

On partitioning multivariate self-affine time series

Christopher M. Taylor, Student Member, IEEE, and Abdellah Salhi

Abstract-Given a multivariate time series, possibly of high dimension, with unknown and time-varying joint distribution, it is of interest to be able to completely partition the time series into disjoint, contiguous subseries, each of which has different distributional or pattern attributes from the preceding and succeeding subseries. An additional feature of many time series is that they display self-affinity, so that subseries at one time scale are similar to subseries at another after application of an affine transformation. Such qualities are observed in time series from many disciplines, including biology, medicine, economics, finance and computer science. This paper defines the relevant multiobjective combinatorial optimization problem with limited assumptions as a biobjective one, and a specialized evolutionary algorithm is presented which finds optimal self-affine time series partitionings with a minimum of choice parameters. The algorithm not only finds partitionings for all possible numbers of partitions given data constraints, but also for self-affinities between these partitionings and some fine-grained partitioning. The resulting set of Pareto-efficient solution sets provides a rich representation of the self-affine properties of a multivariate time series at different locations and time scales.

Index Terms—Partitioning algorithms, genetic algorithms, Pareto optimization, time series analysis, fractals, optimization methods, finance

I. INTRODUCTION

Given a multivariate time series, possibly of high dimension, with unknown and time-varying joint distribution, it is of interest to be able to completely partition the time series into disjoint, contiguous subseries, each of which might be assumed for the purpose of further analysis to have different distributional or pattern attributes from the preceding and succeeding subseries. Whilst in the univariate case it is sometimes quite easy to agree on such a partitioning simply by visual inspection, including the number of partitioned subseries into which it is most appropriate to divide the time series and the location of the transitions between subseries, in the multivariate case visual inspection becomes impossible with more than a few dimensions. Clustering or partitioning of time series data has been widely studied in the machine learning and data mining literature; [41] and [39] contain good introductions to partitioning and clustering algorithms, respectively. In the case of financial data, it has long been

First submitted for review on February 11th, 2016; resubmitted with revisions July 26th, 2016 and January 30th, 2017. Final version submitted 27/3/2017. Christopher M. Taylor is with the Department of Mathematical Sciences, University of Essex, U.K.(e-mail: cmtayl@essex.ac.uk). Abdellah Salhi is with the Department of Mathematical Sciences, University of Essex, U.K.(e-mail: as@essex.ac.uk).

recognized that distribution varies over time, not in a smooth manner but rather in jumps between different states. Many well-established econometric models developed to deal with time-varying distributions nonetheless take the states as given, and do not address the question of identifying the different subseries corresponding to such states. Furthermore, most econometric models with time-varying parameters have difficulty dealing with the multivariate case, especially in high dimensions.

An additional feature of many time series is that they display self-affinity, which we may loosely define as the property that subseries at one time scale are similar to subseries at another after application of an affine transformation. Such observations in the natural world date back at least to work in the mid-20th century on hydrography of the Nile [35] and the length of international borders [60], and a more general theory was advanced in [48]. Many examples have since been found; for example one survey of the literature [59] cites inter alia work on air temperature, river discharge and tree ring spectra; variations in solar luminosity, sedimentation, and the earth's magnetic field; the structure of river networks; growth of plankton and many other flora and fauna; and in the human world, automobile and internet traffic flows.

In economic and financial data, [49, 50] advanced the idea of self-similarity under power laws, later generalized and developed into the Multifractal Model of Asset Returns (MMAR) [51]. Ideas of self-similarity and power laws at work in financial data however have a longer history, at least back to work in the 1930s by R.N. Elliott [23, 24] in which he posits particular patterns occurring at different time scales with a relationship governed by a power law based on the golden ratio. A good deal of the most interesting work on self-affine time series has been conducted within the fields of finance and econometrics, but many of the theoretical findings and tools developed can be of use in analyzing a wider range of data types. However, to our knowledge there is nothing in the literature that addresses the specific problem of actually identifying optimal partitionings of multivariate, self-affine data, as we will describe in this paper.

The limitations shared by most econometric models that allow time-varying parameters in respect of higher dimension time series have already been noted. An additional problem is that most rely on particular assumptions regarding the underlying distributions and processes, and often require large numbers of parameters to be either provided or estimated from the data. This can make it difficult to separate the validity of the model and its assumptions from the particular data set and parameterization applied. An alternative approach is to build a model with as few assumptions and parameters as possible, in particular limiting the number of a priori choice parameters supplied by the modeller. Evolutionary algorithms

(EAs) can be particularly well suited to such an approach if suitably designed, and the main contribution of this paper is the development of a simple theoretical model based on self-affinity of a time series in terms of the similarity of a coarse-grained partitioning of the whole to a fine-grained partitioning of a subseries, and its implementation via a specialized EA with limited choice parameters.

The remainder of this paper is organized as follows. Section II introduces the concepts of self-affinity and partitioning into distinct subseries, or regimes as they are known in econometrics, in univariate and multivariate time series, and discusses some common statistical models. Section III discusses more specific aspects of modelling self-affine multivariate time series of financial data including techniques to reduce computational complexity, and defines the multiobjective optimization problem to be addressed in validly partitioning such time series. Section IV describes the evolutionary algorithm, as well as certain general principles of design when addressing problems with high computational complexity and limited computational resources. Finally, Section V presents some initial empirical observations using both simulated and real data, and the paper concludes with a brief discussion of possible future development of this line of research towards a general framework for multiscale threat analysis, detection and mitigation.

II. BACKGROUND AND MOTIVATION

A. Necessity of a specialized approach

In this paper we present an approach designed specifically to simultaneously identify non-overlapping partitionings of both coarse-grained and fine-grained subseries. This means firstly that we are not attempting to define clusters of overlapping or non-contiguous time series subsequences. Although certain types of clustering algorithm, including the widely-used k-means algorithm, have been categorized as partitional algorithms [27], such algorithms are typically not constrained to produce only partitions that are contiguous along one dimension (i.e. time) but if used for time series data, can cluster together individual observations from many different parts of the entire time series. In terms of taxonomy, we thus draw a distinction between partitioning algorithms of the type set out in this paper and the more general case of clustering algorithms. Hence the approach is also significantly different in this respect to clustering EAs [19, 54, 34].

Furthermore, clustering algorithms commonly have an assumption of decreasing similarity of points in a cluster as distance from a centroid increases, as well as decreasing dissimilarity from points in other clusters as distance from them decreases. The time series partitioning algorithm presented here rather assumes homogeneity within partitions and sharp differences with other partitions, and so is arguably much more suitable for data that displays rapid transitions between different subseries in terms of some measure or set of measures. The combination of a population-based approach and subgroup separation means that useful results can be obtained from a single run, although the stochastic nature of the algorithm still means more useful results may be obtained

from multiple runs. We also make the explicit assumption that data being examined has the property of self-affinity. The particular nature of the objective functions required for simultaneous identification of coarse-grained and fine-grained partitions together with the constraints necessary to produce valid partitions cannot be handled by other EAs not created specifically to solve this particular problem, and requires a specialized approach, as we will see in later sections.

B. Subseries and self-affinity

Consider the general case of a multivariate time series \mathbf{S} with m simultaneous individual time series, each with T time-ordered samples. Initially we know nothing a priori about the distributions of the individual series or about the joint distribution, except that some distributional features may be time-varying, and in particular that \mathbf{S} may be partitioned into two or more disjoint subseries each of which has distinct distributional attributes that distinguish each subseries, or partition, from the preceding or following one, forming a complete and still strictly time-ordered partitioning of \mathbf{S} . We also know that the first and last time-ordered subseries may be incomplete, in the sense that data collected after the T -th sample may still be part of the final subseries, whilst if we could collect data before the first sample, some number of additional, earlier samples might still belong to the first subseries. Hence we may typify such a partitioning $\mathbf{K}_\kappa = \cup_{k=1}^\kappa \{K_k\}$ of \mathbf{S} into κ partitions, or subseries, in terms of a set of $\kappa - 1$ cutpoints $C = \{c_1, c_2, \dots, c_{\kappa-1}\}$, with each $c_k \in [1, T]$, $k = 1 \dots \kappa - 1$. By convention we will consider each cutpoint $c_k, k < \kappa$ to be the last point in a subseries, so that the next point in \mathbf{S} is the first in subseries K_{k+1} , and all the c_k are unique, so that each subseries $K_k \neq \emptyset, k = 1 \dots \kappa$.

