



UNIVERSITÀ DEGLI STUDI DI TRIESTE
DOTTORATO IN FINANZA AZIENDALE, XX CICLO

TESI DI DOTTORATO

**OTTIMIZZAZIONE DEL PORTAFOGLIO CON
STRATEGIE PASSIVE: UN'APPLICAZIONE
DEGLI ALGORITMI DI THRESHOLD
ACCEPTING**

Dottorando:
Dott.ssa Karla Bjelanović

Coordinatore del collegio dei docenti:
Chiar.mo Prof. Maurizio Fanni

RELATORE:
Chiar.mo Prof. Maurizio Fanni

CORRELATORE:
Chiar.mo Prof. Sergio Invernizzi

ANNO ACCADEMICO 2006–2007

Indice

Prefazione	x
1 Premesse finanziarie	1
1.1 Teoria su gestione del portafoglio	1
1.1.1 Richiami sulla gestione del portafoglio	1
1.2 Stili di gestione del portafoglio	7
1.3 Indici di borsa	11
1.3.1 Indici <i>price weighted</i>	12
1.3.2 Indici <i>value weighted</i>	13
1.3.3 Indici <i>unweighted</i>	14
2 Ottimizzazione euristica	15
2.1 Introduzione	15
2.2 Algoritmi evolutivi	18
2.2.1 <i>Ant colonies AC</i>	22
2.3 Metodi di traiettoria	22
2.3.1 <i>Simulated annealing SA</i>	22
2.3.2 <i>Tabu search TA</i>	23
2.4 <i>Threshold accepting</i>	23
2.4.1 L'algoritmo	24
2.4.2 Convergenza del <i>threshold accepting</i>	26
2.4.3 Implementazione del <i>threshold accepting</i>	32
3 Modello per l'<i>index tracking</i>	36
3.1 Il modello matematico	36
3.1.1 Portafogli e rendimenti	37
3.1.2 Confronto dei rendimenti	39
3.1.3 Formalizzazione del problema	39
3.1.4 Un esempio numerico con un indice artificiale	41
3.1.5 Complessità computazionale dello stadio combinatorio	45
3.1.6 Necessità di un'euristica	45

3.1.7	Ottimizzazione mista	46
3.2	Algoritmo	47
3.2.1	Stadio euristico: threshold accepting	47
3.2.2	Stadio classico: metodo di Newton	47
3.2.3	Definizione degli intornoi	48
3.2.4	Implementazione in R	48
3.2.5	Output	53
3.2.6	Ottimizzazione mista: efficacia	56
4	Risultati e conclusioni	58
4.1	Risultati	58
4.2	Conclusioni	74
A	<i>Tracking</i> dell'indice Hang Seng	76
B	<i>Tracking</i> dell'indice S&P 100	86
C	<i>Tracking</i> dell'indice MIB 30	98

Elenco delle figure

1.1	Frontiera efficiente per il portafoglio con due e quattro titoli . . .	3
1.2	Frontiera efficiente e la scelta del portafoglio in base al grado di avversione al rischio	6
3.1	Grafici di $\mathcal{E}(x)$	43
3.2	Il triangolo di Maxwell	44
3.3	Esempio output grafico	55
3.4	Ottimizzazione euristica e ottimizzazione classica locale	57
3.5	Composizione dei portafogli con ottimizzazione euristica e classica	57
4.1	Esempio dell'output di una esecuzione dell'algoritmo	60
4.2	Conferma della diversificazione del portafoglio con pochi titoli	61
4.3	Matrice a barre HSI	65
4.4	Istogramma peso medio	66
4.5	HSBC <i>vs.</i> Hang Seng	67
4.6	Esempio dell'output di una esecuzione dell'algoritmo	69
4.7	Matrice a barre MIB 30	70
4.8	Istogramma peso medio per l'indice MIB 30	71
4.9	Esempio dell'output di una esecuzione dell'algoritmo	72
4.10	Matrice a barre S&P 100	74
4.11	Istogramma peso medio per l'indice S&P 100	75
A.1	Portafoglio con 2 titoli riproducente HSI	77
A.2	Portafoglio con 3 titoli riproducente HSI	78
A.3	Portafoglio con 4 titoli riproducente HSI	79
A.4	Portafoglio con 5 titoli riproducente HSI	80
A.5	Portafoglio con 6 titoli riproducente HSI	81
A.6	Portafoglio con 7 titoli riproducente HSI	82
A.7	Portafoglio con 8 titoli riproducente HSI	83
A.8	Portafoglio con 9 titoli riproducente HSI	84
A.9	Portafoglio con 10 titoli riproducente HSI	85

B.1	Portafoglio con 10 titoli riprodotte S&P 100	87
B.2	Portafoglio con 11 titoli riprodotte S&P 100	88
B.3	Portafoglio con 12 titoli riprodotte S&P 100	89
B.4	Portafoglio con 13 titoli riprodotte S&P 100	90
B.5	Portafoglio con 14 titoli riprodotte S&P 100	91
B.6	Portafoglio con 15 titoli riprodotte S&P 100	92
B.7	Portafoglio con 16 titoli riprodotte S&P 100	93
B.8	Portafoglio con 17 titoli riprodotte S&P 100	94
B.9	Portafoglio con 18 titoli riprodotte S&P 100	95
B.10	Portafoglio con 19 titoli riprodotte S&P 100	96
B.11	Portafoglio con 20 titoli riprodotte S&P 100	97
C.1	Portafoglio con 2 titoli riprodotte MIB 30	99
C.2	Portafoglio con 3 titoli riprodotte MIB 30	100
C.3	Portafoglio con 4 titoli riprodotte MIB 30	101
C.4	Portafoglio con 5 titoli riprodotte MIB 30	102
C.5	Portafoglio con 6 titoli riprodotte MIB 30	103
C.6	Portafoglio con 7 titoli riprodotte MIB 30	104
C.7	Portafoglio con 8 titoli riprodotte MIB 30	105
C.8	Portafoglio con 9 titoli riprodotte MIB 30	106
C.9	Portafoglio con 10 titoli riprodotte MIB 30	107

Prefazione

1. Premessa

La gestione dei fondi consiste nell'investimento di milioni di dollari (o il loro equivalente) in azioni (o quote) di società quotate nelle varie borse mondiali. La fonte di capitale per la gestione dei fondi è normalmente rappresentata dai risparmi, dai versamenti per la previdenza sociale e dai premi assicurativi. L'obiettivo principale dei gestori è quello di generare la crescita del capitale a loro affidato nel medio-lungo periodo. Le strategie di gestione possono sostanzialmente essere identificate in due categorie principali: strategie attive e strategie passive. La gestione attiva è caratterizzata da una maggiore flessibilità del gestore nella scelta delle operazioni del portafoglio (acquisti o vendite) con il fine di scegliere i titoli che durante un arco temporale definito avranno la capacità di generare prestazioni migliori rispetto ad altri titoli. Questa strategia si basa sull'assunzione che i gestori attraverso la loro esperienza e capacità di giudizio riescano a generare valore aggiunto nell'attività di gestione individuando i titoli a più alto rendimento e operando le operazioni di compravendita nei momenti più opportuni (vendite ai massimi, acquisti ai minimi). Una gestione passiva, al contrario, presenta molti più vincoli per il gestore. Le strategie d'investimento passivo prevedono infatti una gestione che ha come obiettivo principale quello di replicare la *performance* del benchmark, senza alcun vincolo di extrarendimento.

Partendo dall'assunto che nel lungo periodo i mercati finanziari tendono ad accrescere il valore degli investimenti, molti ritengono che sia più efficiente non esporsi a rischi tipici alla gestione attiva, normalmente caratterizzata da risultati di *overperformance* rispetto agli indici di riferimento nel breve periodo, ma non sempre vincente durante archi temporali più lunghi (Beasley et al., 2003 [10]). Un ulteriore vantaggio offerto dalla gestione passiva è costituito dal contenimento dei costi, che possono essere anche di molto inferiori rispetto a quelli generati dalla gestione attiva. Di conseguenza sempre più fondi di investimento prevedono di affidarsi ad una strategia di investimento fondata sulla gestione passiva, basando le proprie scelte d'investimento non più su decisioni

derivanti dal giudizio soggettivo umano guidato dalla conoscenza ed esperienza tipica del *fund manager* bensì su modelli matematici capaci di riprodurre e “prevedere” l’andamento degli indici presi a riferimento. Ad esempio (fonte: E.H. Sorenson, K.L. Miller, V. Samak, Allocating between active and passive management, *Financial Analysts Journal* 54 (5) (1998) 183) si stima che nei soli Stati Uniti più di 10^{12} \$ siano investiti in gestioni passive.

La gestione passiva può essere realizzata detenendo in portafoglio tutte le attività finanziarie appartenenti ad un indice di mercato nella stessa proporzione (effettuando dunque movimenti solo quando vengono aggiunti o eliminati titoli dall’indice), oppure, in alternativa, tentando di replicare l’indice stesso con un numero inferiore di titoli utilizzando particolari tecniche di selezione del portafoglio per scegliere i titoli più appropriati. Questa strategia alternativa è comunemente denominata *index tracking*, ed è il tema centrale di questa tesi di dottorato.

Va subito detto che l’attività di *index tracking*, ossia la riproduzione dell’indice, può essere tradotto in un problema di ottimizzazione vincolata. Lo spazio \mathcal{X} degli stati ammissibili (*feasibility space*) è costituito da tutti i possibili portafogli dell’investitore, la funzione obiettivo da minimizzare è il *tracking error*, ossia il differenziale di rendimento tra il portafoglio costruito e l’indice di riferimento, i vincoli sono ad esempio la cardinalità k del portafoglio (con k minore degli n titoli che compongono l’indice di riferimento), i costi di transazione, ecc. Un classico risultato combinatorio ci informa che la cardinalità dello spazio \mathcal{X} dei portafogli ammissibili è $\binom{n}{k} = 2^{n(h(p)+o(1))}$ se $k \sim pn$, dove H è l’entropia $H(p) = -p \log_2 p - (1-p) \log_2 (1-p)$. Questa crescita che, anche per valori bassi di p e quindi di k , è esponenziale, rende il problema computazionalmente complesso.¹ L’oggetto di tale problematica è argomento di studio a livello internazionale da parte di un gruppo di ricercatori, tra cui Beasley, J. E., Meade N., Chang T. J., E. Küllezi, Gilli, M. e P. Winker. La tesi si inserisce in questa linea di ricerca.

2. Risultati

Per una comprensione più immediata del problema in questione, in questa tesi si è scelto di ridurre al minimo le variabili, escludendo in tal modo elementi secondari quali ad esempio i costi di transazione ed i lotti minimi. La struttura del lavoro svolto tuttavia lascia aperta la possibilità, in un secondo tempo, di considerare nel calcolo anche tali variabili.

Considerata la complessità del problema in oggetto, la scelta di un algoritmo euristico come strumento di risoluzione è risultata obbligata. I più efficaci

¹ Ad esempio per $n = 100, k = 5$, si ha $\#\mathcal{X} = 7.52875 \times 10^7$, e per $n = 100, k = 10$ si ha $\#\mathcal{X} = 1.73103 \times 10^{13}$.

tra i diversi tipi di algoritmi euristici sono risultati essere gli algoritmi genetici (Beasley et al., 2003 [10]) e l'algoritmo di *threshold accepting* (Gilli, M. e E. K ellezi 2001 [45]). Di conseguenza, seguendo in particolare Gilli, si   deciso di adottare, per la risoluzione numerica del problema di ottimizzazione il *threshold accepting* (TA). La definizione delle linee guida di questo algoritmo sono state ricalcate da quelle elaborate da Gilli e Winker (2007, [43]), con l'introduzione di alcune varianti che si sono dimostrate particolarmente significative.

L'algoritmo euristico, infatti,   stato raffinato aggiungendo una parte finale che consiste nell'applicazione di un metodo classico di ottimizzazione locale. Questa operazione ci ha consentito di raggiungere il primo risultato principale della tesi. In pratica, l'ottimizzazione locale ha consentito di individuare il fenomeno dei minimi che si trovano sul bordo dell'insieme ammissibile. La conclusione che si evince   la seguente: nella costruzione di un portafoglio ridotto di titoli che ha il fine di replicare un determinato indice non sempre risulta efficace l'inclusione di un insieme allargato di elementi; al contrario si   verificato che l'aggiunta di un titolo poco significativo pu  peggiorne l'azione di *tracking* dello stesso indice. Viene confermata, quindi, matematicamente la supposizione di alcuni analisti come Elton, E. J. e Gruber M. J., Brown, S. J. e Goetzmann, W. N. (2003 [35]) che la diversificazione del portafoglio pu  essere raggiunta con un numero piuttosto basso dei titoli se questi sono ben scelti. In altre parole, la diversificazione del portafoglio non dipende dalla quantit  dei titoli al suo interno, ma dalla loro "qualit ".

Il secondo risultato ottenuto   rappresentato dalla capacit  dell'algoritmo nel determinare se all'interno di un indice oggetto vi siano delle componenti dominanti: una componente di un indice   *dominante* se, intuitivamente, si trova sempre presente nei portafogli di cardinalit  bassa che riproducono l'indice. In tal caso per giungere alla loro individuazione   sufficiente verificare la loro presenza ed il loro peso nei vari portafogli prodotti dall'algoritmo. Ad esempio il fatto che l'indice azionario Hang Seng (Borsa di Hong Kong), sia dominato da alcune sue componenti (HSBC e China Mobile)   ben noto agli analisti,² che riconducono al comportamento dinamico di questi titoli la maggior parte dei movimenti dell'indice.

Non sono state rilevate componenti dominanti nell'indice S&P 100, mentre le analisi sembrano indicare che ce ne siano nel MIB 30, meno significative rispetto al caso di Hang Seng.

²A Hong Kong, la caduta del 6 febbraio 2008   stata causata dal crollo dei titoli *heavy-weights*: HSBC (ticker Bloomberg 5 HK, il suo peso all'interno dell'indice quel giorno era 15.27%), China Mobile (ticker Bloomberg 941 HK con il peso 13,22%) e China Life Insurance Co. (ticker Bloomberg 2628 HK e il peso 4,12%). HSBC ha registrato una flessione del 4.1%, China Life   sceso del 7.8%, mentre China Mobile ha perso il 5.1%. (V. Phani Kumar, MarketWatch)

Durante la stesura della tesi, discutendo con alcuni gestori di Allianz S.p.A., si è evidenziato che questo algoritmo potrebbe avere anche delle applicazioni utili nella gestione attiva. Nelle fasi di mercato particolarmente volatile, la maggior parte dei gestori preferisce assumere una posizione più in linea con mercato evitando, quindi, esposizioni concentrate su particolari titoli con il rischio di subire maggiori perdite. Con questo tipo di strumento a disposizione, vi sarebbe la possibilità di implementare tale strategia difensiva allocando gran parte del capitale in un numero ridotto di titoli altamente correlati con il mercato, senza necessariamente coprire tutte le posizioni del benchmark.

3. Implementazione

Il *threshold accepting* è un algoritmo euristico di ottimizzazione di recente introduzione, ed è sostanzialmente una versione modificata e semplificata dell'algoritmo di *simulated annealing*. Come tutti gli euristici, anche TA parte da una soluzione ammissibile e cerca di migliorarla esplorandone un opportuno intorno. L'algoritmo termina quando non è più possibile trovare delle soluzioni migliori nell'intorno della soluzione corrente, ovvero dopo un prefissato tempo di calcolo o un prefissato numero di operazioni. Per sfuggire alle trappole dei minimi locali, durante l'esecuzione dell'algoritmo, TA "accetta" anche i movimenti in direzione opposta a quella della minimizzazione (*up-hill moves*) purché non superino una certa "soglia" (da cui il nome *threshold accepting*).

Come linguaggio di programmazione è stato scelto il pacchetto statistico R (www.r-project.org). Il pacchetto statistico R sta diventando lo standard per la analisi dei dati in ambito accademico (e non solo). La sua caratteristica di essere liberamente disponibile ed open-source fa sì che i risultati di un lavoro di ricerca sviluppato in R siano *verificabili* da chiunque intenda effettuare un controllo per dare una valutazione del lavoro. A svantaggio, va riportato che di norma R richiede più tempo di cpu rispetto a linguaggi come C++, ma, a posteriori, questo svantaggio è minimo in quanto come si vedrà i tempi tipici di esecuzione sono dell'ordine dell'ora (la workstation usata da J.E. Beasley et al. [10] aveva tempi simili). In sostanza si è privilegiata la trasparenza e la comprensibilità del codice rispetto alla sua efficienza (che è un obiettivo che potrà essere raggiunto in seguito).

L'implementazione effettiva ha utilizzato tre indici di riferimento. Il primo indice scelto è S&P 100 (meno volatile, ³ composto da $n = 100$ titoli, ticker Bloomberg OEX Index). Il secondo indice è Hang Seng (più volatile, ⁴ composto da $n = 40$ titoli, ticker Bloomberg HSI Index). Il terzo indice scelto è MIB

³La deviazione standard di S&P 100 annualizzata su serie storica delle variazioni giornaliere dei prezzi su 5 anni è 8,82%, calcolata all'8 febbraio 2008.

⁴La deviazione standard di Hang Seng annualizzata su serie storica delle variazioni giornaliere dei prezzi su 5 anni è 17,93%, calcolata all'8 febbraio 2008.

30 (media volatilità, ⁵ composto da $n = 30$ titoli, ticker Bloomberg MIB30 Index). I dati utilizzati sono i prezzi settimanali medi del 2006 e 2007 come scaricati da Bloomberg. Nell'implementazione, usando i dati di tutto il 2006 (*in-sample*), viene selezionato, utilizzando il codice in R, un portafoglio con un ridotto numero k di componenti (tipicamente $5 \leq k \leq 10$) appunto per ridurre i costi di transazione. Si immagina poi di investire un capitale fissato convenzionalmente pari a 1000 in data 01/01/2007. Poi, con i dati del 2007, viene confrontato l'andamento di questo portafoglio rispetto all'indice (*out-sample*). I risultati appaiono molto confortanti in quanto, secondo la simulazione, il ribilanciamento del portafoglio selezionato andrebbe di norma effettuato dopo il primo trimestre 2007, ma basterebbe per questo far rigirare il programma. Il codice è stato per scrupolo provato su tre tipi diversi di sistemi operativi (Mac OSX, Windows XP, Solaris 10) che supportano R. L'output è di tipo testuale (viene fornita la composizione del portafoglio "ottimo"), ma anche grafico.

⁵La deviazione standard di MIB 30 annualizzata su serie storica delle variazioni giornaliere dei prezzi su 5 anni è 12,32%, calcolata all'8 febbraio 2008.

RINGRAZIAMENTI

Si ringraziano:

il Prof. Maurizio Fanni perché si è reso disponibile a seguirmi in questo lavoro di ricerca facendomi da Relatore, mi ha consentito piena libertà nella scelta dell'argomento e mi ha sostenuto;

il Prof. Sergio Invernizzi, che si è dimostrato sempre presente e pronto ad incoraggiarmi. In particolare lo ringrazio per avermi fornito delle idee ogni volta originali per l'avvio di nuovi lavori di ricerca. Lo ringrazio per tutto il tempo che ha spontaneamente dedicato al mio progetto rendendo in tal modo possibile il raggiungimento di ottimi risultati;

l'Ing. Simone Salerni per la sua comprensione, per i consigli pratici che mi ha dato e che si sono rivelati di importante spessore scientifico. Ho avuto da lui supporto in qualsiasi momento;

la mia famiglia, mia mamma e mio papà, che hanno creduto in me e hanno fatto dei sacrifici fin dall'inizio della mia carriera scolastica. Grazie, questo risultato di oggi è il coronamento di un nostro sogno;

Giulio e l'intera famiglia Benvenuti, la mia famiglia italiana, per la comprensione, l'aiuto e l'incoraggiamento che da loro ricevo dal primo giorno che ci siamo incontrati.

Grazie a tutti, per essere stati sempre presenti in questo mio percorso di vita.

Capitolo 1

Premesse finanziarie

1.1 Teoria su gestione del portafoglio

1.1.1 Richiami sulla gestione del portafoglio - teoria di Markowitz

Proprietà comune delle opportunità d'investimento è che il loro rendimento reale possa essere diverso da quello atteso, o in altre parole che sono rischiose. Questa nozione di rischio finanziario, definito come potenziale deviazione da un rendimento atteso, include non solo le deviazioni negative del rendimento atteso (*downside risk*), ma anche le deviazioni positive (*upside risk*).

Quando tutte le informazioni a disposizione e tutte le aspettative sui prezzi futuri dei titoli sono incluse nei prezzi correnti degli stessi, allora i prezzi futuri dei titoli e i loro rendimenti possono essere trattati come variabili casuali. Nel caso più semplice, il rendimento del titolo i può essere descritto attraverso una distribuzione normale, quindi tramite rendimento atteso $E(r_i)$ e la sua varianza σ_i^2 .

Confrontando le varie opportunità d'investimento e cercando di metterle assieme per costruire dei portafogli d'investimento, un aspetto importante da considerare è la loro correlazione. In altre parole, ci si chiede se le deviazioni positive di un titolo seguano le deviazioni positive di un'altro titolo oppure se queste deviazioni siano tra di loro indipendenti. Se due titoli non sono positivamente correlati tra di loro, ci possono essere dei momenti in cui le deviazioni di un titolo saranno sopra le aspettative, mentre le deviazioni del secondo titolo saranno sotto le aspettative. Questo fenomeno quindi, tende a ridurre, in parte, il rischio. Quando i singoli titoli possono essere descritti tramite la distribuzione dei loro rendimenti, il rendimento atteso del portafoglio \mathcal{P} (costruito da N titoli) è quindi espresso come la combinazione lineare di rendimenti attesi

dei singoli titoli con i pesi rappresentati dalle quote del capitale investite in essi (al tempo 0):

$$E(r_{\mathcal{P}}) = \sum_{i=1}^N x_i E(r_i) \quad (1.1)$$

e la varianza

$$\sigma_{\mathcal{P}} = \sum_{i=1}^N \sum_{j=1}^N x_i x_j \sigma_{ij} \quad (1.2)$$

dove x_i rappresenta la quota del capitale del titolo i -esimo investito nel portafoglio, quindi $\sum_{i \in \mathcal{P}} x_i = 1$. σ_{ij} rappresenta le covarianze tra i rendimenti dei titoli i e j con $\sigma_{ii} = \sigma_i^2$ e $\sigma_{ij} = \sigma_{ji} = \sigma_i \cdot \sigma_j \cdot \rho_{ij}$ con $\rho_{ij} \in [-1, 1]$ il coefficiente di correlazione.

Supponendo che tutti i titoli nel portafoglio abbiano lo stesso peso, cioè $x_i = x = \frac{1}{N}$, le equazioni (1.1) e (1.2) possono essere riscritte:

$$E(r_{\mathcal{P}}) = \sum_{i=1}^N \frac{1}{N} E(r_i) = \bar{r}_i \quad (1.3)$$

e

$$\sigma_{\mathcal{P}} = \sum_{i=1}^N \sum_{j=1}^N \frac{1}{N} \frac{1}{N} \sigma_{ij} \quad (1.4)$$

$$\sigma_{\mathcal{P}} = \underbrace{\frac{1}{N} \frac{1}{N} \sum_{i=1}^N \sigma_i^2}_{=\bar{\sigma}_i^2} + \frac{N-1}{N} \cdot \underbrace{\frac{1}{N(N-1)} \cdot \sum_{i=1}^N \sum_{i=1, j \neq i}^N \sigma_{ij}}_{\bar{\sigma}_{ij}}$$

$$\sigma_{\mathcal{P}} = \bar{\sigma}_{ij} + \frac{1}{N} \cdot (\bar{\sigma}_i^2 - \bar{\sigma}_{ij})$$

dove \bar{r}_i rappresenta il rendimento medio atteso dei N titoli, σ_i^2 rappresenta la varianza media dei N titoli e $\bar{\sigma}_{ij}$ rappresenta la covarianza media di $N \cdot (N-1)$ titoli $i \neq j$. Più grande è il numero di titoli N , più vicino sarà il valore di varianza del portafoglio alla covarianza media, mentre il rendimento atteso rimarrà invariato. L'effetto di diversificazione sarà maggiore quando la correlazione tra i rendimenti di titoli è bassa. Nel caso di portafoglio di titoli con pesi equivalenti, l'espressione $1 - \frac{\bar{\sigma}_{ij}}{\bar{\sigma}_i^2}$ rappresenta la quantità di rischio che può essere eliminata attraverso la costruzione di un portafoglio di più titoli al posto di un unico titolo.

L'effetto di diversificazione può essere ulteriormente incrementato attraverso l'ottimizzazione dei pesi x_i dei singoli titoli. Consideriamo un caso semplice. Supponiamo di dover costruire un portafoglio di soli due titoli, con la quota x_1 investita nel titolo 1 e $x_2 = 1 - x_1$ investita nel titolo 2. Il rendimento atteso e la varianza per il portafoglio \mathcal{P} costruito con questi due titoli saranno:

$$E(r_{\mathcal{P}}) = x_1 E(r_1) + (1 - x_1) E(r_2) \tag{1.5}$$

e

$$\sigma_{\mathcal{P}}^2 = x_1^2 \cdot \sigma_1^2 + (1 - x_1)^2 \cdot \sigma_2^2 + 2 \cdot x_1 \cdot (1 - x_1) \cdot \sigma_{12} \tag{1.6}$$

$$\sigma_{\mathcal{P}}^2 = x_1^2 \cdot (\sigma_1^2 + \sigma_2^2 - 2 \cdot \sigma_{12}) - 2 \cdot x_1 \cdot (\sigma_2^2 - \sigma_{12} + \sigma_2^2) \tag{1.7}$$

Risolvendo l'equazione (1.5) per x_1 abbiamo che:

$$x_1 = \frac{E(r_{\mathcal{P}}) - E(r_2)}{E(r_1) - E(r_2)} \tag{1.8}$$

Sostituendo (1.8) nella (1.7) si ottiene che la varianza del portafoglio è raggiunta da un'iperbole. Si veda il grafico di sinistra della Figura 1.1.

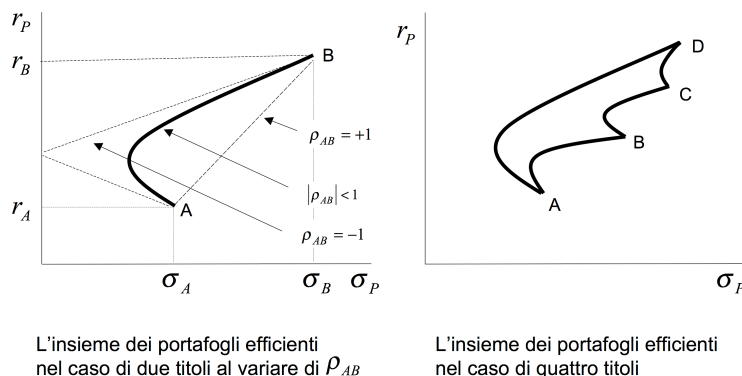


Figura 1.1: Insieme di portafogli efficienti al variare di ρ_{AB} per il portafoglio di due titoli e la frontiera efficiente per il portafoglio di quattro titoli

Aggiungendo ulteriori titoli dentro il portafoglio esiste la possibilità di incrementare la diversificazione. In questo caso l'iperbole (l'insieme dei portafogli efficienti) verrà traslata sempre di più verso la sinistra. Tuttavia, non è solo la quantità dei titoli nel portafoglio a determinare il grado di diversificazione, bensì anche la loro correlazione; citando Markowitz (1952, pagina 89 [66]): “Not only does [portfolio analysis] imply diversification, it implies the ‘right kind’ of diversification for the ‘right reason’ ”. Questo effetto è evidenziato nel

grafico di sinistra della Figura 1.1: minore la correlazioni tra i titoli più accentuata sarà la curvatura della funzione rappresentante l'insieme dei portafogli efficienti. Quando il portafoglio è costruito da più di due titoli, l'insieme dei portafogli efficienti è rappresentato dalla superficie: qualsiasi punto nel grafico a destra della Figura 1.1 rappresenta un portafoglio efficiente per un dato valore di rendimento e rischio e può essere individuato calcolando i pesi x_i .

Harry M. Markowitz è stato il primo ad impostare un modello di ottimizzazione capace di calcolare i pesi x_i dei titoli da allocare in modo tale da minimizzare il rischio per un determinato rendimento. Ha introdotto il criterio di media - varianza, per cui la composizione del portafoglio ottimale (a dato rischio si ottiene il massimo rendimento atteso ovvero la minimizzazione del rischio per ogni livello di rendimento atteso) dipende dal valore atteso di rendimento e della matrice di varianza-covarianza del vettore dei rendimenti attesi, quantità ignote approssimate con i relativi stimatori campionari. Lo studio di Markowitz si basa sull'analisi del processo che genera la domanda e l'offerta di attività finanziarie in funzione del rapporto rischio/rendimento da esse espresso. Il principio base che governa la teoria di Markowitz, è che, al fine di costruire un portafoglio efficiente occorre individuare una combinazione di titoli tale da minimizzare il rischio e massimizzare il rendimento complessivo, compensando gli andamenti asincroni dei singoli titoli. Questo modello è la base della teoria moderna del portafoglio finanziario.

Harry Markowitz considera tutti gli investitori razionali che hanno l'obiettivo di massimizzare l'utilità attesa della loro ricchezza finale $E(\mathcal{U}(w_T))$ al tempo T . Gli investitori sono *price takers* e la decisione dell'investimento viene presa al tempo 0. Nel caso un investitore sia avverso al rischio e non saziato (vuole massimizzare il profitto) la sua funzione d'utilità attesa soddisferà le seguenti proprietà:

$$\frac{\partial \mathcal{U}}{\partial w_T} > 0 \quad (1.9)$$

e

$$\frac{\partial^2 \mathcal{U}}{\partial w_T^2} < 0 \quad (1.10)$$

La ricchezza finale verrà data da $w_T = w_0 \cdot (1 + r_{\mathcal{P}})$, dove w_0 rappresenta la ricchezza iniziale (nota) e $r_{\mathcal{P}}$ è il rendimento del investimento con orizzonte temporale monopériodale $[0, T]$. Se l'utilità dell'investitore è rappresentata da una funzione quadratica del tipo:

$$\mathcal{U}(w_T) = \beta \cdot w_T - \gamma \cdot w_T^2$$

$$\mathcal{U}(w_T) = \beta \cdot w_0 \cdot (1 + r_{\mathcal{P}}) - \gamma \cdot w_0^2 \cdot (1 + r_{\mathcal{P}})^2$$

$$\mathcal{U}(w_T) = \underbrace{(\beta \cdot w_0 - \gamma w_0^2)}_{=a} + \underbrace{(\beta \cdot w_0 - 2 \cdot \gamma w_0^2) \cdot r_{\mathcal{P}}}_{=b} - \underbrace{(\gamma \cdot w_0^2)}_{=c} \cdot r_{\mathcal{P}}^2$$

dove l'unico elemento di rischio è $r_{\mathcal{P}}$, allora l'utilità attesa diventa:

$$E(\mathcal{U}) = E(a + b \cdot r_{\mathcal{P}} - c \cdot r_{\mathcal{P}}^2)$$

$$E(\mathcal{U}) = a + b \cdot E(r_{\mathcal{P}}) - c \cdot E(r_{\mathcal{P}}^2)$$

Sapendo che $E(r_{\mathcal{P}}^2) = \sigma_{\mathcal{P}}^2 + E(r_{\mathcal{P}})^2$, l'utilità attesa può essere scritta nella maniera seguente:

$$E(\mathcal{U}(w_T)) = a + E(r_{\mathcal{P}}) \cdot (b + c \cdot E(r_{\mathcal{P}})) - c \cdot \sigma_{\mathcal{P}}^2 \quad (1.11)$$

Questo implica che l'utilità attesa della ricchezza finale dipende solo da due momenti che caratterizzano il portafoglio (media e varianza). In altre parole, scegliendo un portafoglio con il criterio dell'utilità attesa si individua di nuovo il portafoglio efficiente, di conseguenza esiste compatibilità fra criterio dell'utilità attesa ed criterio di media-varianza.

Il modello di Markowitz si basa sulle condizioni tipiche dei mercati perfetti: assenza di tasse e costi di transazione, perfetta divisibilità dei beni ed esclusione delle vendite allo scoperto. In queste condizioni, l'identificazione del portafoglio ottimale può essere definito come un problema quadratico di ottimizzazione. Risolvendolo si ottengono le quote dei titoli x_i che assicurano il minor rischio possibile $\sigma_{\mathcal{P}}^2$ per un rendimento atteso fissato $r_{\mathcal{P}} = r^*$. L'impostazione del problema è la seguente:

$$\min_{x_i} \sigma_{\mathcal{P}}^2 \quad (1.12)$$

con

$$\sigma_{\mathcal{P}}^2 = \sum_i \sum_j x_i \cdot x_j \cdot \sigma_{ij} \quad (1.13)$$

$$r_{\mathcal{P}} = r^* \quad (1.14)$$

$$r_{\mathcal{P}} = \sum_i x_i \cdot r_i \quad (1.15)$$

$$\sum_i x_i = 1 \quad (1.16)$$

$$x_i \in \mathbb{R}_0^+ \quad \forall i \quad (1.17)$$

Questo problema di ottimizzazione ha una soluzione quando sono soddisfatte le seguenti condizioni tecniche:

$$\min_i r_i \leq r_{\mathcal{P}} \leq \max_i r_i \tag{1.18}$$

$$\sigma_i > 0 \quad \forall i \tag{1.19}$$

$$\rho_{ij} > -1 \quad \forall (i, j) \tag{1.20}$$

$$\exists (i \neq j) \text{ tali che } r_i \neq r_j \tag{1.21}$$

In questo caso, il portafoglio con la varianza minima (*Minimum Variance Portfolio*, MPV) è quello con il minor rischio (si veda la Figura 1.2). Quindi, ha senso cercare la soluzione del problema di ottimizzazione soltanto fissando il rendimento atteso $r^* \geq r_{MPV}$. Ovviamente, gli investitori razionali sceglieranno solamente i portafoglio che si trovano sopra il portafoglio MPV (la linea in grassetto sul grafico a sinistra della Figura 1.2), cioè i portafoglio che si trovano sulla frontiera efficiente. Scegliendo un qualsiasi portafoglio \mathcal{P} che non sia sulla frontiera efficiente, esisterà un'altro portafoglio \mathcal{P}'' che avrà lo stesso rendimento atteso, ma minor rischio. Esisterà anche un portafoglio \mathcal{P}' con lo stesso rischio ma con rendimento atteso più alto. I portafoglio \mathcal{P}' e \mathcal{P}'' si trovano sulla frontiera efficiente.

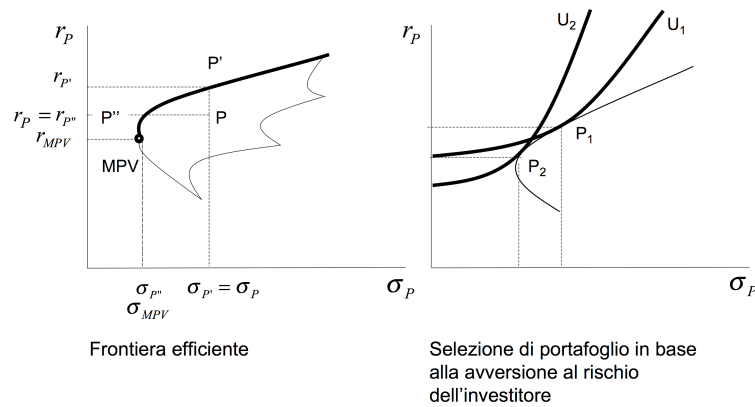


Figura 1.2: La frontiera efficiente e la selezione del portafoglio in base al grado di avversione al rischio dell'investitore

Nel modello di Markowitz, il rendimento atteso dato r^* , è un parametro esogeno e quindi potrebbe risultare anche inferiore a r_{MPV} che non è noto a priori. Questo inconveniente può essere evitato modificando la funzione da minimizzare che diventa il *trade-off* tra rischio e rendimento atteso:

$$\max_{x_i} (\lambda \cdot r_{\mathcal{P}} - (1 - \lambda) \cdot \sigma_{\mathcal{P}}^2) \tag{1.22}$$

La frontiera efficiente si ottiene quindi risolvendo il nuovo problema di ottimizzazione (con la funzione obiettivo modificata) e fissando quindi un nuovo parametro esogeno $\lambda \in [0, 1]$. Se $\lambda = 1$, la soluzione del problema sarà il portafoglio con il rendimento più alto possibile senza tener conto della varianza. Per i valori bassi di λ , il modello dà più peso al rischio del portafoglio e meno al suo rendimento. Quando $\lambda = 0$ la soluzione del problema è il portafoglio MVP.

Data la convessità della frontiera efficiente di Markowitz, il premio per il rischio marginale decresce quando ci sono solamente titoli rischiosi nel portafoglio. Inoltre, la varianza del portafoglio efficiente può essere espressa come la funzione del suo rendimento atteso. Il processo di selezione del portafoglio consiste prima di tutto nella determinazione della frontiera efficiente, e poi nell'identificazione del portafoglio ottimo, rappresentato dal punto sulla curva in cui il premio per il rischio marginale è uguale all'utilità marginale (si veda il grafico di destra della Figura 1.2). Il contributo principale del modello di Markowitz consiste nel fatto che la frontiera efficiente può essere determinata senza conoscere il grado di avversione al rischio dell'investitore.

I vincoli di non negatività ($x_i \in \mathbb{R}_0^+$ per ogni i) rendono il modello di Markowitz un problema NP-completo.¹ (Garey e Johnson 1979, [40]). Nonostante ciò per un numero ragionevolmente piccolo di titoli, N , il problema di ottimizzazione può essere risolto numericamente con ausilio di un software per la risoluzione dei problemi di ottimizzazione non lineare (ottimizzazione quadratica) in un tempo ragionevole (Jagannathan e Ma 2003, [55]).

1.2 Stili di gestione del portafoglio

Le strategie di gestione di portafoglio possono sostanzialmente essere identificate in due categorie principali: strategie attive e strategie passive. La gestione attiva, tipica dei fondi comuni, si pone come obiettivo un rendimento maggiore

¹Nella Teoria della Complessità i problemi NP-completi sono i più difficili problemi in NP (“problemi non deterministici a tempo polinomiale”) nel senso che quasi sicuramente non appartengono a P (l'insieme di quei problemi decidibili in un numero polinomiale di passi da una Macchina di Turing deterministica). La ragione è che, se si potesse trovare un algoritmo in grado di risolvere velocemente (cioè in tempo polinomiale) un qualsiasi problema NP-completo, allora si potrebbe usarlo per risolvere velocemente tutti i problemi NP. La classe di complessità che contiene tutti i problemi NP-completi è spesso indicata con NP-C. Un esempio di problema NP-completo è il problema delle somme parziali, cioè: dato un insieme finito di numeri interi, determinare se esiste un sottoinsieme tale che la somma dei suoi elementi sia zero. È evidente che è facile verificare velocemente se un sottoinsieme è o meno una soluzione del problema, ma non è noto alcun metodo per trovare una soluzione che sia sensibilmente più veloce per verificare tutti i sottoinsiemi.

al *benchmark* di riferimento. Tale proposito è raggiungibile mediante il *market timing* (l'attività del gestore volta ad anticipare il trend del mercato di settori e di singoli titoli), e/o lo *stock picking* (come è chiamata l'azione di selezione dei titoli sui quali si punta).

La gestione passiva è l'opposto della gestione attiva. La gestione passiva è applicata dai gestori dei fondi comuni indicizzati e degli ETF che hanno come obiettivo di base la composizione del portafoglio dei titoli il più possibile in linea con il valore del loro *benchmark*, o l'indice di riferimento. Il risultato pratico è la replica della performance dell'indice, cioè un risultato in linea con il mercato.

Una buona gestione attiva è quella che batte il *benchmark* con un ragionevole livello di volatilità. Anche se non esiste un extrarendimento senza extrarischio, quest'ultimo deve rimanere comunque proporzionato. Una buona gestione passiva è quella che segue il *benchmark* senza aumentare la propria volatilità: se così non avviene, infatti, si sopporta un extrarischio che non è motivato da alcun extrarendimento atteso.

La gestione attiva è più costosa di quella passiva. Infatti, per fare gestione passiva, è sufficiente un software che, pesando a dovere i titoli in portafoglio, replichi la composizione del *benchmark*. Al contrario, per fare una buona gestione attiva, è necessario remunerare un *team* di gestori che, per battere il *benchmark*, strutturi un portafoglio diverso da quello di mercato e lo movimenti per sfruttare di volta in volta le occasioni che si presentano.

Senza dubbio, di maggior interesse, nell'ottica dell'investitore, è scoprire se il gestore sia in grado di sfruttare dalle informazioni in suo possesso al fine di prevedere l'andamento dei mercati per garantire al portafoglio da lui gestito una performance superiore a quella del *benchmark* (al netto dei maggiori costi che la gestione attiva comporta). La letteratura in materia riconosce come punto di riferimento gli studi di Eugene Fama ([37]), che hanno evidenziato come il tentativo di sfruttare le informazioni alla ricerca di *miss-pricing* sia essenzialmente un tentativo vano. I mercati finanziari, almeno quelli più evoluti, sono sostanzialmente efficienti nel senso che “prezzano” i vari titoli presenti sul mercato in modo da riflettere tutte le informazioni disponibili.

Una versione meno “integralista” della teoria, riconducibile a Jensen ([56] e [57]), sostiene che, anche qualora vi fossero “nicchie” (aree di inefficienza) nel mercato, la gestione attiva sarebbe comunque incapace di superare, in termini di performance, il mercato, a causa dei costi di transazione. Diverse ricerche empiriche sembrano confermare tali affermazioni. Mentre Jensen fornisce evidenze empiriche a sostegno della tesi secondo cui i gestori non dispongono di informazioni riservate che consentano loro di ottenere extraperformance, Ippolito ([53]) dimostra la capacità dei fondi di battere il *benchmark*, ma senza

prendere in considerazione i costi di transazione e le commissioni di gestione. Risultati del tutto analoghi a questi ultimi sono stati ottenuti da Cesari e Panetta ([18] e [19]). Da sottolineare come a deporre ulteriormente contro le presunte *superior skills* dei gestori vi siano la possibile scelta di *benchmark* inefficienti con cui confrontare i portafogli ed il fenomeno noto come *survivorship bias*.

Inoltre, il livello dei costi dei fondi comuni, in particolare le commissioni, da molti studiosi è ritenuto eccessivo. Quindi la domanda che si pongono gli studiosi d'oltreoceano da alcuni decenni (e che da qualche anno è oggetto di studio anche nella letteratura italiana finanziaria) è quale sia l'effettivo contributo dei gestori di portafogli alla performance degli stessi. Gli interrogativi, in estrema sintesi, sono i seguenti:

- quale è il reale grado di “attivismo” dei portafogli gestiti?
- la gestione attiva può davvero contribuire in modo significativo alla performance dei fondi, giustificando, in questo modo, gli oneri (fees) che essa comporta?

Il confronto obbligatorio con il *benchmark* nel prospetto informativo e l'evidenza che, nella maggior parte dei casi, la performance del fondo non supera quella del parametro, induce un numero crescente di investitori (soprattutto i più attenti ed informati) e studiosi a mettere in discussione le capacità superiori dei gestori. A ciò si aggiunge la recente diffusione, anche in Europa, di strumenti di investimento collettivo a gestione dichiaratamente passiva (index funds, ETF) che richiedono commissioni sensibilmente inferiori a quelle di un fondo tradizionale. Si comprende facilmente quindi quanto attuale sia l'argomento in discussione, suscettibile di modificare, nei prossimi anni, la stessa struttura dell'industria del risparmio gestito.

