbf{x}_j^*$ is randomly drawn from $\mathcal{J}$.
7: **set** $T := \gamma T$
8: **end while**

Particle Swarm Optimisation

Algorithm 3 Pseudo-code for particle swarm optimisation.

initialisation: randomly generate an initial swarm of particles (feasible solutions) $\mathbf{x}_j$, $j = 1, 2, ..., D$ and velocity vectors $\mathbf{v}_j$, $j = 1, 2, ..., D$. Set values for the parameters υ, c_1, c_2 of the dynamic equation (3) and the population size D. Let $\mathbf{x}_j^b$ be the best solution in the particle's j history, $\mathbf{x}_g$ be the globally optimum solution and $f(\mathbf{x}_j^b)$ and $f(\mathbf{x}_g)$ be the corresponding values of the objective function.
while stopping criteria are not met **do**
  set current population of particles $\mathbf{x}_j^c := \mathbf{x}_j$, $j = 1, 2, ..., D$ and calculate the fitness $f(\mathbf{x}_j^c)$ of each particle.
  for $j = 1$ **to** D **do**
    If $f(\mathbf{x}_j^c) > f(\mathbf{x}_j^b)$, **set** $\mathbf{x}_j^b := \mathbf{x}_j^c$.
    If $f(\mathbf{x}_j^c) > f(\mathbf{x}_g)$, **set** $\mathbf{x}_g := \mathbf{x}_j$.
  end for
  for $j = 1$ **to** D **do**
    repeat
      Generate random vectors $\mathbf{r}_1$, $\mathbf{r}_2$, whose elements are uniformly distributed in $[0, 1]$.
      Update the velocity $\mathbf{v}_j$ and the position $\mathbf{x}_j$ of particle j, using the dynamic equations:

$$\hat{\mathbf{v}}_j := \upsilon \mathbf{v}_j + c_1 \mathbf{r}_1 \circ (\mathbf{x}_j^b - \mathbf{x}_j) + c_2 \mathbf{r}_2 \circ (\mathbf{x}_g - \mathbf{x}_j) \quad (3a)$$

$$\hat{\mathbf{x}}_j := \mathbf{x}_j + \mathbf{v}_j \quad (3b)$$

    until a new feasible solution $\hat{\mathbf{x}}_j$ is found.
    set $\mathbf{x}_j := \hat{\mathbf{x}}_j$.
  end for
end while

Genetic algorithms

Algorithm 2 Pseudo-code for a genetic algorithm

1: **initialisation**: randomly generate an initial population of feasible solutions $\mathbf{x}_j^0$, $j = 1, 2, ..., D$. Select an encoding scheme. Assign values to e_s, the number of elite children and p_c the crossover rate and p_m the mutation rate.

2: **repeat**

3: set current population of individuals $\mathbf{x}_j^c := \mathbf{x}_j$, $j = 1, 2, ..., D$ and calculate the fitness $f(\mathbf{x}_j^c)$ of each solution.

4: Create three types of children for reproducing the next population:

5: (1) Choose the n_s best-ranking solutions of the current population (*elite children*) to be transferred to the next generation

6: (2) Randomly pick a fraction p_c of the population on which the crossover operator is applied to create new *offsprings*.

7: (3) Complete the generation by applying the *mutation* operator to the remaining set of individuals. Each gene is mutated with probability p_m.

8: Compile the new generation of solutions $\mathbf{x}_j$, $j = 1, 2, ..., D$.

9: **until** number of generations limit is reached or no improvement in the objective function for a sequence of consecutive generations is observed.