Our motivation is to discover what we can about the partitions, in particular where the dividing points c_i between partitions may most usefully be placed for the type of analysis in which we are interested; yet we do not necessarily even know yet how many such partitions there are. Indeed, several optimal partitionings \mathbf{K}_κ with different numbers κ of partitions may be considered possibly equally valid, and indeed as we shall see later, if we have more than one metric for considering different partitionings, there may be more than one Pareto optimal partitioning even when the number of partitions is the same. Such a partitioning of a multivariate time series could have many uses for a variety of different types of data. Beyond using such a partitioning for further analysis of the data set, it may in particular be possible to solve a further multiobjective optimization problem of the form

Minimize

$$\mathbf{f}(\mathbf{K}_\kappa(\mathbf{S})) = [\mathbf{f}(\mathbf{S}_1), \mathbf{f}(\mathbf{S}_2) \dots \mathbf{f}(\mathbf{S}_\kappa)]^T, \quad (\text{II.1})$$

where some multiobjective function over ω objectives $f = [f_1(\mathbf{S}), f_2(\mathbf{S}) \dots f_\omega(\mathbf{S})]$ is applied separately to each partitioned subseries $\mathbf{S}_k, k = 1 \dots \kappa$ and a Pareto-efficient solution set is obtained. One application of this would be portfolio optimization, where one could produce portfolios robust to several different sets of market circumstances, but many other applications could be imagined. The problem of identifying

the nature of the last partition, which we may term the current partition, is of particular interest, but as already noted we may have incomplete information, since the true end point c_κ of the final subseries cannot be known at time T as it falls at some later point.

Furthermore, we will consider data that has a particular structural attribute, namely that of self-affinity. Sets with fractal properties are sometimes loosely referred to as self-similar, but in many datasets true scale invariance or statistical self-similarity as posited in [48] is replaced by self-affinity, and [51] reserves the term self-similarity for geometric objects that are invariant under isotropic contraction. The concept of self-affinity is defined in [52] as follows: let $X(t, \omega)$ be a random function defined on $-\infty < t < \infty$, with $\omega \in \Omega$ the set of possible values of the functions. Then such a random function is self-affine with exponent $H > 0$ if for every $h > 0$ and any t_0 ,

$$\begin{aligned} X(t_0 + \tau, \omega) - X(t_0, \omega) &\stackrel{d}{=} \\ \{h^{-H} [X(t_0 + h\tau, \omega) - X(t_0, \omega)]\}, \end{aligned} \quad (\text{II.2})$$

where $X(t_0, \omega) \stackrel{d}{=} Y(t, \omega)$ indicates two random functions with the same joint distributions. The literature generally analyses self-affinity in terms of fractionally integrated Brownian motion (FBM) processes, i.e. $X(t, \omega) = B_H(t, \omega)$.

In contrast, except where noted we will make no assumptions about the underlying data-generating process (DGP) and will rather describe self-affine sets in terms of similarity achieved by making affine transformations equivalent to a restriction of the usual geometric type at different time scales; later we will consider the statistical invariances involved. In the most general sense, however, we will define the property of self-affinity as follows: let $\{F_k, k = 2 \dots \kappa - 1\} = \mathbf{F}_\kappa$ be a fine-grained partitioning of some subseries s of $\bigcup_{k=2}^{\kappa-1} \{K_k\} \equiv \mathbf{S} - \{K_1, K_\kappa\}$, so that the first and last points in s are in $[c_1, c_{\kappa-1}]$ and in general terms s is sufficiently short compared to \mathbf{S} that it can be considered of a different “scale”. Then \mathbf{S} can be described as self-affine if the two-scale partitioning pair $\{\mathbf{K}_\kappa, \mathbf{F}_\kappa\}$ obeys

$$\bigcup_{k=2}^{\kappa-1} \{\mathbf{A}F_k + \mathbf{b}\} \stackrel{\mu}{\approx} \bigcup_{k=2}^{\kappa-1} \{K_k\} \quad (\text{II.3})$$

for at least one such subseries F_k ; that is to say, at least one fine-grained partition of a subseries of \mathbf{S} is by some measure μ similar, after the application of a suitable affine transformation, to a coarse-grained partition of the whole of \mathbf{S} , excluding an initial and a final incomplete subseries. We define the relationship $\stackrel{\mu}{\approx}$ in terms of the similarity, by a measure yet to be defined, of some fine-grained set of subseries $\{F_k, k = 1 \dots \kappa - 2\}$ after such an affine transformation to the coarse-grained set of subseries $\{K_k, k = 2 \dots \kappa - 1\}$; later we will impose limitations on the nature of \mathbf{A} , induced by the choice of measure. The first and last subseries of the coarse-grained partitioning \mathbf{K}_κ are excluded because they are incomplete, so we must always have $\kappa \geq 3$, and \mathbf{F}_κ comprises $\kappa - 2$ fine-grained partitions. Self-affinity should be understood at this stage in a very general sense; it does not necessarily imply that the compared subseries have the same statistical distribution, though this may be so depending on the measure

employed, but might also refer to other attributes, such as the recurrence of certain patterns in the time series data.

Consider initially the univariate case with $\kappa = 3$ so that there is a single complete interior coarse-grained partition, and hence a single fine-grained subseries is considered, so that (II.3) reduces to

$$\{\mathbf{a}F_1 + \mathbf{b}\} \stackrel{\mu}{\approx} \{K_2\}; \quad (\text{II.4})$$

that is, an affinely transformed version of the single fine-grained subseries is similar under μ to the second, i.e. the only interior, coarse-grained subseries (the first and third being incomplete). If we compare (II.2) and (II.4), several key differences emerge. Firstly (II.2) refers to affinity between a continuity of subseries of different lengths starting from t_0 which can be thought of as an expanding moving average, but (II.3) compares the interior part of one coarse-grained partition within the portion of \mathbf{S} that excludes the incomplete initial and final subseries to a single fine-grained subseries which may start and end at any point within \mathbf{S} .

Secondly, (II.2) requires some assumptions about the nature of $X(t, \omega)$ in order to be useful. In the literature $X(t, \omega)$ is usually an FBM process; by contrast, (II.3) makes no assumptions about the underlying process. Thirdly, if we consider a univariate version of (II.3) and take $h = b$, (II.2) requires a particular power law relationship between the translation and linear transformations, namely $a = b^H$; (II.3) does not make any assumptions about the relationship between \mathbf{A} and \mathbf{b} .