Gli *Exchange Traded Funds*, ETF (“fondi indicizzati quotati”) sono una particolare categoria di fondi comuni o Sicav,² le cui quote sono negoziate in Borsa in tempo reale come semplici azioni, attraverso una banca o un qualsiasi intermediario autorizzato. Una delle caratteristiche peculiari di questo

²L'investimento in Sicav è molto diffuso in Italia e spesso si usa indifferentemente il termine fondo comune o Sicav. In realtà, tra i due strumenti esistono alcune importanti differenze, anche se svolgono la medesima attività di gestione collettiva del risparmio. Sicav è l'acronimo di società di investimento a capitale variabile. Si tratta di società per azioni, che hanno per oggetto esclusivo l'investimento collettivo in valori mobiliari (azioni, obbligazioni, ecc.) del patrimonio raccolto mediante l'offerta al pubblico in via continuativa delle proprie azioni. A differenza dei fondi comuni, dove il risparmiatore non ha la possibilità di intervenire nelle scelte di *asset allocation*, nel caso delle Sicav, egli diventa socio; di conseguenza, può partecipare alle assemblee sociali e contribuire, seppur entro certi limiti, alle scelte di allocazione delle risorse.

innovativo strumento è costituita dall'indicizzazione: gli ETF replicano infatti passivamente la composizione di un indice di mercato (geografico, settoriale, azionario o obbligazionario) e di conseguenza anche il suo rendimento. Gli ETF si caratterizzano come una via di mezzo tra un'azione ed un fondo e consentono di sfruttare i punti di forza di entrambi gli strumenti. In particolare, acquistando un ETF, è possibile realizzare la diversificazione e la riduzione dei rischi tipica di un fondo di investimento, scambiando le quote del fondo come una normale azione. La trattazione continua degli ETF rende infatti possibile conoscerne il valore di mercato in ogni istante, a differenza dei fondi comuni di investimento tradizionali, il cui valore viene fissato una sola volta al giorno, generalmente a fine giornata. Gli ETF possono essere comprati e venduti come semplici azioni, tramite un intermediario come una banca, una sim o un broker online, semplicemente inoltrando un ordine con il relativo codice ISIN dello strumento. Con una sola operazione, si acquista/vende l'intero portafoglio di titoli che compone l'indice. Investire in un ETF significa prendere posizione su un intero indice di mercato, diversificando e diminuendo il rischio dell'investimento. L'ETF consente di ottenere un rendimento pari a quello dell'indice *benchmark* di riferimento (a parte un limitato errore, positivo o negativo, definito *tracking error*) in virtù di una gestione passiva e di un particolare meccanismo di funzionamento detto *creation/redemption in kind* (sottoscrizione/rimborso in natura). Il meccanismo *creation/redemption in kind* consente una migliore replica dell'indice e un maggior contenimento dei costi. In pratica, vi è una ponderazione continua del portafoglio rispetto alle variazioni dell'indice di riferimento, per cui se un titolo ottiene una performance superiore agli altri il suo peso nel paniere aumenta proporzionalmente alla sua capitalizzazione. Contrariamente ai fondi tradizionali, gli ETF non prevedono nessuna commisione di "entrata", di "uscita" e di "performance". Viene applicata solo una commissione di gestione annua molto contenuta.

A testimonianza dell'attualità del tema della gestione passiva si riporta una selezione di un'articolo tratto dal de *Il sole 24 ore*, supplemento settimanale *Plus 24* di Sabato 15 Dicembre 2007: ... "In particolare a Piazza Affari, secondo i consulenti di Bcg (Boston Consulting Group), nel quadriennio 2002-2006 gli strumenti finanziari che replicano gli indici (non solo ETF, introdotti in Italia nel 2002, ma anche fondi azionari e obbligazionari passivi) hanno registrato una crescita annuale composta (CAGR) delle masse gestite pari al 142 per cento. Inoltre, i 187 ETF quotati attualmente sul listino milanese hanno visto incrementare il controvalore degli scambi di quasi il 50 per cento, passando dai 17,5 miliardi di euro del 2006 ai 25,8 miliardi di quest'anno (gennaio-ottobre 2007). ... A monte di tale "innamoramento finanziario" vi sono di certo i bassi costi. Niente gestori da pagare e dunque commissioni meno

onerose. Nel report elaborato dall'analista di Morgan Stanley, Doborah Fuhr, emerge che le commissioni totali annue degli ETF oscillano fra 0,07% e 1,11% contro una forchetta ben più larga dei fondi tradizionali (0,39% - 1,91%). A Milano, i "cloni" hanno costi annui tra 0,16% e 0,9 per cento".

La replica di un'indice di mercato può essere realizzata detenendo in portafoglio tutte le attività finanziarie appartenenti a quell'indice di nella stessa proporzione (effettuando dunque movimenti solo quando vengono aggiunti o eliminati titoli dall'indice), oppure, in alternativa, tentando di replicare l'indice stesso con un numero inferiore di titoli utilizzando particolari tecniche di selezione del portafoglio per scegliere i titoli più appropriati. Questa strategia alternativa è comunemente denominata *index tracking*, ed è il tema centrale di questa tesi di dottorato.

1.3 Indici di borsa

Gli indici di Borsa sono un parametro indicativo del valore di uno specifico mercato o di un paniere composto dai titoli più rappresentativi dello stesso. Gli indici vengono calcolati come valore medio ponderato delle quotazioni in borsa di un gruppo di titoli di riferimento.

A seconda del numero di titoli presi in considerazione per calcolare l'indice, è possibile operare una distinzione tra indici generali (che prendono in considerazione tutti i titoli quotati nel mercato di riferimento) e indici parziali (che si limitano invece a considerare un campione significativo dei titoli quotati nel mercato di riferimento).

Il termine "settoriale" si riferisce agli indici che prendono in considerazione titoli di società operanti in un determinato comparto produttivo. La loro principale utilità è quella di fornire un'indicazione sintetica, ma esaustiva, circa lo stato di salute del settore oggetto di analisi.

L'utilità degli indici di Borsa risiede nella loro capacità di costituire un indicatore piuttosto valido delle condizioni economiche, risultando in tal modo un importante fattore di riferimento per investitori istituzionali, gestori di fondi comuni, ecc. La funzione di un indice di Borsa è dunque quella di fornire agli investitori un'impressione generale circa l'andamento medio del mercato. Attraverso gli indici, i risparmiatori possono quindi farsi un'idea complessiva sull'andamento del valore di un investimento. La scelta dell'indice da prendere come riferimento è naturalmente legata agli specifici obiettivi dell'investitore ed alla composizione del suo portafoglio finanziario.

Gli indici di Borsa costituiscono inoltre dei parametri a cui spesso risulta legato il rendimento di alcuni strumenti finanziari (si pensi alle obbligazioni strutturate o ad alcune tipologie di strumenti derivati).

Ci sono tre fattori importanti da considerare nella costruzione dell'indice.

Il primo fattore importante è il campione delle società selezionato per comporre l'indice. È essenziale che questo campione sia in grado di rappresentare il “comportamento” dell'intera popolazione, incorporandolo, quindi, le caratteristiche principali della popolazione di riferimento.

Il secondo fattore è la pesatura di ogni singolo titolo all'interno dell'indice, cioè il meccanismo di ponderazione utilizzato per la loro costruzione. Ci sono tre principali tipi di pesatura utilizzati: *price weighted* (l'indice ponderato rispetto ai prezzi delle attività sottostanti), *market value weighted* (l'indice ponderato per la capitalizzazione di mercato) e *unweighted* o *equal-weighted* (tutti i titoli hanno lo stesso peso all'interno dell'indice).

Il terzo fattore è la procedura computazionale utilizzata per la costruzione dell'indice. La prima alternativa è quella di calcolare la media aritmetica dei titoli. Un'altra è rappresentata dal calcolo dell'indice prendendo in considerazione tutti i cambiamenti avvenuti, sia in base al prezzo che al valore, rispetto all'indice di base. Al posto della media aritmetica può essere utilizzata la media geometrica delle componenti dell'indice.

1.3.1 Indici *price weighted*

L'indice *price weighted* è calcolato attraverso la media aritmetica dei prezzi correnti dei titoli che lo compongono. Questo significa che i movimenti dell'indice sono influenzati dalle variazioni dei prezzi delle sue componenti. Il più famoso indice *price weighted* è Dow Jones Industrial Average (DJIA). Il DJIA è la media aritmetica dei prezzi delle trenta società *blue chips* nel settore industriale. DJIA si calcola nel seguente modo: la somma dei prezzi correnti dei trenta titoli si divide con un divisore che tiene conto del frazionamento delle azioni e dei cambiamenti del campione delle società che compongono l'indice. Questo divisore viene aggiustato in modo tale da tenere invariato il valore dell'indice prima e dopo il frazionamento dei titoli.

$$DJIA_t = \sum_{i=1}^{30} \frac{P_{it}}{D_{agg}}$$

dove:

$DJIA_t$ è il valore del DJIA al giorno t

P_{it} è il prezzo di chiusura del titolo i -esimo al giorno t

D_{agg} è il divisore aggiustato per il giorno t

Si ricorre ad un semplice esempio per chiarire il concetto del divisore aggiustato (si veda tabella 1.1) Il valore iniziale dell'indice viene ipotizzato pari

Titoli	Prezzi prima del frazionamento	Prezzi dopo il frazionamento
A	30	10
B	20	20
C	10	10
Somma	60	40
	60:3=20	40:x=20
		x=2 diventa nuovo divisore

Tabella 1.1: Esempio di calcolo del valore dell'indice *price weighted*

a 20. In seguito al frazionamento quindi, il valore dell'indice deve rimanere 20, e di conseguenza a dover variare sarà il divisore che da 3 si ridurrà a 2.

Nell'indice *price weighted* per definizione i titoli con i prezzi più alti hanno un peso maggiore rispetto a quelli titoli con i prezzi più bassi. Inoltre, i cambiamenti percentuali equivalenti nei prezzi dei titoli all'interno dell'indice non comportano le variazioni equivalenti nel valore dell'indice. Un aumento ad esempio del 10% del prezzo di un titolo che pesa di più nell'indice, porta una variazione maggiore nel valore dell'indice rispetto ad un aumento del 10% di un prezzo del titolo che pesa relativamente di meno.

1.3.2 Indici *value weighted*

L'indice *value weighted* viene calcolato partendo dal valore di mercato totale di tutti i titoli che lo compongono. Tale valore iniziale viene chiamato "base" dell'indice. Successivamente all'indice viene assegnato un valore (generalmente 100) associato al valore precedentemente definito. Si procede poi al calcolo della nuova capitalizzazione per tutti i titoli che compongono il suddetto indice; questo valore viene confrontato con la base iniziale. Un modo da poter determinare la variazione percentuale subita dall'indice:

$$\text{Indice}_t = \frac{\sum P_t Q_t}{\sum P_b Q_b} \cdot \text{Valore iniziale dell'indice}$$

dove:

Indice_t è il valore del DJIA al giorno t

P_t è il prezzo di chiusura del titolo al giorno t

Q_t è il numero di azioni in circolazione al giorno t

P_b è il prezzo di chiusura del titolo al giorno base

Q_b è il numero di azioni in circolazione al giorno base

Per spiegare il concetto, la tabella 1.2 riporta un esempio di calcolo del valore di un indice *value weighted*. Nell'indice *value weighted*, ai titoli con

Titolo	Prezzo	Num d'azioni	Capitalizzazione
Dicembre 31, 2006			
A	10	1.000.000	10.000.000
B	15	6.000.000	90.000.000
C	20	5.000.000	100.000.000
Totale			200.000.000
Base dell'indice 100			
Dicembre 31, 2007			
A	12	1.000.000	12.000.000
B	10	12.000.000	120.000.000
C	20	5.500.000	110.000.000
Totale			242.000.000
		Nuovo valore = $\frac{242.000.000}{200.000.000} \cdot 100$	= 121

Tabella 1.2: Esempio di calcolo del valore dell'indice *value weighted*

la maggiore capitalizzazione viene associato anche un maggiore peso. Una determinata variazione percentuale nella capitalizzazione di una società grande ha un impatto maggiore sul valore dell'indice rispetto alla stessa variazione percentuale subita nella capitalizzazione di una società piccola.

1.3.3 Indici *unweighted*

La principale caratteristica dell'indice *unweighted* è che tutti i titoli che lo compongono hanno lo stesso peso. In questo modo non si dà l'importanza alla dimensione delle società oppure ai diversi prezzi dei titoli. Le variazioni nel valore dell'indice vengono calcolate in base alla media aritmetica delle variazioni percentuali dei prezzi o della capitalizzazione dei titoli che lo compongono. Questo tipo di indice viene di solito costruito per fini sperimentali.

Capitolo 2

Ottimizzazione euristica

2.1 Introduzione

In che modo l'econometria applicata riesce a gestire i problemi di ottimizzazione estremamente complessi dal punto di vista computazionale? Quando si parla di problema di ottimizzazione "difficile" ci si riferisce, ad esempio, a problemi con funzione obiettivo non differenziabile, problemi con "molti" ottimi locali, problemi con un elevato numero di parametri; in altri termini problemi con un insieme delle possibili soluzioni "molto grande" o con elevato numero di possibili configurazioni dei parametri. Un approccio per gestire questo tipo di problemi è quello di cercare la semplificazione fino al punto da poter utilizzare i metodi tradizionali per risolverlo. Tuttavia, nel processo di formulazione del problema, attraverso tali semplificazioni, il rischio è quello di perdere informazioni importanti e di fare modelli troppo semplificati, creati su un problema non più specifico, ma piuttosto generale. Un altro approccio è quello di ricorrere all'ottimizzazione euristica. I modelli di ottimizzazione euristica si distinguono dai modelli classici. Il primo passo è definire l'ottimizzazione euristica. Spesso il termine euristico è legato agli algoritmi che imitano in qualche maniera comportamenti che si ritrovano in natura; ad esempio l'evoluzione naturale della specie tramite incroci e mutazione (algoritmi genetici), il processo di ricottura di scoglimento del ferro (*simulated annealing*) oppure l'autoorganizzazione di una colonia di formiche (*ant colony optimization*). La definizione generale del termine euristico, in realtà si basa sulle proprietà degli algoritmi euristici (2007 [78]):

1. Un algoritmo euristico ha il fine di fornire un'approssimazione stocastica di alta qualità della soluzione di un determinato problema. Questa approssimazione può avvenire in due modi. Il primo è ottenere una soluzione al crescere delle risorse computazionali impiegate per una singola

esecuzione dell'algoritmo. Il secondo modo, invece, porta alla soluzione del problema aumentando il numero delle esecuzioni d'algoritmo.

2. Un algoritmo euristico è robusto al cambiare delle caratteristiche del problema da risolvere. Se un algoritmo euristico infatti, risolve il problema posto in un determinato caso, lo deve risolvere anche al cambiare delle condizioni al contorno. Si dice che l'algoritmo euristico risolve tutta la classe di un determinato problema.
3. Un algoritmo euristico non è sensibile ai aggiustamenti dei parametri del algoritmo nemmeno ai cambiamenti dei vincoli che servono per determinare lo spazio di tutte le soluzioni possibili.
4. Un algoritmo euristico si implementa facilmente per una serie di casi di un problema posto.
5. Un algoritmo euristico non deve contenere gli elementi soggettivi.

Il vantaggio riconosciuto agli algoritmi euristici si identifica nella loro marcata adattabilità ai cambiamenti nei vincoli specifici contenuti nei problemi da ottimizzare. La maggioranza degli algoritmi euristici richiede come unica condizione la possibilità di poter calcolare la funzione obiettivo (la funzione da ottimizzare) nei punti che appartengono allo spazio di tutte le soluzioni possibili. Ai fini del funzionamento degli algoritmi euristici (a differenza dei modelli classici), non sono necessarie nè proprietà universali in riferimento alla funzione obiettivo, nè la differenziabilità di quest'ultima. Alcuni algoritmi euristici trattano anche problemi di ottimizzazione che hanno come spazio di tutte le soluzioni possibili un insieme discreto.

Gli algoritmi euristici sono principalmente metodi computazionali, e si sono di conseguenza evoluti di pari passo allo sviluppo delle scienze informatiche. Uno dei primi contributi in questo campo è da riconoscere a Bock (1958 [15]) e Croes (1958 [23]) che hanno introdotto le prime procedure per la risoluzione del problema del commesso viaggiatore.¹ I maggiori impulsi evolutivi sono riconducibili alla fine degli anni 80 e 90 come conseguenza ad importanti miglioramenti nel campo dell'informatica. Tuttavia, solo negli ultimi anni i personal computer hanno raggiunto capacità sufficienti tali da poter sostenere le procedure di calcolo previste dagli algoritmi euristici.

L'ottimizzazione euristica si suddivide in due grandi classi: metodi costruttivi e metodi di ricerca locale. Alla classe dei metodi costruttivi appartengono

¹Un esempio classico è il problema del commesso viaggiatore. Tale problema consiste in un commesso viaggiatore che deve visitare N città esattamente una volta e quindi ritornare al punto di partenza. Visti i costi di viaggio tra le città il suo problema è come pianificare il suo itinerario in modo da ridurre i costi al minimo.

le reti neurali. Queste reti neurali si basano principalmente sulla simulazione di neuroni artificiali opportunamente collegati. Un esempio delle reti neurali è quello proposto da McCulloch e Pitts (1943 [68]). In questo modello i suddetti neuroni ricevono in ingresso degli stimoli e li elaborano. L'elaborazione può essere anche molto sofisticata ma in un caso semplice si può pensare che i singoli ingressi vengano moltiplicati per un opportuno valore, detto peso. Il risultato delle moltiplicazioni viene sommato e se la somma supera una certa soglia il neurone si attiva determinandone l'uscita. Il peso indica l'efficacia sinaptica della linea di ingresso e serve a quantificarne l'importanza. Un ingresso molto importante avrà un peso elevato, mentre un ingresso poco utile all'elaborazione avrà un peso inferiore. Si può pensare che se due neuroni comunicano fra loro, utilizzando maggiormente alcune connessioni allora tali connessioni avranno un peso maggiore. L'utilizzo più diffuso di tali sistemi si ritrova nei software di OCR, nei sistemi di riconoscimento facciale e più in generale nei sistemi che hanno la funzione di elaborare dati soggetti ad errori o rumore. Essi sono anche uno degli strumenti maggiormente utilizzati nelle analisi di *Data mining* e come mezzo previsionistico dell'analisi finanziaria o meteorologica. Negli ultimi anni è aumentata notevolmente la loro importanza nel campo della bioinformatica nel quale vengono utilizzate per la ricerca di *pattern* funzionali e/o strutturali in proteine e acidi nucleici. Mostrando opportunamente una lunga serie di input (fase di *training* o apprendimento), la rete è in grado di fornire l'output più probabile.

In questo lavoro, si è voluto focalizzare sulla seconda grande classe degli algoritmi euristici: i metodi di ricerca locale. Questi metodi sono dei metodi iterativi che si basano sui concetti di esplorazione della soluzione nella vicinanza della soluzione corrente in ogni passo di elaborazione dell'algoritmo (Ausiello 1999, [8]). In generale, l'algoritmo di ricerca locale deve essere inizializzato prendendo come la soluzione corrente o la soluzione iniziale un qualsiasi elemento appartenente allo spazio in cui si va a ricercare la soluzione. Successivamente, in ogni iterazione viene scelto un elemento appartenente all'intorno della soluzione corrente, che poi in base al criterio di selezione o sostituisce la soluzione corrente (cioè diventa la nuova soluzione corrente) o viene scartato. Per un lungo periodo questi metodi sono stati considerati poco maturi per la loro applicazione nella risoluzione dei problemi complessi. Solo di recente si è assistito alla loro ripresa, grazie ai diversi vantaggi che conseguono al loro utilizzo. Gli algoritmi di ricerca locale sono, infatti, trattabili dal punto di vista computazionale e la loro implementazione risulta particolarmente semplice, sono inoltre caratterizzati da una forte flessibilità ed adattabilità ai cambiamenti delle caratteristiche del problema ed hanno riscosso particolare successo nella risoluzione dei problemi di tipo reale e complesso.

Un classico metodo di ricerca locale che ha come obiettivo quello di trovare il punto di minimo di una funzione $f(x)$ è rappresentato nel seguente pseudo-codice:

Pseudo codice di metodo di ricerca locale

1. Genera la soluzione iniziale x^c
 2. **while** (NOT criterio di stop) **do**
 3. Scegli $x^n \in \mathcal{N}(x^c)$ (che si trova nell'intorno della soluzione corrente)
 4. if $f(x^n) < f(x^c)$ then $x^c = x^n$
 5. **end while**
-
-

Ci sono due tipi di metodi di ricerca locale: metodi di traiettoria *trajectory methods* e metodi basati su una popolazione *population based methods*, spesso chiamati anche algoritmi evolutivi. Il primo tipo, cioè i metodi di traiettoria, si concentrano sul miglioramento di una singola soluzione. Esempi di algoritmi di questo tipo sono: *simulated annealing*, *threshold accepting*, *tabu search*, etc. La seconda categoria di metodi di ricerca locale, cioè gli algoritmi evolutivi aggiornano tutto l'insieme delle soluzioni ammissibili ad ogni iterazione. A questa famiglia appartengono: gli algoritmi genetici, i *differential evolution methods* e le *ant colonies*. Tutti i metodi di ricerca locale hanno delle proprie regole sia per quanto riguarda la scelta degli intorni della soluzione corrente, sia per quanto riguarda l'aggiornamento/miglioramento della soluzione corrente.

2.2 Algoritmi evolutivi

Gli algoritmi evolutivi sono una classe vasta di algoritmi “euristici” particolarmente efficaci nei problemi di ottimizzazione. Essi sono basati su principi che imitano l'evoluzione naturale delle specie. Un algoritmo evolutivo opera su una “piccola” popolazione iniziale di potenziali soluzioni del problema dato, che viene estratta, di norma casualmente, dall'insieme di tutte le soluzioni possibili. Questa popolazione subisce successivamente un processo di selezione basato sulla sopravvivenza degli individui più idonei, in base alla valutazione di una funzione di “fitness”, e l'azione di alcuni operatori “genetici” (mutazione, incrocio, inversione) che simulano a tutti gli effetti l'evoluzione naturale e sono usati per produrre via via migliori approssimazioni della soluzione cercata. Ogni nuova generazione di individui creati grazie alla selezione ed agli operatori genetici sostituisce, parzialmente o totalmente, gli individui della popolazione corrente, e viene poi sottoposta alle stesse operazioni di valutazione

e riproduzione. Dopo alcune generazioni, il migliore degli individui della popolazione avrà elevate capacità di adattamento all'ambiente circostante, e sarà molto più adatto dei precedenti a svolgere il compito per cui è stato progettato. Di norma, le ripetizioni cicliche dei passi appena descritti per creare le nuove generazioni proseguono fino a quando venga soddisfatto un opportuno criterio di stop.

I concetti evolutivi sono stati introdotti in biologia nel XIX secolo da Charles Darwin [29], e sono ancora oggi considerati validi. Negli anni '60, molti informatici studiarono la biologia evolutiva con l'idea che i meccanismi dell'evoluzione potessero essere usati come strumenti di ottimizzazione per i problemi d'ingegneria. Le prime proposte in questa direzione sono dovute a John Holland [51] dell'Università del Michigan, che ha introdotto gli algoritmi genetici, a Lawrence Fogel [39] e ai suoi colleghi dell'Università della California a San Diego, che hanno introdotto la programmazione evolutiva, ed a Ingo Rechenberg [71] dell'Università di Berlino che indipendentemente dagli americani ha introdotto le strategie evolutive. Infine è stata anche sviluppata la programmazione genetica, proposta da John Koza [63] dell'Università Stanford all'inizio degli anni '90. Questo lavoro è stato la base per la creazione di una ampia classe di metodi di ottimizzazione principalmente applicati a problemi di progettazione in ingegneria (come ad esempio la progettazione di aeromobili civili o militari, o la progettazione di molecole ad uso medico). Questi metodi ispirati all'evoluzione naturale sono comunemente chiamati algoritmi evolutivi.

È chiaro che gli algoritmi evolutivi differiscono dai metodi classici di ottimizzazione: elenchiamo le principali differenze. Gli algoritmi evolutivi:

- Possono utilizzare il calcolo parallelo perchè indagano simultaneamente su molte possibilità differenti, anziché esaminare un solo punto dello spazio alla volta.
- Richiedono soltanto la conoscenza della funzione obiettivo, cioè una funzione di selezione basata sulla idoneità, ossia la *fitness* degli individui, e degli operatori genetici da utilizzare.
- *Conseguentemente* (pro), non richiedono operazioni aritmetiche elaborate, ma piuttosto operazioni tipiche del *soft computing*, come la manipolazione delle stringhe simboliche e le operazioni di sorting.
- Sono metodi non deterministici, perchè in parte sono basati su ricerca casuale.
- Mantengono sempre una popolazione di candidati per la soluzione. Soltanto una tra queste è la "migliore" e le altre sono dei campioni che rappresentano altre regioni dello spazio degli stati dove nelle prossime generazioni potrebbe essere trovata una soluzione migliore. In questo modo gli algoritmi evolutivi non

si “intrappolano” attorno un ottimo locale e mantengono sempre la possibilità di trovare un ottimo fuori dall’intorno della soluzione ottima corrente.

- In generale non hanno il concetto della “soluzione ottima”. Una soluzione ottima per un algoritmo evolutivo risulta essere migliore solo in base al confronto rispetto alle altre.
- *Conseguentemente* (contro), non sono collegabili direttamente a risultati teorici di esistenza e/o unicità; in generale, gli algoritmi evolutivi sono ancora privi di una sistemazione teorica soddisfacente che spieghi la loro straordinaria efficacia.

Dato il parallelismo tra la computazione evolutiva e l’evoluzione biologica, per gli algoritmi evolutivi vengono comunemente usati dei termini presi a prestito dalla biologia. Per completezza, può essere utile richiamare il significato che essi assumono nel contesto computazionale. Tutti gli individui sono composti di cellule. Ogni cellula contiene la stessa dotazione di uno o più *cromosomi*. Quindi un *cromosoma* rappresenta un individuo, cioè una soluzione candidata. Un cromosoma può essere idealmente diviso in *geni* che quindi rappresentano le unità di base degli individui. L’insieme dei valori che possono assumere i geni si chiama *allele*. La posizione di un gene nella stringa, cioè la sua posizione all’interno del cromosoma si chiama *locus*. La struttura genetica che rappresenta un individuo si chiama *genotipo*. Le caratteristiche osservate di un individuo come l’organismo vivente si chiama *fenotipo*. L’idoneità o *fitness* di un organismo è definita tipicamente come la probabilità che l’organismo viva abbastanza da riprodursi o come la funzione del numero di discendenti che esso genera.

Ogni algoritmo evolutivo considera una popolazione di m individui per un numero N di iterazioni. Indichiamo con $\{x_1^n, \dots, x_m^n\}$ la popolazione di m individui alla n -esima iterazione. Ogni individuo rappresenta una potenziale soluzione al problema, e ogni soluzione x_i^n con $i \in \{1, \dots, m\}$, $n \in N$, viene valutata mediante l’assegnazione di un valore di idoneità. Poi si procede con la creazione di una nuova generazione della popolazione, rappresentata dall’iterazione $n + 1$, selezionando gli individui più idonei e trasformandoli per mezzo di opportuni operatori genetici: *incrocio*, *mutazione* e *inversione*. L’incrocio scambia parti dei cromosomi imitando la ricombinazione biologica fra due organismi dotati di un solo cromosoma. La mutazione modifica in modo casuale i valori di alcune posizioni del cromosoma. L’inversione cambia l’ordine di una selezione del cromosoma, cambiando quindi l’ordine nel quale i geni sono sistemati.

Un algoritmo evolutivo deve avere le seguenti cinque componenti:

1. Una rappresentazione genetica per le potenziali soluzioni del problema.

2. Un modo per creare una popolazione iniziale di potenziali soluzioni del problema.
3. Una funzione di valutazione che svolge il ruolo dell'ambiente stimando le soluzioni in termini di idoneità.
4. Degli operatori genetici che alterino i cromosomi discendenti.
5. Dei valori per i diversi parametri che vengono usati nell'algoritmo evolutivo (la numerosità della popolazione, la probabilità di applicazione degli operatori genetici).

Gli algoritmi evolutivi hanno molte applicazioni pratiche come ad esempio nel modello di crescita ottima unisetoriale di Ramsey [70]. Questo modello è il prototipo nello studio dell'allocazione intertemporale delle risorse economiche, ed è il punto di partenza della teoria moderna della crescita economica. Più precisamente il problema da risolvere è l'*equazione di Bellman* (1957 [11]), a tempo discreto, deterministica e 1-dimensionale:

$$V(x) = \max_{y \in \Gamma(x)} \{U(x, y) + \beta V(y)\}$$

Questa equazione ha la forma di un'equazione di punto fisso $V = T(V)$ in un opportuno spazio di funzioni. In molte (ma non in tutte) le applicazioni economiche, l'operatore T , nell'ipotesi che il fattore di sconto β sia minore di 1, verifica le condizioni sufficienti di Blackwell (1965 [14]) per l'applicabilità del principio delle contrazioni di Banach-Caccioppoli. Questo rende teoricamente calcolabile un' ϵ -approssimazione della funzione V con una comune iterazione. Ciò però non deve trarre in inganno sulla semplicità del problema. In generale, la funzione incognita $V(x) = V(s_1, s_2, \dots, s_d)$ dipende da una variabile di stato che può essere un vettore a d componenti $x = (s_1, s_2, \dots, s_d)$: l'intero d è detto allora la dimensione dell'equazione; modelli recenti di strutture di vendita (Hall e Rust [52], Aguirregabiria [4]) considerano migliaia di prodotti, per cui dimensioni come $d = 8000$ sono comuni (cioè $V : \mathbb{R}^{8000} \rightarrow \mathbb{R}$). Ora, nel mentre la complessità temporale delle iterazioni "di Bellman" per d fissata è polinomiale, è stato mostrato da Chow e Tsitsiklis [26] che la complessità del calcolo delle ϵ -approssimazioni della funzione V nel caso peggiore può dipendere esponenzialmente da d , rendendo il problema genericamente intrattabile. Applicando l'algoritmo evolutivo per risolvere l'equazione di Bellman nel caso di modello di crescita ottima unisetoriale e confrontando i risultati ottenuti con i risultati ottenuti risolvendo lo stesso problema con i metodi classici [13] il risultato più rilevante è che, nell'ambito della precisione dei numeri-macchina, qui 10^{-16} , l'algoritmo evolutivo porta agli stessi risultati del classico schema

delle iterazioni di Bellman, ma in un tempo di CPU che risulta da 40 a 100 volte inferiore: sostanzialmente minuti invece di ore. Questo conferma che gli algoritmi evolutivi sono uno strumento estremamente potente e promettente non solo nei campi in cui sono ormai comunemente utilizzati (come la progettazione ingegneristica), ma anche nella economia matematica computazionale.

2.2.1 *Ant colonies AC*

Gli algoritmi *ant colonies* prendono spunto e simulano le strategie utilizzate dalle formiche per trovare cibo e riportarlo al loro nido. La formica, infatti, esplora i dintorni in maniera casuale e successivamente, quando trova del cibo, lo trasporta al nido lasciando tracce di feromoni sul terreno che guidano le altre formiche verso la meta. L'intensità delle tracce rilasciate dipende dalla quantità e dalla qualità del cibo a disposizione in quel luogo, nonché dalla distanza del cibo dal nido. Dal momento che le formiche tendono di preferenza a seguire le stesse tracce, il loro comportamento consente di ottimizzare il lavoro. Quando la sorgente di cibo si esaurisce, le tracce di feromoni si indeboliscono e le formiche iniziano a percorrere strade diverse. L'area di ricerca delle formiche corrisponde alle soluzioni possibili, la quantità di cibo è la funzione obiettivo, le tracce di feromoni sono la memoria del processo.

2.3 Metodi di traiettoria

La caratteristica principale dei metodi di traiettoria è il modo in cui viene creata la “nuova” soluzione corrente, cioè quella soluzione che eventualmente andrà a sostituire la soluzione corrente. Questa creazione avviene solamente in base alla conoscenza della soluzione corrente. Non è richiesto nessun altro tipo di informazione. Per questo motivo la difficoltà cruciale quando si cerca di implementare questo tipo di algoritmi consiste nel definire degli intorni della soluzione corrente. Definire degli intorni “efficienti” può essere molto impegnativo e varia da un problema all'altro .

2.3.1 *Simulated annealing SA*

Simulated annealing rappresenta un raffinato tipo di metodi di traiettoria e simula il processo di ricottura di scioglimento dei metalli. Se un materiale solido viene riscaldato oltre il punto di fusione e poi riportato allo stato solido tramite raffreddamento, le sue proprietà strutturali dipenderanno dal processo di raffreddamento. Un algoritmo di *simulated annealing* simula il cambiamento

di energia di un sistema (considerato come un insieme di particelle) sottoposto a raffreddamento, finché non converge ad uno stato solido. Il movimento delle singole particelle dovuto al calore diminuisce col passare del tempo ed allo stesso modo diminuiscono le dimensioni dell'intorno all'interno del quale ricercare un valore ottimale. Come in tutti gli algoritmi di ricerca locale, il miglioramento della soluzione corrente passando da x^c a x^n è sempre accettato. Tuttavia, l'algoritmo accetta anche una soluzione peggiorativa di quella corrente, però soltanto con una data probabilità. Questa probabilità che rappresenta il range di peggioramento della soluzione corrente diminuisce ad ogni esecuzione dell'algoritmo fino ad arrivare al valore zero. Il criterio di stop dell'algoritmo avviene esaurendo un numero prefissato di iterazioni oppure nel caso in cui per un determinato numero di iterazioni consecutive non ci sia un miglioramento della soluzione corrente x^c .

2.3.2 *Tabu search TA*

La caratteristica degli algoritmi del tipo *tabu search* è rappresentato dall'uso sistematico di una memoria che ci garantisce l'impossibilità di ripetere le nostre scelte in un prefissato periodo di tempo. Con questi algoritmi si accetta di muoversi anche verso soluzioni peggiori purché siano ammissibili. L'algoritmo genera una *tabulist* di soluzioni proibite da non prendere più in considerazione. La scelta dell'intorno nell'ambito del quale andare ad effettuare la ricerca successiva non dipende, dunque solo dalla soluzione corrente, ma anche dalla storia pregressa.

In questo lavoro viene data particolare enfasi agli algoritmi *threshold accepting* che sono trattati in maggiore dettaglio nel seguente paragrafo.

2.4 *Threshold accepting*

Il *Threshold accepting* è un algoritmo euristico di ottimizzazione di recente introduzione, ed è sostanzialmente una versione modificata e semplificata dell'algoritmo *simulated annealing*. L'algoritmo di *threshold accepting* è stato introdotto per la prima volta da Dueck e Scheuer nel 1990 ([30]) come la versione deterministica di *simulated annealing* per risolvere il problema del commesso viaggiatore. I due fisici hanno creato anche una versione non deterministica del *threshold accepting* che si è dimostrata migliore rispetto alla versione deterministica. Come tutti gli algoritmi euristici, anche TA parte da una soluzione ammissibile e cerca di migliorarla esplorandone un opportuno intorno. L'algoritmo termina quando non è più possibile trovare delle soluzioni migliori nell'intorno della soluzione corrente, ovvero dopo un prefissato tempo di

calcolo o un prefissato numero di operazioni. Per sfuggire alle trappole dei minimi soluzioni locali, durante l'esecuzione dell'algoritmo, TA “accetta” anche i movimenti in direzione opposta a quella della minimizzazione (*up-hill moves*) purché non superino una certa “soglia” (da cui il nome *threshold accepting*)

2.4.1 L'algoritmo

Sia $f : \mathcal{X} \rightarrow \mathbb{R}$ un'istanza nel problema di ottimizzazione. Supponiamo che \mathcal{X} sia un insieme discreto². Il compito dell'algoritmo è quello di trovare almeno una tra potenzialmente molte soluzioni ottimali diverse, cioè trovare un elemento \mathcal{X}_{min} , dove

$$\mathcal{X}_{min} = \{x \in \mathcal{X} | f(x) = f_{opt}\}$$

e

$$f_{opt} = \min_{x \in \mathcal{X}} f(x)$$

L'algoritmo di *threshold accepting* è una procedura sofisticata dell'algoritmo di ricerca locale. Sono due le principali caratteristiche che permettono all'algoritmo di essere classificato tra gli algoritmi di ricerca locale. La prima riguarda lo spazio di ricerca della soluzione \mathcal{X} che deve ammettere una struttura degli intorni, che preveda l'esistenza di una mappa $\mathcal{N} : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ tale che per ogni $x \in \mathcal{X}$ esiste un suo intorno $\mathcal{N}(x) \subset \mathcal{X}$, dove $2^{\mathcal{X}}$ rappresenta l'insieme di tutti sottoinsiemi di \mathcal{X} . La seconda proprietà soddisfatta dall'algoritmo consiste nel processo di ricerca del risultato; la partenza del processo deve essere casuale, scegliendo tra un qualsiasi elemento $x^c \in \mathcal{X}$, a cui viene assegnata una probabilità $1/|\mathcal{X}|$ (se l'insieme \mathcal{X} è un insieme finito). Ci possono essere anche altri modi per generare $x^c \in \mathcal{X}$. Ad esempio, la scelta della soluzione di partenza può ricadere su un elemento $x^c \in \mathcal{X}$ con buone caratteristiche per essere una soluzione ottimale per la quale $f(x^c)$ sia abbastanza piccola. Di seguito verrà dimostrato che la scelta della soluzione iniziale non è un elemento essenziale per un corretto comportamento dell'algoritmo, soprattutto se il numero di iterazioni dell'algoritmo è elevato e gli altri requisiti dell'algoritmo sono soddisfatti. Una scelta del punto di partenza “mirata” non è addirittura nemmeno consigliabile. Se l'algoritmo inizia con una buona soluzione trovata ad esempio utilizzando qualsiasi altro algoritmo di ottimizzazione, questa soluzione potrebbe corrispondere ad un minimo locale della funzione da minimizzare mettendo alla “dura” prova il *threshold accepting* già all'inizio della sua ricerca. Infatti, tutti i parametri dell'algoritmo devono essere scelti in modo

²Per l'applicazione del *threshold accepting* vedasi Winker [77], capitolo 15. Le principali caratteristiche di *threshold accepting* discreto e *threshold accepting* continuo sono simili.

tale da poter garantire che l'algoritmo non venga bloccato tra i minimi locali. Questo obiettivo è realizzato durante l'esecuzione dell'algoritmo accettando come soluzioni possibili anche quelle meno buone. Di conseguenza, inizialmente l'algoritmo spazia ovunque dentro lo spazio delle soluzioni possibili. Quindi non si ha nessun tipo di guadagno partendo da una soluzione buona, anzi con i numerosi esperimenti e esecuzioni dell'algoritmo si è verificato che è meglio partire con un punto scelto a caso. Per maggiori approfondimenti su questo tema si veda Winker 2001 [77]. Una volta scelto il punto di partenza x^c , il *threshold accepting* si comporta in modo simile agli altri algoritmi di ricerca locale, svolgendo un numero elevato di iterazioni. A ciascuna iterazione, l'algoritmo prima genera un nuovo elemento x^n che sta nell'intorno $\mathcal{N}(x^c)$ della soluzione corrente x^c . Poi, in base ad un prefissato criterio viene deciso se il nuovo elemento x^n diventa la soluzione di partenza per la successiva iterazione, o se viene mantenuta la soluzione corrente anche per la prossima iterazione. Ci possono essere vari modi per generare un nuovo elemento x^n appartenente all'intorno $\mathcal{N}(x^c)$. L'elemento x^n può essere scelto arbitrariamente da $\mathcal{N}(x^c)$ con la probabilità $\frac{1}{|\mathcal{N}(x^c)|}$ oppure certi elementi x^n dell'insieme $\mathcal{N}(x^c)$ possono essere selezionati con maggiore probabilità. Questa probabilità maggiore di scelta per certi elementi dell'insieme $\mathcal{N}(x^c)$ di solito è legata ai valori di funzione obiettivo $f(x^n)$: ad esempio si può calcolare il valore di f per tutti gli elementi dell'insieme $x^n \in \mathcal{N}(x^c)$ e poi privilegiare quelli elementi che hanno i valori di f minori. Una volta stabilito il processo di selezione di x^n dall'insieme $\mathcal{N}(x^c)$, entra in gioco la particolarità del *threshold accepting*. Mentre gli algoritmi classici di ricerca locale accettano che x^n diventi la soluzione corrente per la successiva iterazione solo se questa rappresenta un miglioramento vero rispetto alla soluzione corrente, cioè se $f(x^n) \leq f(x^c)$, gli algoritmi *threshold accepting*, al contrario sono caratterizzati da un processo decisionale più complesso. Il *threshold accepting* accetterà x^n come nuova soluzione corrente se e solo se $f(x^n) - f(x^c) \leq T$, dove T rappresenta un valore di soglia non negativo definito in precedenza. Quindi, data la soglia $T > 0$ durante un'iterazione del *threshold accepting*, l'algoritmo, o passerà dalla soluzione corrente x^c a una nuova soluzione corrente x^n , oppure manterrà la soluzione corrente x^c . Il *threshold accepting* accetterà x^n come nuova soluzione corrente del problema se in termini di funzione obiettivo questa soddisferà almeno uno dei seguenti criteri: 1) genera un risultato migliore rispetto a x^c ; 2) il peggioramento rispetto a x^c non supera la soglia di tolleranza prefissata T . In altre parole, la soglia T quantifica il peggioramento della soluzione che viene accettata dall'algoritmo. Per questo motivo, a differenza degli altri algoritmi di ricerca locale, il *threshold accepting* richiede la predeterminazione di un maggiore numero di parametri di definizione dell'algoritmo stesso. In particolare, è necessaria la definizione della successione di soglie *threshold heights*, T_i . Dal punto di vista teorico, ad

ogni iterazione dell'algoritmo dovrebbe corrispondere una sola soglia T_i . Nella pratica, la successione delle soglie è più ridotta e più iterazioni dell'algoritmo fanno riferimento ad una stessa soglia. Inoltre, la successione delle soglie è decrescente e converge allo zero. Quindi esisterà un i_0 tale che per ogni $i \geq i_0$, x^n sarà accettata come una nuova soluzione corrente se e solo se rappresenta veramente un miglioramento della soluzione corrente x^c . Il pseudo-codice che rappresenta la logica dell'algoritmo *threshold accepting* nel linguaggio di programmazione è riportato di seguito:

Threshold accepting	
1.	Genera la sequenza dei threshold $T_i, i = 0, \dots, I_{max}$
2.	Genera la soluzione corrente x^c
3.	Scegli $x^n \in \mathcal{N}(x^c)$
4.	Calcola $\Delta f = f(x^n) - f(x^c)$
5.	Se $\Delta f \leq T_i$ assegna $x^c = x^n$
6.	Se $i \leq I_{max}$ incrementa $i, i = i + 1$ e torna allo step 3.
7.	Altrimenti x^c è l'output dell'algoritmo

Dal codice sopra riportato si nota come il criterio di *stop* sia dato dall'esaurimento di tutte le iterazioni I_{max} . Inoltre, il numero di soglie T_i è esattamente uguale al numero di iterazioni I_{max} . Alcuni valori di soglia T_i potranno coincidere determinando diverse iterazioni dell'algoritmo con il medesimo valore di soglia. È possibile che la soluzione di minimo globale venga individuata anche con valori di soglia T_i elevati; in questi casi, il valore T_i troppo alto ne determina l'automatica eliminazione. Per questo motivo, durante l'esecuzione dell'algoritmo viene memorizzata non solo l'ultima soluzione corrente x^c , ma anche la migliore tra le soluzioni individuate. Se $f(x_{opt}) \leq f(x^c)$, allora il risultato dell'algoritmo viene migliorato sostituendo x^c con x_{opt} .

2.4.2 Convergenza del *threshold accepting*

Nel processo di risoluzione di problemi di ottimizzazione di varia natura, hanno trovato impiego diversi tipi di algoritmi euristici. Non esiste, in effetti, un modello universale migliore degli altri. Un criterio decisivo nel processo di scelta dell'algoritmo da adottare per risolvere un problema specifico risiede nell'analisi del comportamenti asintotico dello stesso algoritmo.