We can now define the principal problem, which we can cast as a biobjective minimization problem. Let \mathbf{d}_{K_κ} be a $(\kappa - 1) \times 1$ vector of values of a distance metric $d(K_k, K_{k-1}) \in [0, 1]$, $k = 2 \dots \kappa$, and $g(\mathbf{d}_{K_\kappa})$ be some summarizing function that returns a non-negative scalar, so that

$$f_1(\mathbf{S}) = g(\mathbf{d}_{K_\kappa}). \quad (\text{II.5})$$

Also let

$$f_2(\mathbf{S}) = h(-\mathbf{d}_{F_\kappa}), \quad (\text{II.6})$$

where \mathbf{d}_{F_κ} is a vector of the same distance metric between each of the $\kappa - 2$ pairs of sequential subseries at the fine-grained and coarse-grained levels, excluding the incomplete first and last subseries of the coarse-grained partition:

$$\mathbf{d}_{F_\kappa} = [d(F_2, K_2) \dots d(F_{\kappa-1}, K_{\kappa-1})]^T, \quad (\text{II.7})$$

and $h(-\mathbf{d}_{F_\kappa})$ is a summarizing function of the negative of the distance vector, so that it indicates similarity between the coarse-grained and fine-grained subseries. Then the problem for a partition into a given number of subseries κ is the following biobjective problem:

Minimize

$$\begin{aligned} \mathbf{f}(\mathbf{S}) &= [f_1(\mathbf{S}), f_2(\mathbf{S})] \\ &= [g(\mathbf{d}_{K_\kappa}) h(-\mathbf{d}_{F_\kappa})]. \end{aligned} \quad (\text{II.8})$$

The only explicit assumptions here are that \mathbf{S} may be validly partitioned into κ distinct coarse-grained subseries, the first and last of which may be incomplete; that the coarse-grained

partitioning may be typified as similar to some fine-grained partitioning of a subseries of \mathbf{S} ; and that the differences between the subseries of the coarse-grained partitioning and the similarities between the sequential subseries of the coarse-grained and fine-grained partitionings respectively both form metric spaces metrizable by the same metric d . We make no other assumptions about the actual distributions or DGPs of \mathbf{S} or its subseries. Of course (II.8) allows the possibility that there may be many valid fine-grained partitionings of different subseries that show similarity to any given coarse-grained partitioning, and in forming the Pareto frontier the biobjective function will make use of functions g, h which in some sense summarize the distributions of all solutions found.

C. Regimes in econometrics and finance

Having established the framework for a general multivariate time series \mathbf{S} and an unspecified metric d , we now consider metric spaces (\mathbf{S}, d) where d in some way metrizes the time-varying variance-covariance structure of \mathbf{S} , so that we are now making a more specific assumption, namely that changes over time in \mathbf{S} may be identified and partitioned with reference to changes in variance and covariance of subseries of \mathbf{S} . Such cases have been extensively studied in econometrics, where the partitioned subseries are commonly referred to as regimes, and we will use the terms partition, regime and subseries interchangeably henceforth. However most of the literature concentrates on univariate data or data with only a very few variables, and rather than seeking to identify the regimes, concentrates on analysis of the distributional qualities of the data with the regimes taken as given. A very brief summary of a large literature follows, which although principally from the areas of econometrics and finance, contains principles and techniques which may shed light on processes observed in many types of data.

1) *Volatility regimes*: Although there is no consistent definition of a regime, the terminology comes largely from seminal papers by Hamilton [31, 32, 33] which consider the case where κ possible regimes exist from which a particular observation y_t may be drawn, and an unobserved state variable s_t which takes an integer value $(1 \dots \kappa)$ such that y_t depends on the current and most recent m autoregressive lags of y_t , the current and most recent m values of s_t and a vector of parameters θ ; that is:

$$p(\mathbf{y}_t | s_t, s_{t-1}, \dots, s_{t-m}, \mathbf{y}'_{t-1}, \mathbf{y}'_{t-2} \dots \mathbf{y}'_{t-m}) \quad (\text{II.9})$$

$$\equiv p(\mathbf{y}_t | \mathbf{z}_t; \theta),$$

$$z_t \equiv (s_t, s_{t-1}, \dots, s_{t-m}, \mathbf{y}'_{t-1}, \mathbf{y}'_{t-2} \dots \mathbf{y}'_{t-m}).$$

Note that there is an assumption that the data is stationary, and it may be necessary to transform the data, for example by taking differences, in order to ensure stationarity. A vector autoregression may be generalized such that the constant terms, the covariance matrix and the autoregressive coefficients may all be functions of the state s_t , and the transition between states is modelled as a Markov chain. Although the subsequent literature explores many of the possibilities of this general formulation, most often the number of states is small (2 or 3)

and a change in level or in volatility are considered more often than a change in autoregressive structure. An earlier paper [32] considered the proposal that there might be an occasional shift in the constant term around which a scalar fourth-order regression clusters, and many later papers find strong evidence for volatility clustering in financial time series.

Examples of volatility regimes in the literature include the finding in [28] that real interest rates can be partitioned in the time domain into regimes separated by sharp jumps caused by structural breaks such as the oil shocks of the 1970s; the analysis in [15] of volatility regime switches in world stock markets; and the study in [29] of changes in volatility regime evidenced by the behaviour of the VIX volatility index. Econometricians have continued to develop a testing methodology for such regime shifts, and the most commonly used test is the iterated cumulative sum of squares (ICSS) of [38], later modified in [63] to be more robust to heteroskedastic and leptokurtic distributions of returns. Such tests however concentrate on detecting structural breaks in volatility in a univariate sense; they cannot analyze shifts in the overall covariance structure of the returns of a set of assets, such as the constituents of an index.

2) *Time varying models: ARCH and GARCH*: In recent years substantial research has been devoted to applying the self-affine FBM processes described in Section II-B to the regime-switching paradigm outlined in Section II-C. By way of background, it had long been recognized that standard statistical models which assumed financial time series to be formed from independent, normally distributed random variables do not adequately describe the data. [49] and [26] found that return distributions did not conform to the standard normal distribution but were leptokurtic and [8] noted that returns were also asymmetric. As a result, the autoregressive conditionally heteroskedastic (ARCH) model of [25] and its various generalizations and adaptations have been used extensively to model financial data, as have the stochastic volatility (SV) models of [55] and others.

To recap, the original ARCH model of [25] considers a first-order autoregressive (AR(1)) process of the form

$$y_t = \gamma y_{t-1} + \epsilon_t, \quad (\text{II.10})$$

where $E[\epsilon] = E[y] = 0, V(\epsilon) = \sigma^2$, but the conditional mean $E[y_t | y_{t-1}] = \gamma y_{t-1}$. By introducing heteroskedasticity, that is, time-varying variance, we find the ARCH(1) model:

$$y_t = \epsilon_t h_t^{1/2} \quad (\text{II.11})$$

$$h_t = h(y_{t-1}, \alpha) = \alpha_0 + \alpha_1 y_{t-1}^2, \quad (\text{II.12})$$

where α is a vector of unknown parameters, and the ARCH(p) model follows from a generalization of h_t to p lags. The GARCH model [10] offered a generalization of ARCH which has been shown to outperform empirically and can be more parsimonious than a higher-order ARCH model. The GARCH(p, q) model defines:

$$h_t = \alpha_0 + \sum_{i=1}^q \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^p \beta_i h_{t-i}, \quad (\text{II.13})$$

so that GARCH($0, q$) \equiv ARCH(q). The general null hypothesis is that $\alpha = \beta = \mathbf{0}$.

Multivariate extensions to the ARCH/GARCH family exist, but are generally hard to specify and estimate in higher dimensions.

3) The Multifractal Model of Asset Returns: Likelihood-based estimation of Markov-switching processes in the statistics literature predates the treatment of regimes in finance outlined in Subsection II-C1 [13]. Stochastic regime-switching models involve the conditional mean and variance being dependent on an unobserved and time-variant latent state which can change markedly and quickly. The latent state $M_t = \{m^1, m^2, \dots, m^d\}$ gives rise to a model of returns

$$r_t = \mu(M_t) + \sigma(M_t) \varepsilon_t. \quad (\text{II.14})$$

The Markov chain M_t is governed by a transition matrix A , the components a_{ij} of which represent the probability that state j will follow state i . In ([51]), the authors introduced a new model based on FBM incorporating new scaling properties. They note that whilst FBM captures long memory in the persistent case where $1/2 < H < 1$, $H = 1/2$ representing ordinary Brownian motion, it captures neither the fat tails of the distribution of returns nor the time-variant volatility clustering often observed in financial data. Consider first a generalization of (II.2):

$$\begin{aligned} X(t+c\Delta t) - X(t) &\stackrel{d}{=} \\ M(c)[X(t+\Delta t) - X(t)], \quad c > 0, \end{aligned} \quad (\text{II.15})$$

where X and M are independent random functions. With certain restrictions on the distribution of the process, this implies the scaling rule:

$$E(|X(t)|^q) = c(q)t^{\tau(q)+1}, \quad (\text{II.16})$$

where $\tau(q)$ and $c(q)$ are both deterministic functions of q , and $\tau(q)$ is referred to as the scaling function. In the special case of self-affine processes, the scaling function $\tau(q)$ is linear and fully determined by its index H , and such processes are referred to as unifractal or uniscaling; in the more general case, where $\tau(q)$ is non-linear, the processes are referred to as multiscaling.

Now let

$$X(t) = \ln P(t) - \ln P(0), \quad 0 \leq t \leq T, \quad (\text{II.17})$$

where $P(t)$ is the price of a financial asset at time t . Note that $X(t)$ is generally assumed to be a stationary process in the econometrics literature. Then under the MMAR, it is assumed that

$$X(t) \equiv B_H[\theta(t)], \quad (\text{II.18})$$

where stochastic trading time $\theta(t)$ is the cumulative distribution function of a multifractal measure defined on $[0, T]$, $B_H(t)$ is FBM with self-affinity index H , and $B_H(t)$ and $\theta(t)$ are independent. Under these assumptions, $X(t)$ is multifractal, with scaling function

$$\tau_X(q) = \tau_\theta(Hq). \quad (\text{II.19})$$

Trading time $\theta(t)$ can be thought of as a time deformation process, squeezing or stretching the price action, and trading time controls the moments of the distribution of $X(t)$.

4) Markov-Switching Multifractal (MSM) models: [11] introduce a discretized version of trading time, in which prices are driven by a first-order Markov state vector with k components:

$$\begin{aligned} M_t &= \{M_{1,t}; M_{2,t}; \dots; M_{k,t}\} \in \mathbb{R}_+^k, \quad (\text{II.20}) \\ E[M_{k,t}] &= 1. \end{aligned}$$

Returns are modelled as

$$r_t = \sigma \left(\prod_{i=1}^k M_t^{(i)} \right)^{1/2} \cdot \mu_t, \quad (\text{II.21})$$

where σ is the unconditional standard deviation and μ_t the mean return at time t . The multipliers in (II.20) differ in their transition probabilities γ_k but not in their marginal distribution M , and $M \geq 0, E[M] = 1$. The Markov property guarantees that the distribution of the future multipliers depends only on the current multipliers. The transition probabilities are specified as

$$\begin{aligned} \gamma_k &= 1 - (1 - \gamma_1)^{b^{k-1}} \\ , \quad \gamma_1 \in (0, 1), b \in (0, \infty), \end{aligned} \quad (\text{II.22})$$

and this process is referred to as Markov-Switching Multifractal (MSM) [12]. MSM is the main statistical technique embodying and generalizing the idea of self-affinity in financial time series in use today and indeed has become one of the more popular models in financial econometrics overall. The model can be estimated using maximum likelihood (ML) methods and the log-likelihood has a parsimonious closed form, making estimation relatively simple, albeit for restricted cases. [45] introduced GMM estimation techniques which addresses the case where the distributional model requires an infinite state space, as in the lognormal model proposed in [51], as well as addressing the computational complexity implied by some choices of distribution, for example the $2^k \times 2^k$ transition matrix required by the binomial model proposed in [11]. However the author notes that GMM is itself relatively computationally intensive compared to ML methods.

A sizeable body of empirical work has grown up in recent years around various approaches to the estimation of MSM models on different data sets using differing underlying distributions, including findings in *inter alia* [6, 12, 16, 30, 36, 37, 42, 43, 47, 46, 56, 62, 70]. However there is evidence that not all financial data fits well with the hypothesis that the DGP for financial data is FBM; for example [58] finds anomalies in high-frequency data, especially when stock index data from emerging markets is analysed. Also, [18] claims that a new approach, discrete autoregressive stochastic volatility (DSARV) outperforms MSM, and critique the high-dimensional state space inherent to the MSM approach.

A multivariate extension has been proposed [14], but in the natural generalization of the univariate MSM, the number of parameters therefore grows at least as fast as a quadratic function of the number of assets and so like multivariate GARCH is potentially highly computationally expensive, as well as complicated to analyse, for larger number of assets; hence the

authors propose a factor model simplification rather than the true multivariate model. A bivariate model based on univariate decomposition [14, 36, 44] has also been studied, although the approach in [44] requires different volatility parameters at high and low frequencies, and [42] considered an extension of the GMM approach to multivariate models. A further approach to multivariate multifractals in the general literature uses operator fractional Brownian motion (OFBM) [22, 1]. In general however, as with most competing econometric models with time-varying volatility, most studies consider only the bivariate case or at most a handful of assets.

However [57] finds that the degree of multifractality displayed by different stocks is positively correlated to their depth in the hierarchy of cross-correlations, as measured through a correlation based clustering algorithm (DBHT, [64]). Since we know that such hierarchies demonstrate persistence in related correlation-based clustering techniques but nonetheless vary over time [53, 66], this further suggests the possibility that the correlation structure of financial markets may itself display multifractal characteristics.

III. PARTITIONING BASED ON REALIZED COVARIANCE

A. Correlation of subperiod covariances

Consider again a $T \times m$ multivariate time series \mathbf{S} consisting of individual, in general correlated series s_j , $j = 1 \dots m$, where $s_{jt} = \ln v_{jt} - \ln v_{jt-1}$, $t = 2 \dots T$, the v_{jt} being the raw data observations. We assume that \mathbf{S} is globally weakly stationary and that a valid partitioning \mathbf{K}_κ into κ non-overlapping and contiguous subseries subseries is locally jointly ergodic, meaning that over sufficient time each mean $\mu_j \rightarrow 0$ and that \mathbf{S} may have joint distributional features that vary over time but are similar within a given partition K_k , $k = 1 \dots \kappa$. In particular, we assume initially that at a particular level of scaling, the cross-sectional vector of log differences $r_{j,t}$ of the m time series at time t obeys

$$[s_{1t}, s_{2t} \dots s_{mt}]^T = \boldsymbol{\mu}_k + \boldsymbol{\Sigma}_k \boldsymbol{\epsilon}_t, \quad (III.1)$$

$$t \in K_k, \quad k \in [1, \kappa],$$

where $K_k \in \mathbf{K}_\kappa$ represents a given subseries of the partition, $\boldsymbol{\mu}_s$ is a vector of the subseries-dependent means for each individual series of log differences s_j , $\boldsymbol{\Sigma}_k$ is the subseries-dependent variance-covariance matrix and $\boldsymbol{\epsilon}_t$ is a vector of random innovations for each s_j at time t with unknown distribution. Let us further assume that the variances and covariances are sufficiently locally stable for the purpose of forming a valid partitioning.

We might consider a distance metric of the form

$$d(S_A, S_B) = \sqrt{\frac{1}{2}(1 - \rho_{AB})}, \quad (III.2)$$

where S_A, S_B are any two subseries of \mathbf{S} , and ρ_{AB} is a standard matrix correlation between two covariance matrices, so that $d(S_A, S_B) \in [0, 1]$. There would however be several problems with using ordinary covariance matrices for such calculations in an EA with a large population and many generations. Firstly, the computational complexity scales in theory quadratically with m , although in practice the use of

parallel processing and efficient algorithms reduces this scaling substantially. More seriously, the covariance matrices must be recalculated for each and every instance of the metric. In the next subsection we examine the potential for a technique from financial econometrics to considerably decrease the computational complexity.