In questo senso, supponiamo che le iterazioni dell'algoritmo possano essere infinite; interessa capire se l'algoritmo sia in grado di convergere all'ottimo globale almeno in termini teorici. Anche se l'accertamento che questa condizione sia soddisfatta non è sufficiente per capire se l'algoritmo sia adatto per

un determinato problema, per lo meno offre un'indicazione significativa sul suo comportamento. Un forte segnale per un'algoritmo non adatto per un determinato problema è, infatti, la sua incapacità di convergere all'ottimo globale anche per un numero infinito di esecuzioni.

Nel caso degli algoritmi classici di ricerca locale non esistono risultati teorici non banali ³ tali da dimostrarne la loro convergenza (si veda il lavoro di Winker 2001 [77]). I risultati di convergenza, invece, esistono per gli algoritmi *simulated annealing* e *threshold accepting*. Si può dimostrare che l'algoritmo *threshold accepting* è in grado di ottenere un'approssimazione dell'ottimo globale quanto vicino si voglia con la probabilità prossima ad uno nel momento in cui il numero d'iterazioni dell'algoritmo tende ad infinito. Ovviamente la successione delle soglie (*threshold heights*), deve essere scelta in modo corretto.

Definiamo lo spazio di tutte le soluzioni possibili con $\mathcal{X} = \{x_1, \dots, x_n\}$. Supponiamo che f sia la funzione obiettivo da minimizzare sull'insieme \mathcal{X} . L'algoritmo di ricerca locale potrà essere descritto attraverso le catene di Markov. Si possono definire:

1. La distribuzione di probabilità iniziale $\pi^{(0)}$ che descrive la scelta per la soluzione iniziale $x^c = x^{(0)}$, dove

$$\pi_i^{(0)} = P(x^{(0)} = x_i) \geq 0$$

per ogni $i = 1, \dots, n$; e

$$\sum_{i=1}^n \pi_i^{(0)} = 1$$

2. Le probabilità di transizione

$$p_{ij}^{(k)} = \tilde{p}_{ij}^{(k-1,k)} = P(x^{(k)} = x_j | x^{(k-1)} = x_i)$$

per ogni $i, j = 1, \dots, n$

Supponiamo che la scelta della soluzione iniziale avvenga in maniera casuale sull'insieme \mathcal{X} con la probabilità $\frac{1}{|\mathcal{X}|}$. Si ha che $\pi_i^{(0)} = \frac{1}{|\mathcal{X}|} = \frac{1}{n}$ per ogni $i = 1, \dots, n$. Introduciamo, quindi, la seguente matrice $Q = (q_{ij})$ che descrive la struttura degli intorni dell'insieme \mathcal{X} :

$$q_{ij} = \begin{cases} \frac{1}{|\mathcal{N}(x_i)|} & \text{se } x_j \in \mathcal{N}(x_i) \\ 0 & \text{se } x_j \notin \mathcal{N}(x_i) \end{cases}$$

³Qualsiasi algoritmo di ottimizzazione applicato ad un insieme finito di possibili soluzioni genererà prima o poi la soluzione.

Si ha che $q_{ij} = 0$ se x_j non appartiene all'intorno $\mathcal{N}(x_i)$ di x_i . Altrimenti, con l'ipotesi di equiprobabilità, ogni soluzione candidata viene scelta dall'intorno di x_i con la probabilità $q_{ij} = \frac{1}{|\mathcal{N}(x_i)|}$. Data la matrice Q e un valore di soglia T_{k-1} le probabilità di transizione $p_{ij}^{(k)}$ si possono rappresentare tramite la matrice di transizione $P^{(k)}$:

$$p_{ij}^{(k)} = \begin{cases} q_{ij} = 0 & \text{se } j \notin \mathcal{N}(x_i) \\ q_{ij} & \text{se } j \in \mathcal{N}(x_i), j \neq i \text{ e } f(x_j) - f(x_i) \leq T_{k-1} \\ 0 & \text{se } j \in \mathcal{N}(x_i), j \neq i \text{ e } f(x_j) - f(x_i) > T_{k-1} \\ 1 - \sum_{j' \neq i} p_{ij'}^{(k)} & \text{se } j = i \end{cases}$$

Data la distribuzione di probabilità iniziale $\pi^{(0)}$ e le probabilità di transizione $p_{ij}^{(k)}$, in linea di principio è possibile calcolare la distribuzione di probabilità dopo k iterazioni dell'algoritmo:

$$\pi^{(k)} = \pi^{(0)} P^{(1)} P^{(2)} \dots P^{(k)}$$

Per le applicazioni pratiche, sarebbe interessante effettivamente poter calcolare $\pi^{(k)}$. Inoltre, sarebbe anche interessante calcolare la probabilità di individuare una soluzione appartenente all'insieme delle soluzioni che si avvicinano alla soluzione globale x^{opt} durante l'esecuzione dell'algoritmo. Purtroppo occorre constatare che questi calcoli possono essere eseguiti solamente per insiemi con un numero ridotto di elementi, a causa della loro estrema complessità operativa al crescere del numero di questi. Per tale motivo, analisi asintotiche delle catene di Markov nel contesto dei problemi di ottimizzazione si concentrano soprattutto su $\sum_{x_i \in \mathcal{X}_{min}} \pi_i^{(k)}$ per k che tende all'infinito. Viene inoltre analizzata anche la convergenza della probabilità $\sum_{x_i \in \mathcal{X}_{sigma}} \pi_i^{(k)}$ dove \mathcal{X}_{sigma} indica l'insieme di tutti $x \in \mathcal{X}$ tali che per un fissato $\sigma > 0$ soddisfano $f(x) - f_{opt} \leq \sigma$. Sarebbe naturale aspettarsi che queste probabilità siano uguali o si avvicinino molto ad uno. Data la distribuzione di probabilità iniziale $\pi_i^{(0)} = \frac{1}{|\mathcal{X}|}$, la condizione necessaria per ottenere questo risultato è che sia garantita l'esistenza di un cammino che collega qualsiasi due x_i e x_j all'interno dell'insieme \mathcal{X} , cioè:

$$\forall x_i, x_j \in \mathcal{X} \exists l \in \mathbb{N} - \{0\}, k_0, \dots, k_l \in \{1, \dots, n\} \text{ tali che}$$

$$x_i = x_{k_0}, \dots, x_{k_l} = x_j \text{ e } \forall l' \geq 1 \text{ si ha che } x_{k_{l'}} \in \mathcal{N}(x_{k_{l'}})$$

In altre parole, la condizione necessaria è che gli insiemi \mathcal{X} e $\mathcal{N}(x)$ siano connessi.

Supponiamo che d_{ij} rappresenti la lunghezza del cammino più corto che collega x_i e x_j . Chiamiamo D il prodotto di tutte le lunghezze d_{ij} , $D = \prod_{i,j \in \{1, \dots, n\}} d_{ij}$.

Allora per ogni $i, j \in \{1, \dots, n\}$ la probabilità che il cammino che collega x_i e x_j abbia lunghezza minore o uguale a zero è positiva, cioè:

$$\left(Q^D\right)_{ij} > 0 \quad (2.1)$$

Per l'insieme connesso \mathcal{X} è possibile calcolare il valore di soglia che serve per raggiungere x_j partendo da x_i con il cammino x_{k_0}, \dots, x_{k_l} :

$$T_{x_{k_0}, \dots, x_{k_l}} = \sum_{l'=1}^l \max\{0, f(x_{l'}) - f(x_{l'-1})\}$$

In generale, esiste più di un cammino che collega x_i e x_j all'interno di \mathcal{X} . Sia $\Lambda_{i \rightarrow j}$ l'insieme di tutti quei cammini. Si può, allora, definire la soglia minima che serve per raggiungere x_j partendo da x_i :

$$T_{i \rightarrow j} = \min_{\Lambda_{i \rightarrow j}} T_{x_{k_0}, \dots, x_{k_l}}$$

Definiamo inoltre $\tilde{T} = \max_{i,j} \text{con} T_{i \rightarrow j}$ che rappresenta il valore di soglia massimo che ci permette di arrivare da $x_j \in \mathcal{X}$ a $x_i \in \mathcal{X}$.

Seguendo il lavoro di Winker [77], per dedurre la convergenza dell'algoritmo *threshold accepting*, bisogna prima dimostrare la convergenza di *simulated annealing*. La convergenza del *threshold accepting* viene come conseguenza di tale dimostrazione.

Convergenza di *simulated annealing*

Utilizzando la matrice Q definita prima, la matrice di transizione per il *simulated annealing* diventa la seguente:

$$p_{ij}^{(k)} = \begin{cases} q_{ij} = 0 & \text{se } j \notin \mathcal{N}(x_i) \\ q_{ij} & \text{se } j \in \mathcal{N}(x_i), j \neq i \text{ e } f(x_j) - f(x_i) < 0 \\ q_{ij} \exp\left(\frac{f(x_i) - f(x_j)}{C_k}\right) & \text{se } j \in \mathcal{N}(x_i), j \neq i \text{ e } f(x_j) - f(x_i) \geq 0 \\ 1 - \sum_{j' \neq i} p_{ij'}^{(k)} & \text{se } j = i \end{cases} \quad (2.2)$$

con C_{k-1} il valore della successione di raffreddamento (*cooling sequence*) al tempo $k - 1$. Supponiamo per il momento che $C_k = C$, cioè sia una costante. In questo caso tutte le matrici di transizione $P^{(k)}$ saranno tra di loro uguali. Indichiamo con $P = P^{(k)}$ questa matrice di transizione. Siccome \mathcal{X} è un insieme convesso vale la (2.1). Inoltre,

$$p_{ij}^{(k)} > 0 \iff q_{ij} > 0$$

Di conseguenza $P^D > 0$ con D definita come nell'equazione (2.1). Questa disuguaglianza vale componente per componente. Seguendo il lavoro di Kemeny e Snell (1976, pagina 69 [61]) questa condizione equivale a dire che P rappresenta una matrice di transizione regolare. Allora, per il teorema di Markov di convergenza ergodica per le matrici di transizione regolari (si veda Kemeny e Snell, 1976, pagina 71 [61]) si ha che:

$$\pi^{(k)} = \pi^{(0)} P^{(1)} \dots P^{(k)} = \pi^{(0)} P^{(k)} \quad (2.3)$$

converge ad una distribuzione di probabilità d'equilibrio π indipendente da $\pi^{(0)}$ per k che tende all'infinito. Inoltre, P^k converge ad una matrice di probabilità con tutte le righe uguali a π . Abbiamo quindi:

$$\lim_{k \rightarrow \infty} \pi^{(k)}(x \in \mathcal{X}_{\min}) = \sum_{x \in \mathcal{X}_{\min}} \pi(x) \quad (2.4)$$

dove $\pi^{(k)}$ indica la distribuzione di probabilità dopo k applicazioni di matrice di transizione P sulla distribuzione iniziale di probabilità π^0 . Scegliendo la costante C abbastanza piccola, la probabilità nell'equazione (2.4) si avvicina ad uno (Hajek, 1988, pagina 312 [50]). Questa proprietà ci garantisce la convergenza ad una soluzione globale ottima quando il numero di iterazioni dell'algoritmo tende all'infinito.

Ora bisogna ottenere il medesimo risultato per una successione di segmenti di Catene di Markov di lunghezza finita (Aarts et al, 1997, pagina 104 [1]). La successione di segmenti di lunghezza finita di Catene di Markov con le matrici di transizione tra loro diverse (a causa dei diversi valori di *cooling parameter* C_k) si chiama Catena di Markov non omogenea⁴. Supponiamo che le matrici di transizione p_{ij}^k siano definite come nella 2.2 e supponiamo che la successione dei valori di raffreddamento C_k , con $k = 0, 1, 2 \dots$ soddisfa le seguenti due condizioni:

$$\begin{aligned} \lim_{k \rightarrow \infty} C_k &= 0 \\ C_k &\geq C_{k+1}, \quad k = 0, 1, \dots \end{aligned} \quad (2.5)$$

Ora bisogna dimostrare che la Catena di Markov non omogenea definita sopra sia fortemente ergodica⁵, cioè che la distribuzione di probabilità $\pi^{(k)}$ converga

⁴Una Catena omogenea di Markov è, appunto, un processo markoviano nel quale la probabilità di transizione dipende unicamente dallo stato del sistema immediatamente precedente e non anche dal tempo t , ed è pertanto detto omogeneo. Più in generale, si dimostra che in una Catena di Markov omogenea si ha che la probabilità di passare da uno stato ad un altro in t è costante nel tempo.

⁵Una Catena di Markov si definisce ergodica se e solo se per ogni istante iniziale t_0 e per ogni condizione iniziale di probabilità $p(t_0)$ esiste ed è indipendente da t_0 e $p(t_0)$ il limite della probabilità per tempi infiniti $P = \lim_{t \rightarrow \infty} p(t)$.

all'interno della distribuzione. Il teorema numero 7 nel Aarts et al. (1997, pagina 107 [1]) ci garantisce che questo accade quando la *cooling sequence* C_k soddisfa:

$$C_k \geq \frac{\Gamma}{\log(k + k_0)}, \text{ per } k = 0, 1, \dots \quad (2.6)$$

con $\Gamma > 0$ e $k_0 > 2$. In questo caso:

$$\lim_{k \rightarrow \infty} P(x_k \in \mathcal{X}_{\min}) = 1 \quad (2.7)$$

Questo significa che asintoticamente, quando il numero d'iterazioni tende all'infinito, una adeguata implementazione di *simulated annealing* trova la soluzione ottimale con la probabilità che converge ad uno. La letteratura su questo argomento presenta varie approssimazioni per il Γ in base all' problema di ottimizzazione che si cerca di risolvere Gidas (1985, [42]), Hajek (1988, [50]), Gelfand e Mitter (1991, [41]).

Convergenza di *threshold accepting*

Basandosi sui risultati di convergenza di *simulated annealing*, Althöfer e Koschnik (1991, [7]) hanno ottenuto dei risultati per la convergenza di *threshold accepting*. In particolare hanno dimostrato che *simulated annealing* può essere descritto come un involucro convesso⁶ di *threshold accepting*. Utilizzando questo risultato, la convergenza di *threshold accepting* è una conseguenza di convergenza di *simulated annealing* (Althöfer e Koschnik, 1991, pagina 188 [7]).

Per ogni quadrupla (ϵ, a, n, δ) con $\epsilon > 0$, $a > 0$, $n \in \mathbb{N}$, $n \geq 3$ e $\delta > 0$, esiste una sequenza di soglie $(T_k)_{k=0}^{K-1} \in \{\frac{i\delta}{(n-1)(n-1)} | i \in \mathbb{N}\}$ di una certa lunghezza $K = K(\epsilon, a, n, \delta)$ tale che l'applicazione di *threshold accepting* a qualsiasi problema che possa essere descritto tramite l'insieme \mathcal{X} finito, che non sia orientato e che rappresenti un grafo connesso con meno di n vertici, e sia rappresentabile per mezzo di una funzione obiettivo a valori reali

$$f : \mathcal{X} \longrightarrow [0, a] \quad (2.8)$$

si ha che

$$P(f(x_K) - f_{opt} \geq 1 - \epsilon) \quad (2.9)$$

⁶Si definisce involucro convesso di un qualsiasi insieme I l'intersezione di tutti gli insiemi convessi che contengono I . Siccome l'intersezione di insiemi convessi è a sua volta convessa, una definizione alternativa di involucro convesso è il più piccolo insieme convesso contenente I . Se l'insieme I è sottoinsieme di uno spazio vettoriale reale, il suo involucro convesso si può costruire come l'insieme di tutte le combinazioni convesse di punti di I , cioè tutti i punti del tipo $\sum_{j=1}^n \lambda_j x_j$, dove gli x_j sono punti di I e λ_j sono numeri reali positivi a somma 1, ovvero $\sum_{j=1}^n \lambda_j = 1$. Ovviamente, se I è convesso, il suo involucro è I stesso.

In altre parole, dato un problema di ottimizzazione su un'insieme di n elementi con una funzione obiettivo che ha come codominio l'intervallo $[0, a]$ e dei parametri δ e ϵ scelti, denominati anche *quality parameters*, esiste sicuramente una successione di soglie finite con delle caratteristiche importanti. Queste soglie, se utilizzate per l'implementazione dell'algoritmo *threshold accepting* garantiscono il raggiungimento della soluzione del problema. Questa soluzione, con la probabilità di $1 - \epsilon$, non si scosterà dalla soluzione ottima per più di δ , cioè l'algoritmo troverà la soluzione del problema tanto vicina alla soluzione ottima quanto si desidera, con la probabilità vicino ad uno se il numero d'iterazioni dell'algoritmo è abbastanza grande.

2.4.3 Implementazione del *threshold accepting*

I risultati di convergenza asintotica del *threshold accepting*, come accennato in precedenza, sono una buona base di partenza per la scelta del *threshold accepting* come l'algoritmo risolutivo per un determinato problema di ottimizzazione. Tuttavia, questi risultati non bastano per garantire lo stesso comportamento dell'algoritmo quando questo viene applicato ad un problema reale, quindi la sua esecuzione comprende un numero finito di iterazioni. Ci sono tre fasi importanti da seguire nell'implementazione dell'algoritmo per ottenere un risultato soddisfacente.

- La prima fase consiste nella definizione del problema. Il problema deve essere ben formulato e deve avere la struttura di un problema di ottimizzazione. In particolare, bisogna definire la funzione obiettivo e lo spazio di tutte le soluzioni possibili.
- La seconda fase consiste nell'analisi dettagliata dello spazio di tutte le possibili soluzioni. Di solito vengono imposti dei vincoli al problema di ottimizzazione che ci aiutano a limitare lo spazio di ricerca della soluzione.
- La terza fase è quella più importante. In questa fase viene definita la struttura degli intorno all'interno dello spazio di ricerca. Solo questa fase è tipica per il *threshold accepting* perchè in questa fase vengono definiti i parametri dell'algoritmo. In particolare, vengono definite le soglie, il numero di iterazioni dell'algoritmo per ciascuna soglia oppure qualche altro criterio di stop dell'algoritmo. Inoltre, bisogna specificare il numero n_{exec} , cioè quante volte occorre eseguire l'algoritmo, il numero n_{rounds} , cioè il numero totale delle soglie e il numero n_{steps} , cioè quante volte viene eseguito l'algoritmo per un dato valore di soglia. In questo modo il numero totale d'iterazioni per un n_{exec} è dato da $I = n_{rounds} \times n_{steps}$.

Fase 1: Definizione del problema di ottimizzazione

La prima fase consiste nella trasformazione di un problema reale in un problema di ottimizzazione. Il tipico problema di ottimizzazione è espresso nella forma seguente:

$$\max_{x \in \mathcal{X}} f(x) \quad (2.10)$$

dove \mathcal{X} è possibilmente un sottoinsieme discreto di \mathbb{R}^n . Qualora l'insieme \mathcal{X} fosse un'insieme non convesso, la generazione della soluzione corrente e delle soluzioni possibili che stanno nell'intorno della soluzione corrente avviene in seguente modo. Come spazio di ricerca viene preso l'intero insieme \mathbb{R}^n e poi viene aggiunta una penalità alla funzione obiettivo per tutti gli $x \notin \mathcal{X}$. Per poter esplorare l'intero \mathcal{X} , la penalità applicata alla funzione obiettivo inizialmente ha un valore piccolo e viene man mano incrementata al crescere del numero di iterazioni. Alla fine, il valore di penalità è abbastanza alto e quindi ci garantisce che la soluzione trovata sia sicuramente ammissibile.

Fase 2: Definizione degli intorni

Gli intorni della soluzione corrente devono essere scelti accuratamente. La funzione obiettivo dovrebbe avere un comportamento locale nell'intorno $\mathcal{N}(x^c)$ di x^c . Questo significa che per gli elementi $x^n \in \mathcal{N}(x^c)$ i valori della funzione $f(x^n)$ dovrebbero essere più vicini al valore $f(x^c)$ e non a $f(x^r)$ con $x^r \in \mathcal{X}$ scelto a caso, cioè $x^r \notin \mathcal{N}(x^c)$. Inoltre, gli intorni non devono essere né troppo grandi né troppo piccoli. Esiste un *trade-off* tra un intorno grande che ci garantisce la presenza degli elementi diversi da x^c al suo interno e un intorno piccolo dove la funzione obiettivo ha veramente un comportamento locale.

Se $\mathcal{X} \subset \mathbb{R}^d$, la definizione degli intorni è diretta. Per ogni $x \in \mathcal{X}$ e ogni $\epsilon \geq 0$, l'intorno in \mathbb{R}^d è dato da:

$$\mathcal{N}_\epsilon(x) = \{\tilde{x} \in \mathbb{R}^d \mid \|x - \tilde{x}\| < \epsilon\} \quad (2.11)$$

dove $\|\cdot\|$ indica la metrica euclidea in \mathbb{R}^d . Proiettando queste sfere di raggio ϵ su \mathcal{X} si ottiene la struttura degli intorni in \mathcal{X} :

$$\mathcal{N}(x) = \{\tilde{x} \in \mathcal{X} \mid \|x - \tilde{x}\| < \epsilon\} \quad (2.12)$$

Se \mathcal{X} invece è un insieme discreto che non può essere immerso in uno spazio vettoriale reale, per definire gli intorni si può utilizzare la distanza di Hamming⁷. La distanza di Hamming d_H tra due elementi x_1 e x_2 appartenenti a \mathcal{X} è

⁷Nella teoria dell'informazione, la distanza di Hamming tra due stringe di uguale lunghezza è il numero di posizioni nelle quali i simboli corrispondenti sono diversi. In altri termini, la distanza di Hamming misura il numero di sostituzioni necessarie per convertire una stringa nell'altra, o il numero di errori che hanno trasformato una stringa nell'altra.

data da:

$$d_H(x_1, x_2) = \sum_{j=1}^d |x_{1j} - x_{2j}| \quad (2.13)$$

dove $x_i = (x_{i1}, \dots, x_{id})$ per $i = 1, 2$ rispettivamente. Quindi gli intorni possono essere definiti come sfere di raggio ϵ fissando semplicemente la massima distanza di Hamming d_H^{\max} :

$$\mathcal{N}(x) = \{\tilde{x} \in \mathcal{X} \mid d_H(x, \tilde{x}) \leq d_H^{\max}\} \quad (2.14)$$

Fase : Definizione delle soglie

La definizione delle soglie dipende dalla topologia della funzione obiettivo nell'intorno della soluzione corrente. La topologia può essere dedotta dalla funzione di distribuzione cumulativa *CDF*, *cumulative distribution function* delle distanze $\Delta = |f(x^n) - f(x^c)|$ con i punti x^c scelti in maniera casuale per un totale di n_{steps} volte. Come proposto da Winker e Fang (1997, [79]), le soglie T_i possono essere calcolate come quantili che corrispondono ai percentili equidistanti $P_i = \frac{n_{rounds} - i}{n_{rounds}}$, $i = 1, \dots, n_{rounds}$. Il seguente pseudo-codice rappresenta questa procedura:

Generazione della threshold sequence

1. Scegli a caso $x^c \in \mathcal{X}$
 2. For $i = 1$ to n_{steps} do
 3. Calcola $x^n \in \mathcal{X}(x^c)$ e $\Delta_i = |f(x^n) - f(x^c)|$
 4. $x^c = x^n$
 5. End for
 6. Calcola la funzione di distribuzione cumulativa F di Δ_i , $i = 1, \dots, n_{steps}$
 7. Calcola la successione delle soglie $T_r = F^{-1}\left(\frac{n_{rounds} - r}{n_{rounds}}\right)$, $r = 1, \dots, n_{rounds}$
-
-

Scelta di n_{steps} , n_{rounds} e n_{exec}

La scelta dei parametri n_{steps} , n_{rounds} e n_{exec} dipende dal problema che si vuole risolvere. Tipicamente il numero n_{steps} è molto più grande del numero n_{rounds} . Il numero minimo per n_{rounds} suggerito da Winker e Gilli [43] è 10. Il numero totale di iterazioni per un'esecuzione dell'algoritmo è $I = n_{rounds} \times n_{steps}$. Per ottenere i risultati migliori è consigliabile far processare l'algoritmo svariate volte per poi scegliere quale soluzione del problema la migliore soluzione di tutte le esecuzioni. Per decidere il numero di esecuzioni

dell'algoritmo, si fissa per prima cosa la qualità c della soluzione che vogliamo ottenere. Si ha che:

$$f_{sol} = \min_{r=1, \dots, n_{exec}} f(\Psi^{I,r}) \leq c \quad (2.15)$$

dove $f(\Psi^{I,r})$ rappresenta il valore della funzione obiettivo per la soluzione ottenuta nel r -esima esecuzione dell'algoritmo con I iterazioni. Si associa, successivamente, una variabile casuale bernoulliana z alla soluzione $f(\Psi)$ di una singola esecuzione dell'algoritmo, dove $z = 1$ se $f(\Psi^I) \leq c$ e $z = 0$ altrimenti. Si consideri poi una variabile casuale x che conta il numero di soluzioni all'interno dell'insieme di tutte le soluzioni (quindi di cardinalità n_{exec}) che soddisfano $f(\Psi^I) \leq c$. È noto che x è una variabile binominale $x \sim \mathcal{B}(n_{exec}, p)$ con $p = Prob(z = 1)$. La probabilità π di ottenere una soluzione tale che

$$f_{sol} \min_{r=1, \dots, n_{exec}} f(\Psi^{I,r}) \leq c \quad (2.16)$$

è data da

$$\pi = 1 - Prob(x = 0) \quad (2.17)$$

Siccome $x \sim \mathcal{B}(n_{exec}, p)$ abbiamo che $Prob(x = 0) = (1 - p)^{n_{exec}}$ e quindi per trovare il numero n_{exec} è sufficiente risolvere la seguente disuguaglianza:

$$\pi - 1 \leq (1 - p)^{n_{exec}} \quad (2.18)$$

Il seguente pseudo-codice riassume l'implementazione di *threshold accepting* con più esecuzioni:

Threshold accepting con più esecuzioni	
1.	Inizializza n_{exec} , n_{rounds} e n_{steps}
2.	Genera la successione di soglie T_r
3.	For $k = 1$ to n_{exec} do
4.	Genera a caso la soluzione corrente $x^c \in \mathcal{X}$
5.	For $r = 1$ to n_{rounds} do
6.	For $i = 1$ to n_{steps} do
7.	Genera $x^n \in \mathcal{X}(x^c)$ e calcola $\Delta = f(x^n) - f(x^c)$
8.	Se $\Delta < T_r$ allora $x^c = x^n$
9.	End for
10.	End for
11.	$\zeta_k = f(x^c)$, $x_{(k)}^{sol} = x^c$
12.	End for
13.	$x^{sol} = x_{(k)}^{sol}$ dove k è tale che $\zeta_k = \min\{\zeta_1, \dots, \zeta_{n_{exec}}\}$

Capitolo 3

Un modello matematico per *l'index tracking*

3.1 Il modello matematico

Gestione passiva ha come obiettivo la replica di un'indice di mercato. Questo obiettivo può essere realizzato detenendo in portafoglio tutte le attività finanziarie appartenenti a quell'indice di nella stessa proporzione oppure, in alternativa, tentando di replicare l'indice stesso con un numero inferiore di titoli utilizzando particolari tecniche di selezione del portafoglio per scegliere i titoli più appropriati. Questa strategia alternativa è comunemente denominata *index tracking*. L'attività di *index tracking* può essere tradotta in un problema di ottimizzazione vincolata. Il problema di ottimizzazione richiede la definizione di una funzione obiettivo. In questo caso, si tratta di scegliere una funzione che rappresenta in maniera appropriata il *tracking error*, cioè che misura la differenziale di rendimento tra il portafoglio costruito e l'indice di riferimento. Inoltre, bisogna determinare l'insieme dei vincoli: ad esempio la cardinalità k del portafoglio (con k minore degli N titoli che compongono l'indice di riferimento), i costi di transazione, i lotti minimi, vincoli sulla liquidità, ecc. Per una comprensione più immediata del problema in questione, in questa tesi si è scelto di ridurre al minimo le variabili, escludendo in tal modo elementi secondari quali ad esempio i costi di transazione ed i lotti minimi, concentrandosi quindi, sul vincolo più importante e difficile da trattare: vincolo di cardinalità del portafoglio.

In questo modello il tempo è discreto: $t = t_1 - 1, t_1, t_1 + 1, t_1 + 2, \dots, t_2$. L'intervallo temporale complessivo in esame è $[t_1, t_2]$, di ampiezza $T = t_2 - t_1$ unità di tempo (tipicamente settimane o giorni); il tempo $t = t_1 - 1$ serve da riferimento iniziale. Indichiamo con I_t il valore dell'indice al tempo t .

Supponiamo che l'indice abbia N componenti Y_i ($i = 1, \dots, N$), e sia $p_{i,t}$ il valore di una quota della componente Y_i al tempo t (in Euro, USD, yen, ...). Matematicamente quindi, i *dati* del problema sono:

- Una *matrice* $p = p_{i,t}$, con N colonne indicate in $i = 1, \dots, N$, e $t_2 - t_1 + 2$ righe indicate in $t = t_1 - 1, \dots, t_2$; possiamo indicare con p_t la riga di p di indice t (un vettore-riga N -dimensionale).
- Un *vettore-colonna* $I = I_t$ di dimensione $t_2 - t_1 + 2$ indicato in $t = t_1 - 1, \dots, t_2$.

3.1.1 Portafogli e rendimenti

Definizione 3.1.1 Un portafoglio è un vettore non-negativo $x \in \mathbb{R}_+^N$ la cui componente i -esima x_i indica il numero di quote della componente Y_i che sono inserite nel portafoglio.¹ Chiameremo *composizione di un portafoglio* x la N -pla binaria $s \in \{0, 1\}^N$ definita da $s_i = 1 \iff x_i > 0$. La *dimensione di un portafoglio* x è

$$n = \#\{i \mid x_i > 0\} = \sum_{i=1}^N s_i \quad (3.1)$$

Il valore di un portafoglio x al tempo t è allora

$$v_t = (x \mid p_t) = \sum_{i=1}^N x_i p_{i,t} = \sum_{\{i \mid s_i=1\}} x_i p_{i,t} \quad (3.2)$$

dove $(\cdot \mid \cdot)$ è il prodotto scalare euclideo. La *distribuzione di un portafoglio* x (al tempo t) è il vettore d_t definito da $d_{t,i} = x_i p_{i,t} / v_t$. Ovviamente, per ogni t , si ha $\sum_{i=1}^N d_{t,i} = 1$

Se il portafoglio x viene costituito al tempo $t = t_1$, il capitale necessario è evidentemente $C = v_{t_1} = \sum_{i=1}^N x_i p_{i,t_1}$. Poniamo convenzionalmente $C = 1$, in modo che la distribuzione del capitale in % fra le varie componenti coincida con il vettore $d_{t_1} = (x_i p_{i,t_1})_{i=1,N}$. Se \mathcal{P}_n è l'insieme dei portafogli con esattamente n componenti non-nulle, con $0 < \varepsilon = n/N < 1$, per individuare $x \in \mathcal{P}_n$ è sufficiente assegnare

- la *composizione* s , di dimensione N ,
- un vettore y , di dimensione n , indicante le quote, definito da $y_i = x_{k_i}$ essendo $k_1 < k_2 < \dots < k_n$ gli indici delle componenti uguali ad '1' in s .

¹ Indichiamo come d'uso con \mathbb{R}_+ l'insieme dei numeri reali $x > 0$.

Introduciamo ancora qualche notazione che ci consentirà di definire meglio il problema di cui ci occupiamo. Indichiamo con Σ_n^N l'insieme di cardinalità $\binom{N}{n}$ costituito dalle N -ple binarie con n valori '1' ed $N - n$ valori '0'. La composizione di un portafoglio $x \in \mathcal{P}_n$ è allora un'applicazione

$$\text{comp} : x \in \mathcal{P}_n \longmapsto s \in \Sigma_n^N$$

che coincide matematicamente con la funzione *segno*:

$$\text{comp}(x)_i = \text{sgn}(x)_i = \begin{cases} -1 & \text{se } x_i < 0 \\ 0 & \text{se } x_i = 0 \\ 1 & \text{se } x_i > 0 \end{cases}$$

Definizione 3.1.2 *Il rendimento unitemporale (rendimento lordo) del portafoglio x al tempo t ($t_1 \leq t \leq t_2$) è*

$$\mathcal{R}_t(x) = \log(v_t/v_{t-1}) \quad (3.3)$$

Analogamente si definisce il rendimento uniperiodale dell'indice al tempo t

$$\mathcal{Q}_t = \log(I_t/I_{t-1}) \quad (3.4)$$

Ricordiamo che $\mathcal{R}_t(x)$ è il tasso composto continuo al quale l'investimento di un capitale v_{t-1} all'epoca $t-1$ produrrebbe, dopo 1 unità di tempo, il valore finale $v_t = v_{t-1}e^{\mathcal{R}_t(x)}$. Per il calcolo di $\mathcal{R}_t(x)$ per ogni $t \in [t_1, t_2]$ è necessario non solo conoscere i prezzi per $t = t_1, t_1 + 1, t_1 + 2, \dots, t_2$, ma anche per $t = t_1 - 1$.

Osservazione 3.1.3 *Il rendimento uniperiodale $\mathcal{R}_t : \mathbb{R}_+^N \rightarrow \mathbb{R}$ è una funzione positivamente omogenea di grado 0, ossia per ogni $\lambda > 0$ riesce*

$$\mathcal{R}_t(\lambda x) = \mathcal{R}_t(x) \quad (3.5)$$

Questa ambiguità nella definizione dei portafogli può essere eliminata, per un portafoglio dato, in vari modi. In modo più vicino alla applicazione finanziaria, possiamo utilizzare il vincolo (bilaterale) di bilancio, ossia la assegnazione del valore totale C del portafoglio x all'epoca iniziale t_1 :

$$\sum_{i=1}^N x_i p_{i,t_1} = C \quad (3.6)$$

Dal punto di vista matematico-teorico possiamo restringerci alla sfera della norma del massimo,

$$\|x\|_\infty = \max_{i=1,N} x_i = 1 \quad (3.7)$$

ridefinendo x come $x/\|x\|_\infty$. Dal punto di vista della implementazione numerica invece conviene porre uguale a 1 l'ultima componente non-nulla

$$x_{k_n} = 1 \quad (3.8)$$

ridefinendo x come x/x_{k_n} , ed è questa la scelta che verrà usata in seguito.

3.1.2 Confronto dei rendimenti

Per confrontare il rendimento del portafoglio x con quello dell'indice occorre quindi valutare in qualche modo la “distanza” fra i vettori T -dimensionali $\mathcal{R}(x)$ e \mathcal{Q} . La stima della varianza campionaria \tilde{S}^2 dell'errore $\mathcal{R}(x) - \mathcal{Q}$ non è una buona scelta, in quanto se un portafoglio x costantemente sottoperforma l'indice si ha, per ogni t , che $\mathcal{R}_t(x) = \mathcal{Q}_t - \text{costante}$, e quindi la varianza è $\tilde{S}^2 = 0$. Una scelta ragionevole è una distanza classica in \mathbb{R}^T indotta da una norma α , o dalla norma del massimo. Scegliendo quest'ultima porremo:

$$\mathcal{E}(x) = \frac{1}{T} \max_{t \in [t_1, t_2]} |\mathcal{R}_t(x) - \mathcal{Q}_t| \quad (3.9)$$

Per una norma α , con $\alpha > 1$, avremmo invece:

$$\mathcal{E}(x) = \frac{1}{T} \left(\sum_{t=t_1}^{t_2} |\mathcal{R}_t(x) - \mathcal{Q}_t|^\alpha \right)^{1/\alpha} \quad (3.10)$$

La soluzione di un medesimo problema di ottimizzazione dipende dalla scelta della norma.² In questo lavoro la norma del massimo è preferibile per i due motivi seguenti:

- drastica riduzione dei tempi di calcolo rispetto alla norma α ,
- mantiene il rendimento del portafoglio *uniformemente* vicino a quello dell'indice (e non solo in α -media), consentendo ad esempio di “uscire” dall'investimento a condizioni indipendenti dal tempo.

Introduciamo un'ulteriore distanza $\mathcal{W}(x)$ che rappresenta la media delle deviazioni dei rendimenti del portafoglio costruito rispetto all'indice durante il periodo $[t_1, t_2]$. In termini tecnici questa quantità si chiama *excess return* ed è data da:

$$\mathcal{W}(x) = \frac{1}{T} \left(\sum_{t=t_1}^{t_2} (\mathcal{R}_t(x) - \mathcal{Q}_t) \right) \quad (3.11)$$

3.1.3 Formalizzazione del problema

Per formalizzare il problema di cui ci occupiamo, fissiamo i parametri N (cardinalità dell'indice) e n (cardinalità del portafoglio). Il problema è allora minimizzare il *trade-off* tra il *tracking error* e *excess return*:

$$\min_{x \in \mathcal{P}_n} \lambda \mathcal{E}(x) - (1 - \lambda) \mathcal{W}(x) \quad (3.12)$$

² Ad esempio, se l'istanza è trovare quale dei due punti $A = (1.3, 1.3, 1.3)$, $B = (1.0, 1.5, 1.1)$ è più vicino a $U = (1, 1, 1)$, nella norma euclidea ($\alpha = 2$) il più vicino è B , ma in quella del massimo il più vicino è A .

con $\lambda \in [0, 1]$ In questo lavoro, noi consideriamo solamente il caso in cui $\lambda = 1$ e quindi cerchiamo semplicemente di effettuare un *tracking* puro dell'indice senza voler ottenere un extra rendimento.

Il problema diventa allora

$$\min_{x \in \mathcal{P}_n} \mathcal{E}(x) \quad (3.13)$$

che, introducendo la composizione s , si può scomporre in due stadi:

$$\min_{s \in \Sigma_n^N} \min_{\{x \mid \text{comp}(x)=s\}} \mathcal{E}(x) \quad (3.14)$$

Il problema è quindi di tipo misto: una ottimizzazione di tipo combinatorico, su un insieme finito di cardinalità $\binom{N}{n}$, ed una di tipo classico “statico” di una funzione $\mathbb{R}_+^n \rightarrow \mathbb{R}$, con vincoli di disuguaglianza stretta $x_i > 0$, dove, usando la (3.5), si può abbassare ad $n - 1$ la dimensione dato che una delle variabili può essere (e conviene che sia) posta = 1.

Esistenza. La esistenza di soluzione non è assicurata a priori, in quanto, pur essendo la funzione \mathcal{E} continua, non è definita su un insieme compatto (per via delle disuguaglianze strette). Se invece si pongono vincoli di disuguaglianza debole $x_i \geq 0$, questi, assieme al vincolo (di uguaglianza) di bilancio o alla normalizzazione $\max(x_i) = 1$, definiscono un sottoinsieme compatto, per cui lo stadio classico ha l'esistenza assicurata a norma del teorema di Weierstrass,³ (si veda Giusti, [47]). Dato che però in questo caso la soluzione può essere sul bordo, la composizione della soluzione può non essere la s assegnata, ma una $s' < s$ (dove, fra composizioni r ed s , la disuguaglianza $r \leq s$ significa che $r_i = 1 \Rightarrow s_i = 1$). È quindi assicurata la esistenza per

$$\min_{s \in \Sigma_n^N} \min_{\{x \mid \text{comp}(x) \leq s\}} \mathcal{E}(x) \quad (3.15)$$

Unicità. I teoremi classici non assicurano a priori la unicità di soluzione, neppure nello stadio classico, in quanto tipicamente la funzione \mathcal{E} non risulta convessa. Dal punto di vista della applicazione finanziaria questa non-unicità è ovvia, in quanto se l'indice ha due componenti con serie storiche dei rendimenti e le varianze uguali, un portafoglio ottimo può indifferentemente selezionare l'una o l'altra.

³Teorema di Weierstrass ci dice che se $f : K \rightarrow \mathbb{R}$ sia una funzione continua ed K sia un insieme chiuso e limitato, allora f ha massimo e minimo assoluti.

3.1.4 Un esempio numerico con un indice artificiale

Esemplifichiamo con $N = 3$, $n = 2$ ai fini di comprendere meglio il problema. Avremo $\Sigma_n^N = \Sigma_2^3 = \{(1, 1, 0), (0, 1, 1), (1, 0, 1)\}$. Supponiamo che siano assegnati i dati seguenti L'indice $\text{Index} = I$ è costruito ponendo (artificialmente)

t	Y_1	Y_2	Y_3	Index
0	36.8	55.3	27.3	38.075
1	37.8	57.1	28.1	39.225
2	39.0	62.7	29.1	41.655
3	38.6	61.9	28.6	41.090
4	39.2	65.2	28.9	42.365
5	38.7	67.6	29.4	43.185

Tabella 3.1: Esempio numerico con un indice artificiale

$$I = 0.25 Y_1 + 0.30 Y_2 + 0.45 Y_3$$

I tempi iniziale e finale sono rispettivamente $t_1 = 1$ e $t_2 = 5$. I dati della prima riga servono per il calcolo dei rendimenti al tempo iniziale. I portafogli $x \in \mathcal{P}_2$ sono gli elementi delle tre facce dell'ottante positivo di \mathbb{R}^3 (senza gli spigoli, dove sarebbe $n = 1$). Si ha

$$v_t = x_1 p_{1,t} + x_2 p_{2,t} + x_3 p_{3,t} \quad (3.16)$$

con uno dei tre addendi uguale a zero, e

$$\mathcal{R}_t(x) = \log \left(\frac{x_1 p_{1,t} + x_2 p_{2,t} + x_3 p_{3,t}}{x_1 p_{1,t-1} + x_2 p_{2,t-1} + x_3 p_{3,t-1}} \right) \quad (3.17)$$

Dobbiamo considerare 3 portafogli corrispondenti alle 3 possibili composizioni.

Composizione $s = (1, 1, 0)$. Il portafoglio con questa composizione, normalizzato, è del tipo $x = (1, x_2, 0)$, per il quale quindi $\mathcal{E}(x)$ diventa funzione del solo x_2 , e precisamente:

$$\mathcal{E}(x) = \frac{1}{5} \max \left\{ \left| \log \frac{37.8 + 57.1 x_2}{36.8 + 55.3 x_2} - 0.0297 \right|, \left| \log \frac{39.0 + 62.7 x_2}{37.8 + 57.1 x_2} - 0.0601 \right|, \left| \log \frac{38.6 + 61.9 x_2}{39.0 + 62.7 x_2} + 0.01366 \right|, \left| \log \frac{39.2 + 65.2 x_2}{38.6 + 61.9 x_2} - 0.0305 \right|, \left| \log \frac{38.7 + 67.6 x_2}{39.2 + 65.2 x_2} - 0.0192 \right| \right\}$$

Usando un algoritmo standard di minimizzazione per funzioni di una sola variabile si ottiene $\min \mathcal{E}(x) = 0.000955614$ per $x = (1, 0.7343, 0)$. Confrontiamo il rendimento dell'Index con quello del portafoglio selezionato: La distanza

t	1	2	3	4	5
Index	0.029756	0.0601071	-0.0136566	0.0305578	0.0191707
$Y_1 + 0.7343 Y_2$	0.029553	0.0645014	-0.0116793	0.0353358	0.0143926

Tabella 3.2: Esempio del confronto di rendimento del portafoglio con quello dell'indice

massima si ha per $t = 4$ quando appunto si ha $\frac{1}{5}|0.0353358 - 0.0305578| = 0.000955614$. Si osservi che la funzione $\mathcal{E}(x)$ da minimizzare non è differenziabile né convessa.

Composizione $s = (0, 1, 1)$. Questo caso è del tutto simile al precedente, e si ha $\min \mathcal{E}(x) = 0.0010051$ per $x = (0, 0.2587, 1)$.