B. Realized covariance

There is always a tension in considering sampling frequency between the desire to gain potentially greater accuracy, especially in complex time-varying frameworks, from a higher sampling rate against the inevitable increase in noise. As higher-frequency financial data has become more freely available, one approach to this has been to use realized volatility, where time-varying volatility is estimated as a series of observations for sub-periods (typically one trading day), the latter in turn being estimated from intraday data sampled at 5-minute or more frequent intervals, so that an estimator of RV [2, 3, 4, 5] is:

$$\psi_t^2 = \sum_{n=1}^N s_{nt}^2, \quad t = 1..T, \quad (III.3)$$

where s_{nt} is the n th return observation of N total on day t ; this very simple formulation is derived from arguments using stochastic calculus. Through aggregation, the use of RV can reduce the complexity inherent in the use of high-frequency data. As such, it is used as an input to many models, including GARCH, but can also be used in and of itself as a non-parametric modelling device; [7] proposes RV as a third dynamic volatility model class in its own right, alongside SV and GARCH, and use RV estimates both as an input to SV and GARCH and on their own to estimate volatility dynamics on the WIG20 stock index and the EUR/PLN exchange rate. The authors conclude that in general, models perform better using progressively higher sampling rates for RV, but that this is also true simply using RV on its own.

A natural extension of RV is realized covariance:

$$RC_{ijt}^{(n)} = \sum_{n=1}^N s_{int} s_{jnt}, \quad i, j = 1 \dots m, \quad (III.4)$$

$$t = 1..T$$

Realized variance/covariance has the useful property that it simply sums over periods of aggregation:

$$RC_{ijT} = \sum_{t=1}^T \sum_{n=1}^N s_{int} s_{jnt}. \quad (III.5)$$

In practice this means much simpler recalculation of the realized covariance matrices for a given subseries compared to calculating ordinary covariances. We can then conveniently specify a single partitioned $\frac{1}{2}m(m+1) \times \kappa$ array consisting of the realized variances and covariances for each subseries stacked into column vectors $\boldsymbol{\sigma}_k$, $k = 1 \dots \kappa$:

$$\begin{array}{cccc} RC_{11t_1} & \dots & RC_{11t_\kappa} \\ \vdots & \ddots & \vdots \\ RC_{mmt_1} & \dots & RC_{mmt_\kappa} \\ RC_{21t_1} & \dots & RC_{21t_\kappa} \\ \vdots & \ddots & \vdots \\ RC_{mm-1t_1} & \dots & RC_{mm-1t_\kappa} \\ = [\sigma_1^K \dots \sigma_k^K \dots \sigma_\kappa^K]. \end{array} \quad (\text{III.6})$$

$$= [\sigma_1^K \dots \sigma_k^K \dots \sigma_\kappa^K]. \quad (\text{III.7})$$

We can then simply define the correlation of covariance matrices for successive coarse-grained partitions as:

$$\rho_{K_k, K_{k-1}} = \text{corr}(\sigma_k^K, \sigma_{k-1}^K), k = 2 \dots \kappa, \quad (\text{III.8})$$

and similarly the correlation of the correlation of a fine-grained partition as

$$\rho_{F_k, K_k} = \text{corr}(\sigma_k^F, \sigma_k^K), k = 2 \dots \kappa - 1, \quad (\text{III.9})$$

where the variances and covariances of the fine-grained partitions, labelled σ_k^F , are constructed analogously with (III.7). It then remains only to specify suitable metric summarization functions g and h that capture enough of the distribution of the correlations. We will use:

$$g(\mathbf{d}_{K_k, K_{k-1}}) = \sqrt{\frac{1}{4} (\overline{\rho_g} + \sup \{\rho_{K_k, K_{k-1}}\} + 2)}, \quad (\text{III.10})$$

$$\overline{\rho_g} = \frac{1}{\kappa - 1} \sum_{k=2}^{\kappa} \rho_{K_k, K_{k-1}};$$

$$h(\mathbf{d}_{F_k, K_k}) = \sqrt{\frac{1}{4} (2 - \overline{\rho_h} - \inf \{\rho_{F_k, F_{k-1}}\})}, \quad (\text{III.11})$$

$$\overline{\rho_h} = \frac{1}{\kappa - 2} \sum_{k=2}^{\kappa-1} \rho_{F_k, K_k}.$$

Both $g, h \in [0, 1]$, and these formulations are suitable for use in a minimization problem per (II.8), given that we wish to simultaneously minimize the similarity between successive coarse-grained subseries and the difference between the coarse-grained subseries and some set of fine-grained subseries.

C. Formulation of the optimization problem

We are now ready to fully state the computable final form of our optimization problem. Consider a set of m data series each of length T , with the observations grouped into equal periods of length n of log differences of observations. Form a $(T-1) \times m$ matrix which can be completely partitioned into $\kappa \geq 3$ non-overlapping and contiguous coarse-grained subseries K_k , $k = 1 \dots \kappa$, and call this partitioning \mathbf{K}_κ , noting that the first and last subseries K_1 and K_κ are considered incomplete. Further, let \mathbf{F}_κ be a fine-grained partition of $\kappa-2$ sub-subseries F_k , $k = 2 \dots \kappa-1$ of a subseries in $\cup_{k=2}^{\kappa-1} \{K_k\}$, which is sufficiently short as to be considered of a different scale to \mathbf{S} as a whole.

We further require a minimum number of observations t_{min} in a given sub-subperiod in order for statistics such as covariance to be meaningful, and hence it is also a requirement that the length $\ell(F_k)$ of each fine-grained sub-subseries be at least t_{min} . In order to maintain scale differentiation we further require that the length $\ell(K_k)$ of each coarse-grained interior subseries be at least $(\kappa-2)^2 \cdot t_{min}$, the idea being that any coarse-grained subseries could at least contain a fine-grained partition which has all its sub-subseries of minimum length t_{min} . For the first and last incomplete coarse-grained subseries, we only require that they are of length at least t_{min} . Hence for a given T, t_{min} , we can calculate a maximum feasible number of coarse-grained subseries:

$$\kappa_{MAX} = \left\lfloor \sqrt{\frac{T-1-2 \cdot t_{min}}{t_{min}}} + 2 \right\rfloor, \quad (\text{III.12})$$

$$T > 0, 0 < t_{min} \leq \left\lfloor \frac{T-1}{2} \right\rfloor.$$

We wish firstly to minimize the similarity between each successive coarse-grained subseries K_k , $k = 2 \dots \kappa$ and the preceding subseries; note that we do not require dissimilarity of non-contiguous subseries, so a partition in which K_k is statistically very similar to $K_{k-\alpha}, \alpha > 1$ is permissible in this scheme. Next, for some fine-grained partition, we wish to minimize the dissimilarity of each sub-subseries F_k , $k = 2 \dots \kappa-1$ to its counterpart interior coarse-grained subseries K_k , $k = 2 \dots \kappa-1$. We might also consider a third objective, namely the maximization of the dissimilarity of successive fine-grained sub-subseries, but this is implied by the first two objectives, and is omitted in order to simplify the problem. Hence the biobjective minimization problem is:

Minimize

$$f(\mathbf{S}) = [g(\mathbf{d}_{K_k, K_{k-1}}), h(\mathbf{d}_{F_k, K_k})], \quad (\text{III.13})$$

subject to:

$$\ell(K_k) \geq (\kappa-2)^2 \cdot t_{min}, k = 2 \dots \kappa-1; \quad (\text{III.14})$$

$$\ell(K_k) \geq t_{min}, k = 1, k = \kappa; \quad (\text{III.15})$$

$$\ell(F_k) < \ell(K_k), k = 2 \dots \kappa-1, \quad (\text{III.16})$$

where $g(\mathbf{d}_{K_s, K_{s-1}}), h(\mathbf{d}_{F_k, K_s})$ are as defined in (III.10), (III.11).

IV. A SPECIALIZED EVOLUTIONARY ALGORITHM FOR PARTITIONING SELF-AFFINE MULTIVARIATE TIME SERIES

A. The computational complexity trade-off

When brute force methods which guarantee finding the global optimum are too computationally expensive and we wish to avoid use of greedy algorithms which guarantee finding only a local optimum, we often use iterative methods such as evolutionary algorithms (EAs) which although they may not find the true global optimum within a limited time frame given finite computational resources, will find a useful approximation depending on the computational resources available. Furthermore EAs are often designed so that the

Algorithm 1 Random initial assignment of constrained cutpoints (coarse-grained version)

Step 0: initialize a $T \times 1$ vector \mathbf{v} of true binary values and set the first and last t_{min} entries to false, indicating these zones are unavailable for cutpoints

Step 1: randomly assign a cutpoint to any point in \mathbf{v} with a true value and set the up to $(\kappa - 2)t_{min}$ entries $v_k, k \geq 1$ preceding the cutpoint and the up to $(\kappa - 2)t_{min}$ entries $v_k, k \leq T$ including and following the cutpoint to false, i.e $[\min\{1, c_k - (\kappa - 2)t_{min} + 1\}, \max\{c_k - (\kappa - 2)t_{min}, T\}] = false$;

Step 2: loop to Step 1 until $\kappa - 1$ valid cutpoints are found.

solution at each generation is at least as good as that provided at the previous generation, and so the top level problem common to all such procedures may be stated as:

Minimize

$$z = \frac{T_C}{C_T} \quad (\text{IV.1})$$

subject to

$$\mathbf{f}(\Omega) \leq \theta. \quad (\text{IV.2})$$

Here T_C is the computational cost for the algorithm; this could be measured in computation time if resources are fixed, proportion of resources used if resources are shared, or even a monetary cost if for example resources are paid for on a time-per-node basis. C_T is the value of the solution according to some measure, which might be a theoretical measure of progress, or else could be measured in more concrete ways, for example as a cost saving in a logistics problem or a trading profit in a financial application. The set of constraint parameters Ω is a subset of the parameters used in IV.1; $\mathbf{f}(\Omega)$ is often nonlinear (and may be combinatorial). This naturally includes all of the constraints used in the underlying optimization problem, but in addition could include parameters which have upper bounds set in the vector of constants θ governing not only cost or time or share of resources but also available memory, for example, including aspects such as the maximum size of individual arrays or maximum total size of all stored arrays. These parameters, which depending on algorithm design may also include population size or number of runs or number of generations, may be included in both the numerator and denominator of IV.1.

It is (IV.1) which will form the highest level of analysis for our optimization algorithm; we will take C_T to be (II.8), whilst in practice, we may not be able to calculate z a priori from analysis of the algorithm and computing resources, but may need to rely on empirical estimation. We will return to these issues as we examine the structure of our EA.

Algorithm 2 Random assignment of initial constrained cutpoints (fine-grained version)

Step 0: initialize a $(t_2 - t_1 + 1) \times 1$ vector \mathbf{v} of true binary values, where $[t_1, t_2] \in \mathbf{R}$, $t_{min} < t_1 < t_2 - t_{min} \leq T - 2$. t_{min} corresponds to a subinterval of a coarse-grained partition in Algorithm 1;

Step 1: randomly assign a cutpoint to any point in \mathbf{v} with a true value and set the up to t_{min} entries $v_k, k \geq 1$ preceding the cutpoint and the up to t_{min} entries $v_k, k \leq T$ including and following the cutpoint to false, i.e $[\min\{1, c_k - t_{min} + 1\}, \max\{c_k - t_{min}, T\}] = false$;

Step 2: loop to Step 1 until $\kappa - 3$ valid cutpoints are found.

B. General aims of the EA

Following [17], we note four general goals common in design of (a posteriori) multiobjective EAs:

- 1) Preservation of nondominated points;
- 2) Progress towards points on PF_{true} , the Pareto front (PF) representing the global optimal multiobjective solution set;
- 3) Maintenance of diversity of points on PF_{known} , the set of currently known nondominated solutions;
- 4) Provide the decision maker (DM) with a limited number of PF points on termination of the algorithm.

The first goal may be attained by virtue of the operation of genetic operators or through explicit or implicit elitism strategies. The second implies that successive generational PFs should themselves be nondominated with respect to previous PFs and should if possible be better than previous PFs, in that they either contain additional points that fill out the previous PF or contain points that dominate one or more points from the previous PF. The third implies that points on the PF should not be crowded into a small number of regions, as this may indicate similar crowding into particular regions of the representation space, i.e. a lack of diversity in search directions, and also does not provide the DM with a diverse set of combinations of objective values. The final goal simply indicates that solution sets with a large number of points may be counterproductive in not sufficiently distinguishing between solutions from the DM's standpoint, and may indicate deficiencies in the fitness function or other aspects of EA design. We will refer to these goals in describing the EA in following sections.

C. Functional description of the EA

1) *Representation:* Each individual is represented by a pair of integer vectors of $\kappa - 1$ cutpoint locations, one coarse-grained and one fine-grained, with the cutpoint representing the first point in each coarse-grained subseries or fine-grained sub-subseries.

2) *Initialization and specialization in selection of fine-grained subseries:* Random initialization of the population N of initial solutions is performed using Algorithms 1 and 2. However, were we to leave a single, unrestricted population, we should leave the evolutionary selection process open to

two undesirable features. Firstly, as is usual with unrestricted EAs, solutions would quickly crowd into certain sections of the solution space, potentially leading to premature convergence. Typical countermeasures to this crowding effect [17] include:

- A weight-vector approach, where different weights are applied to bias the search and move solutions away from neighbours;
- Niching, where a penalty is applied to the fitness of solutions based on the number of solutions sharing some neighbourhood. MOEAs employing versions of this strategy include NSGA [65];
- Crowding/clustering, where solutions are subject to selection based on a crowdedness metric, as used in NSGA-II [21];
- Relaxed dominance, as in the ϵ -dominance technique used in [40];
- Restricted mating, where a minimum distance is required for recombination between any two given individuals.

In some other approaches, diversity may be intrinsic to the EA design; for example [69] finds that the decomposition technique that forms the basis of MOEA/D leads to a good chance of producing a uniform distribution of Pareto solutions.

In this EA, we subdivide the population into a number of subgroups, and restrict the range $[t_1, t_2]$ within which each fine-grained subseries may be initialized, but with the requirement that the whole of S is covered, so that these ranges usually overlap. No restriction is placed on the coarse-grained subseries, which are initialized over the whole of S . Since crossover is only allowed within a given subgroup, and mutation is also restricted (see Subsection IV-C7 below), the algorithm is forced to try to find solutions involving fine-grained sub-subseries within the initial range unless mutation changes the available range. This maintains diversity, but it also solves the problem that in the absence of such subdivision, selection would over time push the fine-grained sub-subseries towards the same scale as the coarse-grained subseries, yielding a trivial solution set. This approach differs somewhat from the Island Model of [68] and similar distributed EAs in that no migrations of individuals between subgroups or crossover between individuals from different subgroups is allowed, and the subgroups overlap, potentially to an increasing degree under the operation of mutation over numerous generations, which to some extent substitutes for migration (see Subsection IV-C5).

The number of subgroups is set to

$$I = \left\lceil \sqrt{T/t_{min}} \right\rceil; \quad (\text{IV.3})$$

initial cutpoints for the coarse-grained partitions are determined randomly but subject to the length constraints (III.14),(III.15), and initial fine-grained partitions are set by first selecting two successive points from a random permutation of available coarse-grained cutpoints as the start and end points and then selecting fine-grained cutpoints randomly, subject to constraint (III.16). This initialization scheme is designed, subject to the amount of data available, to produce a wide variety of initial fine-grained partitions, in terms of both size and location, which also cover the entire data set.

On average the length of the initial fine-grained partitions decreases with increasing κ . Note that initialization is in effect a constrained a priori multiobjective optimization process in itself, given the need to balance variety and total coverage of the fine-grained partitions and obey all length constraints, and in practice Algorithms 1,2 requires careful programming to ensure valid results, especially if solutions are sought entailing large numbers of partitions.

3) *Fitness function and invariance properties of correlation:* We will now consider the suitability of metrics of the type developed in Section III for developing a fitness function for use in the EA. For the purpose of summarizing the distributional properties of a given subseries, variance-covariance has the advantages of familiarity and well-understood properties, and as already set out, realized covariance has additional advantages in terms of simplicity and speed of computation. Furthermore, there are specific properties of invariance in respect of the standard coefficient of correlation between two sets of subseries variances-covariances that will prove most useful in relation to the value of our fitness function in assessing similarity and difference under affine transformations.