Composizione $s = (1, 0, 1)$. In questo caso possiamo cercare un portafoglio normalizzato del tipo $x = (x_1, 0, 1)$ (ossia decidendo di comprare solo Y_1 e Y_3 , ed in particolare x_1 quote di Y_1 e $x_3 = 1$ quota di Y_3). Si verifica però un fenomeno inatteso: la funzione $\mathcal{E}(x) = \mathcal{E}(x_1)$ è una funzione crescente su $x_1 \geq 0$, per cui anche il possesso di una quota piccolissima di Y_1 peggiora la distanza dall'indice. Formalmente, la funzione $\mathcal{E}(x) = \mathcal{E}(x_1)$ non ha minimo col vincolo $x_1 > 0$, mentre con $x_1 \geq 0$ si raggiunge il minimo $\min \mathcal{E}(x) = 0.0050277$ in $x_1 = 0$, ossia va ridotto il portafoglio all'acquisto di solo $n = 1$ componente, ossia la Y_3 .

La Figura 3.1 illustra i grafici di $\mathcal{E}(x)$ nei tre casi analizzati.

Quindi, a conclusione di questo esempio, la soluzione ottima è il portafoglio $x = (1, 0.7343, 0)$, quello relativo alla composizione $s = (1, 1, 0)$. Moltiplicando

s	(1, 1, 0)	(0, 1, 1)	(1, 0, 1)
$\mathcal{E}(x)$	0.0009556	0.0010051	> 0.0050277

Tabella 3.3: Il valore della funzione obiettivo nei tre casi ipotizzati

componente per componente il portafoglio ottimo $x = (1, 0.7343, 0)$ per i prezzi $p_1 = (37.8, 57.1, 28.1)$ all'epoca iniziale $t_1 = 1$ otteniamo $x p_1 = (37.8, 41.9285, 0)$, che rappresenta la valuta investita, per un capitale totale investito di $C =$

79.7285. In percentuale, la distribuzione del capitale è $d_1 = (47.4\%, 52.6\%, 0.00\%)$.

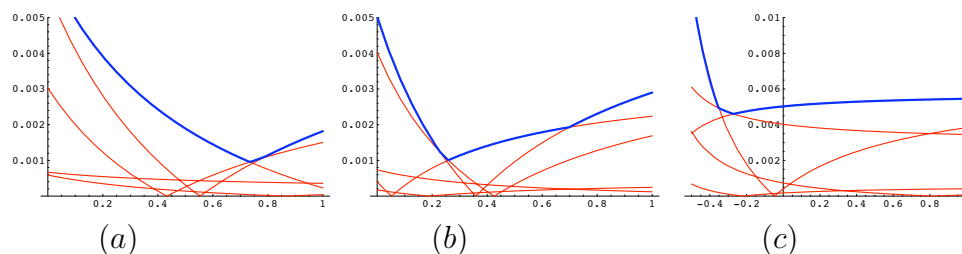


Figura 3.1: Le curve tracciate con linea grossa (—) rappresentano il grafico di $\mathcal{E}(x)$ nei tre casi analizzati: (a) $s = (1, 1, 0)$, portafoglio ottimo: $x = (1, 0.7343, 0)$, $\mathcal{E}(x) = 0.0009556$; (b) $s = (0, 1, 1)$, portafoglio ottimo: $x = (0, 0.2587, 1)$, $\mathcal{E}(x) = 0.0010051$; (c) $s = (1, 0, 1)$, portafoglio ottimo: $x = (0, 0, 1)$, $\mathcal{E}(x) = 0.0050277$. I grafici tracciate con linea sottile (—) in ogni figura sono i cinque rendimenti uniperiodali.

Osservazione 3.1.4 *Non sempre è meglio aggiungere un titolo in più nella costruzione del portafoglio.* L'esempio precedente relativo alla composizione $s = (1, 0, 1)$ contrasta apertamente la credenza comunemente accettata in finanza che il portafoglio costruito con più titoli è più diversificato rispetto al portafoglio costruito con un numero minore di titoli. Infatti in questo caso, quel che accade è che il minimo libero (non vincolato da $x_1 > 0$) di $\mathcal{E}(x_1)$, è pari a 0.00460247, e si raggiunge per $x_1 = -0.271043$, ossia per un numero *negativo* di quote di Y_1 , impossibile nel nostro modello che non ammette la vendita allo scoperto.⁴

Il controesempio trovato non stupisce se si riflette sulla analogia della rappresentazione dei colori in un computer con il sistema RGB. Nei display dei computer si “dipinge” un singolo pixel assegnando tre numeri R, G, B, che indicano approssimativamente la intensità desiderata di luce dei tre colori primari: rosso (Red), verde (Green) e blu (Blue). Nel sistema RGB si riproduce un colore con combinazioni dei tre colori primari (similmente a come noi vogliamo riprodurre un indice con una combinazione di solo alcuni dei suoi titoli). È però ben noto che il sistema RGB non può riprodurre tutti i colori dello spettro della luce visibile. Per esempio il ciano (Cyan) è generato in RGB come sintesi additiva di 0% di rosso, 100% di verde, 100% di blu (codice html: #00ffff).

⁴ Sarebbe naturalmente possibile partire con un portafoglio che *venda* allo scoperto $-x_1 = 0.271043$ quote di Y_1 e *compri* $x_3 = 1$ quote di Y_3 , ma non è questa la strategia tipica di un investimento passivo.

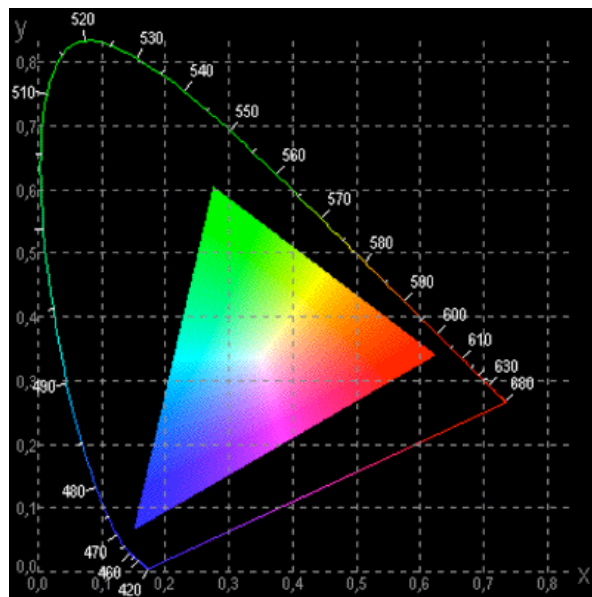


Figura 3.2: Il triangolo di Maxwell. I colori primari R,G,B sono ai vertici, il bianco è al centro, ed è la somma dei tre colori primari in equal proporzione (codice html: `#ffffff`). Il ciano è sul bordo del triangolo a sinistra, per raggiungere il ciano spettrale servirebbe “uscire” dal triangolo con una quantità negativa di rosso. Immagine: <http://www.efg2.com/Lab/Graphics/Colors/Chromaticity.htm>

Tuttavia, in questo “portafoglio” di colori primari l'osservatore umano riconosce il ciano come tinta, ma non come saturazione: il ciano spettrale (ossia la luce pura di lunghezza d'onda 485–500 nm) è diverso. L'unico modo di ottenere il ciano spettrale sarebbe di *togliere* del rosso, che è impossibile. La Figura 3.2 rappresenta il triangolo di Maxwell, ossia i colori riproducibili con combinazione dei colori primari.

3.1.5 Complessità computazionale dello stadio combinatorio

Abbiamo analizzato lo stadio di ottimizzazione classica. Lo stadio combinatorio non comporta problemi di esistenza, dato che si tratta del minimo di un insieme finito (ci possono essere ovviamente più punti di minimo). Ci sono però problemi di complessità computazionale, relativi alla cardinalità $\binom{N}{n}$ delle composizioni. Poniamo $0 < \varepsilon = n/N < 1$. Approssimando il fattoriale con la formula di Stirling,⁵ (Sgarro, [74]) otteniamo

$$\binom{N}{n} \approx \frac{1}{\sqrt{2\pi}} \left(N^{\frac{1}{2}+N} (\varepsilon N)^{-\frac{1}{2}-\varepsilon N} ((1-\varepsilon)N)^{-\frac{1}{2}-(1-\varepsilon)N} \right) = \frac{1}{\sqrt{2\pi\varepsilon(1-\varepsilon)N}} 2^{Nh(\varepsilon)} \quad (3.18)$$

dove $h(\varepsilon) = -(\varepsilon \log_2(\varepsilon) + (1-\varepsilon) \log_2(1-\varepsilon)) > 0$ è l'entropia. Quindi, per ε fissato, $\binom{N}{n}$ cresce esponenzialmente⁶ con N , con velocità di crescita massima per $\varepsilon = 1/2 = \operatorname{argmax}_{0 < \varepsilon < 1} h(\varepsilon)$. È quindi naturale e ragionevole ricercare un portafoglio con componenti non-nulle che costituiscano una “piccola” percentuale delle componenti dell'indice, in modo che il coefficiente $h(\varepsilon)$ sia “basso”, ed eventualmente valutare come ottimizzare la scelta della cardinalità n del portafoglio.

3.1.6 Necessità di un'euristica

Consideriamo ad esempio il caso concreto dell'indice S&P 100, con $N = 100$, e supponiamo di volerlo replicare con un portafoglio di $n = 10$ titoli, sui dati di 52 settimane. Da quanto abbiamo visto, un algoritmo *deterministico* prevede

⁵In matematica l'approssimazione di Stirling o formula di Stirling è un'approssimazione per fattoriali grandi. Deve il suo nome al matematico scozzese James Stirling (1692-1770).

La formulazione corretta è: $\lim_{n \rightarrow +\infty} \frac{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n}{n!} = 1$ che viene scritto spesso come: $n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$. Per valori elevati di n il secondo membro della formula fornisce una buona approssimazione di $n!$ che si può calcolare rapidamente e facilmente. Ad esempio la formula per $30!$ fornisce l'approssimazione $2,6452 \times 10^{32}$, mentre un valore più preciso è $2,6525 \times 10^{32}$; in questo caso si ha una discrepanza minore dello 0,3%.

⁶ Ad esempio, $\binom{100}{5} = 7.52875 \times 10^7$, e $\binom{100}{10} = 1.73103 \times 10^{13}$.

l'esecuzione di $\binom{100}{10} = 1.73103 \times 10^{13}$ minimizzazioni (vincolate) di funzioni di 9 variabili, praticamente impossibile (volendo contenere l'esecuzione entro un mese di tempo-macchina occorrerebbe eseguire ogni ottimizzazione statica in $0.15\mu s$). Conseguentemente si rende necessario utilizzare un algoritmo di ottimizzazione *euristico*.

3.1.7 Ottimizzazione mista

Resta possibile, una volta che l'euristica abbia indicato un portafoglio ottimo x^{opt} , utilizzare dalla euristica la informazione relativa alla composizione $s = \text{comp}(x^{\text{opt}})$ del portafoglio ottimo (ossia la informazione su *quali* componenti comprare) per costruire la funzione obiettivo $\mathcal{E}(x)$ definita sui soli portafogli x tali che $\text{comp}(x) = s$. Si può poi utilizzare dalla euristica la informazione relativa alla distribuzione del portafoglio ottimo (ossia la informazione su *quante* quote delle componenti comprare) per definire le condizioni iniziali di un metodo iterativo numerico classico di minimizzazione.

Sia infatti $x^{\text{opt}} \in \mathbb{R}_+^N$ il portafoglio ottimo generato dall'algoritmo di ottimizzazione euristico. Sia n la sua dimensione e siano $k_1 < k_2 < \dots < k_n$ gli indici delle sue componenti non-nulle. Il vettore $y^{\text{opt}} \in \mathbb{R}_+^n$ che definisce le quote è quindi definito da $y_i^{\text{opt}} = x_{k_i}^{\text{opt}}$. Consideriamo la matrice P dei prezzi relativi alle sole componenti selezionate definita da

$$P_{ti} = p_{k_i,t}$$

per $t_1 - 1 \leq t \leq t_2$ e per $1 \leq i \leq n$. È agevole allora definire la funzione da minimizzare come

$$f(y) = \frac{1}{T} \sum_{t=t_1}^{t_2} |\log(P_t|y) - \log(P_{t-1}|y) - Q_t|$$

Non è però conveniente applicare un metodo di minimizzazione classico, a causa della (3.5). Conviene prima normalizzare dividendo ad esempio per y_n (che si ritiene diverso da 0). Si ridefinisce allora l'obiettivo ponendo

$$F(y_1, y_2, \dots, y_{n-1}) = f(y_1, y_2, \dots, y_{n-1}, 1)$$

e si utilizza un metodo iterativo numerico classico di minimizzazione locale fra i molti disponibili nei software (Gradiente, Gradiente coniugato, Levenberg-Marquardt, Newton, Quasi-Newton, ecc.) per minimizzare $F : \mathbb{R}^{n-1} \rightarrow \mathbb{R}$ con la condizioni iniziale $y = (y_i^{\text{opt}})_{i=1, n-1}$.

Come si vedrà nel Capitolo successivo, questa tecnica mista, proposta per la prima volta in questo lavoro, consente miglioramenti interessanti.

3.2 Algoritmo

3.2.1 Stadio euristico: threshold accepting

Possiamo allora presentare lo pseudo-codice dell'algoritmo di threshold accepting che verrà utilizzato per lo stadio di minimizzazione euristica.

Threshold accepting	
1.	Inizializza $n_{\text{exec}}, n_{\text{rounds}}, n_{\text{steps}}$
2.	Generazione della sequenza dei threshold
2.1.	for $k = 1, K$ do
2.2.	Scegli a caso $x^1, x^2 \in \mathcal{X}$
2.3.	Calcola $\Delta_k = f(x^2) - f(x^1) $
2.4.	end for
2.5.	Calcola la funzione empirica di distribuzione F di Δ
2.6.	for $r = 1, n_{\text{rounds}} - 1$ do
2.7.	Calcola il quantile τ_r di ordine $(n_{\text{rounds}} - r)/n_{\text{rounds}}$
2.8.	end for
2.9.	Definisci $\tau_{\text{rounds}} \leftarrow 0$
3.	for $n = 1, n_{\text{exec}}$ do
4.	Esecuzione dei rounds
4.1.	for $r = 1, n_{\text{rounds}}$ do
4.2.	Inizializza a caso x^c , calcola $F^c = f(x^c)$
4.3.	for $n = 1, n_{\text{steps}}$ do
4.4.	Scegli random $x^n \in \mathcal{N}(x^c)$
4.5.	Calcola $F^n \leftarrow f(x^n)$
4.6.	if $F^n < F^c + \tau_r$ then
4.7.	$x^c \leftarrow x^n$
4.8.	$F^c \leftarrow F^n$
4.9.	end if
4.10.	end for
4.11.	end for
5.	Memorizza esecuzione $\xi_n \leftarrow x^c, \Phi_n \leftarrow F^c$
6.	end for
7.	Scegli n^* t.c. $\Phi_{n^*} = \min\{\Phi_n \mid n = 1, \dots, n_{\text{exec}}\}$
8.	output ξ_{n^*} (portafoglio tracking)

3.2.2 Stadio classico: metodo di Newton

Come algoritmo classico “statico” di minimizzazione locale scegliamo ad esempio un metodo iterativo del tipo di Newton.

Minimizzazione locale	
1.	Definisci la soluzione iniziale $y = (y_i^{\text{opt}})_{i=1, n-1}$ (sez. 3.1.7)
2.	Definisci l'obiettivo $F : \mathbb{R}^{n-1} \rightarrow \mathbb{R}$
3.	Esegui la minimizzazione locale
3.1.	while i criteri di stop non si verificano do
3.2.	aggiorna la soluzione y
3.3.	end while
4.	output y (portafoglio tracking), $F(y)$ (minimo raggiunto)

3.2.3 Definizione degli intorni

Nello stadio di minimizzazione euristica, è cruciale la definizione di intorno $\mathcal{N}(x^c)$ della soluzione corrente. Per appoggiarci alla intuizione geometrica, proseguiamo con l'esempio di dimensione bassa, $N = 3$, $n = 2$. La intuizione geometrica ci consente allora di individuare la nozione di ϵ -intorno $\mathcal{N}(x)$ di un portafoglio x . La unità di misura di ϵ è quella della valuta (euro, ...). Prendiamo ad esempio $x = (x_1, x_2, 0)$: se questo vettore, che appartiene alla faccia $x_3 = 0$, è di ϵ "dentro" alla faccia, ossia se $\min\{x_1 p_{1,t}, x_2 p_{2,t}\} > \epsilon$, allora il suo ϵ -intorno è costituito dai portafogli

$$x' = (x_1 - \Delta/p_{1,t}, x_2 + \Delta/p_{2,t}, 0) \quad (3.19)$$

con $-\epsilon < \Delta < \epsilon$. In questo caso sostanzialmente effettuiamo uno switch di Δ unità di valuta da Y_1 a Y_2 . Se invece fosse ad esempio $x_1 p_{1,t} \leq \epsilon$, saremmo "vicini" alla frontiera fra la faccia $x_3 = 0$ e la $x_1 = 0$, per cui l' ϵ -intorno di x sarebbe costituito in questo caso *anche* dai portafogli

$$x' = (0, x_2, x_1 p_{1,t}/p_{3,t}) \quad (3.20)$$

nel questo caso sostanzialmente abbiamo venduto tutte le x_1 quote possedute di Y_1 , realizzando $x_1 p_{1,t}$, per comprare $x_3 = x_1 p_{1,t}/p_{3,t}$ quote di Y_3 . Si noti come questa definizione di intorni assicuri che ad ogni istante t di tempo i portafogli appartenenti ad uno stesso intorno di x conservano lo stesso valore che x ha al tempo t .

3.2.4 Implementazione in R

Perché R

R è un software specializzato nell'analisi statistica dei dati. R è un ambiente statistico scaricabile gratuitamente da Internet sul sito di *The R Project for Statistical Computing* il cui indirizzo Internet è *www.r-project.org*. Esso è il

frutto del lavoro collettivo svolto da un gruppo, sempre più folto, di ricercatori in campo statistico ed informatico a livello mondiale. R più che un software statistico può essere definito come un ambiente, costituito da una varietà di strumenti, orientato alla gestione, all'analisi dei dati e alla produzione di grafici, basato sul linguaggio S creato da AT&T Bell Laboratories. Il pacchetto R è essenzialmente il clone *freeware* del pacchetto commerciale S-plus e inoltre, a differenza di MSEXcel, ha delle potenzialità di programmazione enormi, associate tuttavia ad una difficoltà di utilizzo dato che è completamente basato sull'uso di comandi non grafici (*command line*). Le sue caratteristiche principali possono essere così descritte:

- semplicità nella gestione e manipolazione dei dati
- disponibilità di una suite di strumenti per calcoli su vettori, matrici ed altre operazioni complesse
- accesso ad un vasto insieme di strumenti integrati per l'analisi statistica
- produzione di numerose potenzialità grafiche particolarmente flessibili
- possibilità di adoperare un vero e proprio linguaggio di programmazione orientato ad oggetti che consente l'uso di strutture condizionali e cicliche, nonché di funzioni create dall'utente

Codice

```
#--- Importa i dati -----
rm(list=ls())
indexname <- "< inserisci nome indice >"
setwd("< inserisci working directory >")
data <- read.table("nomefile.txt", sep="\t", dec=".", fill=TRUE, header=TRUE)

nrow <- dim(data)[1] # numero righe
ncol <- dim(data)[2] # numero colonne
#--- salva names
saved.names <- names(data)
#--- rimuovi gli attributi di names
names(data) <- NULL
m <- matrix(as.numeric(as.matrix(data)), nrow, ncol)
w <- m[,1]          # time (weeks)
I <- m[,ncol]      # index
#--- trova le componenti con NA (intervalli)
no.nas <- c()
for (j in 2:(ncol-1)) {#
```

```

nas <- which(is.na(m[,j]));
if (length(nas)!=0)
  print(
    paste("warning: component",j,"= NA on [",min(nas),",",max(nas),"]")
    else no.nas<-c(no.nas,j)
  )#
#--- cancella componenti con NA -----
p <- m[,no.nas]          # prezzi delle componenti
N <- length(no.nas)     # cardinalita' delle componenti senza NA
#===== Parametri =====
kk      <- 1500
lambda  <- 1
nsteps  <- 1000
nrounds <- 10
nexec   <- 500
n       <- 5
lambda  <- 1
t1      <- 2
t2      <- 53
c0      <- 100
#===== Generazione della threshold sequence =====
cpu0 <- proc.time()[3]
#--- calcolo del rendimento dell'indice
rI <- log(I[t1:t2]/I[t1:t2-1])
D <- rep(NA,kk)
for (k in (1:kk)){#
x <- matrix(rep(0,2*N),N,2)
s <- matrix(c(sample(N,n),sample(N,n)),n,2)
r <- matrix(runif(2*n),n,2)
x[s[,1],1] <- c0*(r[,1]/margin.table(r,2)[1])/p[1,s[,1]]
x[s[,2],2] <- c0*(r[,2]/margin.table(r,2)[2])/p[1,s[,2]]
F <- c(0,0)
for (j in 1:2) {#
v <- rep(0,t2-t1+1);
for (t in (t1-1):t2) {v[t] <- x[s[,j],j] %% p[t,s[,j]]};
rP <- log(v[t1:t2]/v[t1:t2-1]);
E <- max(abs(rP-rI))/(t2-t1);
R <- sum(rP-rI)/(t2-t1);
F[j] <- lambda*E - (1-lambda)*R};
D[k] <- abs(F[1]-F[2])}
quant <- round(quantile(D,((nrounds-1):1)/nrounds),5)
tau <- c(as.vector(quant),0)
cpu1 <- proc.time()[3]

```

```

print(paste("Calcolata la threshold sequence in ",cpu1-cpu0, "sec."),
      quote=FALSE)
#=== Minimizzazione euristica =====
#--- opzioni grafiche -----
par(mfrow=c(2,3))
curve(ecdf(D)(x),-0.0001,0.001,col="blue",
      main="Threshold sequence",xlab="",ylab="");
points(tau,ecdf(D)(tau),pch=19,col="red");
#-----
optF <- rep(NA,nexec)
optx <- matrix(NA,N,nexec)
cpu2 <- proc.time()[3]
#-----
temp1 <- proc.time()[3];
for (r in (1:nexec)) {# OPEN1
if (r==2) {print(paste("Completamento atteso entro:",
  nexec*ceiling((proc.time()[3]-temp1)/60),"min"),quote=FALSE)};
for (j in (1:nrounds)){# OPEN2
#---- genera a caso la soluzione corrente xc, calcola f(xc) -----
xc <- rep(0,N)
s <- sample(N,n)
q0 <- c0/n
xc[s] <- q0/p[t1,s]
vc <- rep(0,t2-t1+1)
for (t in (t1-1):t2) {vc[t] <- xc[s] %*% p[t,s]};
rPc <- log(vc[t1:t2]/vc[t1:t2-1]);
Ec <- max(abs(rPc-rI))/(t2-t1);
Rc <- sum(rPc-rI)/(t2-t1);
Fc <- lambda*Ec - (1-lambda)*Rc;
for (i in (1:nsteps)) {# OPEN3
#--- genera xn in un intorno di xc -----
xn <- xc
nz <- sort(s[sample(n,2)]);
z <- which(xc==0)[sample(N-n,1)];
vz <- xc[nz]*p[1,nz];
sw <- runif(1,.25,.75)*q0;
if (sw < vz[1]) {
  xn[nz[1]] <- (vz[1]-sw)/p[1,nz[1]];
  xn[nz[2]] <- (vz[2]+sw)/p[1,nz[2]]} else {
  xn[nz[1]] <- 0;
  xn[z] <- vz[1]/p[1,z]};
#--- calcola f(xn) -----
vn <- rep(0,t2-t1+1);

```

```

for (t in (t1-1):t2) {vn[t] <- xn[s] %*% p[t,s]};
rPn <- log(vn[t1:t2]/vn[t1:t2-1]);
En <- max(abs(rPn-rI))/(t2-t1);
Rn <- sum(rPn-rI)/(t2-t1);
Fn <- lambda*En - (1-lambda)*Rn;
#--- aggiorna -----
if (Fn < Fc + tau[j]) {xc <- xn; Fc <- Fn; s <- which(xc!=0)}
} # CLOSE3
} # CLOSE2
#--- output -----
optx[,r] <- xc
optF[r] <- Fc
} # CLOSE1
cpu3 <- proc.time()[3]
#-----
v <- rep(0,t2-t1+1)
u <- which(optF==min(optF))[1]
Fo <- optF[u]
xo <- optx[,u]
xe <- xo
s <- which(xo!=0)
s0 <- s # copia di s
#=== Minimizzazione locale classica =====
y <- xo[s]/xo[s][n]
yr <- y[1:(n-1)]
fr <- function(x){
lpx <- log(p[(t1-1):t2,s] %*% c(x,1));
max(abs(lpx[t1:t2]-lpx[t1:t2-1]-rI))}
locmin <- nlm(fr,yr)
Fo <- locmin$minimum/(t2-t1)
xo[s] <- c(locmin$estimate,1)
zoneg <- which(xo<0)
if (length(zoneg) > 1) {
xo[zoneg] <- 0 #--- azzero i negativi
s <- which(xo!=0) #--- ridefinisco s
print(paste(c("componenti negative",saved.names[no.nas[zoneg]])) }
#=====
#--- plot -----
hist(optF,col="yellow",breaks="Sturges",
main=paste("min F =",round(Fo,6)),xlab="min F",ylab="Frequenza")
plot(xo*p[t1,]/sum(xo*p[t1,]),type="h",
main="Distribuzione (%)",xlab="componenti",ylab="",ylim=c(0,1))
for (t in 1:(t2-t1+1)) {v[t] <- xo[s] %*% p[t1+t-1,s]}

```

```

track <- matrix(c(I[t1:t2]/I[t1],v/v[1]),t2-t1+1,2)
matplot(track, main="Tracking (in-sample)", xlab="time",ylab="",
        pch = 1:2, type = "o", col = c("gray","red"))
#-----
t3 <- t2;
out <- 26
t4 <- t3+out;
rI.out <- log(I[t3:t4]/I[t3:t4-1]) # length = out+1
v.out <- rep(0,t4-t3+1) # length = out+1
for (t in 1:(t4-t3+1)) {v.out[t] <- xo[s] %*% p[t3+t-1,s]}
shadow <- matrix(c(I[t3:t4]/I[t3],v.out/v.out[1]),t4-t3+1,2)
matplot(shadow, main="Out-of-sample",xlab="time",ylab="",
        pch = 1:2, type = "o", col = c("gray","red"))
pie.s <- round(xo[s]*p[t1,s]/c0,4)
names(pie.s) <- saved.names[no.nas[s]]
pie(pie.s,main=indexname)
#-----
print(indexname,quote=FALSE)
data.frame(parameters=c("restarts","rounds","steps","ntfolio"),
        values=c(nexec,nrounds,nsteps,n))
print(paste("CPU: threshold:",cpu1-cpu0, "sec; ",nexec,"esecuzioni:",
        round((cpu3-cpu2)/60),"min"),quote=FALSE);
print("Dopo lo stadio euristico")
data.frame(numero=s0,componente=saved.names[no.nas[s0]],
        pesi=round(xe[s0]*p[t1,s0]/sum(xe[s0]*p[t1,s0]),4))
print("Dopo la minimizzazione locale")
data.frame(numero=s,componente=saved.names[no.nas[s]],
        pesi=round(xo[s]*p[t1,s]/sum(xo[s]*p[t1,s]),4))
#-----

```

3.2.5 Output

Per illustrare l'output del programma, esemplifichiamo con l'indice Hang Seng nel periodo da $t_1 = 06/01/2006$ a $t_2 = 29/12/2006$ (un anno) per il *tracking (in-sample)*, e nel periodo da $t_3 = t_2$ a $t_4 = 16/11/2007$ per la simulazione (*out-of-sample*). Le componenti contenenti dell'indice che non erano sempre presenti nell'indice durante il periodo di *tracking*, nel codice definite come NA, sono state eliminate dal data base dell'indice per semplicità. Nel *tracking* del Hang Seng ci sono due titoli con queste caratteristiche: ICBC (ticker Bloomberg 1398 HK) e Bank of China (ticker Bloomberg 3988 HK). Fissiamo in questo esempio $n = 7$.

Output testuale

L'output testuale del programma comprende principalmente:

- La verifica della conclusione del calcolo dei thresholds:
[1] Calcolata la threshold sequence in 4.13 sec.
- Una previsione di massima del tempo di esecuzione:
[1] Completamento atteso entro: 98 min
- La conferma dell'indice in esame, ad esempio:
[1] Hang Seng 40
- La stampa dei parametri utilizzati:

```

parameters values
1  restarts    500
2   rounds     10
3    steps   1000
4   ntfolio     7

```

- Il riassunto del tempo effettivo di esecuzione:
[1] CPU: threshold: 4.13 sec; 500 esecuzioni: 98 min
- La composizione e la distribuzione del portafoglio ottimo dopo entrambi gli stadi di ottimizzazione:

```

numero componente  pesi
1      1      X1.HK 0.0424
2     12      X2.HK 0.0850
3     20     X293.HK 0.0409
4     25     X388.HK 0.0231
5     26      X4.HK 0.1616
6     28      X5.HK 0.4404
7     38     X941.HK 0.2066

```

Output grafico

L'output grafico consiste in una finestra con 6 immagini (vedi Figura 3.3):

- Grafico della cdf (funzione empirica di distribuzione) delle distanze della funzione obiettivo `abs(F[1]-F[2])` per i portafogli scelti in maniera casuale per un totale di `kk` volte. Le soglie `tau` possono essere calcolate come quantili che corrispondono ai percentili equidistanti `quant <- round(quantile(D,((nrounds-1):1)/nrounds),5)`

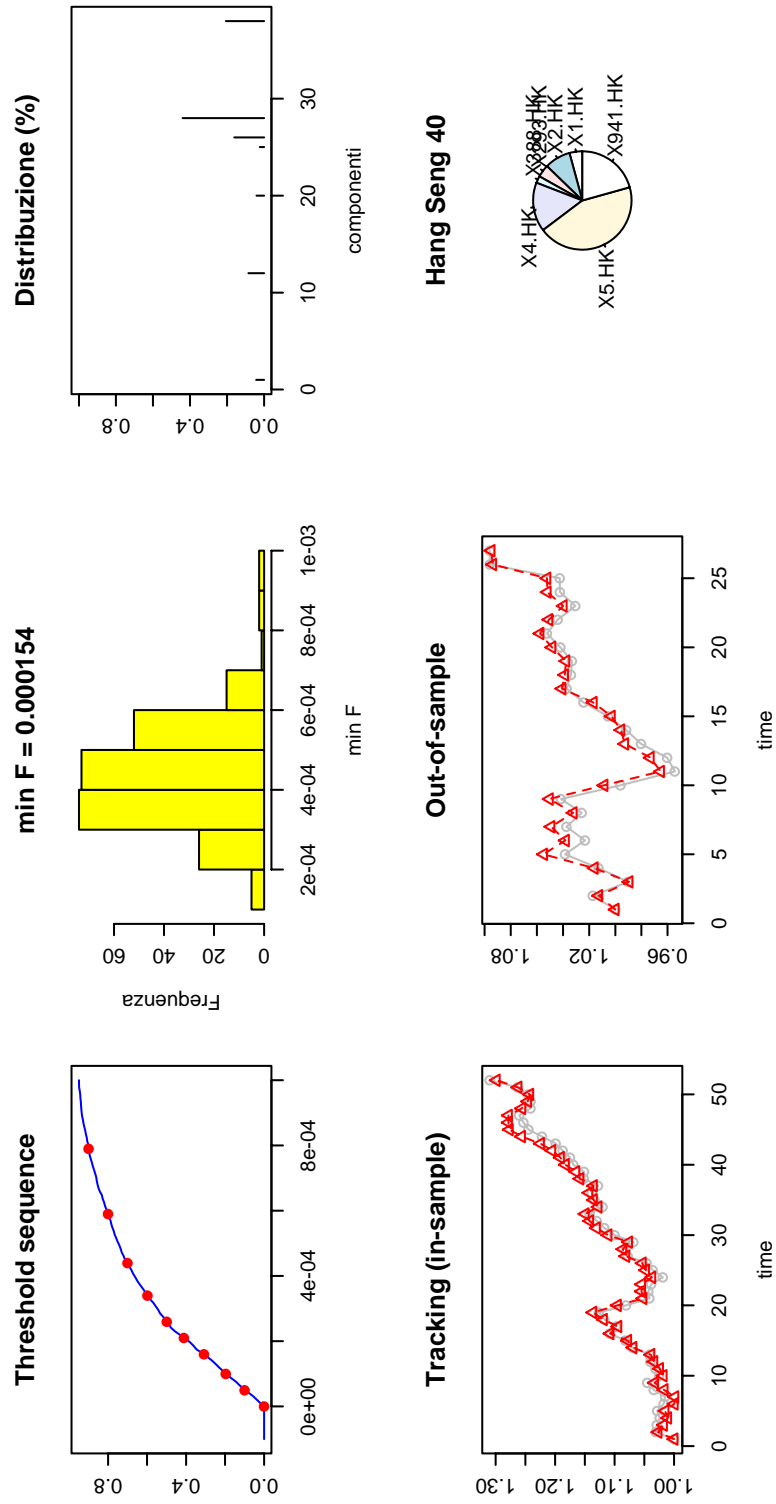


Figura 3.3: Esempio output grafico

- Istogramma degli n_{exec} valori minimi della funzione obiettivo \mathcal{E} ottenuti nelle n_{exec} esecuzioni, con indicazione dell'ottimo $\min F$ di tali valori.
- Diagramma a barre verticali della distribuzione del portafoglio ottimo. In ascissa il numero d'ordine della componente dell'indice (da 1 a N), in ordinata la % di capitale investito nella corrispondente componente (vedi anche il diagramma a torta nella sesta sottofigura).
- Grafici del rendimento dell'indice ($-\circ-$) e del portafoglio tracking ($-\triangle-$). Periodo *in-sample* da t_1 a t_2 .
- Grafici del rendimento dell'indice ($-\circ-$) e del portafoglio tracking ($-\triangle-$). Periodo *out-of-sample* da $t_3 = t_2$ a t_4 .
- Diagramma a torta della distribuzione del capitale investito nel portafoglio ottimo, con indicazione del nome della componente (vedi anche il diagramma a barre verticali nella terza sottofigura).

3.2.6 Ottimizzazione mista: efficacia

Per esemplificare l'importanza di far seguire uno stadio di ottimizzazione classica allo stadio di ottimizzazione euristica, illustriamo sempre il caso del portafoglio tracking dell'Indice Hang Seng con $n = 7$ componenti. L'ottimizzazione euristica ha fornito $\min \mathcal{E} = 0.000167$, e la successiva fase di minimizzazione classica locale ha diminuito questo valore a $\min \mathcal{E} = 0.000154$. Vedi anche la Figura 3.4 e la Figura 3.5.

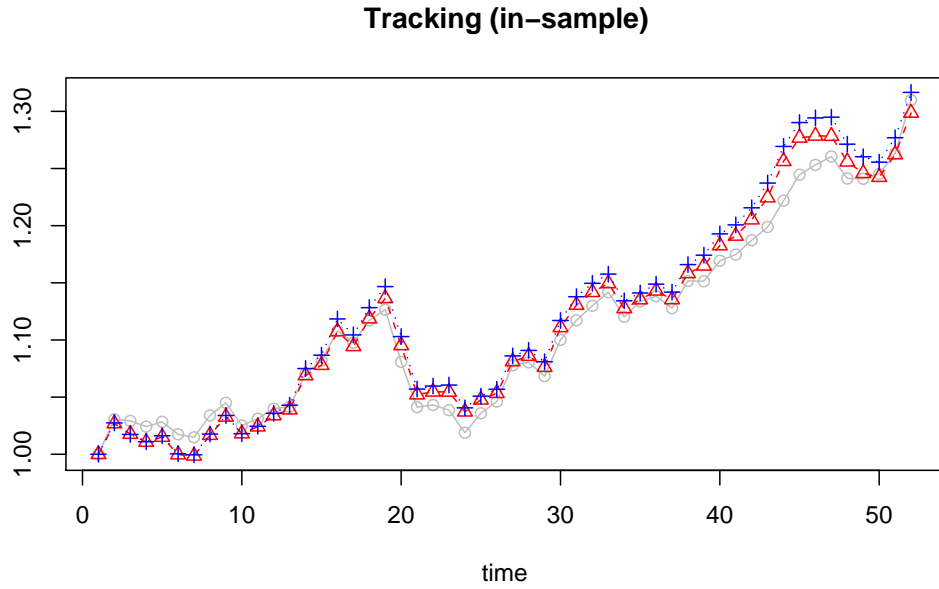


Figura 3.4: Grafici del rendimento dell'Indice (Hang Seng) ($- \circ -$), del portafoglio tracking con $n = 7$ generato dal solo stadio di ottimizzazione euristica ($- + -$) e del portafoglio generato dopo lo stadio di ottimizzazione classica locale ($- \triangle -$). Periodo: $t_1 = 06/01/2006$, $t_2 = 29/12/2006$.

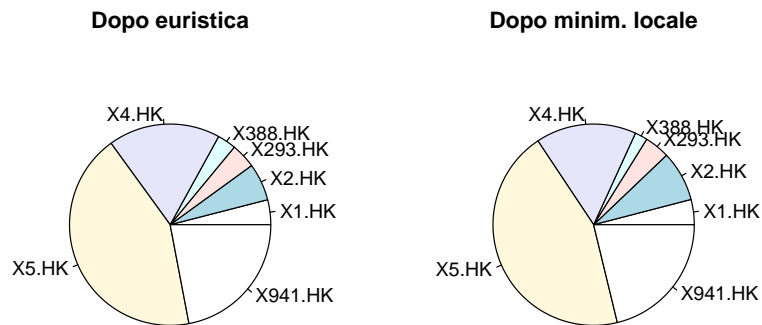


Figura 3.5: Composizioni dei portafogli di cui alla Figura 3.4.

Capitolo 4

Risultati e conclusioni

4.1 Risultati

L'algoritmo è stato implementato utilizzando tre indici di riferimento, utilizzando come linguaggio di programmazione il pacchetto statistico R.

Questo tipo di linguaggio ha la particolarità di essere liberamente disponibile ed *open-source*; permette, cioè, a chiunque di verificare liberamente i risultati ottenuti dalla presente ricerca, consentendone l'agevole valutazione.

Il primo indice scelto è l'S&P 100 (meno volatile,¹ composto da $n = 100$ titoli, ticker Bloomberg OEX Index). L'indice S&P100, introdotto nel 1983 (lanciato a base 50 nel gennaio 1976), rappresenta un sottoindice del più ampio indice S&P 500. I suoi componenti sono 100 tra i titoli americani maggiormente rappresentativi; generalmente tra i maggiori in termini di capitalizzazione e con maggiore stabilità nell'S&P 500. Storicamente il *turnover* dei titoli contenuti nell'S&P 100 è stato minore rispetto a quello che caratterizza l'S&P 500. Tali titoli vengono selezionati in base ad un criterio di bilanciamento per settore e rappresentano una capitalizzazione di mercato aggregata pari al 57% della capitalizzazione dell'S&P 500 e circa il 45% della capitalizzazione del mercato azionario statunitense. L'indice attualmente (febbraio 2008) capitalizza circa USD 7,090 miliardi ed è costituito da titoli azionari appartenenti ai seguenti settori industriali: Materie prime (6 società), Utilities (5 società), Finanziari (18 società), Beni di consumo di base (13 società), Beni di consumo discrezionali (10 società), Prodotti medico-sanitari (11 società), Telecom (3 società), Energia (8 società), Information technology (12 società) e Industriali (14 società).

¹La deviazione standard di S&P 100 annualizzata su serie storica delle variazioni giornaliere dei prezzi su 5 anni è 8,82%, calcolata all'8 febbraio 2008.

I dati utilizzati nell'implementazione sono i prezzi settimanali medi del 2006 e 2007 (più precisamente i dati 2007 sono fino al 09/11/2007) forniti da Bloomberg. Usando i dati di tutto il 2006 (*in-sample*), viene selezionato, utilizzando il codice in R, un portafoglio con un ridotto numero k di componenti (tipicamente $10 \leq k \leq 20$), appunto per ridurre i costi di transazione. Si immagina poi di investire un capitale fissato convenzionalmente pari a 1000 in data 01/01/2007. Poi, attraverso i dati del 2007, viene confrontato l'andamento di questo portafoglio rispetto all'indice S&P 100 (*out-sample*). I risultati appaiono molto confortanti in quanto, secondo la simulazione, il ribilanciamento del portafoglio selezionato andrebbe di norma effettuato appena dopo il primo trimestre 2007; basterebbe a questo proposito far semplicemente rigirare il programma. Il codice è stato per scrupolo testato su tre tipi diversi di sistemi operativi (Mac OSX, Windows XP, Solaris 10) che supportano R. L'output è sia di tipo testuale (viene fornita la composizione del portafoglio "ottimo") che grafico (per un esempio di output grafico si veda Figura 4.1).

L'introduzione di ottimizzazione mista, trattata nel precedente capitolo e proposta per la prima volta in questo lavoro, si è dimostrata particolarmente significativa. L'algoritmo euristico, infatti, è stato raffinato sviluppandone ulteriormente la fase finale attraverso l'aggiunta dell'applicazione di un metodo classico di minimizzazione locale. Quest'ultima viene eseguita con l'algoritmo iterativo del tipo di Newton. Questa operazione ha consentito di raggiungere il primo risultato principale della tesi. In pratica, l'ottimizzazione locale ha permesso di individuare il fenomeno dei minimi che si trovano sul bordo dell'insieme ammissibile. In questo caso, infatti, accade che il minimo libero (non vincolato dalla condizione che il numero di quote da investire in un titolo debba essere positivo) sia ottenibile con un numero *negativo* di quote. Tale risultato non è accettabile nella gestione passiva e, di conseguenza, neanche nel nostro modello che non ammette la vendita allo scoperto. Questo argomento viene affrontato in dettaglio nel precedente Capitolo con l'ausilio di un esempio numerico costruito con un indice artificiale. La conclusione che si evince è la seguente: nella costruzione di un portafoglio ridotto di titoli che ha il fine di replicare un determinato indice, non sempre risulta efficace l'inclusione di un insieme allargato di elementi; al contrario si è verificato che l'aggiunta di un titolo poco significativo peggiora l'azione di *tracking* dello stesso indice. Il fenomeno è rappresentato nella Figura 4.2. Viene confermata, quindi, matematicamente la supposizione di alcuni analisti (Elton et al. 2003 [35]) secondo cui la diversificazione del portafoglio può essere raggiunta con un numero piuttosto basso dei titoli se questi sono scelti in maniera corretta. In altre parole, la diversificazione del portafoglio non dipende dalla quantità dei titoli al suo interno, bensì dalla loro "qualità".

Nella tabella 4.1 viene rappresentata la composizione e la distribuzione del

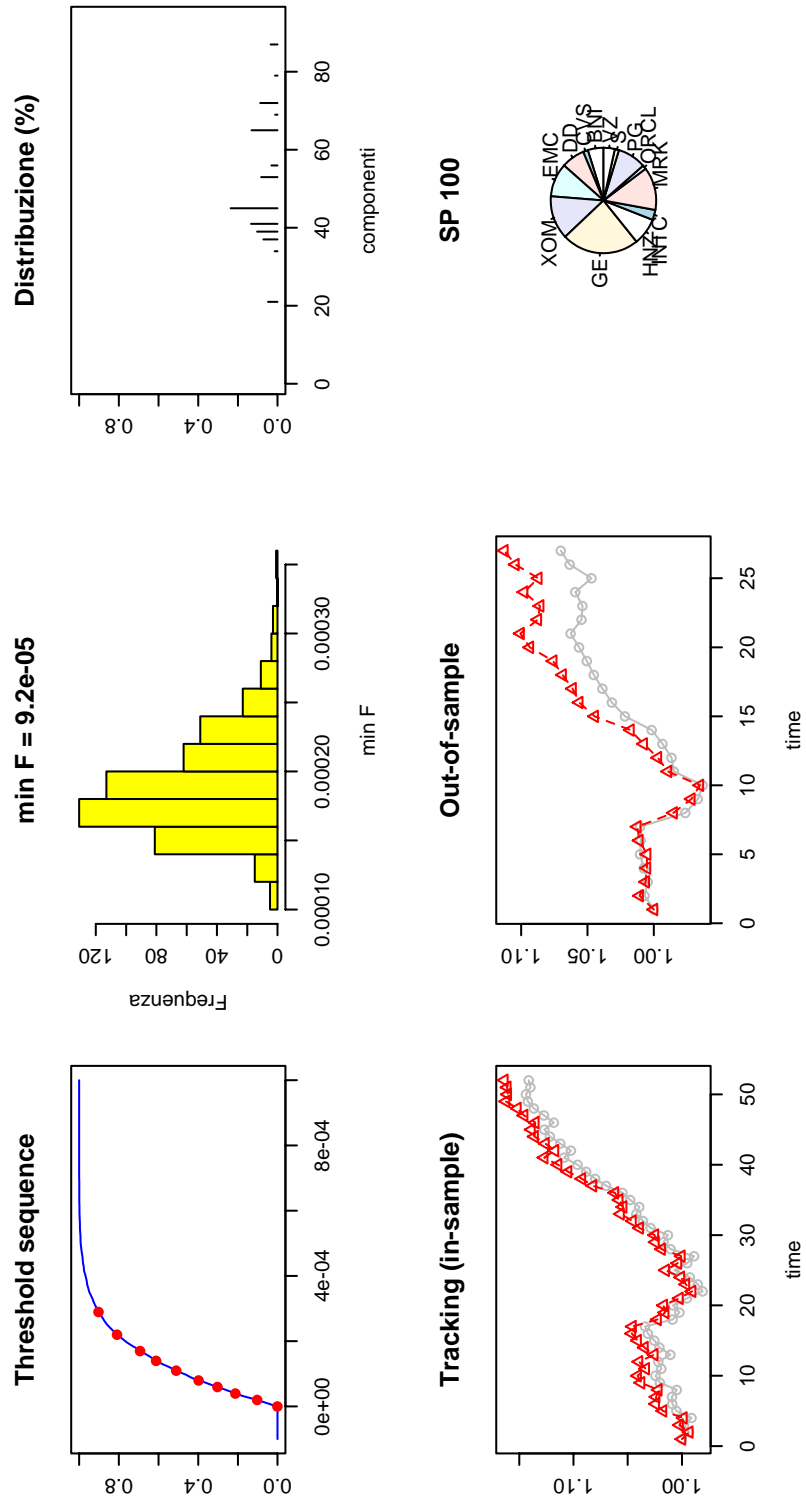


Figura 4.1: Portafoglio con 13 titoli riprodotte S&P 100

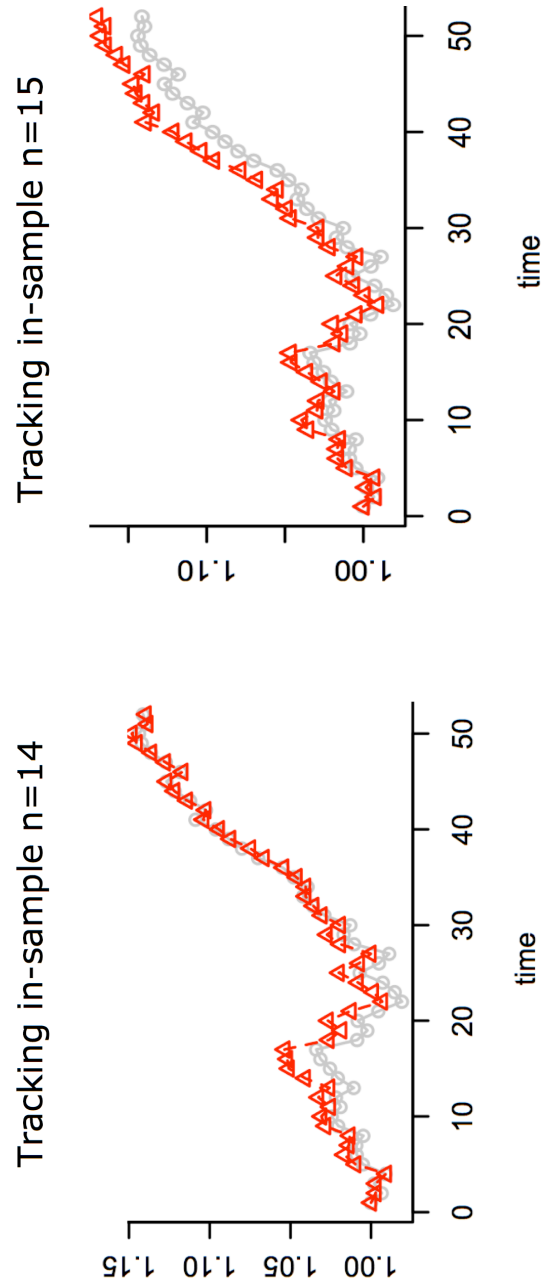


Figura 4.2: Confronto dell'*index tracking* di S&P 100 con $n = 14$ e $n = 15$ titoli

portafoglio ottimo dopo entrambi gli stadi di ottimizzazione per l'indice S&P 100 con $n = 15$ titoli. L'output di sinistra rappresenta il risultato ottenuto con l'algoritmo euristico, l'output di destra invece è il risultato ottenuto in seguito all'ulteriore ottimizzazione (l'ottimizzazione locale).

	numero	componente	pesi		numero	componente	pesi
1	7	MO	0.0006				
2	16	BAC	0.0830	1	16	BAC	0.0981
3	21	BNI	0.0125	2	21	BNI	0.0050
4	22	CPB	0.0422	3	22	CPB	0.0442
5	37	DD	0.0804	4	37	DD	0.0724
6	41	XOM	0.1111	5	41	XOM	0.1064
7	54	HD	0.0324	6	54	HD	0.0346
8	55	HON	0.0459	7	55	HON	0.0597
9	57	IBM	0.1769	8	57	IBM	0.1808
10	58	IP	0.0082	9	58	IP	0.0029
11	65	MRK	0.0254	10	65	MRK	0.0137
12	69	ORCL	0.0192	11	69	ORCL	0.0212
13	70	PEP	0.1695	12	70	PEP	0.1736
14	80	TGT	0.1303	13	80	TGT	0.1245
15	87	VZ	0.0623	14	87	VZ	0.0629

Tabella 4.1: Pesi delle componenti nel portafogli $n = 15$ titoli riprodotte S&P 100 prima e dopo ottimizzazione locale

In pratica, effettuando un'ottimizzazione con vincolo di cardinalità $n = 15$ titoli risulta che per riprodurre l'indice S&P 100 è meglio costruire un portafoglio con $n = 14$ titoli. L'aggiunta di una quantità minima di un'ulteriore componente (in questo caso si tratta della settima componente dell'indice, ticker Bloomberg MO, che dopo l'ottimizzazione euristica ha un peso nel portafoglio ottimo del 0.06%) peggiora il *tracking* dell'indice. Si verifica l'effetto del minimo sul bordo dell'insieme ammissibile e quindi proseguendo con la minimizzazione locale, si ottiene il portafoglio ottimale che contiene solamente $n = 14$ titoli. Questo accade perchè il minimo della funzione *tracking error* si raggiunge per un numero negativo di quote del titolo MO che vengono, di conseguenza, poste uguali a zero. È interessante osservare la composizione del portafoglio ottimo ottenuto con questo processo. Se lo si confronta con il portafoglio che replica l'indice ottenuto con la minimizzazione vincolata, con vincolo $n = 14$ titoli, si nota facilmente come i due portafogli non siano uguali. Questo spiega il peggioramento del *tracking* dell'indice passando dal caso con $n = 14$ ad $n = 15$ titoli. Le componenti ed i pesi del "vero" portafoglio ottimo

che replica l'indice S&P 100 con $n = 14$ titoli sono rappresentati nella tabella 4.2.

	numero	componente	pesi
1	1	MMM	0.0656
2	22	CPB	0.0712
3	28	C	0.0173
4	37	DD	0.1427
5	41	XOM	0.0687
6	60	JPM	0.0496
7	64	MDT	0.0558
8	66	MER	0.0761
9	72	PG	0.1141
10	78	SO	0.0654
11	81	TXN	0.0262
12	85	UTX	0.0487
13	91	WFC	0.1258
14	92	WY	0.0728

Tabella 4.2: esi delle componenti nel portafogli $n = 14$ titoli riprodotte S&P 100 prima e dopo ottimizzazione locale

Il secondo indice scelto è Hang Seng (più volatile, ² composto da $n = 40$ titoli, ticker Bloomberg HSI Index). L'indice Hang Seng, costituito nel luglio del 1964 (pubblicamente è stato lanciato il 24 novembre 1969), è il più diffuso indicatore dell'andamento della Borsa di Hong Kong. Comprende 43 società quotate (la ricerca ha tuttavia preso in considerazione solamente 40 componenti escludendone i seguenti: *China Overseas Land and Investments Ltd* (ticker Bloomberg 688 HK), *China Shenhua Energy Co.Ltd* (ticker Bloomberg 1088 HK), *Petro China Co.Ltd* (ticker Bloomberg 857 HK) entrati nell'indice dopo l'ottobre 2007) ed è ponderato per la capitalizzazione di mercato dei suoi componenti, quotati e negoziati in Hong Kong Dollars. Tali componenti sono scelti tra le prime società che costituiscono il 90% del valore (aggregato) di mercato tra quelle quotate nella Borsa di Hong Kong (SEHK), tra le prime che generano il 90% del fatturato aggregato dello stesso mercato e che siano quotate da almeno 24 mesi (tale condizione permette delle deroghe nel caso si tratti di società con un valore di mercato particolarmente elevato, nel qual caso la quotazione storica minima è i 3 mesi). Le componenti sono suddivise per settore in quattro sottoindici: Commercio ed Industria, Finanziario, Utilities

²La deviazione standard di Hang Seng annualizzata su serie storica delle variazioni giornaliere dei prezzi su 5 anni è 17,93%, calcolata all'8 febbraio 2008.

(Pubblici servizi), Immobiliare. L' Hang Seng comprende oltre a società della Repubblica Popolare Cinese (le cosiddette *H-shares*) anche imprese straniere, in particolare britanniche (differenziandosi dal suo sottoindice *Hang Seng China Enterprises* che appunto comprende solamente società registrate nella repubblica Popolare Cinese e quotate nella Borsa di Hong Kong). L'indice attualmente (febbraio 2008) capitalizza circa HKD 11.7 trillion (Euro 1,018 miliardi).

Nell'implementazione dell'algoritmo nella replica di Hang Seng, i dati utilizzati, come per S&P 100, sono i prezzi settimanali medi del 2006 e 2007 (dati 2007 sono fino al 09/11/2007) ottenuti da Bloomberg. Anche in questo caso, utilizzando i dati di tutto il 2006 (*in-sample*) viene selezionato un portafoglio con un ridotto numero k di componenti (tipicamente $2 \leq k \leq 10$). Si immagina, successivamente, di investire un capitale fissato convenzionalmente pari a 1000 in data 01/01/2007. L'andamento di questo portafoglio viene confrontato con l'indice (*out-sample*) sulla base dei dati del 2007. I risultati ottenuti sono confortanti; analizzando le risposte ottenute dalla replica di questo indice, si è riscontrata la capacità dell'algoritmo di determinare se all'interno di un indice oggetto vi siano delle componenti dominanti. La *dominanza* di una componente è intuitivamente verificata qualora la stessa componente sia costantemente presente nei portafogli di bassa cardinalità che riproducono l'indice. In tal caso, per giungere all'individuazione della componente dominante è sufficiente verificare la sua presenza ed il suo peso nei vari portafogli prodotti dall'algoritmo. Ad esempio, il fatto che l'indice azionario Hang Seng (Borsa di Hong Kong), sia dominato da alcune sue componenti (HSBC e China Mobile) è ben noto agli analisti,³ che attribuiscono al comportamento dinamico di questi titoli la maggior parte dei movimenti dell'indice. L'output grafico rappresentato nella Figura 4.3, ottenuto utilizzando i risultati dell'elaborazione dell'algoritmo *threshold accepting* per replicare l'indice Hang Seng con soli 7 titoli, evidenzia il concetto delle componenti dominanti.

Il grafico (Figura 4.3) rappresenta una matrice “a barre” di 500 righe e 38 colonne. Le colonne da 1 a 38 corrispondono ordinatamente alle 38 componenti di Hang Seng utilizzate per l'esecuzione dell'algoritmo. Ogni riga corrisponde ad una esecuzione del programma e rappresenta in un codice-colore il portafoglio ottimo con numero titoli $n = 7$ fornito da quella esecuzione. Le 31 componenti assenti (cioè le componenti con peso uguale a 0) sono rappresenta-

³A Hong Kong, la caduta del 6 febbraio 2008 è stata causata dal crollo dei titoli *heavy-weights*: HSBC (ticker Bloomberg 5 HK, il suo peso all'interno dell'indice quel giorno era 15.27%), China Mobile (ticker Bloomberg 941 HK con il peso 13,22%) e China Life Insurance Co. (ticker Bloomberg 2628 HK e il peso 4,12%). HSBC ha registrato una flessione del 4.1%, China Life è sceso del 7.8%, mentre China Mobile ha perso il 5.1%. (V. Phani Kumar, MarketWatch)

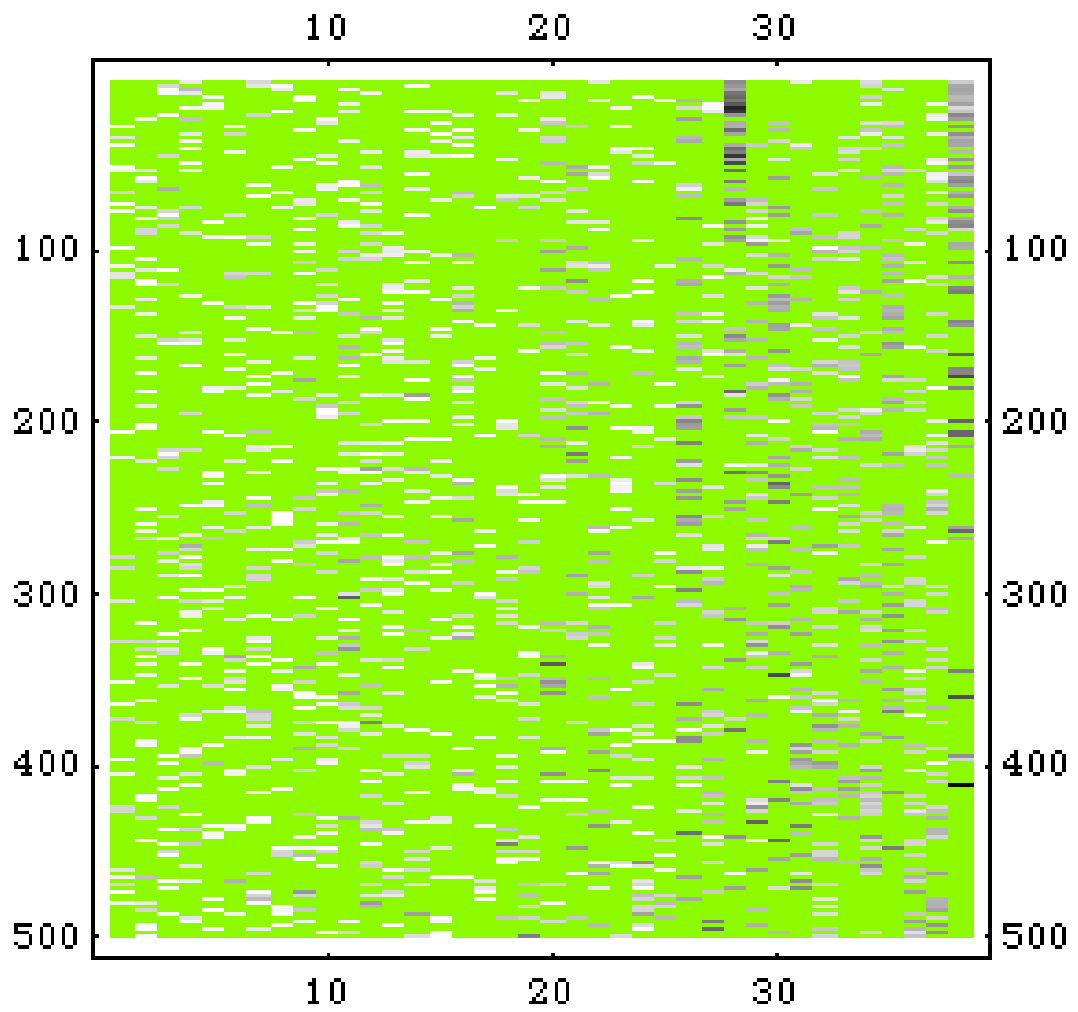


Figura 4.3: Matrice a barre per l'indice Hang Seng

te dallo sfondo verde, le 7 componenti scelte sono rappresentate per mezzo di trattini in scala di grigio, proporzionalmente al loro peso (più scuro=maggiore peso). Le esecuzioni sono ordinate per valori crescenti della funzione obiettivo, per cui il portafoglio ottimo fornito dal complesso delle 500 esecuzioni è quello della riga uno. Si nota come al decrescere della funzione obiettivo si ottengano bande scure sempre più numerose in riferimento alle componenti dominanti numero 28 e 38.

L'istogramma del peso medio, nella Figura 4.4, rappresenta il peso medio delle 38 componenti nelle migliori 100 delle 500 esecuzioni. Le componenti sono elencate in ascissa per peso medio decrescente. Si osserva come le componenti dominanti, cioè le componenti 28 (HSBC, ticker Bloomberg 5 HK) e 38 (China Mobile, ticker Bloomberg 941 HK) abbiano nelle migliori 100 esecuzioni un peso medio rispettivamente del 24.42% e del 15.41%, dove con esecuzioni migliori si intendono quelle con funzione obiettivo più bassa. Le successive 5 componenti più pesanti sono le componenti numero 30, 35, 26, e 21, rispettivamente con pesi pari a 5.58%, 4.47%, 3.86%, 3.26% e 3.15%.

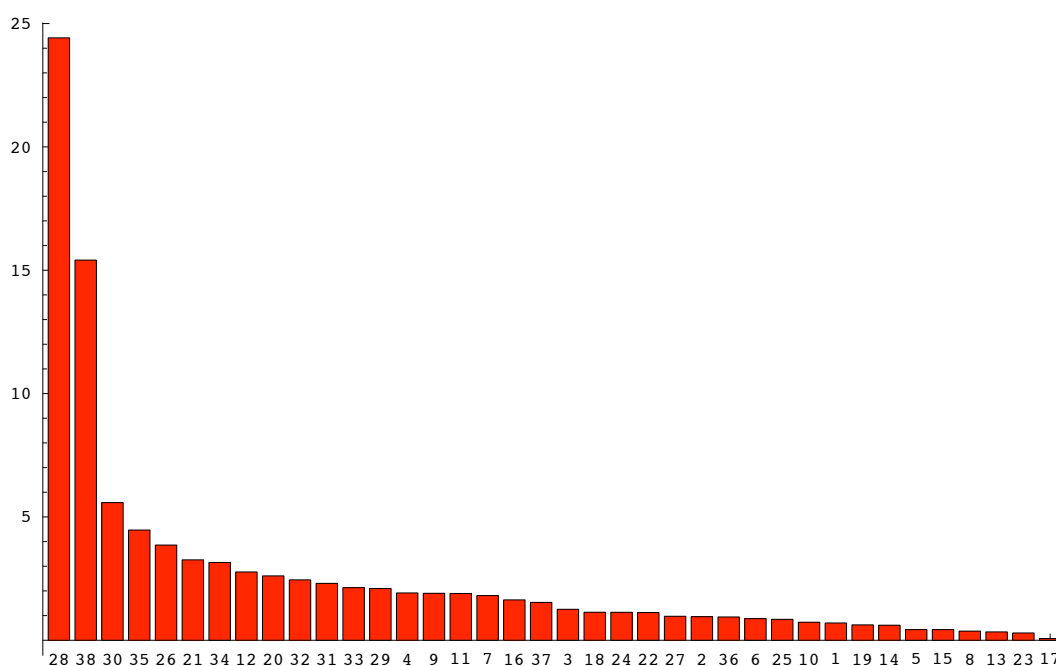


Figura 4.4: Istogramma peso medio

I risultati ottenuti confermano quindi, l'ipotesi "intuitiva" della definizione di una componente dominante, cioè la presenza costante di una componente dominante nei portafogli di bassa cardinalità che riproducono l'indice. I portafogli ottimi di cardinalità k con $2 \leq k \leq 10$ che replicano l'indice Hang Seng

risultano contenere sempre queste due componenti (HSBC e China Mobile). Il risultato ottenuto è riportato nella tabella 4.3

Per un'ulteriore illustrazione della dominanza di HSBC all'interno dell'indice Hang Seng si veda la Figura 4.5 che riporta l'andamento del titolo HSBC ($-\triangle-$) vs. l'andamento del HSI ($-\circ-$). Si nota come l'andamento sia molto simile a quello registrato dall'indice.

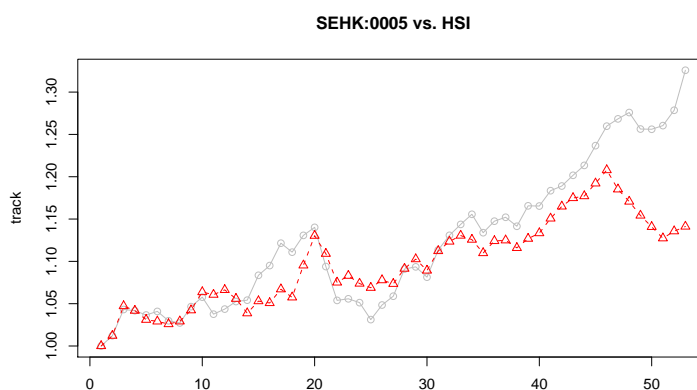


Figura 4.5: HSBC vs. Hang Seng, in pratica un portafoglio con un solo titolo

Un esempio di output grafico tipico per le esecuzioni dell'algoritmo nella replica di Hang Seng è riportato nella Figura 4.6.

Il terzo indice scelto è MIB 30 (media volatilità,⁴ ticker Bloomberg MIB30 Index). Mib 30 è l'indice composto dai 30 titoli più liquidi e a maggior capitalizzazione quotati sulla Borsa di Milano (l'indice è ponderato per la capitalizzazione di mercato dei suoi componenti). Il lancio a base 10,000 risale al 31 dicembre 1992. La revisione del paniere di titoli avviene ogni sei mesi. L'indice attualmente (febbraio 2008) capitalizza circa Euro 508 miliardi.

Anche per questo indice i dati utilizzati sono i prezzi settimanali medi del 2006 e 2007 ottenuti da Bloomberg, con la differenza, rispetto ai dati utilizzati per S&P 100 e Hang Seng, che sono stati considerati i dati dell'intero periodo 2007. Come per le prime due applicazioni, attraverso i dati del 2006 (*in-sample*) viene selezionato un portafoglio con un ridotto numero k di componenti (tipicamente $2 \leq k \leq 10$). Si immagina poi di investire un capitale fissato convenzionalmente pari a 1000 in data 01/01/2007. Poi, con i dati del 2007, viene confrontato l'andamento di questo portafoglio rispetto all'indice (*out-sample*).

⁴La deviazione standard di MIB 30 annualizzata su serie storica delle variazioni giornaliere dei prezzi su 5 anni è 12,32%, calcolata all'8 febbraio 2008.

numero titoli	2	3	4	5	6	7	8	9	10
nome componente									
1 HK	0	0	0	0.20	0	0	0.21	0	0
101 HK	0	0	0	0	0	0	0	0	0.06
1038 HK	0	0	0	0	0	0	0	0	0
11 HK	0	0	0	0	0	0.07	0.18	0	0
1199 HK	0	0	0	0	0	0	0	0	0
12 HK	0	0	0	0	0	0	0	0	0
13 HK	0	0	0	0	0.20	0.18	0	0.21	0.15
144 HK	0	0	0	0	0	0	0	0	0
16 HK	0	0	0	0	0	0	0	0	0
17 HK	0	0	0	0	0	0	0	0	0
19 HK	0	0	0	0.23	0.14	0	0	0	0
2 HK	0	0	0	0	0	0	0	0	0.05
2038 HK	0	0	0	0	0	0	0	0	0
23 HK	0	0	0.11	0	0	0	0	0	0
2318 HK	0	0	0	0	0	0	0	0	0
2388 HK	0.35	0.43	0	0	0	0	0.09	0	0
2628 HK	0	0	0	0	0	0	0	0	0
267 HK	0	0	0	0	0	0	0	0	0
291 HK	0	0	0	0	0	0	0	0	0
293 HK	0	0	0	0	0	0	0	0.06	0.02
3 HK	0	0	0	0	0	0	0.06	0	0.08
330 HK	0	0	0	0	0	0.07	0	0	0
3328 HK	0	0		0	0	0	0	0	0
386 HK	0	0	0	0.15	0	0	0	0	0
388 HK	0	0	0	0	0	0	0	0	0
4 HK	0	0	0.28	0	0	0	0	0	0.08
494 HK	0	0	0	0	0	0	0	0.03	0
5 HK	0.65	0.42	0.42	0.24	0.31	0.33	0.20	0.36	0.22
551 HK	0	0	0	0	0.08	0	0	0	0
6 HK	0	0	0	0	0	0	0	0	0
66 HK	0	0	0	0	0	0.07	0	0	0.01
762 HK	0	0	0	0	0	0	0	0	0
8 HK	0	0	0	0	0	0	0	0	0
83 HK	0	0	0	0	0	0.12	0.04	0.05	0
883 HK	0	0	0	0	0.11	0	0.08	0.10	0.10
906 HK	0	0	0	0	0	0	0	0.02	0
939 HK	0	0	0	0	0	0	0	0	0
941 HK	0	0.15	0.18	0.18	0.17	0.17	0.14	0.17	0.22
totale	1	1	1	1	1	1	1	1	1

Tabella 4.3: Pesì delle varie componenti nei portafogli di dimensioni $n = 2, \dots, 10$ riproducenti HSI

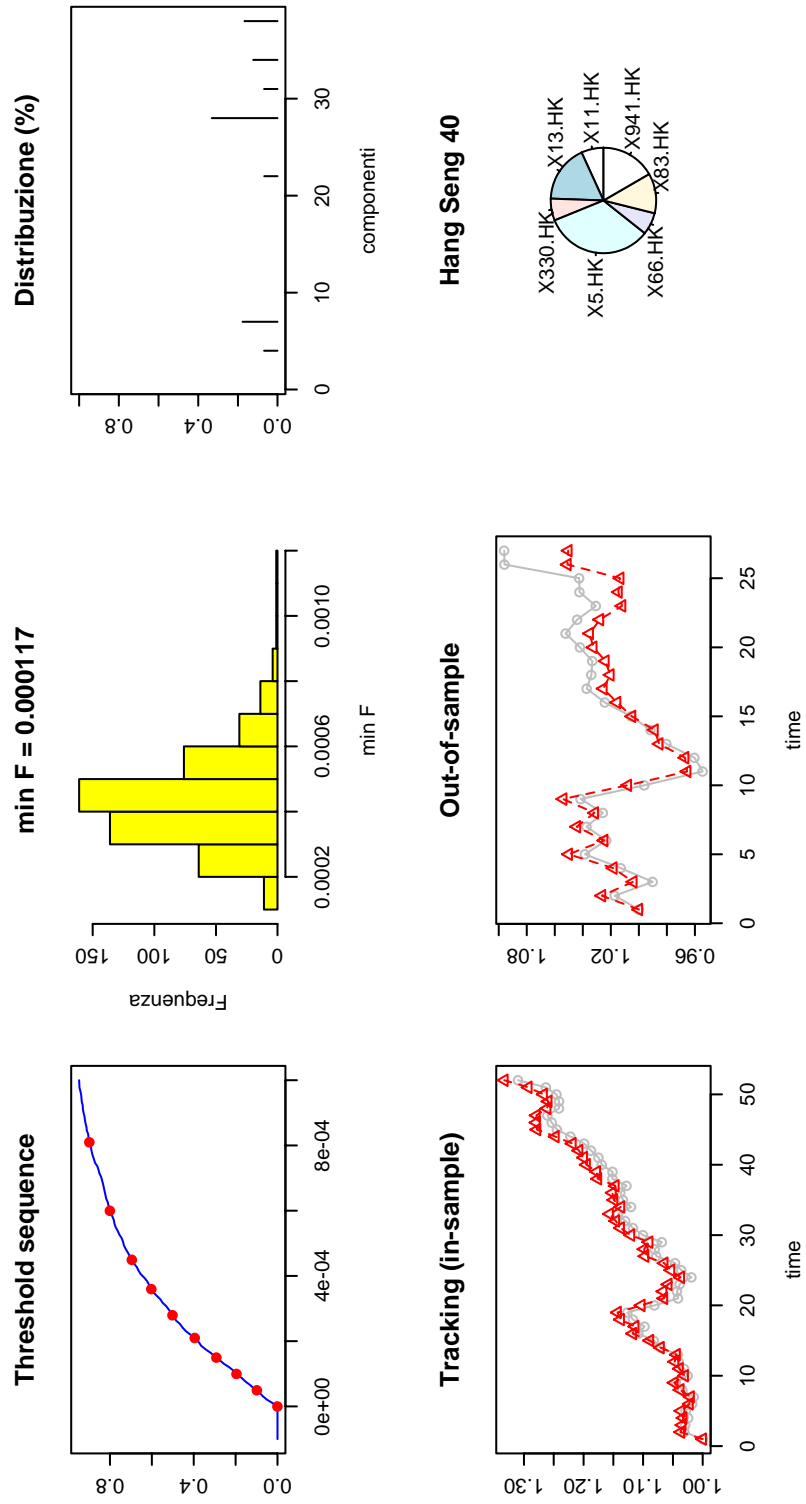


Figura 4.6: Portafoglio con 7 titoli riprodotto HSI

L'output grafico tipico per le esecuzioni dell'algoritmo nella replica di MIB 30 è riportato nella Figura 4.9.

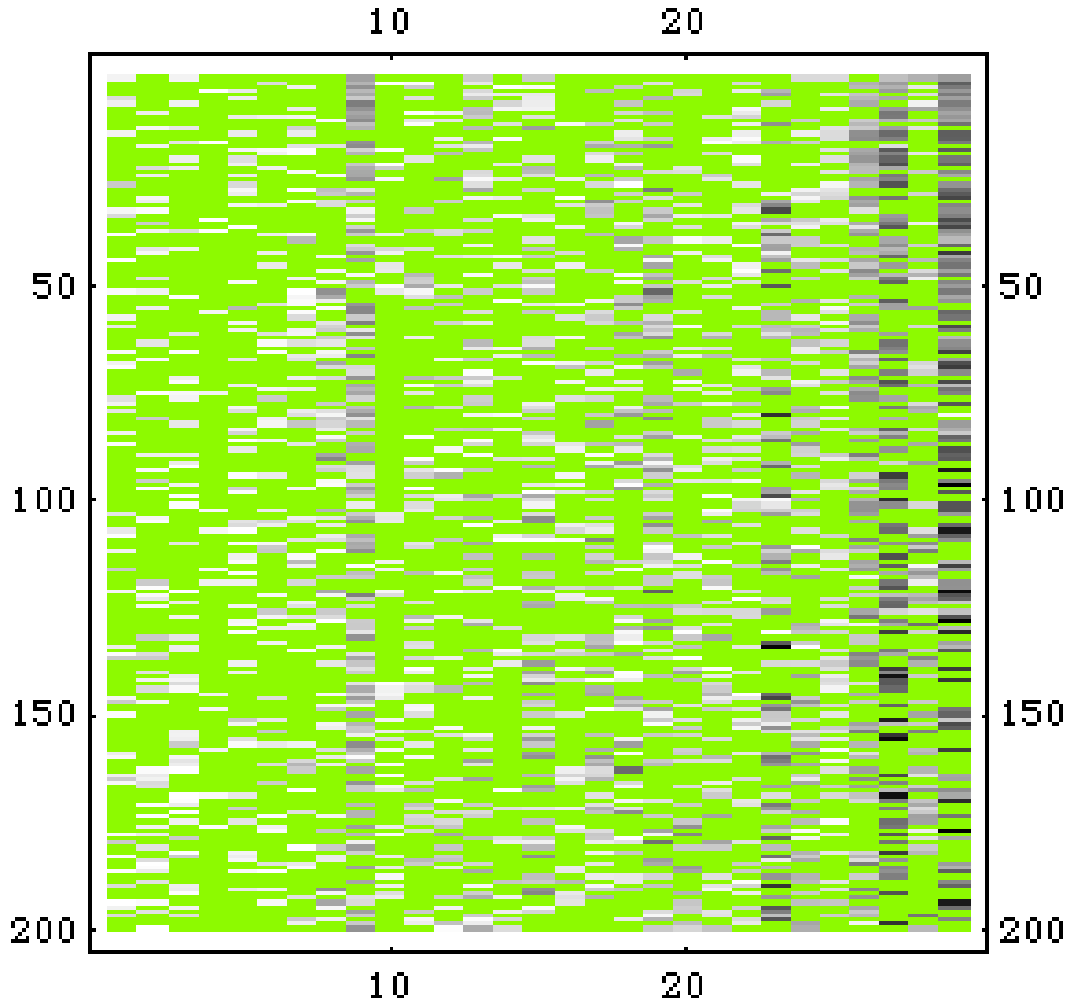


Figura 4.7: Matrice a barre per l'indice MIB 30

Le analisi dettagliate indicano che ci sono tre componenti dominanti nella composizione dell'indice MIB 30: Terna (ticker Bloomberg TRN), Unicredit (ticker Bloomberg UCG) e ENI (ticker Bloomberg ENI). Questa dominanza è, tuttavia, meno evidente rispetto all'indice Hang Seng. La figura 4.7 riporta la matrice "a barre" di 200 righe e 29 colonne che mette in evidenza le tre componenti dominanti. Le colonne da 1 a 29 corrispondono ordinatamente alle 29 componenti di MIB 30 utilizzate per l'esecuzione dell'algoritmo. Ogni riga corrisponde ad una esecuzione del programma e rappresenta in un codice-colore il

portafoglio ottimo con numero titoli $n = 10$ fornito da quella esecuzione. Le 19 componenti assenti (cioè le componenti con peso uguale a 0) sono rappresentate dallo sfondo verde, le 10 componenti scelte con una barra in una scala di grigio proporzionale al suo peso (più scuro = più peso). Anche in questo caso, le esecuzioni sono ordinate per valori crescenti della funzione obiettivo, per cui il portafoglio ottimo fornito dal complesso delle 200 esecuzioni è quello della riga uno. Si nota come al decrescere della funzione obiettivo siano sempre più numerose le bande scure in riferimento alle componenti dominanti numero 9 (ENI), 27 (TRN) e 29 (UCG).

L'istogramma del peso medio, nella Figura 4.11, rappresenta il peso medio delle 29 componenti nelle migliori 200 delle 500 esecuzioni. Le componenti sono elencate in ascissa per peso medio decrescente. Si osserva come alle componenti dominanti, cioè le componenti 29, 27 e 9, nelle migliori 200 esecuzioni siano associati i maggiori pesi medi.

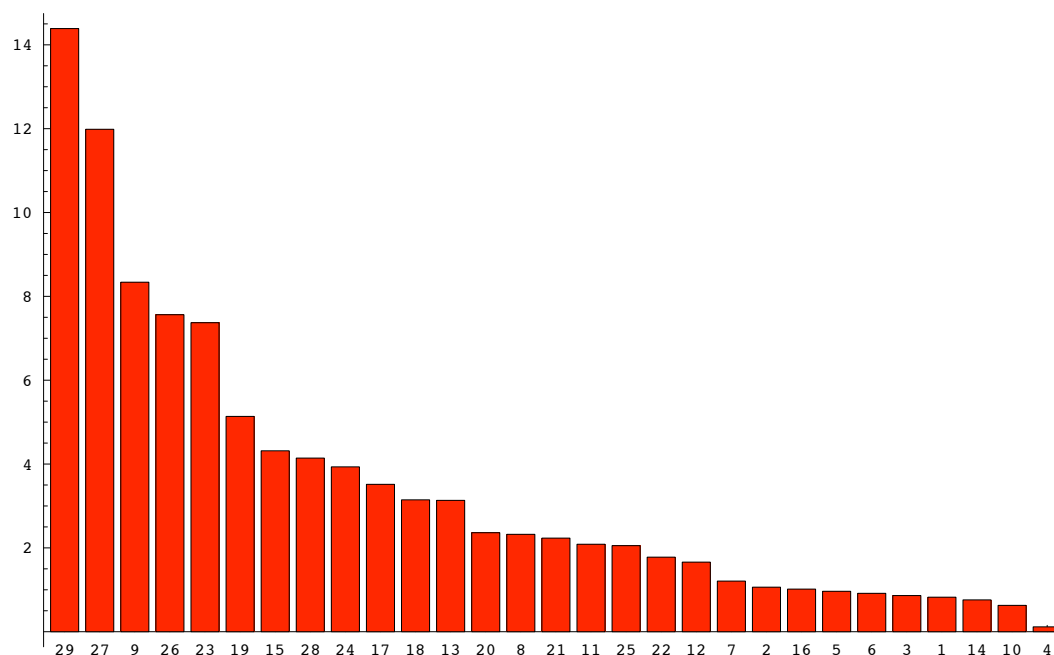


Figura 4.8: Istogramma peso medio per l'indice MIB 30

I portafogli ottimi di cardinalità k con $2 \leq k \leq 10$ che replicano l'indice MIB 30 sono riportati nella tabella 4.4.

Non sono state rilevate componenti dominanti nell'indice S&P100. L'istogramma di peso medio mostra solo le prime 50 componenti in termini di peso (la rappresentazione di tutte le componenti è stata omessa per motivi di visualizzazione grafica). Si nota come il peso medio sia distribuito tra tutte le

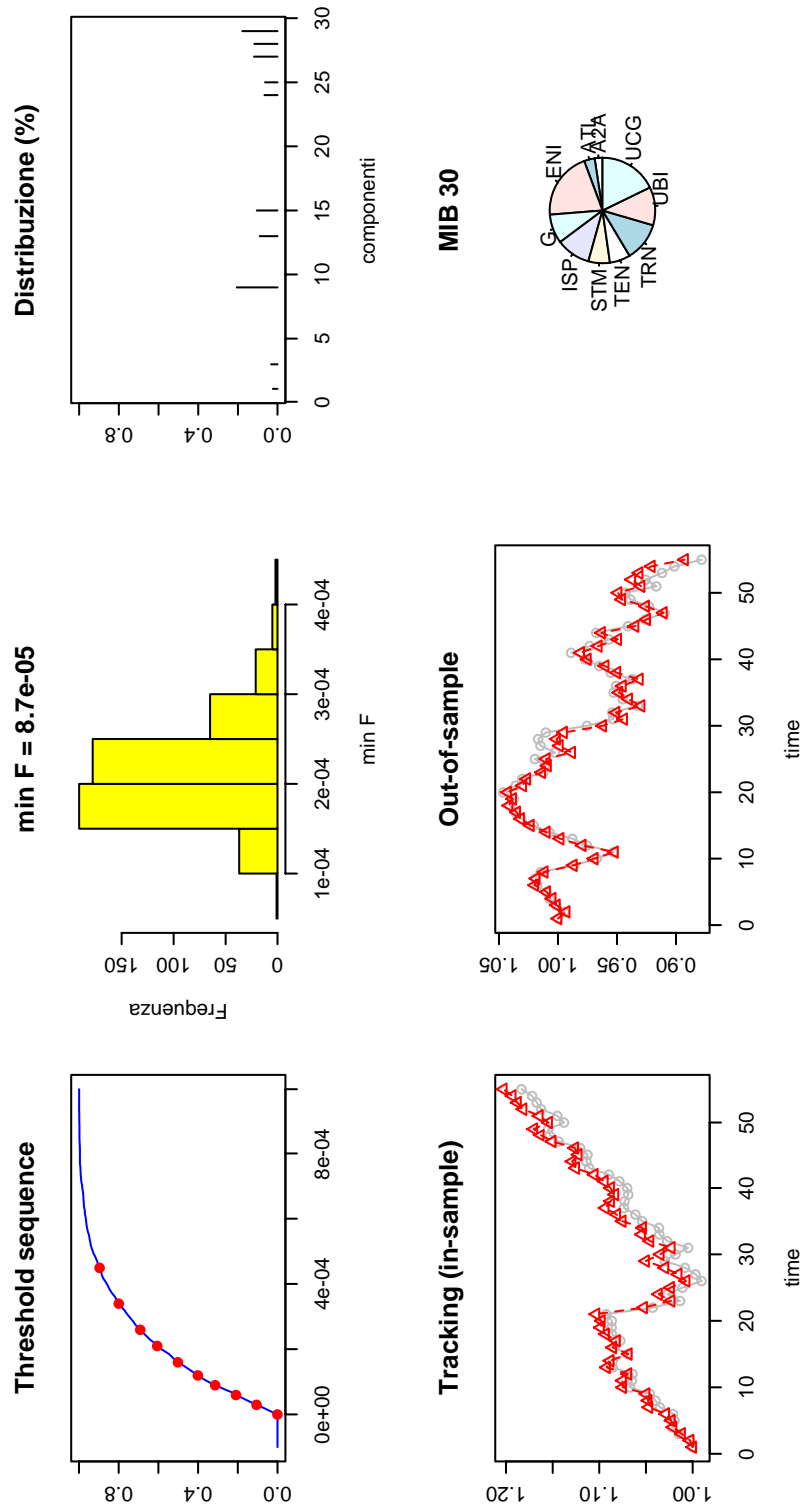


Figura 4.9: Portafoglio con 10 titoli riprodotte MIB 30

numero titoli	2	3	4	5	6	7	8	9	10
nome componente									
A2A	0	0	0	0	0	0	0	0	0.02
AL	0	0	0	0	0	0.11	0.13	0.05	0
ATL	0	0	0	0	0	0	0	0	0.03
BIL	0	0	0	0	0	0	0	0	0
BMPS	0	0	0	0	0	0	0.04	0	0
BPM	0	0	0	0	0	0	0	0.08	0
EDN	0	0	0	0	0	0	0	0	0
ENEL	0	0	0	0	0.36	0	0	0	0
ENI	0	0	0.26	0.16	0	0	0.24	0.18	0.21
F	0	0	0	0	0.09	0	0.06	0.034	0
FNC	0	0	0.26	0	0	0.06	0.09	0	0
FSA	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0.18	0	0.09
IFL	0	0	0	0	0	0	0	0	0
ISP	0	0	0	0.20	0.25	0.07	0.14	0	0.1
LTO	0	0	0	0.05	0	0	0	0.06	0
LUX	0	0	0	0.19	0	0	0	0.07	0
MB	0	0	0	0	0.13	0	0	0	0
MS	0	0.29	0	0	0.11	0.17	0.12	0	0
PC	0	0	0.11	0	0.06	0	0	0	0
PLT	0	0	0	0	0	0	0	0	0
SPM	0	0	0	0	0	0	0	0	0
SRG	0	0	0	0.41	0	0	0	0	0
STM	0	0	0	0	0	0	0	0	0.07
TEN	0	0	0	0	0	0.06	0	0	0.06
TIT	0	0	0	0	0	0	0	0.13	0
TRN	0.53	0.27	0	0	0	0.27	0	0.15	0.12
UBI	0	0	0	0	0	0	0	0	0.12
UCG	0.47	0.45	0.37	0	0	0.27	0	0.25	0.18
totale	1	1	1	1	1	1	1	1	1

Tabella 4.4: Pesì delle varie componenti nei portafogli di dimensioni $n = 2, \dots, 10$ riproducenti MIB 30

componenti. L'unica componente con un presenza più marcata (Figura 4.10 e Figura 4.11) è la numero 86 (US BANCORP, ticker Bloomberg USB), il suo peso medio, però, dell'ordine di 4%, non può essere considerato significativo.