Recall firstly the basic property of the correlation coefficient that

$$\text{corr}(a_1x_1 + b_1, a_2x_2 + b_2) = \text{corr}(x_1, x_2) \quad (\text{IV.4})$$

provided that $a_1a_2 > 0$. Now let X and Y be sets of observations, with the latter an affinely-transformed copy of the former, and $\rho_{X,Y} = \text{corr}(\sigma_X, \sigma_Y)$ be the correlation between the set of m realized variances and $m(m-1)/2$ realized covariances for X and Y , stacked into column vectors σ_X and σ_Y as defined in (III.7); then from (IV.4),

$$\text{corr}([\mathbf{a} \cdot \sigma_X + \mathbf{b}], [\mathbf{c} \cdot \sigma_Y + \mathbf{d}]) = 1, \quad (\text{IV.5})$$

where \mathbf{a}, \mathbf{c} and \mathbf{b}, \mathbf{d} are non-zero real-valued vectors. This leaves $\rho_{X,Y}$ invariant to affine transformations. In particular, $\rho_{X,Y}$ is invariant to stretching or squeezing in either the frequency or amplitude domains. Note that since we are dealing with log differences of the form $r_{j,t} = \ln v_{i,t} - \ln v_{j,t-1}$, then with regard to the original data, applying a translation will change the slope of the time series trajectory, whilst applying a scalar linear transformation $\mathbf{r}_j \mapsto a_j \cdot \mathbf{r}_j$ is equivalent to applying a nonlinear, power law effect to the original observations.

These properties are inherited by the metrics used in our constrained objective (III.13). In the case that successive coarse-grained subseries in a given partitioning are all identical except for affine transformations of the types specified, then $g(\mathbf{d}_{K_k, K_{k-1}}) = 1$, whilst if successive fine-grained sub-subseries are all identical to their reference coarse-grained subseries except for affine transformations of the types specified, then $h(\mathbf{d}_{F_k, K_k}) = 0$. More importantly, in the data if successive coarse-grained subseries are close but for such affine transformations, in other words the underlying DGPs are quite similar, then the objective function f_1 should have values close to 0, whilst if the underlying DGPs are quite different, objective values will approach 1 as the correlation approaches -1. For f_2 , the value will approach 1 if the correlation of

the relevant pair of coarse-grained and fine-grained subseries approaches -1, and will be close to 0 if they are quite similar.

Hence our biobjective fitness function will be (III.13), with the functions defined as per (III.10) and (III.11), and the constraints (III.14),(III.15),(III.16) are dealt with as far as possible by obeying these constraints at all stages of the programming.

4) Fitness function evaluation and the set of Pareto fronts: After an initial calculation of T vectors σ_t , each containing sets of $m(m+1)/2$ stacked realized variance and covariance entries $RC_{ijt}^{(n)}$, fitness function evaluation using the form outlined above requires for each subgroup with population N_g at each generation, only κ summations of vectors, as per (III.5), which is far simpler in computational terms than recalculating covariance matrices repeatedly. It is also guaranteed that all individuals in the population at all generations will be feasible, provided all the original σ_t have valid entries.

We can then find the Pareto front for each subgroup. For a minimization problem with k objectives, let M be the set of at most k points p_i^{\min} out of n total which have the minimum values for each individual objective f_i ; that is:

$$M = \{p_i^{\min} = \min \{f_i(p_j)\}, \quad (IV.6) \\ i = 1 \dots k, j = 1 \dots n\}.$$

Define Pareto dominance in the usual fashion:

$$p_A \succ p_B \iff \quad (IV.7)$$

$$f_i(p_A) \geq f_i(p_B) \forall i \text{ and} \quad (IV.8) \\ \exists i | [f_i(p_A) > f_i(p_B)].$$

This leads to the definition of the nondominated set:

$$PF = \{p \in \Omega : p \not\succ p' \forall p', p \neq p'\}. \quad (IV.9)$$

It follows that

$$\|p_j\|_1 > \|\sup M\|_1 \iff p_j \notin PF \quad (IV.10)$$

where $\|p\|_1$ is the taxicab norm of the objective function values, that is

$$\|p_j\|_1 = \sum_{i=1}^k |f_i(p_j)|, \quad (IV.11)$$

and $\|\sup M\|_1$ is the norm of the lowest values for each objective of any of the points in M ; all such points must be dominated by at least one point in M , and we can immediately eliminate them from consideration as members of the PF. In the biobjective minimization case, this then implies that if we first take the point with the lowest global value of f_1 , all points with a higher value for f_2 than that point can be excluded; we then find the point with the next lowest value for f_1 from the points not dominated by the first point, and so on.

We do not need to sort the points at any stage, only find the suprema of successively smaller nondominated sets. Taking the number of points on the final PF as n_{PF} and the number of remaining nondominated points at each iteration as m_i^* the complexity is thus

$$O \left(m + \sum_{i=2}^{n_{PF}} m_i^* \right) \leq O(n_{PF} \cdot M). \quad (IV.12)$$

Algorithm 3 Fast biobjective Pareto front algorithm

Step 0: initialize with m pairs of fitness function values $\{f_1, f_2\}$;

Step 1: find and archive the point p_1^{\min} with the lowest value for f_1 ;

Step 2: eliminate all points with a value for f_2 more than or equal to $f_2(p_1^{\min})$;

Step 3: loop to Step 1 whilst any points remain.

This compares favourably with the $O(kM^2)$ complexity of exhaustive algorithms. Because ours is a biobjective problem using continuous values and all individuals are unique, we can use a particularly fast and simple algorithm (3) to find the PF. The algorithm works as a simplification of the following more general case with m objectives, and does not need to calculate norms.

Each subgroup thus yields its own PF at each generation and these are stored and added to the next generation only for the purpose of calculating the new PF, which will change only if new points are found that dominate points on the old PF so that the location of the PF changes and the number of points thereon may shrink, or if additional nondominated points are found, so that the number of points on the PF grows. We can apply Algorithm 3 a second time to these points if we wish to find a global PF for the union of populations all the subgroups, and the result will be the same as if we had calculated the global PF directly from this total population, but a further advantage of the subgroup approach is that computation is generally faster if we compute in two stages. However the union of the subgroup PFs is itself of interest, as discussed in Subsection V-D.

5) Permutated biobjective tournament selection : Selection is performed per Algorithm 4; note that the number of participants in each tournament, τ , is randomized so that selection pressure [9] for each subgroup at each generation is also random. Individuals may compete in more than one tournament and all individuals will compete at least once; individuals in the population PF found in IV-C4 will win all tournaments in which they take part; using complete permutations of the population also removes the chance that elite individuals fail to participate in any tournament and obviates the need for explicit elitism. Note also that elite individuals are stored and hence will remain nondominated until the EA terminates unless new individuals emerge that dominate them.

6) Permutated affine crossover: Crossover is performed as per Algorithm 5. All individuals are subject to crossover and existing individuals are passed intact only if two copies are sampled for the same tournament, although individuals on the PF are ultimately recorded and stored; hence there is no parameter associated with crossover probability. The spacing conditions referred to are those used in Algorithms 1,2. In practice, programming Algorithm 5 in such a way that each new individual is guaranteed to have a feasible pair of sets of cut points obeying all constraints may be very computationally

Algorithm 4 Permuted tournament selection

Step 0: initialize with objective values of current population, including those on the PF;

Step 1: randomly permute the current population;

Step 2: Take the next $\tau = X$ of \tilde{N} remaining individuals in the permutation, where X is uniform random $\in [2, \tilde{N}]$; use Algorithm 3 to find non-dominated individuals, and add copies of these to the list to be passed to genetic operators;

Step 3: while individuals remain in the permutation which have not competed, loop to Step 2;

Step 4: whilst the list of copies of individuals to be passed is smaller than the twice the desired population N , loop to Step 1.