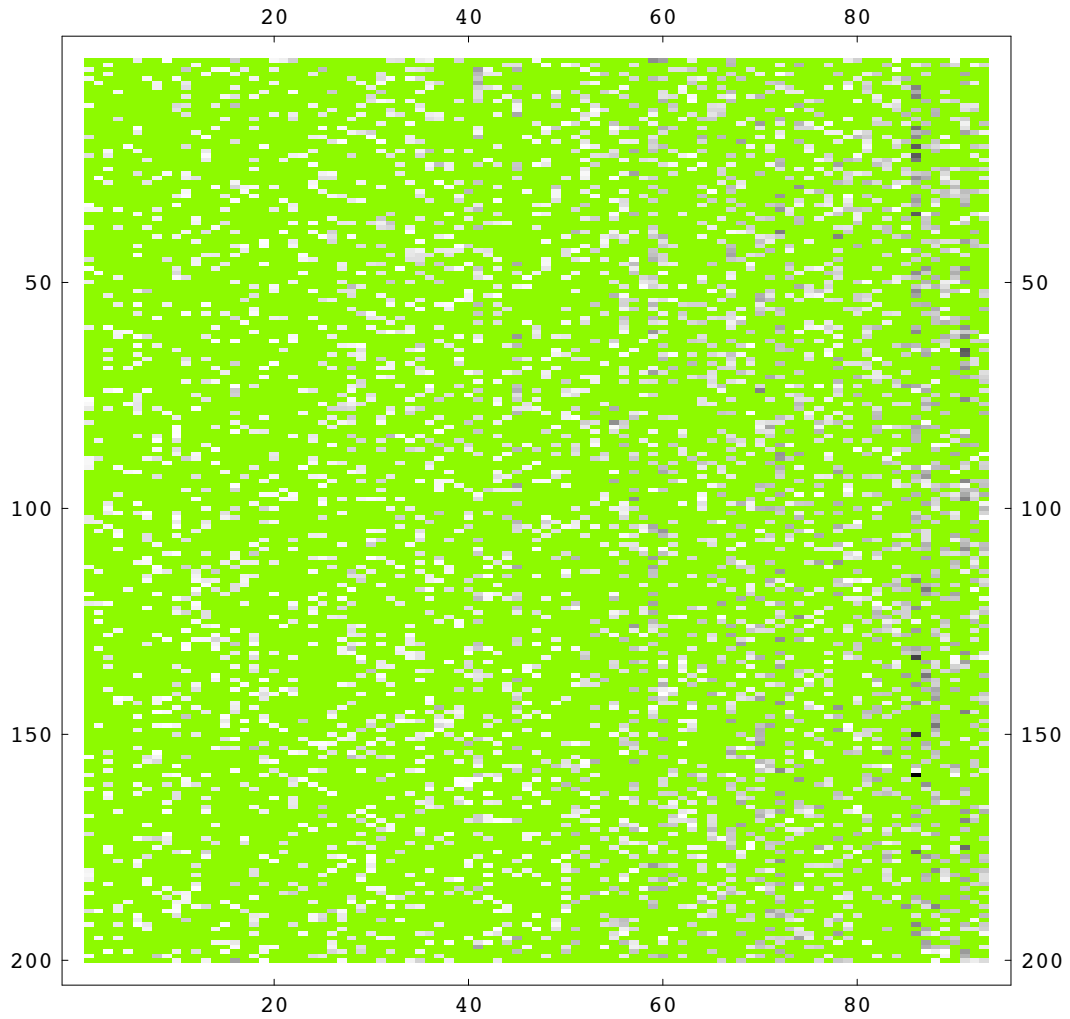


Figura 4.10: Matrice a barre per l'indice S&P 100

4.2 Conclusioni

Il presente lavoro ha affrontato il tema della gestione passiva, e nello specifico lo sviluppo di un processo preposto a simulare l'andamento di un determinato indice di riferimento. In quest'ambito si è proceduto all'applicazione di un

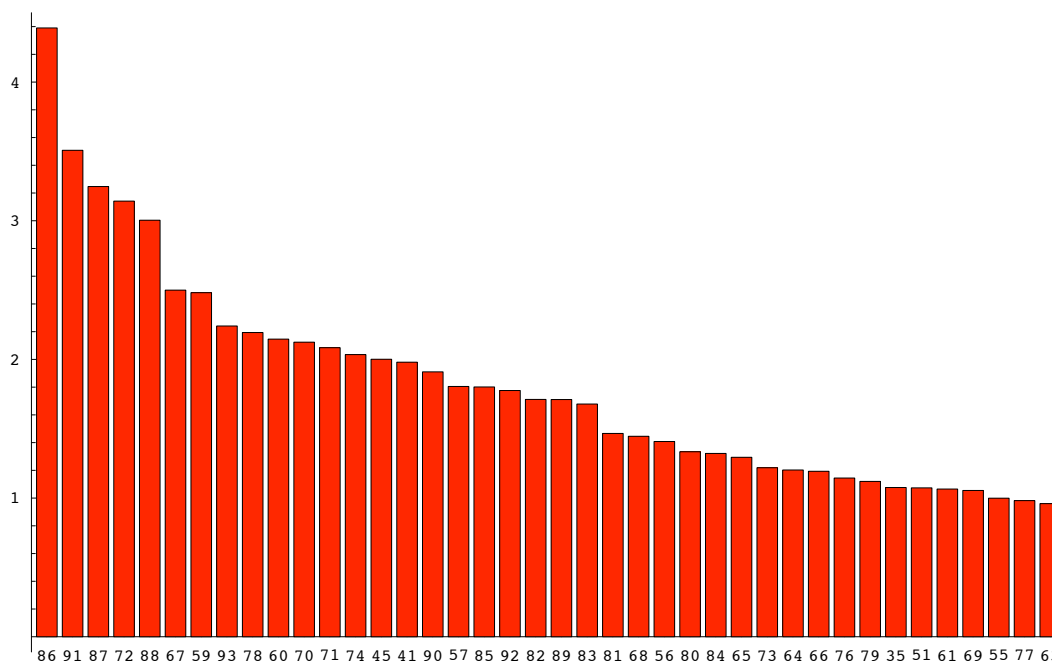


Figura 4.11: Istogramma peso medio per l'indice S&P 100

algoritmo euristico, *threshold accepting*, al fine di verificarne, attraverso vari esperimenti condotti sugli indici MIB 30, S&P 100 e Hang Seng, l'efficacia nella funzione di *index tracking*. È stato prodotto un codice in R che è in grado di replicare l'indice di riferimento con un portafoglio costituito da un numero limitato di titoli (tra 10% e 20% del totale delle componenti dell'indice) con un errore nell'ordine di 10^{-5} . Il risultato è confortante poichè un numero limitato di titoli in portafoglio equivale anche alla riduzione dei costi di transazione. Inoltre, l'algoritmo utilizzato si è dimostrato in grado di determinare se all'interno di un indice oggetto vi fossero delle componenti dominanti. L'algoritmo euristico è stato ulteriormente raffinato con lo sviluppo della fase finale attraverso l'aggiunta di un'ulteriore applicazione, sfruttando un metodo iterativo classico di minimizzazione. In questo modo è stata confermata, matematicamente, la supposizione di alcuni analisti: la diversificazione del portafoglio può essere raggiunta con un numero piuttosto basso di titoli quando questi siano scelti correttamente. Nel campo delle applicazioni pratiche, che riscontrano numerosi utilizzi nel mondo della finanza (sia nella gestione passiva che attiva), rimangono ancora da testare alcune varianti dell'algoritmo caratterizzate da ulteriori vincoli nel problema di ottimizzazione. La struttura del presente lavoro svolto, a questo proposito, si mantiene aperta a tale opportunità.

Appendice A

Risultati delle esecuzione del *tracking* dell'indice Hang Seng

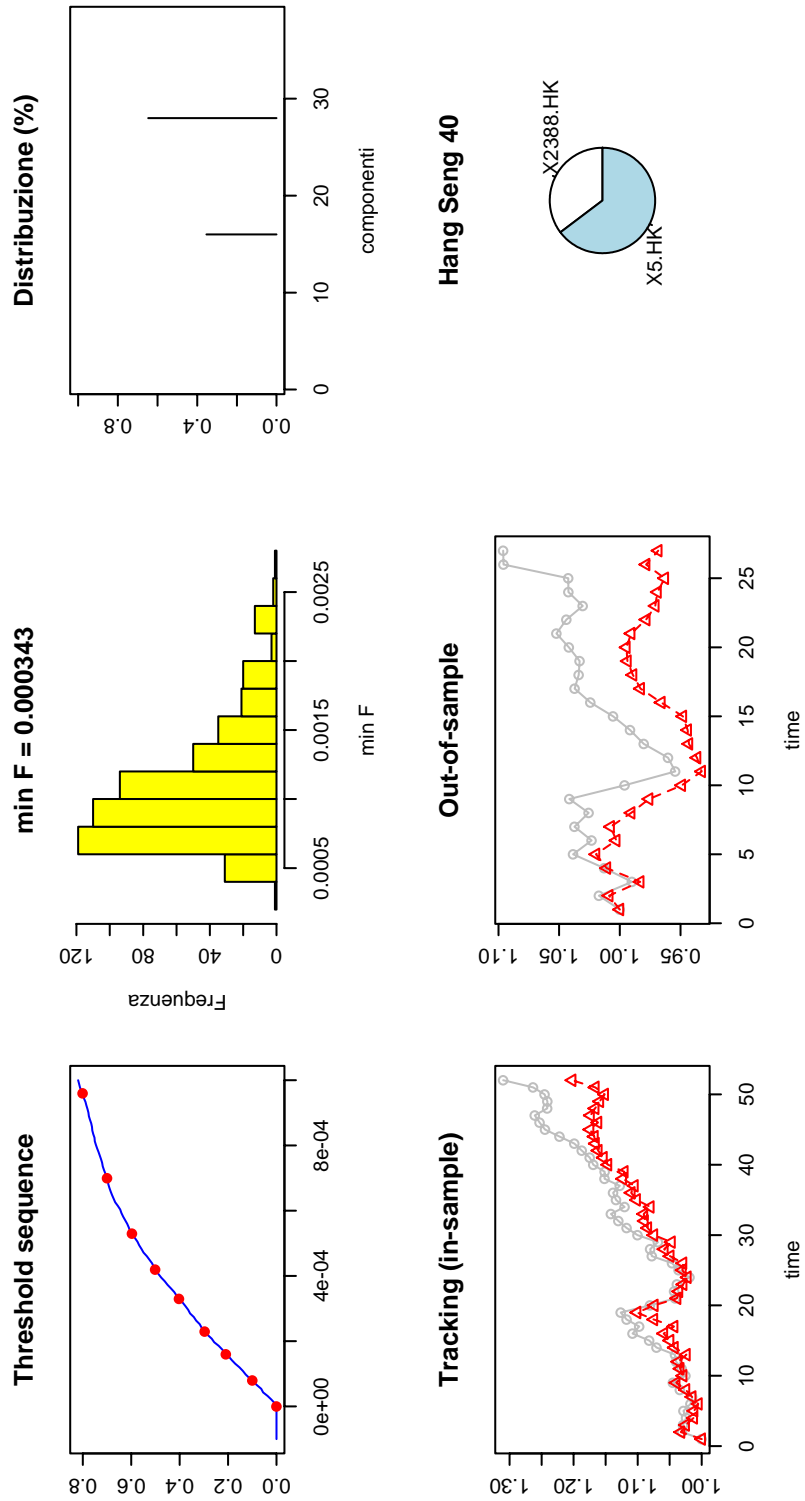


Figura A.1: Portafoglio con 2 titoli riprodotte l'indice Hang Seng

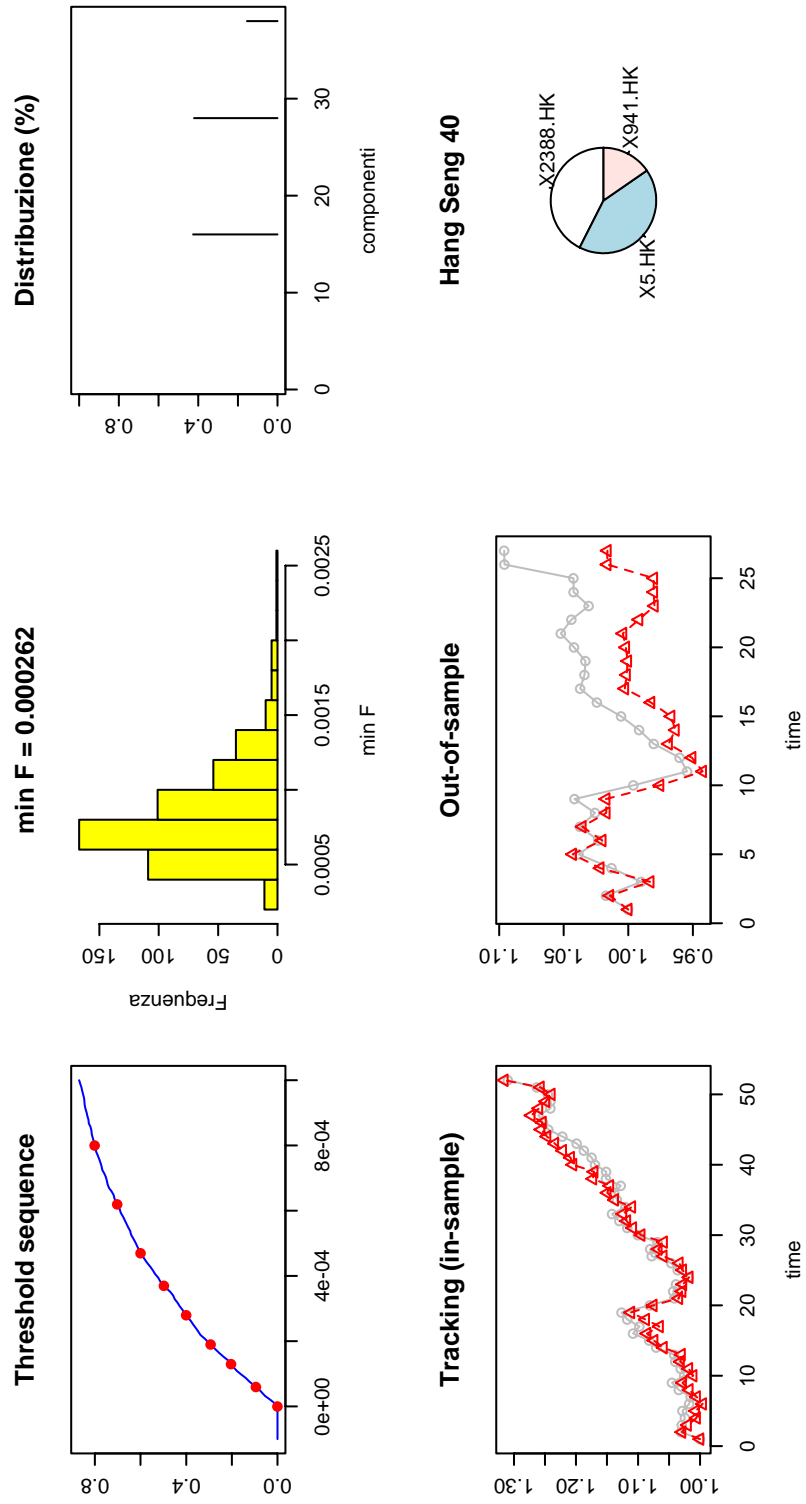


Figura A.2: Portafoglio con 3 titoli riprodotto l'indice Hang Seng

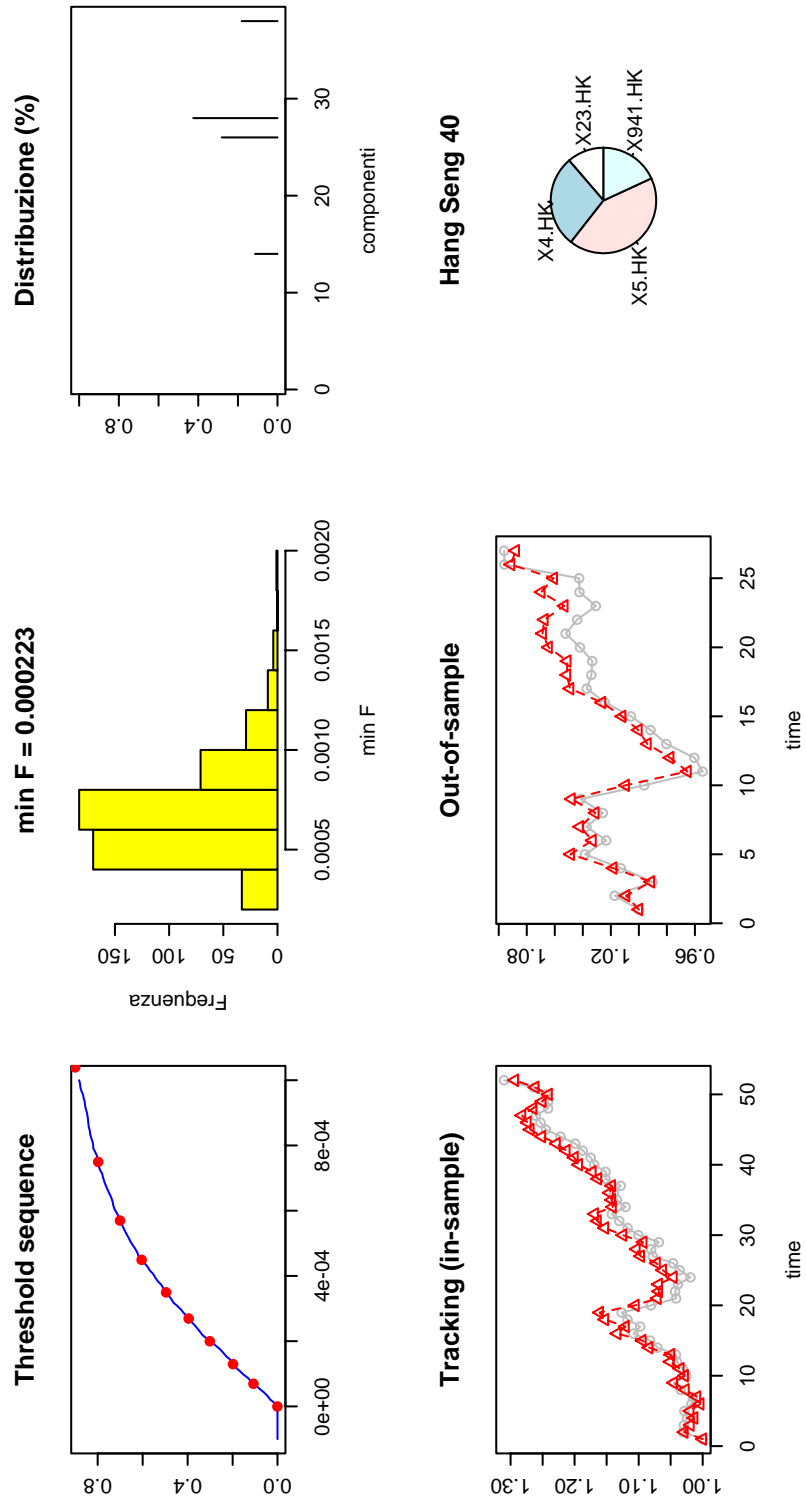


Figura A.3: Portafoglio con 4 titoli riprodotte l'indice Hang Seng

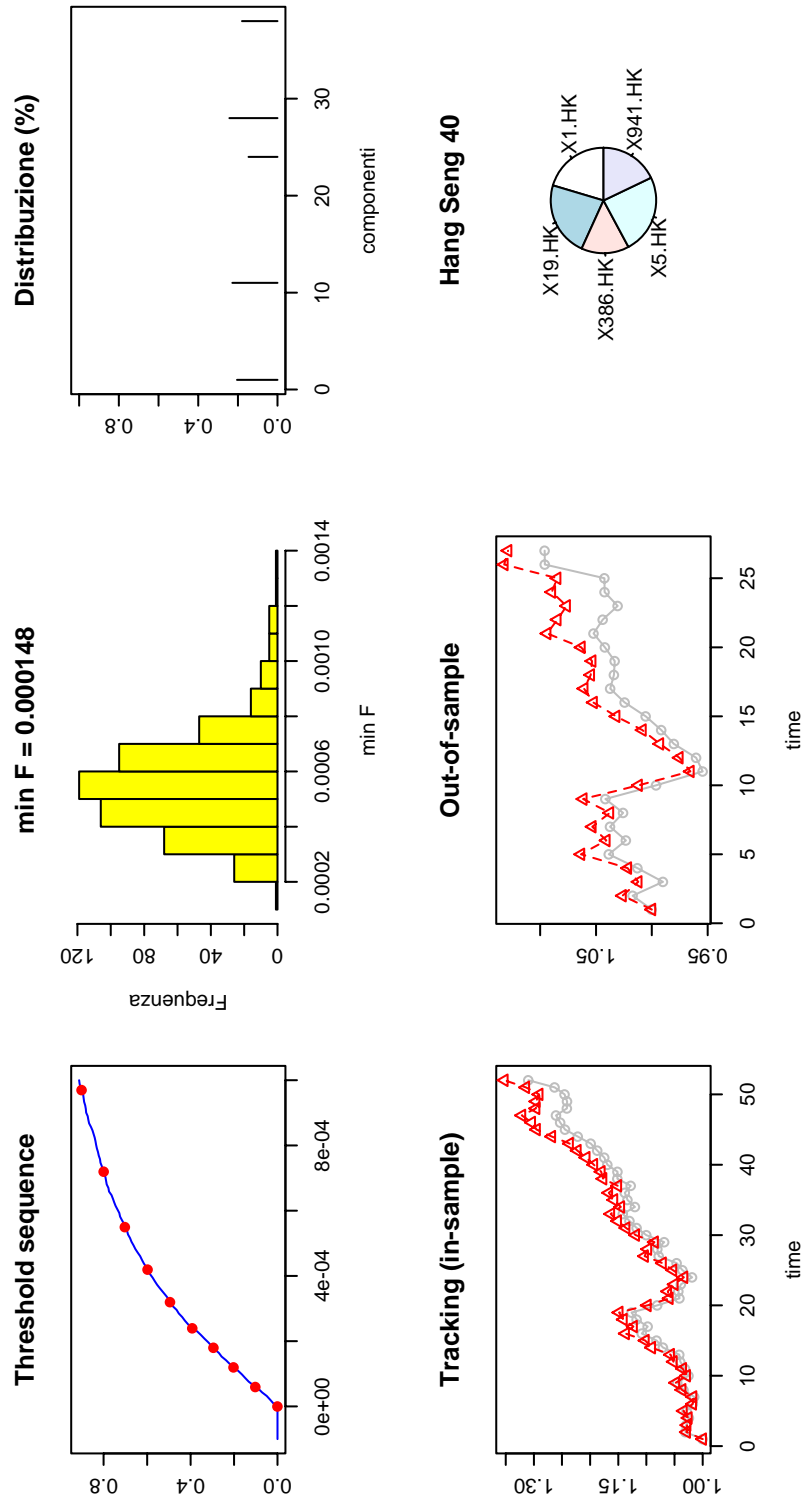


Figura A.4: Portafoglio con 5 titoli riprodotte l'indice Hang Seng

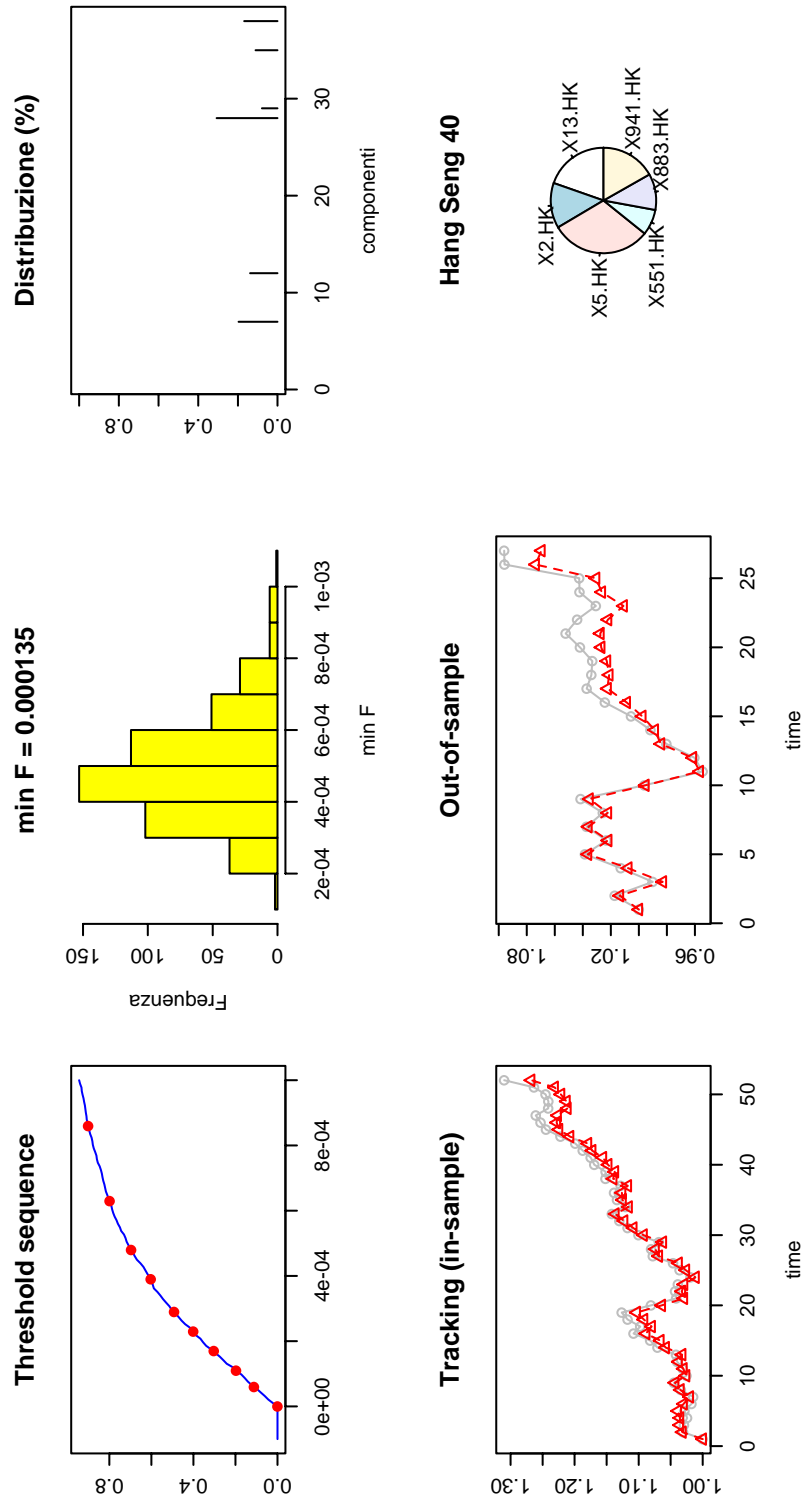


Figura A.5: Portafoglio con 6 titoli riprodotto l'indice Hang Seng

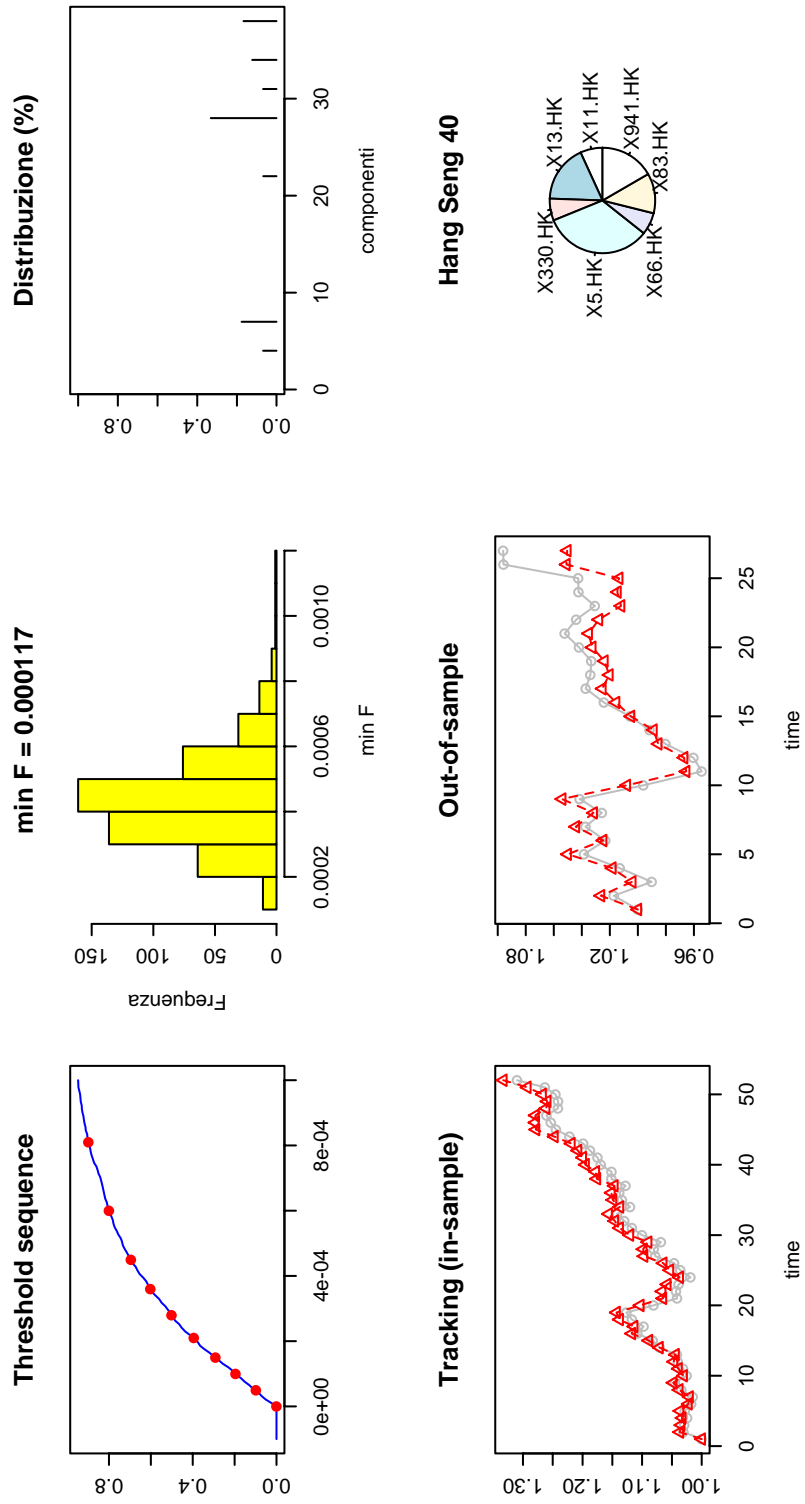


Figura A.6: Portafoglio con 7 titoli riprodotte l'indice Hang Seng

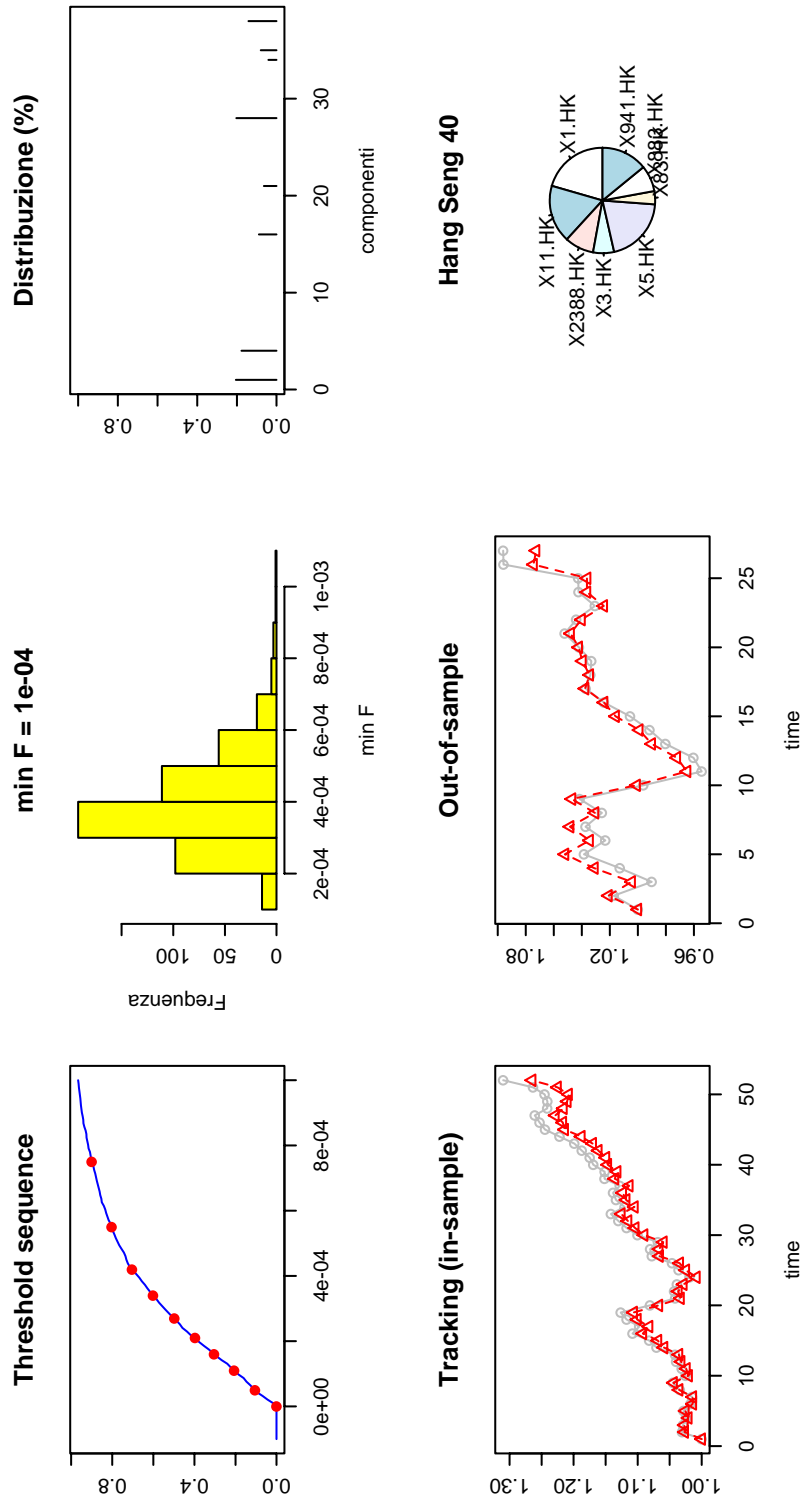


Figura A.7: Portafoglio con 8 titoli riprodotte l'indice Hang Seng

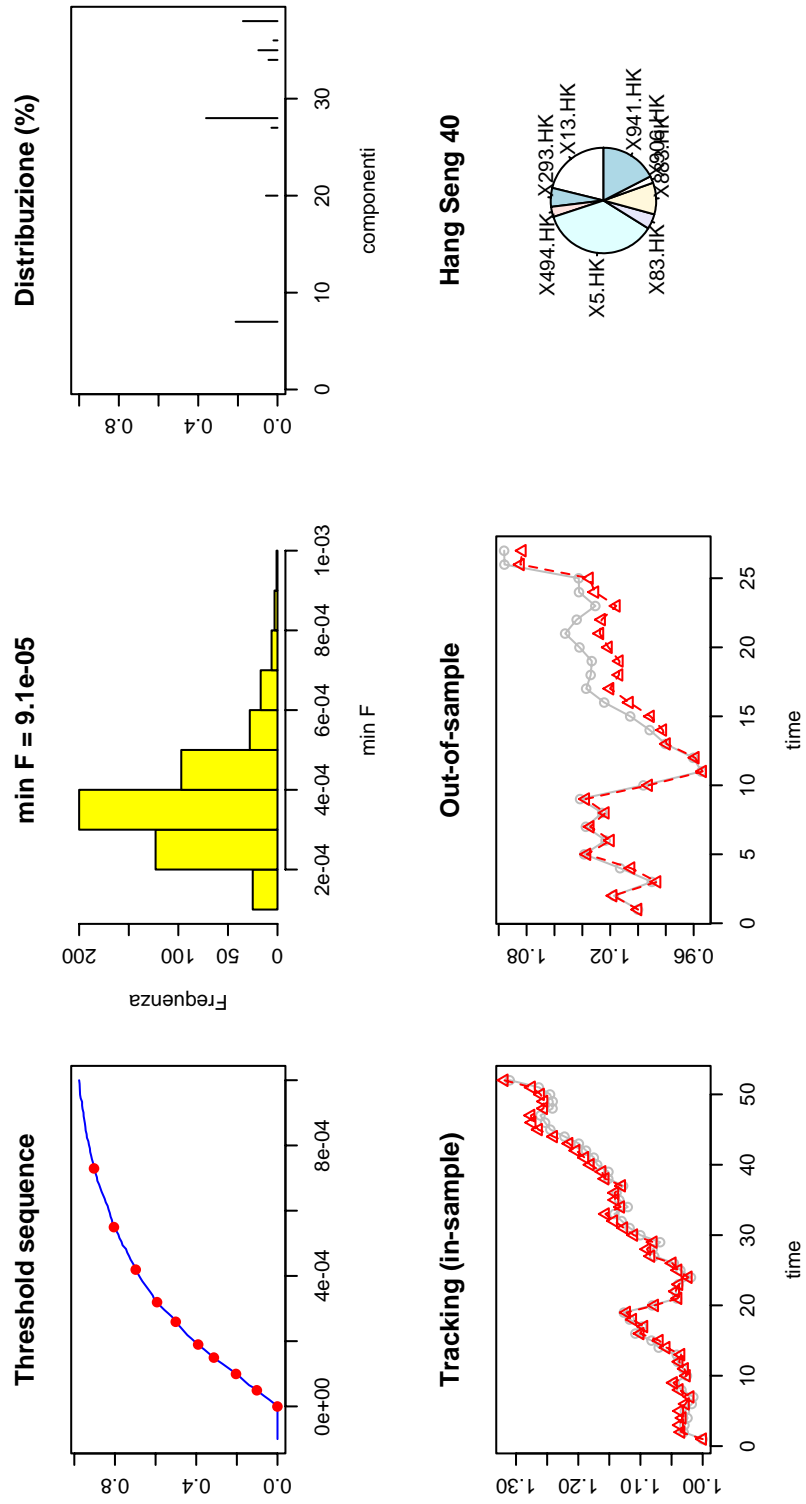


Figura A.8: Portafoglio con 9 titoli riprodotto l'indice Hang Seng

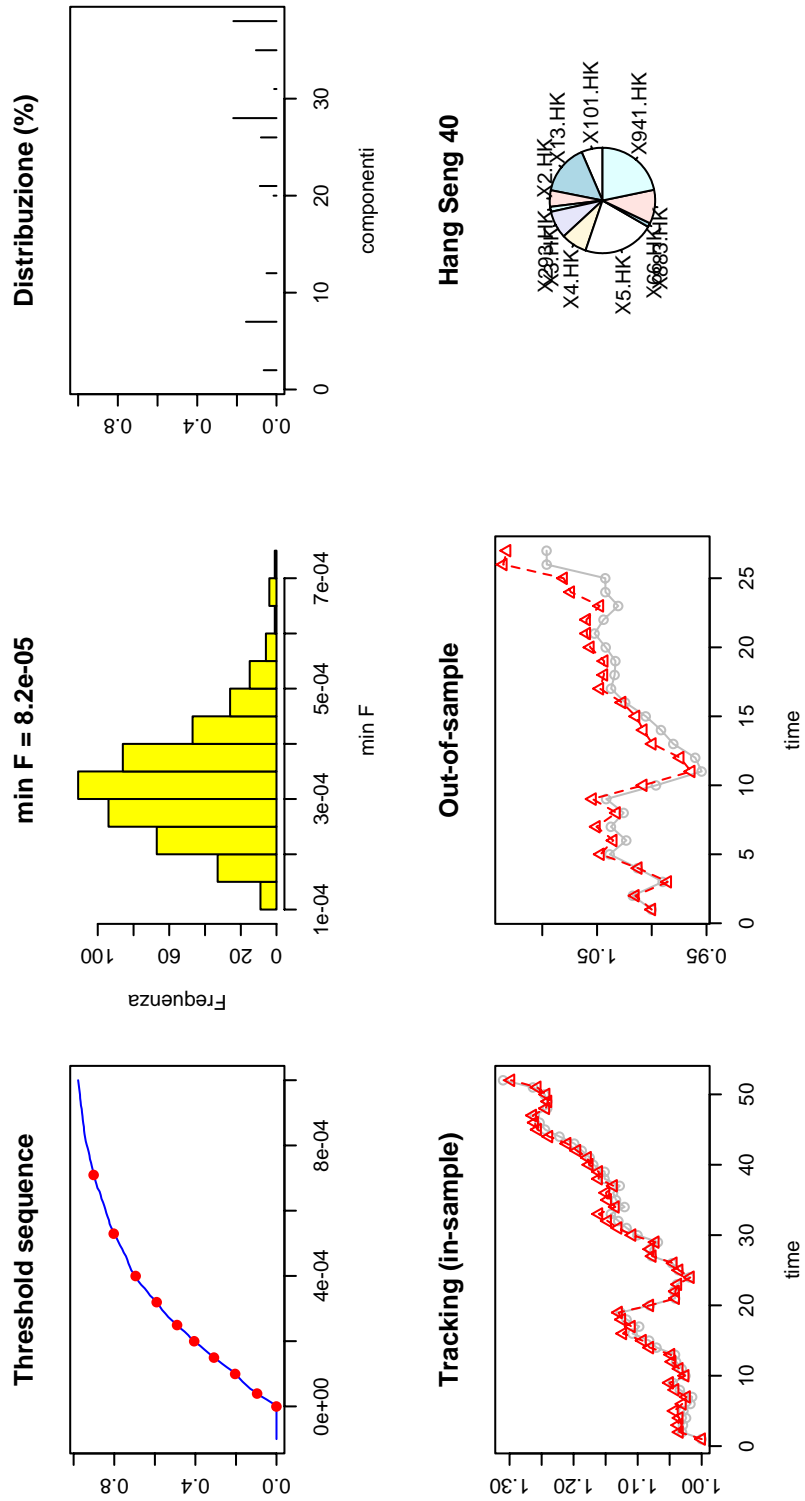


Figura A.9: Portafoglio con 10 titoli riprodotto l'indice Hang Seng

Appendice B

Risultati delle esecuzione del *tracking* dell'indice S&P 100

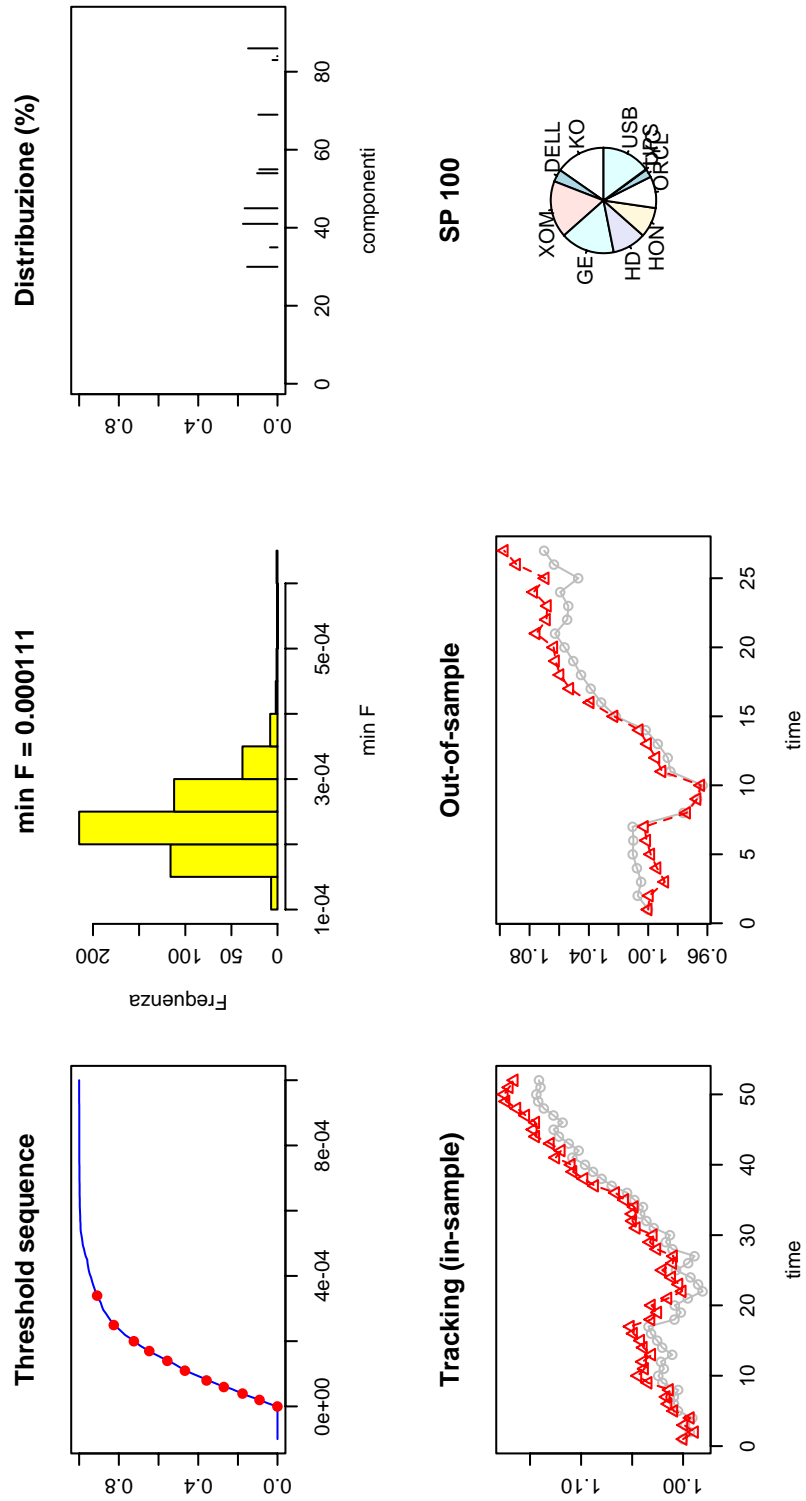


Figura B.1: Portafoglio con 10 titoli riprodotte l'indice S&P 100

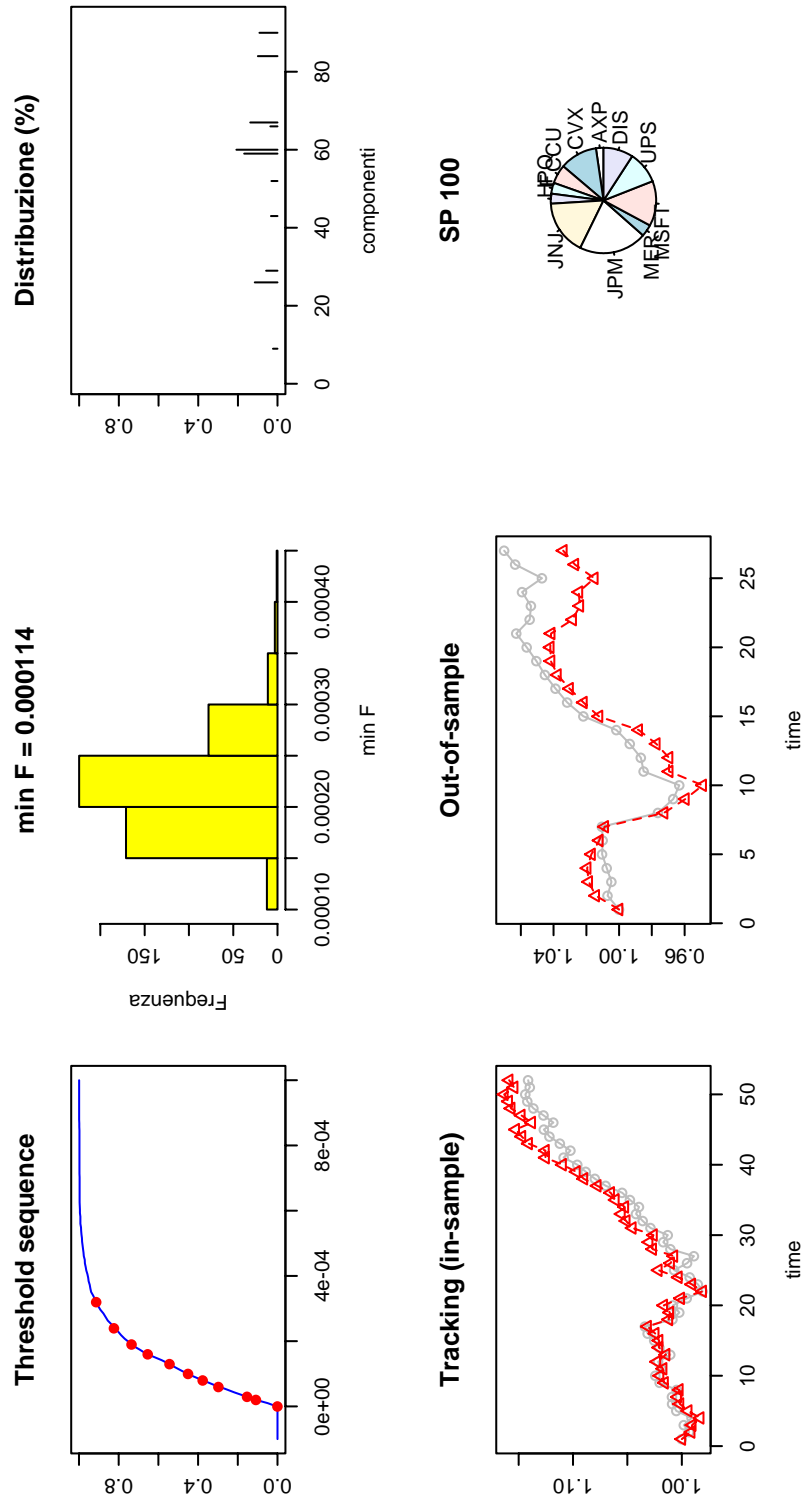


Figura B.2: Portafoglio con 11 titoli riprodotte l'indice S&P 100

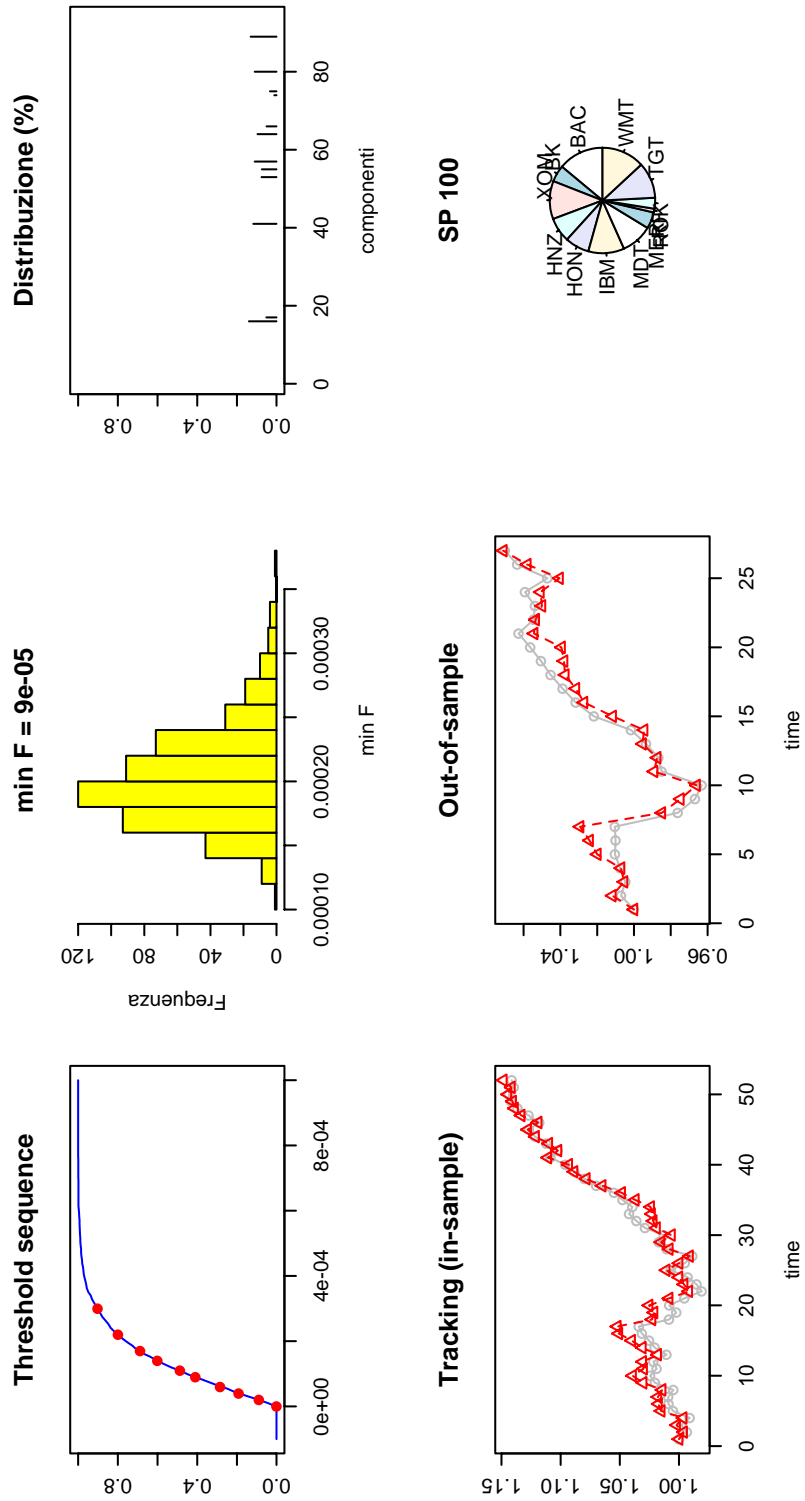


Figura B.3: Portafoglio con 12 titoli riprodotto l'indice S&P 100

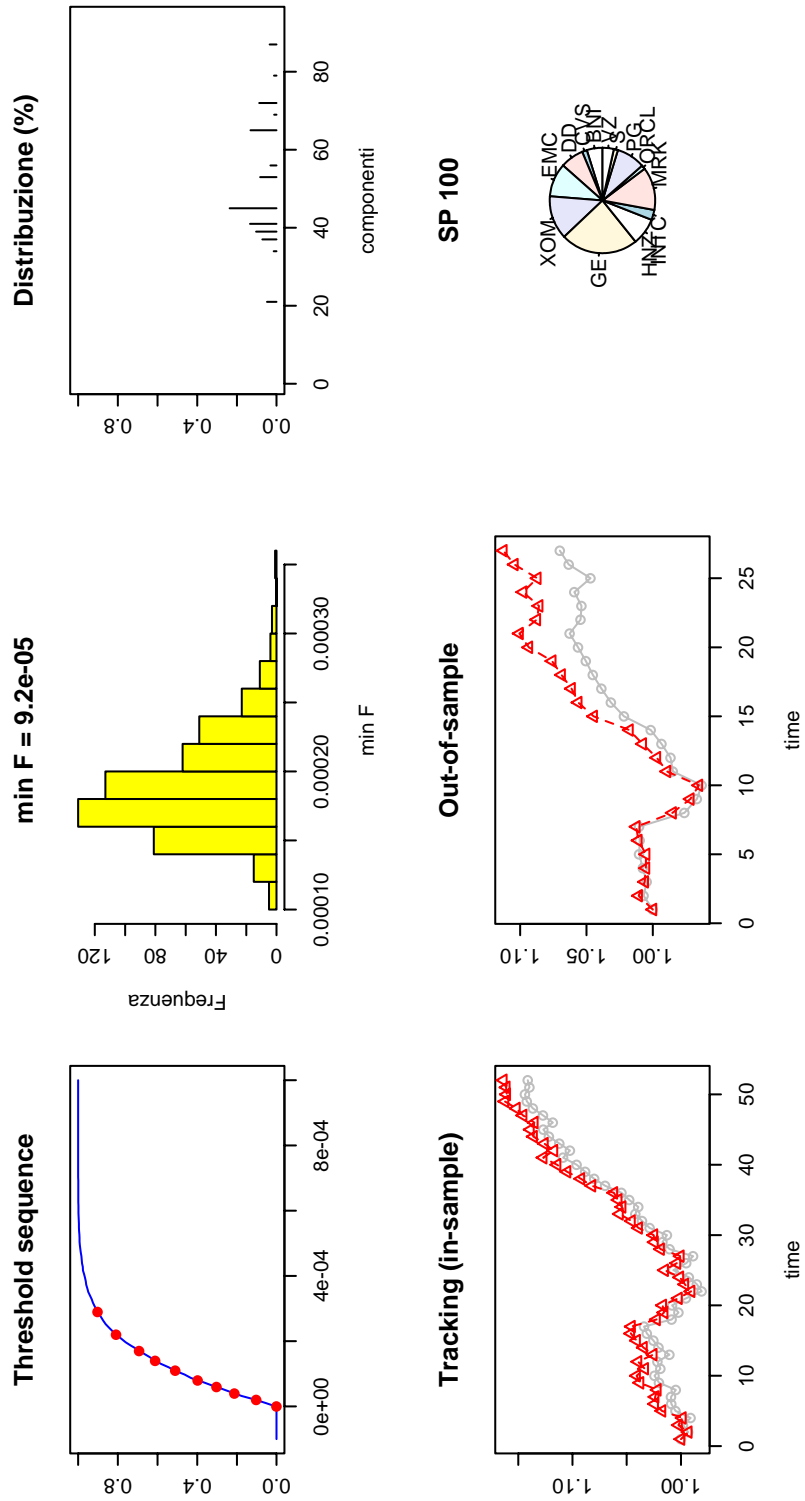


Figura B.4: Portafoglio con 13 titoli riprodotto l'indice S&P 100

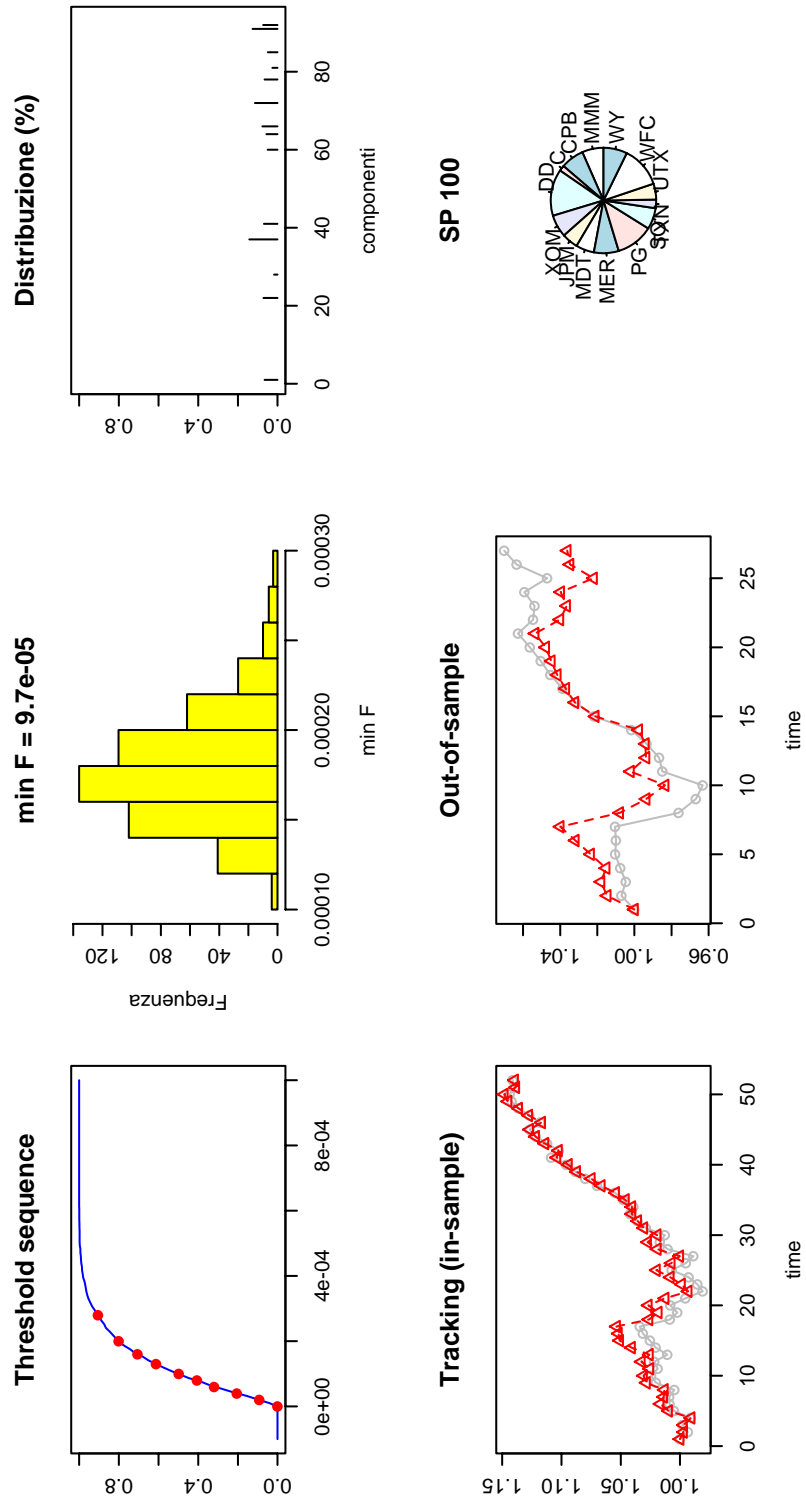


Figura B.5: Portafoglio con 14 titoli riprodotte l'indice S&P 100

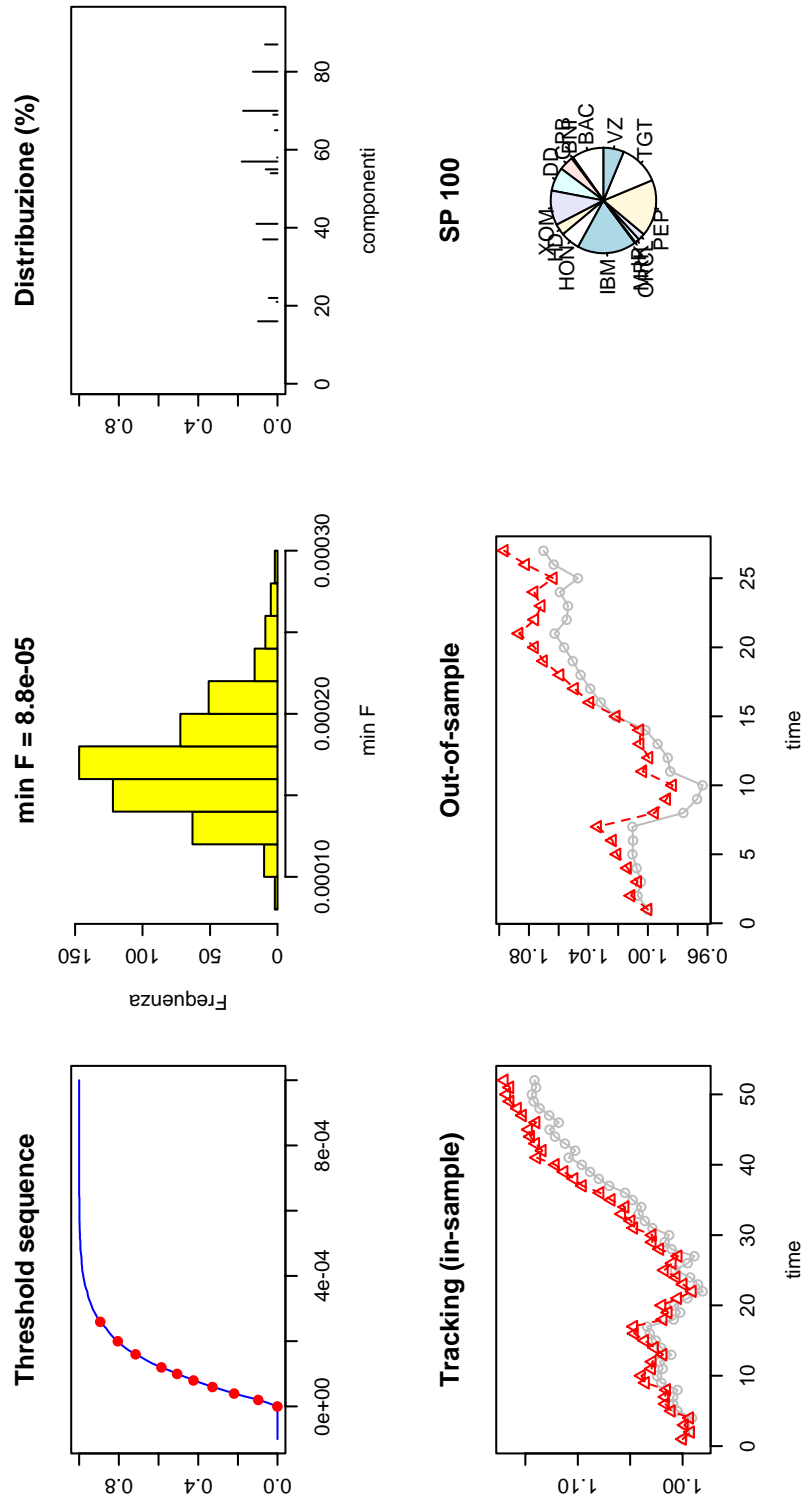


Figura B.6: Portafoglio con 15 titoli riprodotte l'indice S&P 100

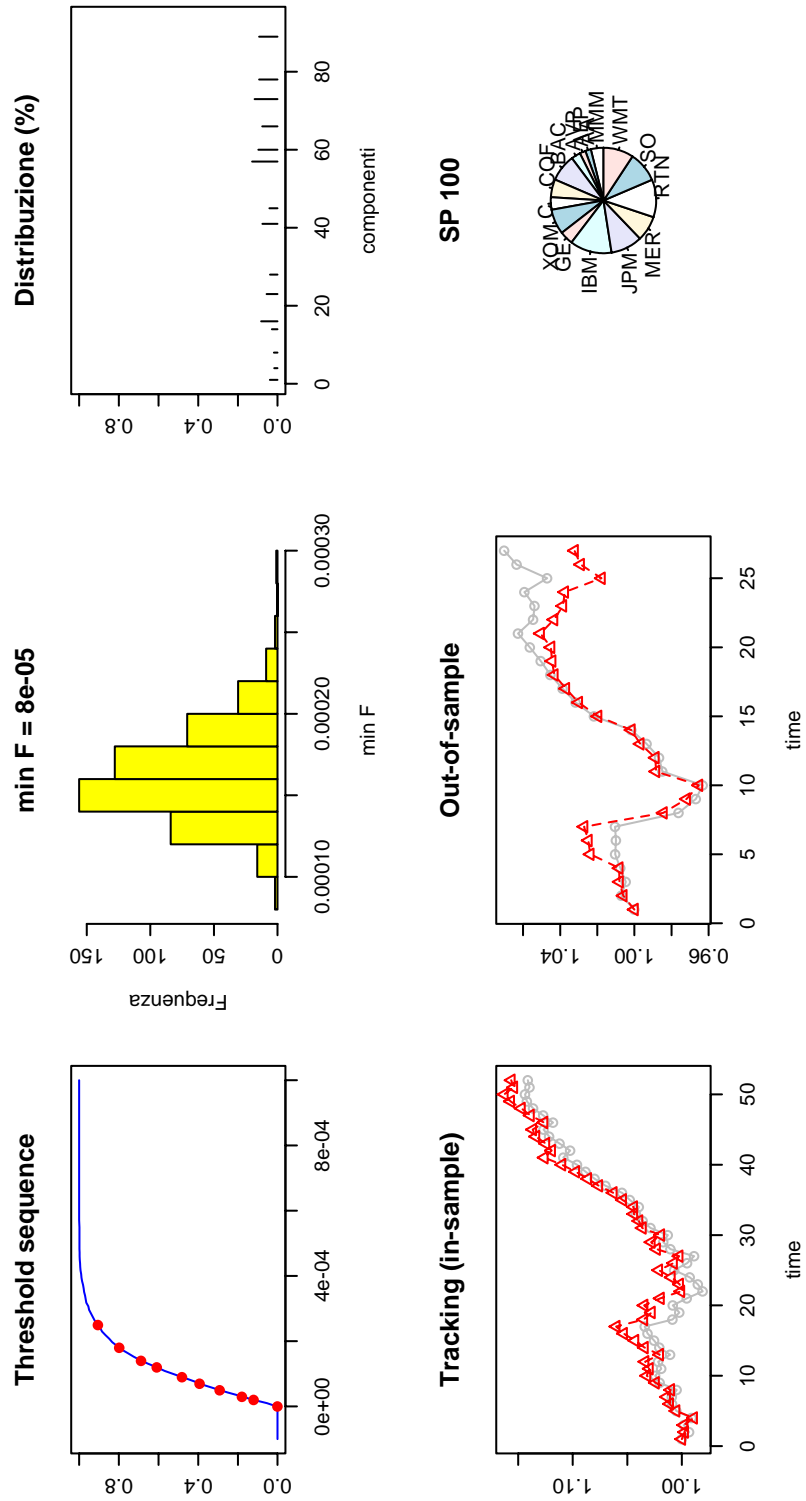


Figura B.7: Portafoglio con 16 titoli riprodotto l'indice S&P 100

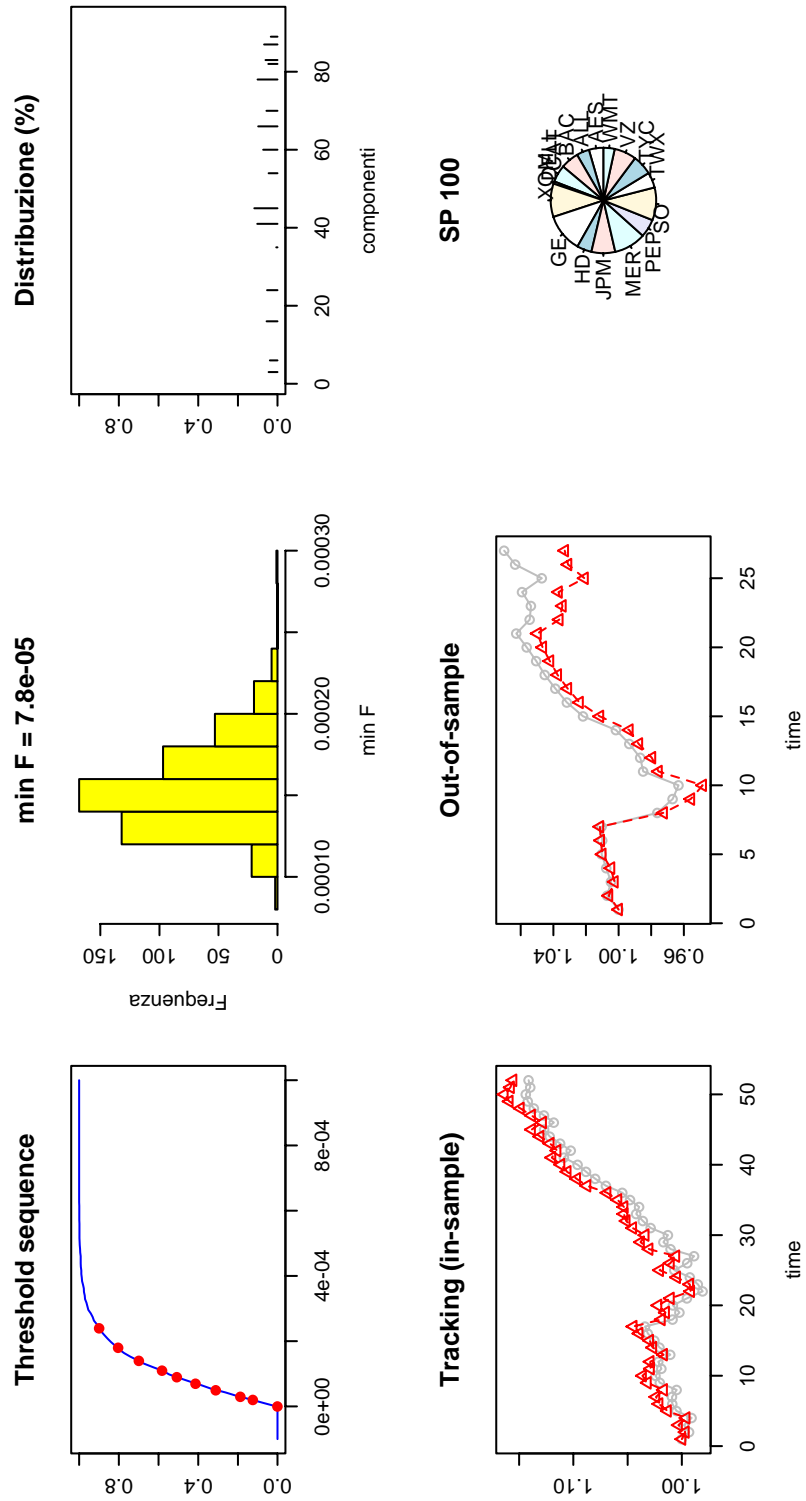


Figura B.8: Portafoglio con 17 titoli riprodotto l'indice S&P 100

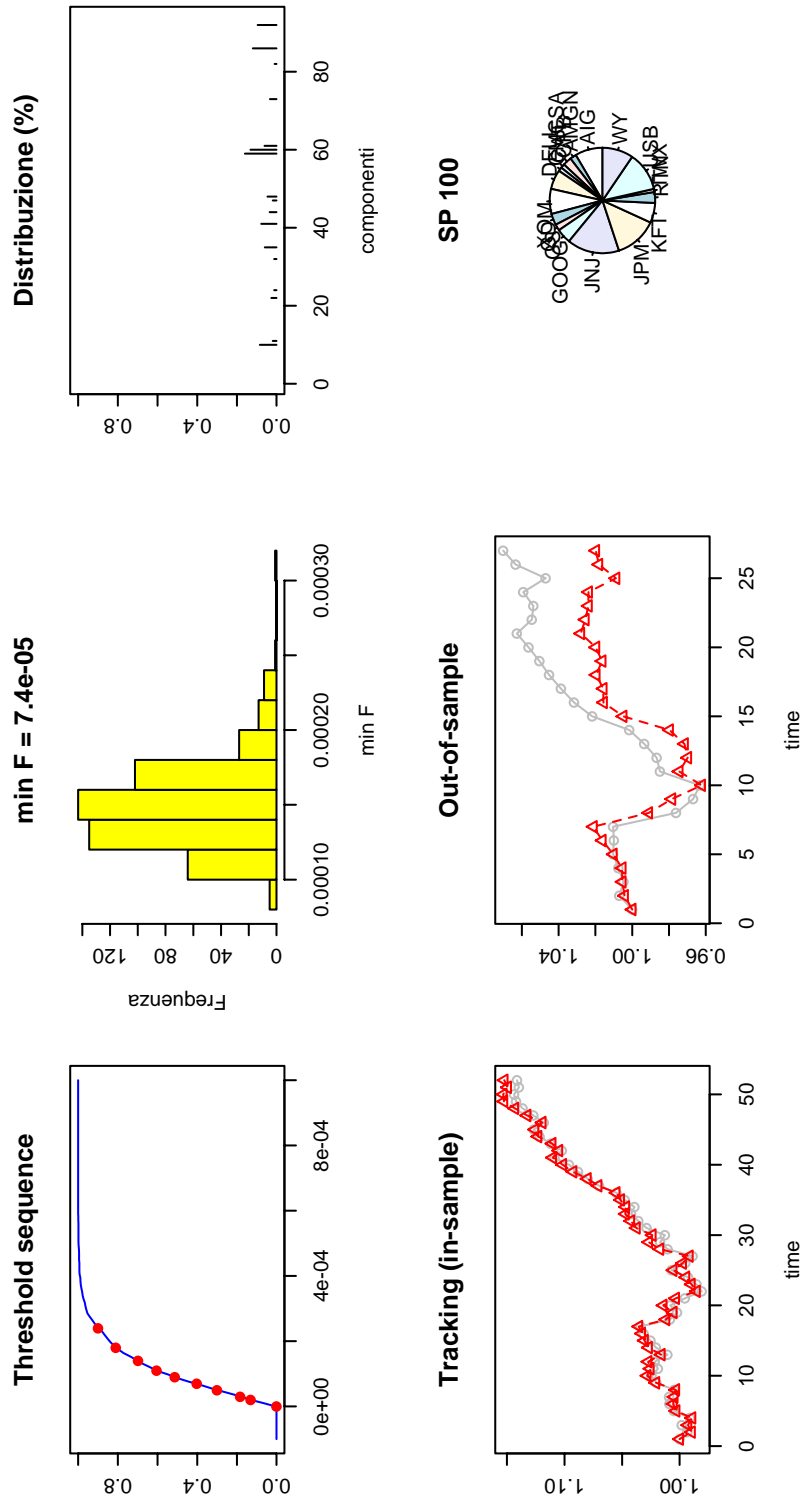


Figura B.9: Portafoglio con 18 titoli riprodotte l'indice S&P 100

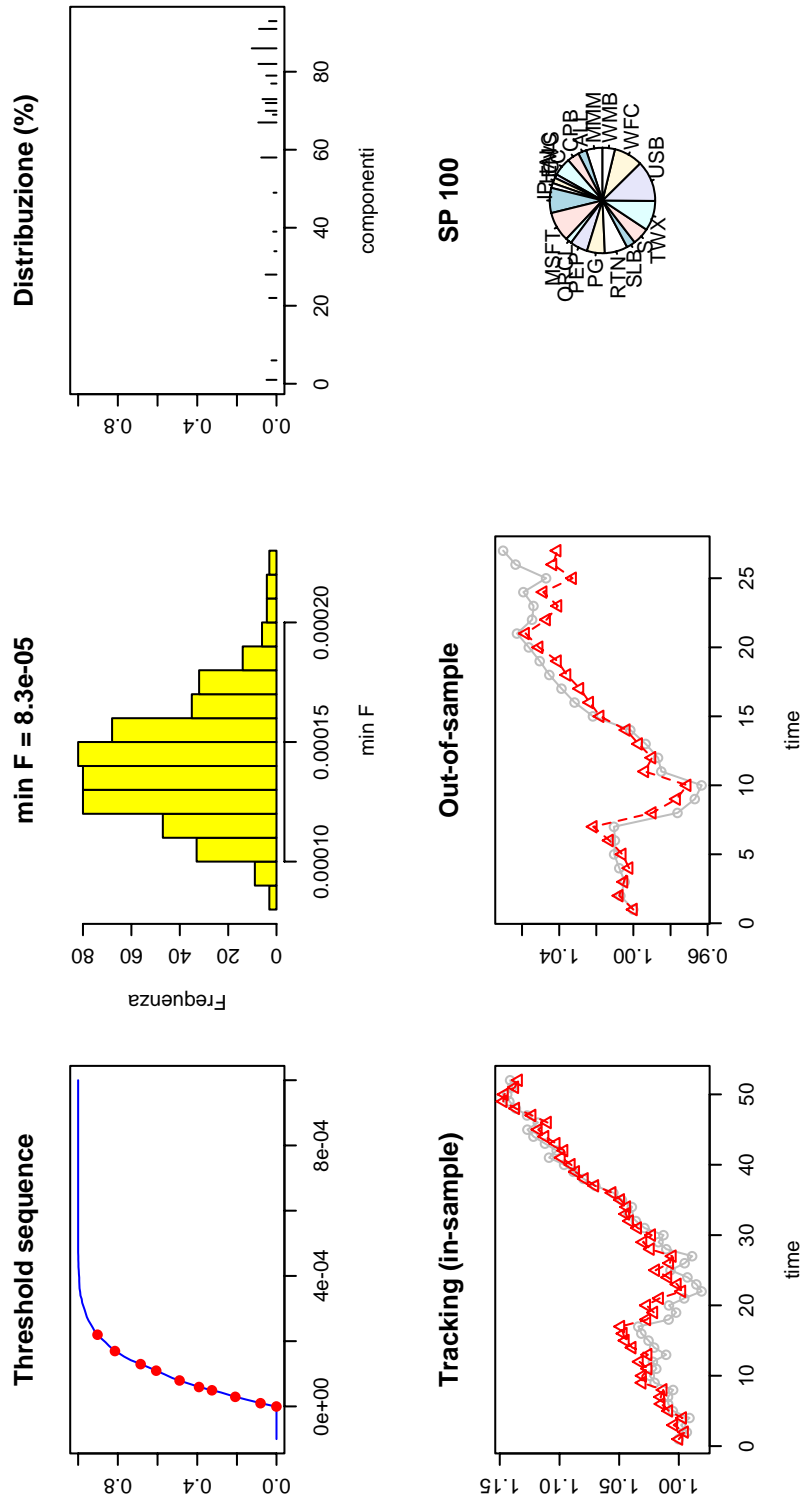


Figura B.10: Portafoglio con 19 titoli riprodotto l'indice S&P 100

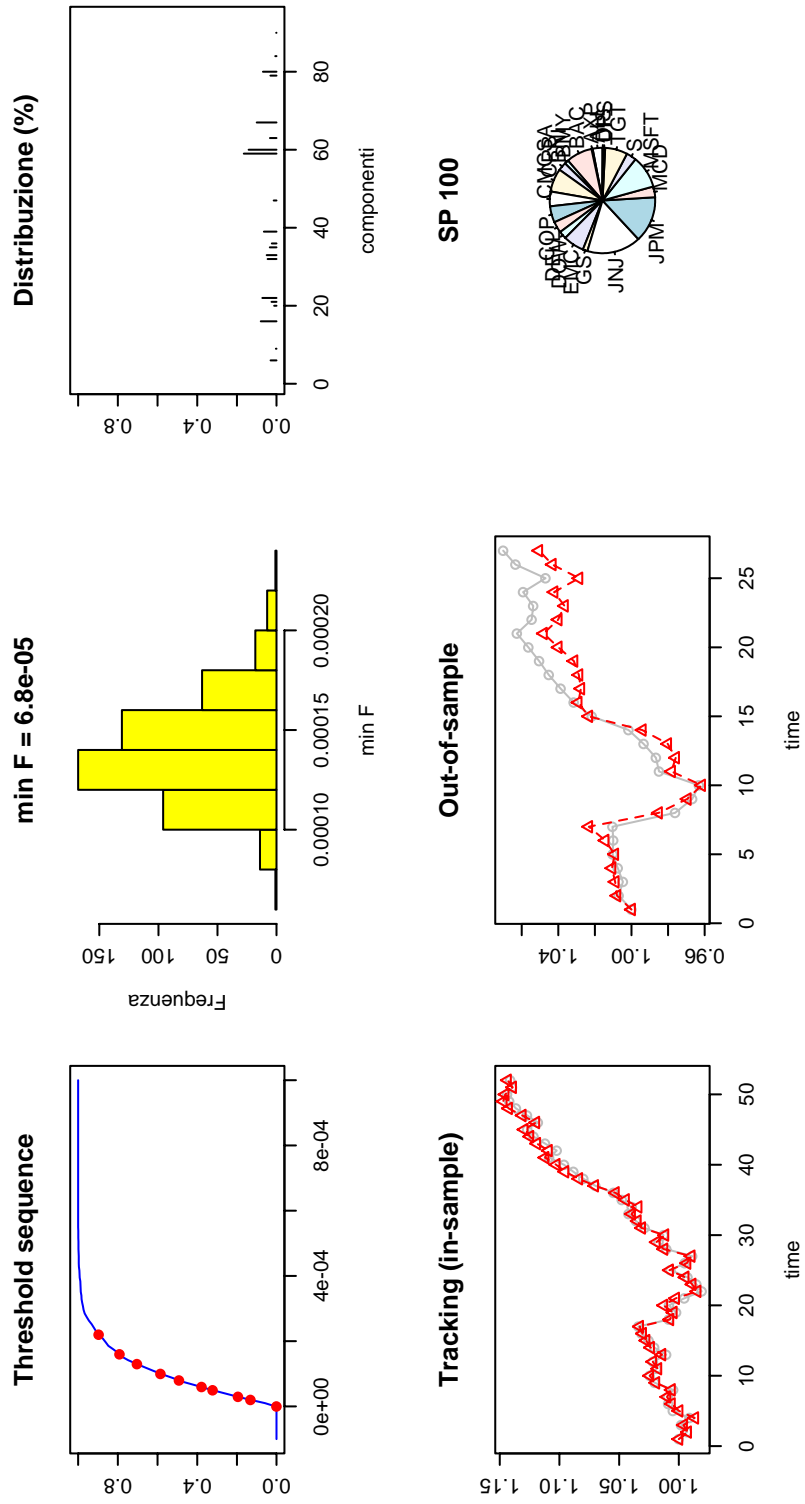


Figura B.11: Portafoglio con 20 titoli riprodotte l'indice S&P 100

Appendice C

Risultati delle esecuzione del *tracking* dell'indice MIB 30

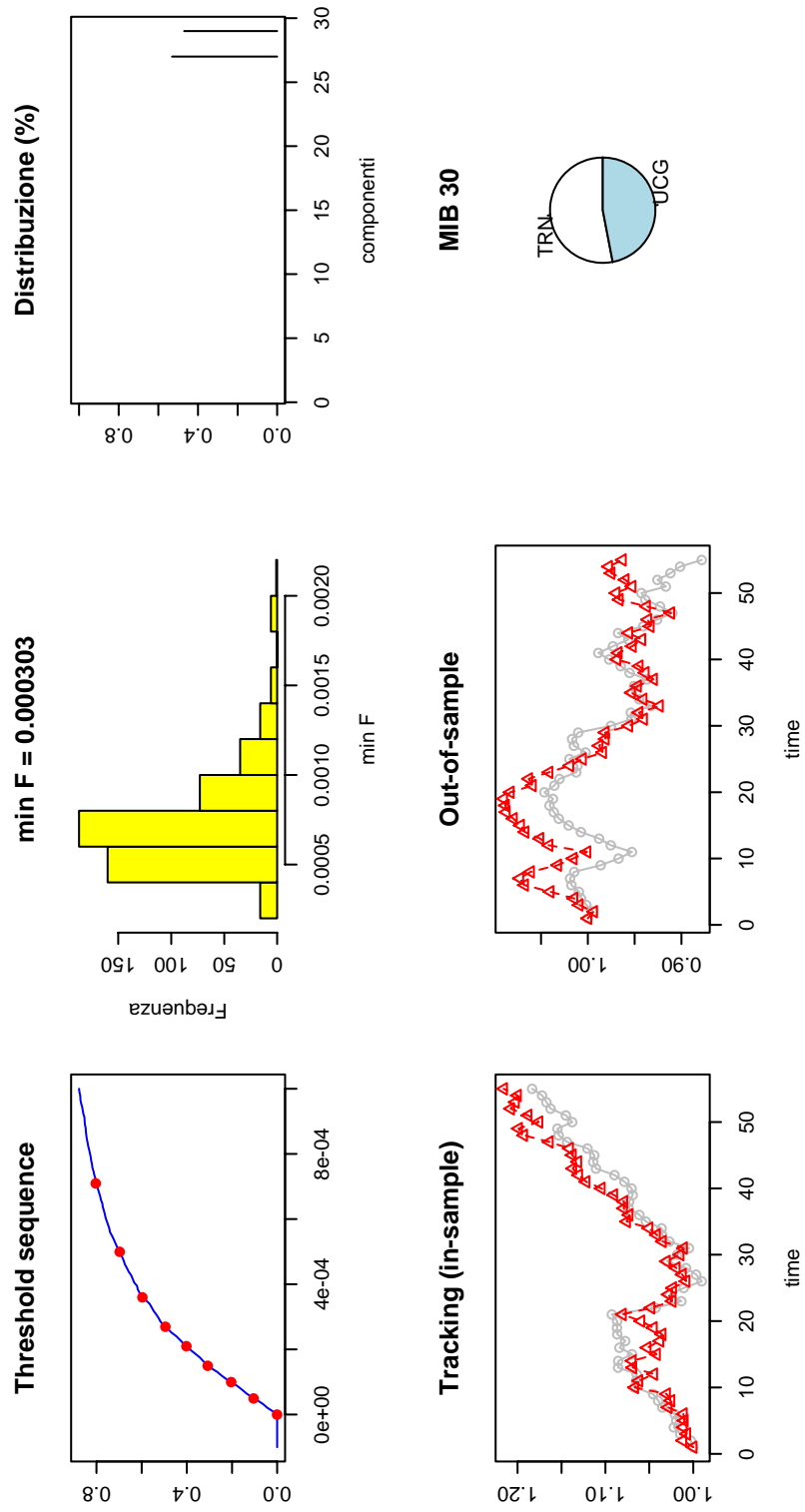


Figura C.1: Portafoglio con 2 titoli riprodotte l'indice MIB 30

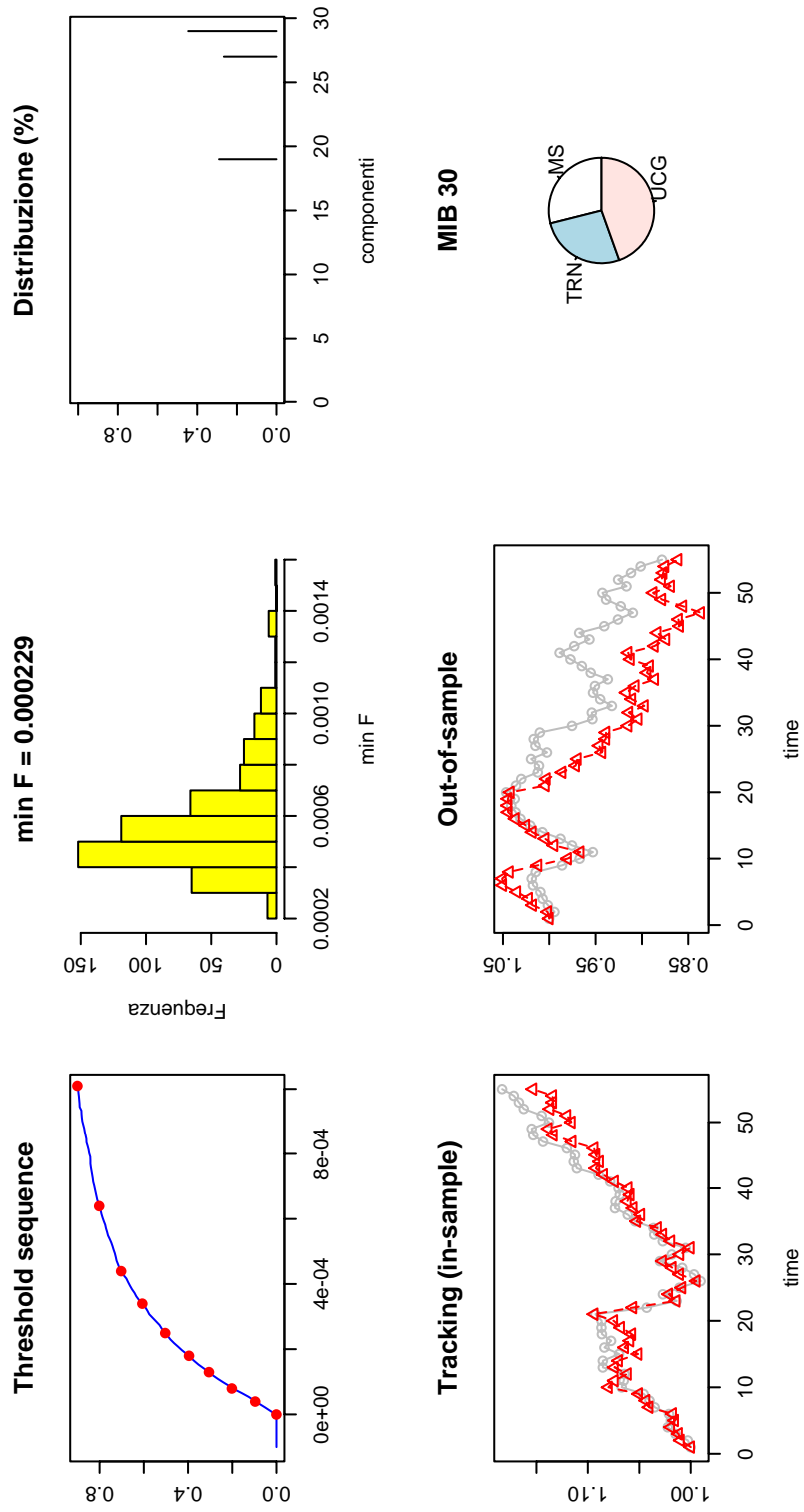


Figura C.2: Portafoglio con 3 titoli riprodotto l'indice MIB 30

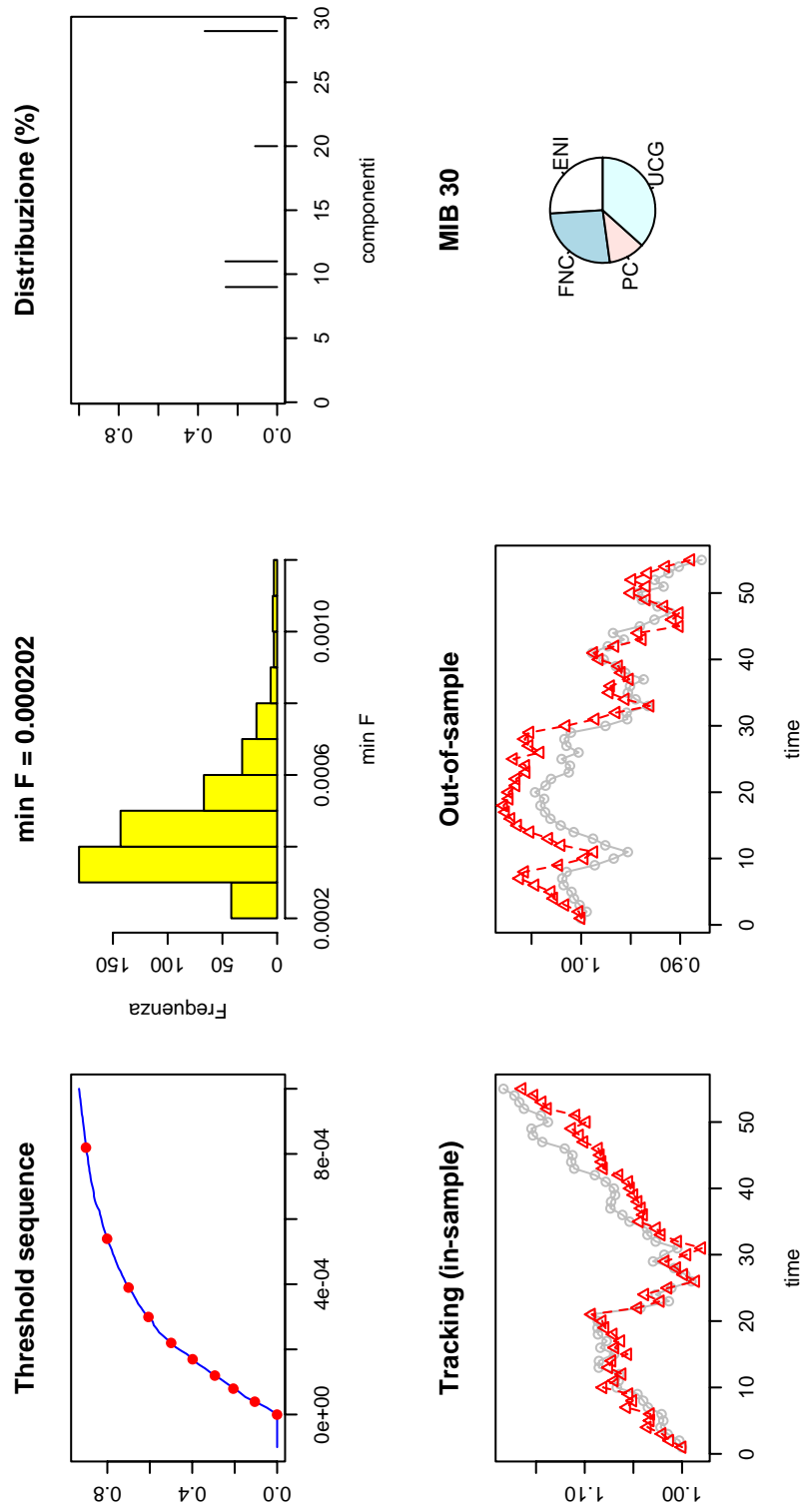


Figura C.3: Portafoglio con 4 titoli riprodotto l'indice MIB 30

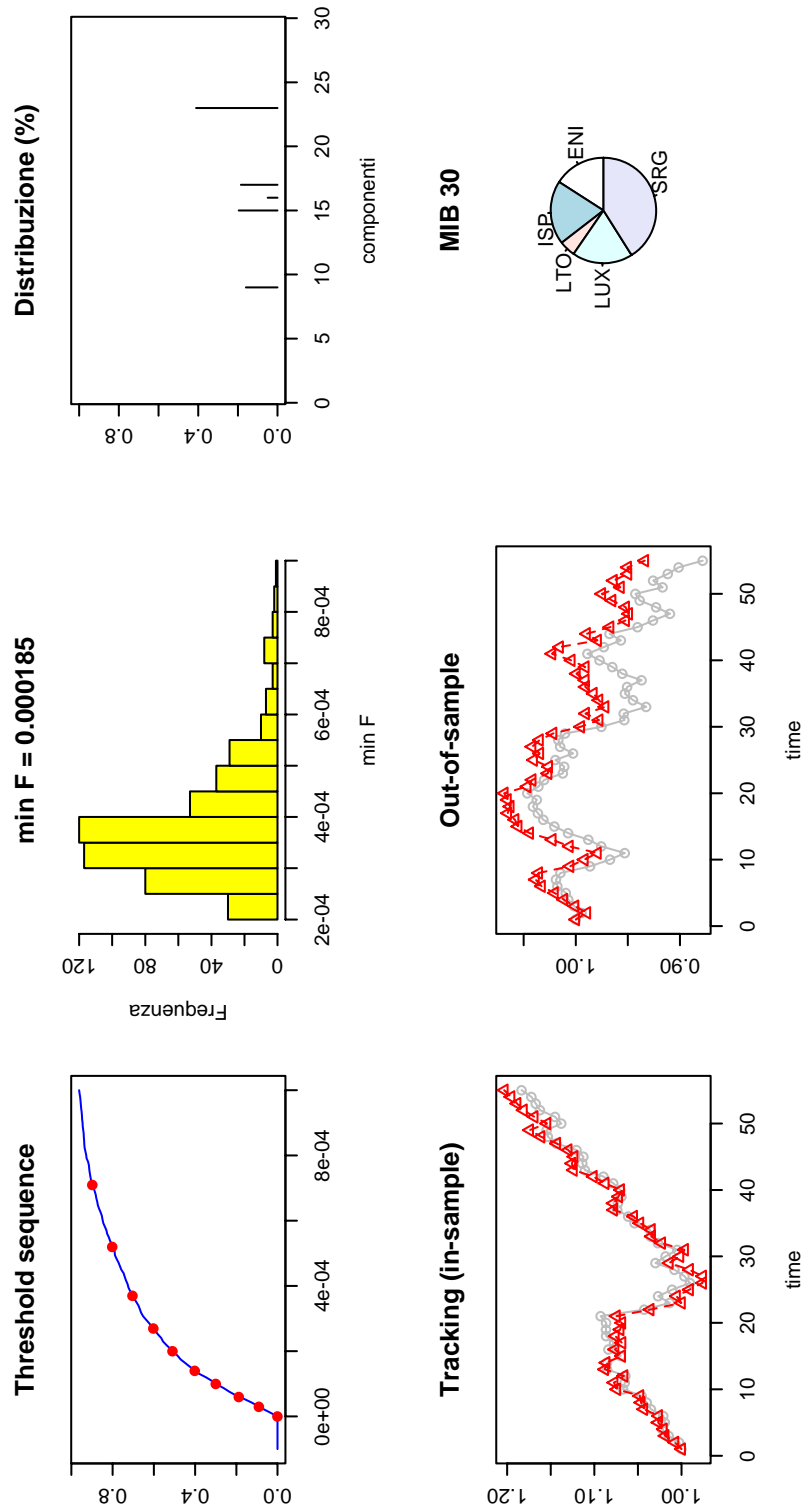


Figura C.4: Portafoglio con 5 titoli riprodotto l'indice MIB 30

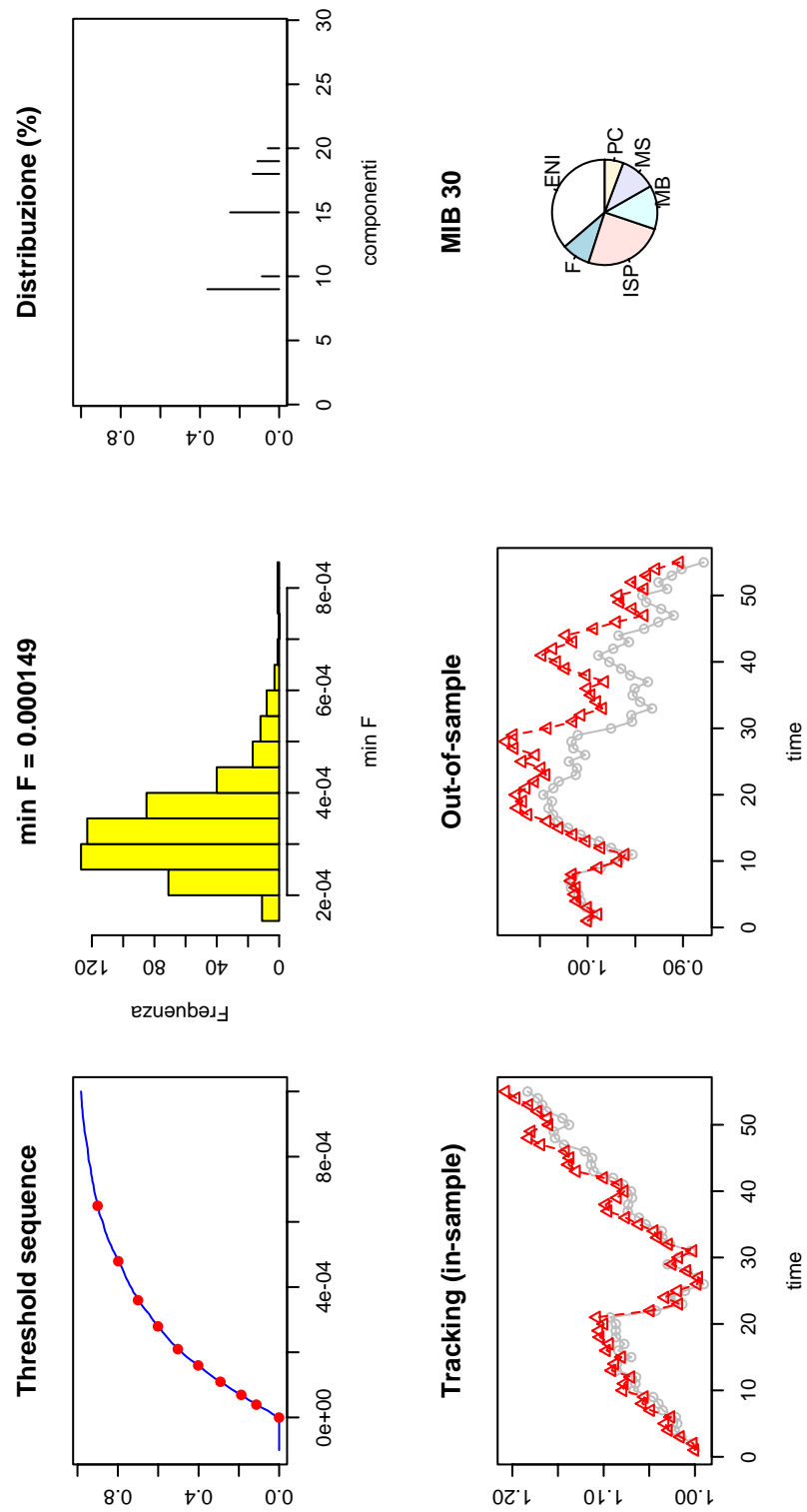


Figura C.5: Portafoglio con 6 titoli riprodotte l'indice MIB 30

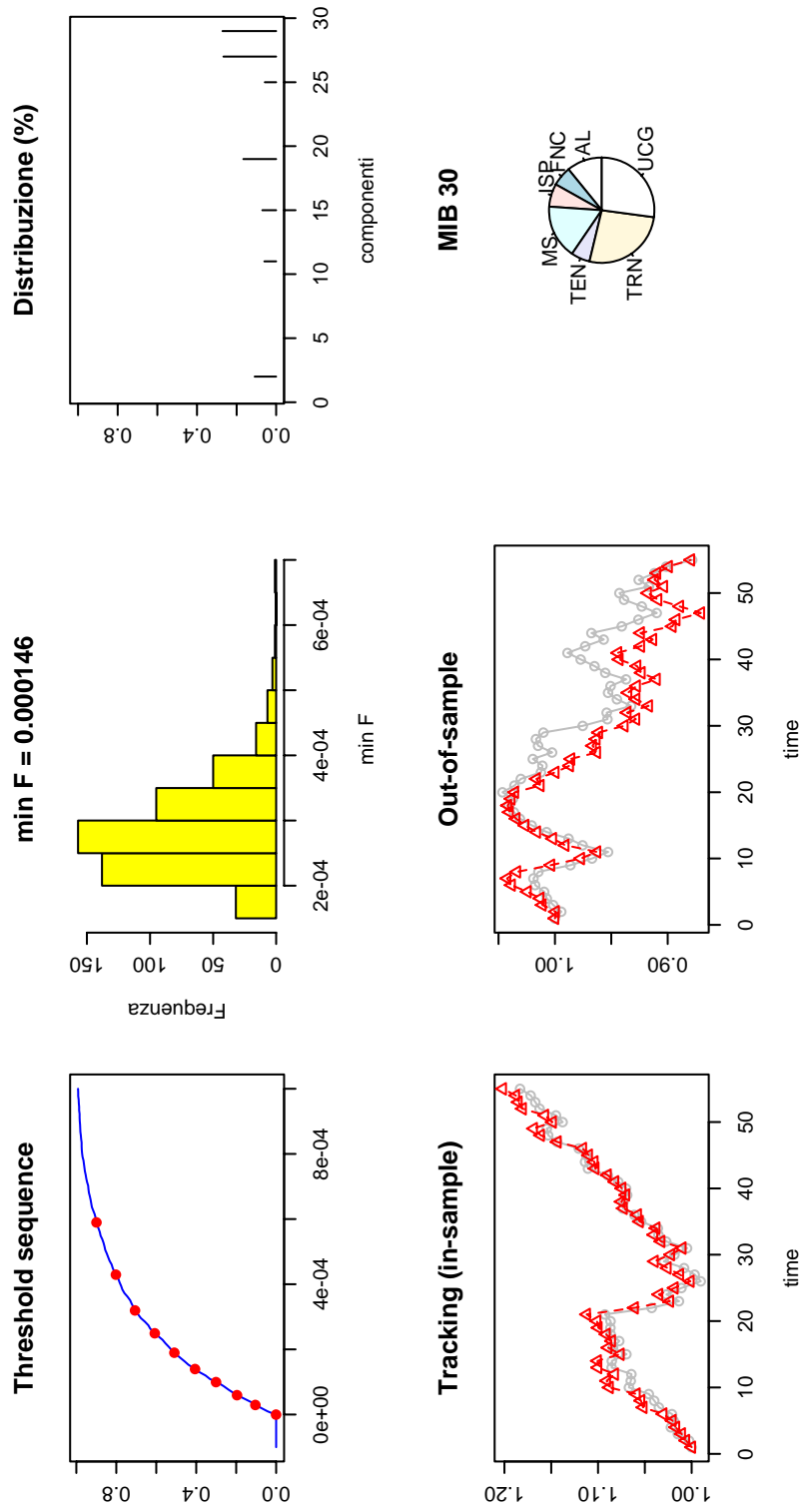


Figura C.6: Portafoglio con 7 titoli riprodotto l'indice MIB 30

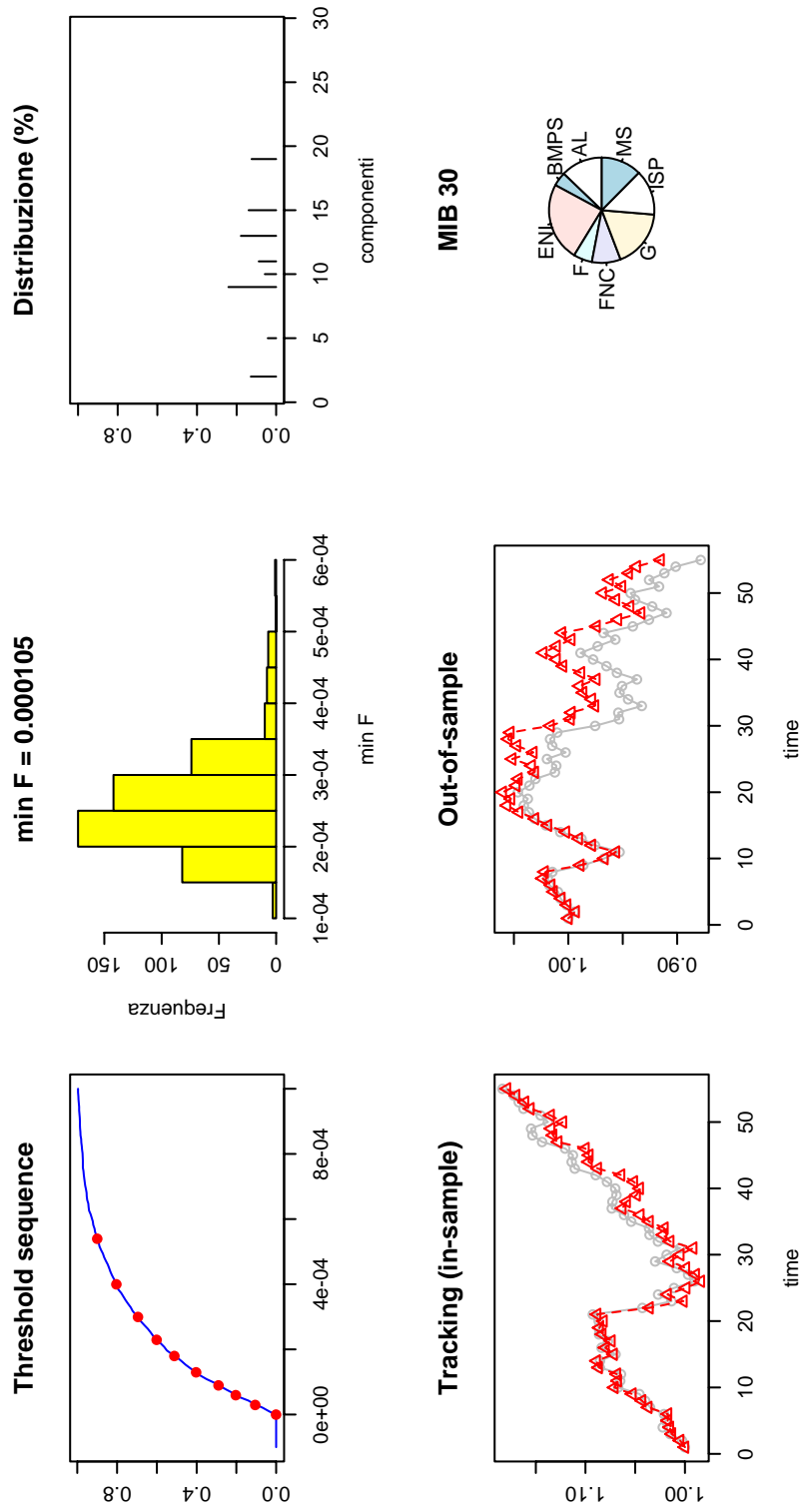


Figura C.7: Portafoglio con 8 titoli riprodotte l'indice MIB 30

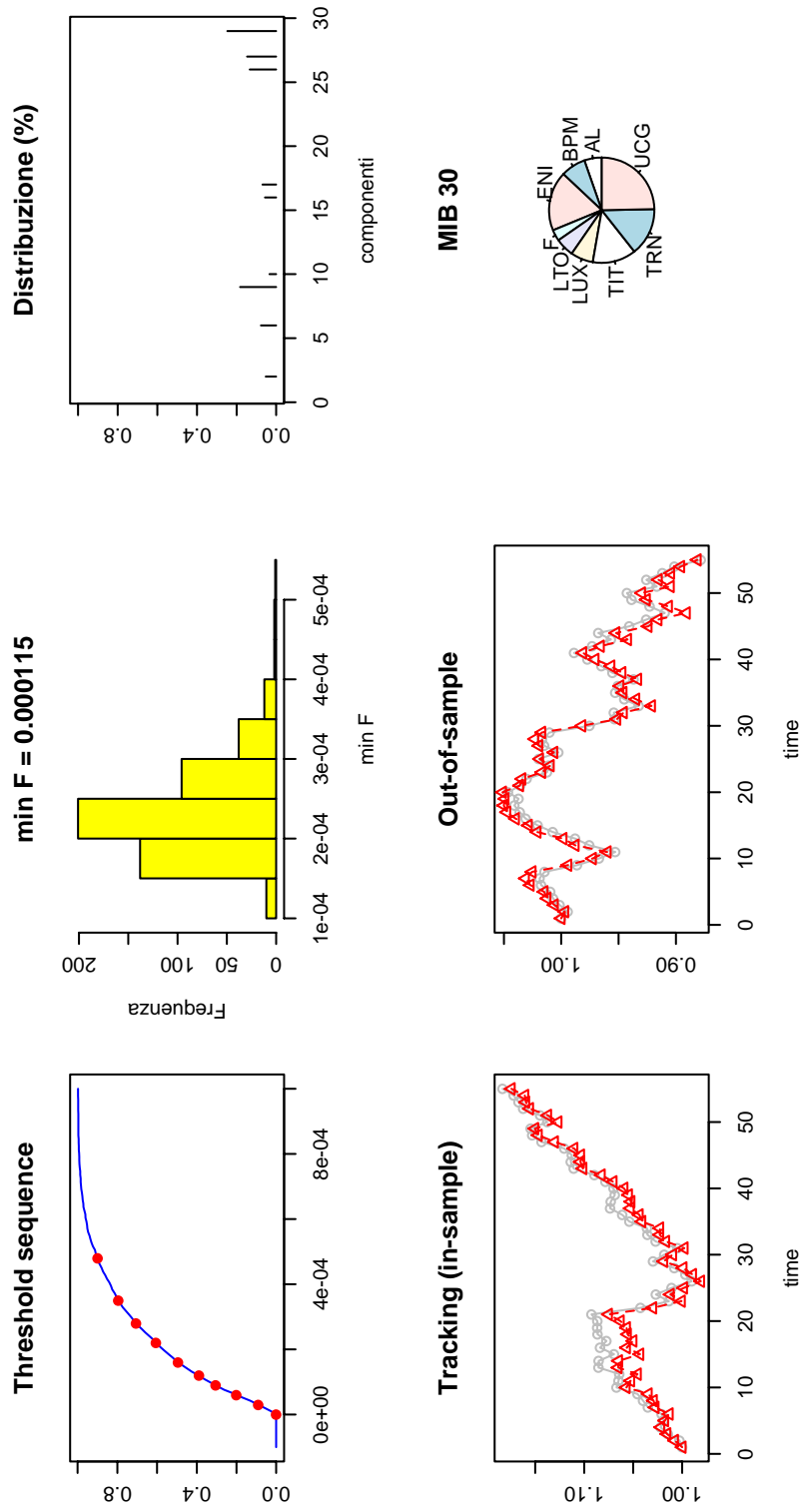


Figura C.8: Portafoglio con 9 titoli riprodotte l'indice MIB 30

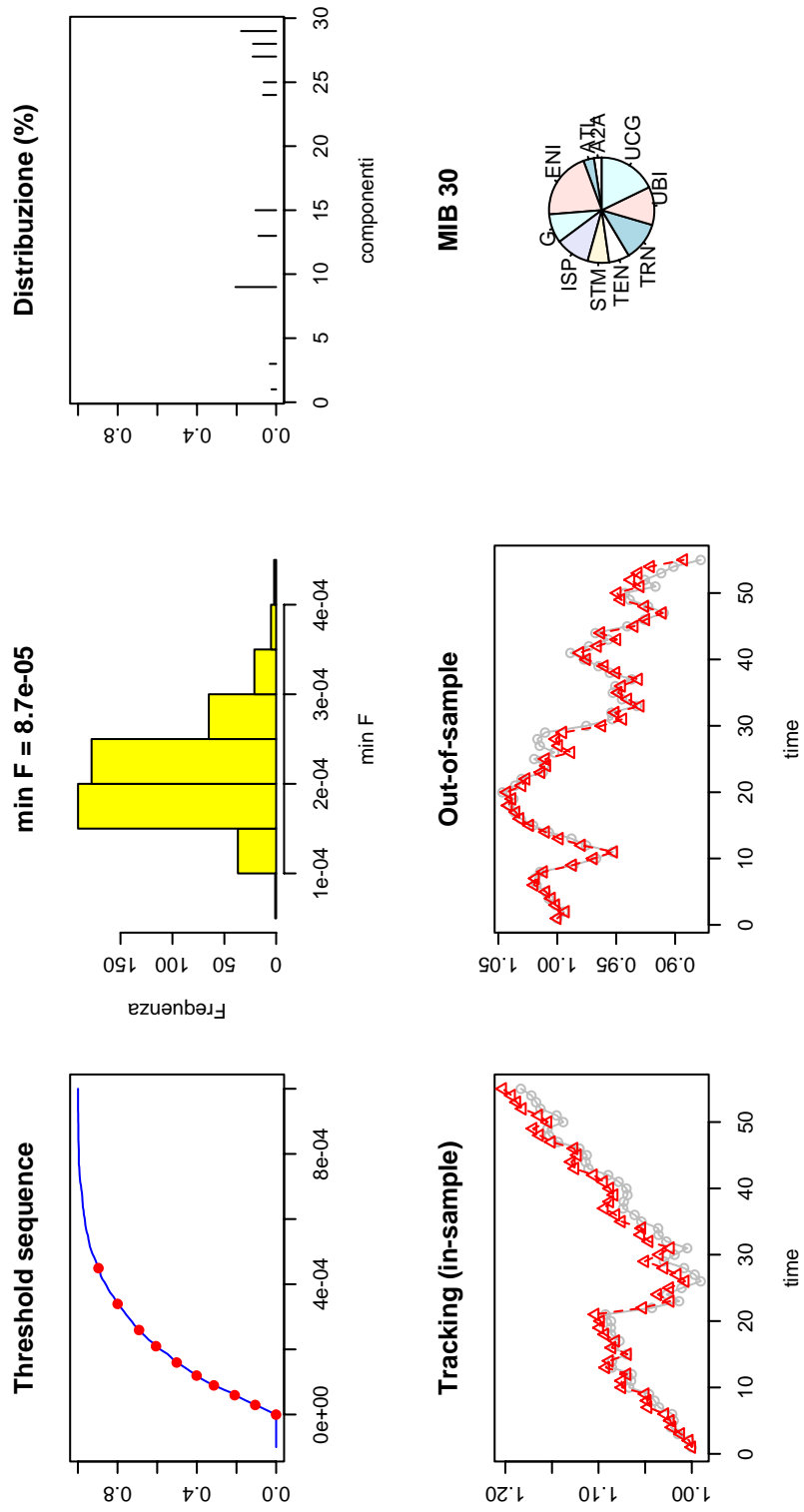


Figura C.9: Portafoglio con 10 titoli riprodotte l'indice MIB 30

Bibliografia

- [1] Aarts, E. H. L., Korst, J. H. M. and Van Laarhoven, P. J. M. 1997. *Simulated annealing*: in E. Aarts and J. K. Lenstra (eds), *Local Search in Combinatorial Optimization*, Wiley, Chister.
- [2] Aarts, E. H. L. and Lenstra, J. K. 1997. *Local search in combinatorial optimization: Introduction*: Wiley, Chichester.
- [3] Aarts, E. H. L., Lenstra, J. K., van Laarhoven, P. J. M. and Ulder, N. L. J. 1994. *A computational study of local search algorithms for job shop scheduling*: *Journal of Computing* 6.
- [4] Aguirregabiria, V. 1999. *The Dynamics of Markups and Inventories in Retailing Firms*: *The Review of Economic Studies*, 66: 275-308.
- [5] Aho, A. V., Hopcroft, J. E. and Ullman, J. D. 1974. *The Design and Analysis of Computer Algorithms*: Addison-Wesley, Reading, MA.
- [6] Althöfer, I. and K. - U. Koschnik 1991. *On the convergence of threshold accepting*: *Applied Mathematics and Optimization* 24, 183-195.
- [7] Althöfer, J. and Koschnik, K. - U. 1991. *On the convergence of threshold accepting*: *Applied Mathematics and Optimization* 24.
- [8] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A. and Protasi, M. 1999. *Complexity and Approximation*: Springer, Berlin.
- [9] Ausiello, G. and Protasi, M. 1995. *Local search, reducibility and approximability of NP-optimization problems*: *Information Processing Letters* 54: 73-79.
- [10] Beasley, J. E., Meade N., Chang T. -J. 2003. *An evolutionary heuristic for the index tracking problem*: *European Journal of Operational Research* 148: 621-643.

- [11] Bellman, R. 1957. *Dynamic Programming*. Princeton, N. J. : Princeton Univ. Press.
- [12] Biethahn, J. and Nissen, V. 1995. *Evolutionary Algorithms in Management Application*: Springer, Berlin.
- [13] Bjelanovic, K. 2004. *Scelte Intertemporali e Programmazione Dinamica: Modelli e Problemi*: Tesi di Laurea in Matematica, relatore Chiar.mo Prof. Sergio Invernizzi.
- [14] Blackwell, D. 1965. *Discounted dynamic programming*: Annals of Mathematical Statistics 36: 226-235.
- [15] Bock, F. 1958. *An algorithm for solving traveling salesman and related network optimization problems*: 14th ORSA meeting, St. Louis.
- [16] Bogle, J. C. 1992. *Selecting equity mutual funds*: The Journal of Portfolio Management 18(2): 94 - 100.
- [17] Bölte, A. and Thonemann, U. W. 1996. *Optimizing simulated annealing schedules with genetic programming*: European Journal of Operational Research 92: 402-416.
- [18] Cesari R., e Panetta F., 1998. *Portafogli gestiti e tassi di rendimento: alcune considerazioni*:Bancaria, n. 1.
- [19] Cesari, R., e Panetta, F., 1998. *Style, Fees and Performance of Italian Equity Funds*: Temi di discussione, n. 325, Banca d'Italia.
- [20] CFA Institute 2007. *Corporate Finance and Portfolio Management*: CFA Program Curriculum, Volume 6, Level I.
- [21] CFA Institute 2007. *Derivatives and Alternative Investments*: CFA Program Curriculum, Volume 4, Level I.
- [22] CFA Institute 2007. *Equity and Fixed Income*: CFA Program Curriculum, Volume 5, Level I.
- [23] Croes, G. A. 1958. *A method for solving traveling salesman problems*: Operations Research 6, 791-812.
- [24] Chan, A. e Chen, C. R. 1992. *How well do asset allocation mutual fund managers allocate assets*: The Journal of Portfolio Management 18(3): 81-91.

- [25] Chang, T. J., Meade N., Beasley J. E. and Sharaiha Y. M. 2000. *Heuristics for cardinality constrained portfolio optimization*: Computers and Operations Research 27, 1271 - 1302.
- [26] Chow, C. S. e Tsitsiklis, J. N. 1989. *The Complexity of Dynamic Programming*: Journal of Complexity, Vol. 5, No 4 : 466-488.
- [27] Dahl, H., Meeraus, A. e Zenios, S. A. 1993. *Some financial optimization models*. *Risk management*: S. A. Zenios (Ed.), Financial Optimization, Cambridge University Press.
- [28] Damodaran, A. 2002. *Investment Valuation. Tools and Techniques for Determining the Value of Any Asset*: Wiley Finance.
- [29] Darwin, C. 1859. *The Origin of Species*: John Murray, London.
- [30] Dueck, G. and T. Scheuer 1990. *Threshold accepting: A general purpose algorithm appearing superior to simulated annealing*: Journal of Computational Physics 90, 161-175.
- [31] Dueck, G. and Winker, P. 1992. *New concepts and algorithms for portfolio choice*: Applied Stochastic Models and Data Analysis 8: 159-178.
- [32] Dueck, G. and Wirsching, J. 1991. *Threshold accepting algorithms for 0-1 knapsack problems*: Fourth European Conference on Mathematics in Industry.
- [33] Elton, E. J. e Gruber M. J. 1995. *Modern Portfolio Theory and Investment Analysis*: Wiley, New York.
- [34] Elton, E. J., Gruber M. J. e Blake, C. R. 1996. *The persistence of risk-adjusted mutual fund performance*: Journal of Business 69 (2): 133-157.
- [35] Elton, E. J. e Gruber M. J., Brown, S. J. and Goetzmann, W. N. 2003. *Modern Portfolio Theory and Investment Analysis*: Wiley and Sons, Inc., Hoboken, NJ, 6th edn.
- [36] Fabozzi, F. J. e Anrott, R. D. 1988. *Asset Allocation, a Handbook of Portfolio Policies, Strategies and Tactics*: Probus.
- [37] Fama E., 1972. *Components of investment performance*: Journal of Finance 27.

- [38] Fanni, M. 2000. *Manuale di finanza dell'impresa*: Biblioteca di Finanza diretta da Maurizio Fanni, Giuffrè Editore.
- [39] Fogel, L. J., Owens, A. J. e Walsh, M. J. 1966. *Artificial Intelligence through Simulated Evolution*: John Wiley and Sons, New York.
- [40] Garey, M. and Johnson, D. 1979. *Computers and Intractability - A Guide to the Theory of Incompleteness*: H. W. Freeman and Company, San Francisco.
- [41] Gelfand, S. B. and Mitter, S. K. 1991. *Weak convergence of Markov Chain sampling methods and annealing algorithms to diffusions*: Journal of Optimization Theory and Applications 68.
- [42] Gidas, B. 1985. *Nonstationary Markov Chains and convergence of the annealing algorithm*: Journal of Statistical Physics 39.
- [43] Gilli, M. and P. Winker 2007. *Heuristic optimization Methods in Econometrics*: University of Geneva, Switzerland and University of Giessen, Germany.
- [44] Gill, P. E., Murray, W. and Wright, M. H. 1991. *Numerical Linear Algebra and Optimization*: Vol. 1, Addison-Wesley, Redwood, CA.
- [45] Gilli, M. and E. Küllezi 2001. *Threshold Accepting for Index Tracking*: University of Geneva, Switzerland.
- [46] Gilli, M. 2004. *An Introduction to Optimization Heuristics*: Seminar at the University of Cyprus, Department of Public and Business Administration, September 2004.
- [47] Giusti, E. 1996. *Analisi matematica 1*: Bollati Boringhieri.
- [48] Goffe, W. L., Ferrier G. D. and Rogers J. 1994. *Global optimization of statistical functions with simulated annealing*: Journal of Econometrics 60 (1-2), 65-99.
- [49] Gruber, M. J. 1996. *The growth in actively managed mutual funds*: Journal of Finance 51(3):783-810.
- [50] Hajek, B. 1988. *Cooling schedules for optimal annealing*: Mathematics of Operations Research 13: 311-329.
- [51] Holland, J. H. 1975. *Adaption in Natural and Artificial Systems*: The University of Michigan Press, Ann Arbor, Michigan.

- [52] Hall, G. J. e Rust, J., 1999. *An Empirical Model of Inventory Investment by Durable Commodity Intermediaries*: Cowles Foundation Discussion Papers 1228, Cowless Foundation, Yale University.
- [53] Ippolito, R. A., 1989. *Efficiency with Costly Information: A Study of Mutual Fund Performance, 1965-1984*: Quarterly Journal of Economics, Vol. 104, n. 1.
- [54] Jarmo T. Alander 2000. *An Indexed Bibliography of Genetic Algorithms in Economics*: Department of Information Technology and Production Economics. University of Vaasa Finland.
- [55] Jagannathan, R., and Ma, T. 2003. *Risk Reduction in Large Portfolios: Why Imposing the Wrong Constraints Helps*: The Journal of Finance, 58 (4), 1651-1638.
- [56] Jensen M.C., 1968. *The Performance of Mutual Funds in the Period 1945-1964*: Journal of Finance.
- [57] Jensen, M., 1968. *The Performance of Mutual Funds in the Period 1945-1964*: Journal of Finance, Vol. 23, n. 1. (gennaio-febbraio),
- [58] Jerell, M. E. and Campione W. A. 2001. *Global optimization of econometric functions*: Journal of Global Optimization 20 (3-4), 273-295.
- [59] Jones, A. E. W. and Forbes, G. W. 1995. *An adaptive simulated annealing algorithm for global optimization over continuous variables*: Journal of Global Optimization, 6:1-37.
- [60] Judd, K. L. 1998. *Numerical Methods in Economics*: The MIT Press: 297-299; 399-443.
- [61] Kemeny. J. G. and Snell, J. L. 1976. *Finite Markov Chains*: Springer, New York.
- [62] Knuth, D. E. 1981. *The Art of Computer Programming*: 2nd Edn, Addison-Wesley, Rading, MA.
- [63] Koza, J. R. 1992. *Genetic Programming*: The MIT Press, Cambridge, Massachusetts.
- [64] Malkiel, B. G. e Radisich, A. 2001. *The growth of index funds and the pricing of equity securities*: The Journal of Portfolio Management 27(2): 9-21.

- [65] Maringer, D. 2005. *Portfolio Management with Heuristic Optimization*: Springer. Dordrecht.
- [66] Markowitz, H. 1952. *Portfolio Selection*: Journal of Finance 7: 77-91.
- [67] Reeves, C. R. e Rowe, J. E. 2003. *Genetic Algorithms - Principles and Prospectives. A Guide to GA Theory*: Kluwer Academic Publishers.
- [68] Mc Culloch Warren S. and Walter Pitts 1943. *A logical calculus of the ideas immanent in nervous activity*: The Bulletin of Mathematica Biophysics, 5: 115-133.
- [69] Norris, J. R. 1997. *Markov Chains*: Cambridge University Press, Cambridge.
- [70] Ramsey, F. P. 1928. *A Mathematical Theory of Saving*. The Economic Journal 38: 543-559.
- [71] Rechenberg, I. 1973. *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*: Fromman - Holzboog Verlag, Stuttgart.
- [72] Revuz, D. 1984. *Markov-Chains*: 2nd Edn, North-Holland, Amsterdam.
- [73] Rudd, A. 1980. *Optimal selection of passive portfolios*: Financial Management 57-66.
- [74] Sgarro, A. *Entropia nella statistica oggettiva e soggettiva*.
- [75] Sorenson E. H., Miller K. L. and Samak V. 1998. *Allocating between active and passive management*: Financial Analyst Journal 54 (5) 183.
- [76] Tettamanzi, A. e Tomassini, M. 2001. *Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems*: Springer - Verlag Berlin. Heidelberg.
- [77] Winker, P. 2001. *Optimization Heuristics in Econometrics: Application of Threshold Accepting*: Wiley. Chichester.
- [78] Winker, P. and D. Maringer 2007. *The threshold accepting optimization algorithm in economics and statistics*. In: *Optimization, Econometric and Financial Analysis*(E. J. Kontoghiorghes and C. Gatu, Ed.): Vol. 9 of Advances in Computational Managment Science. Springer. Belin. pp.107-125.

- [79] Winker, P. and Kai-Tai Fang 1997. *Application of threshold accepting to evaluation of the discrepancy of a set of points*: SIAM Journal on Numerical Analysis 34, 2028-2042.