Algorithm 5 Affine crossover

Step 0: initialize with list from Algorithm 4;

Step 1: randomly split list into 2 parent lists;

Step 2: take the next parent from each list; if both parents are copies of the same individual, add an untransformed copy of that individual to the next generation and move to the next entry pair;

Step 3: merge the sets of coarse-grained and fine-grained cut points from each parent;

Step 4: check which cutpoints in the combined lists (if any) violate no spacing conditions;

Step 5: randomly remove one cutpoint not included in those found in step 4 (if any) or otherwise randomly remove any one cutpoint;

Step 6: loop to Step 4 until the required number of cut points for the number of subperiods κ is reached and all spacing conditions are met;

Step 7: if left with insufficient cutpoints, randomly add back deleted cutpoints which do not violate spacing conditions;

Step 8: loop to Step 1 until all pairs of parents have been subject to crossover.

expensive and it may be necessary to allow through a certain number of infeasible individuals to be caught and eliminated by error checking later. The effect of combining sets of cutpoints in this way is to stretch or squeeze the coarse-grained subseries and fine-grained sub-subseries in a way tantamount to applying offsetting affine transformations in the time domain so that the new set of cutpoints does not change the length or location of the specialized subperiod on which the fine-grained cuts are determined. It is worth noting that the crossover scheme implemented in Algorithm 5 is highly specialized and it is not possible to replicate its operation using general purpose EAs. Employing a more standard crossover scheme, even with many constraints, results in an unacceptable number of infeasible individuals at each generation.

7) *Mutation:* A simple random point mutation is applied to the whole representation of the individuals in each generation, so that either either the coarse-grained and fine-grained part of the representation is affected, per Algorithm 6. If an individual is selected for mutation, one cut point is deleted and another one randomly inserted in such a way that spacing conditions

Algorithm 6 Mutation

Step 0: initialize with new population generated from Algorithm 5;

Step 1: randomly select individuals based on the mutation probability (IV.13);

Step 2: randomly select one cutpoint for each selected individual, delete these cutpoints and replace with new valid cutpoints subject to length constraints; for the fine-grained partitions, allow new cutpoints up to t_{min} before and after the original range of the partition;

Step 3: loop to Step 2 until mutation complete.

are again met. A difference with crossover is that a small time window of length t_{min} is added to the beginning and end of the list of valid locations in the specialization subperiod for the purpose of randomly determining the location of the new cut-point, so that it is possible for the fine-grained sub-subseries to migrate in the time domain. The mutation threshold probability should be set quite low, so that most mutation will typically still occur within the original specialization parameters, and this migration is typically slow and not of great magnitude unless the number of generations is large. This migration is the only mechanism by which the original specialization locations can be changed, but the algorithm needs to prevent the tendency for the scale of the fine-grained subseries to simply expand over many generations towards the scale of the coarse-grained subseries, leading to favouring of individuals with fine-grained subseries almost identical to their coarse-grained subseries. In the algorithm, the mutation rate is fixed as

$$mutRate = \frac{I}{N}, \quad (\text{IV.13})$$

so that on average just one population member per subgroup will be subject to mutation in each generation. Note that the number of subgroups I is itself a function of the length of the dataset, per (IV.3).

D. Parameters, parallelization and the specialized nature of the EA

A notable feature of the construction of the EA is that it has very few choice input parameters; in fact the only choices are:

- N , the total population size;
- G , the number of generations;
- t_{min} , the minimum number of observations in any sub-period.

In practice, given limited computational resources, whilst results can be expected to improve with higher N, G , the first is limited by available memory and if parallel processing is used, the available number of cores, and the second by processor speed and the number of cores used. The third parameter t_{min} can generally be left to the minimum meaningful value of 2, as it was in all the studies in Section V, but might need to be adjusted for very large and complex problems if it is found that too many infeasible individuals are generated. This problem did not come up in our testing for this paper, however.

On the subject of parallelization, in general the specialized EA shares attributes with many other EAs, namely that:

- Multiple runs can be executed in parallel, with the same or different parameters;
- Many operations within a single generation can be parallelized;
- The whole set of operations on a given subgroup within a single generation can be parallelized;
- Generations themselves cannot be parallelized within a given run, by the fundamental nature of the evolutionary process.

The first and fourth aspects are standard and require no further explanation. As to the second, all of the key algorithms as set out in Algorithm 7 below for each generation can be parallelized. Furthermore, certain aspects, such as the evaluation of the fitness function and tournament selection, are also tractable to GPU computation by virtue of their simplified nature, though this is not currently implemented. Finally as to the third, operations on different subgroup can be entirely parallelized across generations, unless any adjustment to the groups is required from generation to generation, for example changing population sizes in an adaptive manner; such intergenerational adjustments were not implemented in the EA.

As noted, the algorithm permits only 3 choice input parameters, and the values used for these is generally driven by practical matters including available time and computational resources as well as the size and complexity of the data used, rather than any speculation on the part of the user about what parameter settings might produce the best results. For this reason, if multiple runs are conducted, it should only be to increase confidence in the optimality of results, and it may well be that better results are to be obtained by using available computation time and resources to increase N, G , rather than performing more runs, with the caveat that this may also depend on the nature of the data.

Otherwise as regards selection and genetic operations, the use of permuted biobjective tournament selection (Subsection IV-C5) means that all individuals in each generation are sampled for at least one tournament and have the same chance of being sampled for more than one tournament, depending on the number of victors produced by the initial permutation. Rather than the number of participants per tournament τ being set arbitrarily (many algorithms uses $\tau = 2$), τ is randomized, and as a consequence so is selection pressure. In crossover (IV-C6), individuals are passed as is only if two copies are assigned as parents in a particular pairing, and all individuals passed from tournament selection are assigned as parents, so there is no crossover probability parameter. Finally, the mutation probability is determined by population size and length of the dataset (IV.13).

It is also worth emphasizing that the EA is highly specialized in several respects, but most notably in the initialization, the type of fitness function and its method of calculation, the affine crossover algorithm and the design of mutation to avoid infeasible individuals. Some of these innovations can be expected to improve performance for this very specific task against other types of algorithms, but others, in particular the crossover and mutation algorithms, are designed to avoid

Algorithm 7 Evolutionary algorithm for partitioning self-affine multivariate time series

- Step 0:** load $(T - 1) \times M$ array of log differences of the multivariate time series S . Generate initial population divided into subgroups using Algorithms 1 and 2;
- Step 1:** calculate fitness function and find subgroup PFs using Algorithm 3;
- Step 2:** perform tournament selection using Algorithm 4;
- Step 3:** perform crossover using Algorithm 5;
- Step 4:** perform mutation using Algorithm 6;
- Step 5:** loop to Step 1 until convergence conditions met or maximum number of generations reached.
-

the population being swamped with infeasible or very low fitness individuals, which is what should be expected if general purpose EAs are used for this problem. This, together with the issue of how to set the many parameters general purpose EAs generally require, makes comparison with existing EAs very tricky in practice, and for these reasons we have not included any such comparisons in this paper.

The entire EA is summarized as Algorithm 7.

V. TESTING WITH SIMULATED AND REAL DATA

A. Formation of the simulated data

Testing with real multivariate data, especially of higher dimension, is problematic as in general we do not know how to partition the data, this being precisely the problem the techniques developed in this paper are designed to address. A first step is to formulate a set of data we know more about, specifically one in which we know that the data inherently displays self-affinity. The approach used in testing is to “stitch” together subseries which are individually self-affine and which repeat patterns from one subseries to the next, but which are each sufficiently different to the preceding subseries in terms of the coarse-grained metric (III.10) that the EA can detect a change in subseries.

The fundamental building blocks of this approach are multivariate fractionally integrated time series (FITS), which use FBM processes in their construction. The approach used is the p-model set out in [20], and the implementation is adapted from [67]. The p-model itself produces only stationary random time series and has one parameter, $p \in [0, 1]$, which is associated with increasingly peaked series as it approaches 0 or 1, and calmer series as it approaches 0.5. To create nonstationary series, the result of the p-model is filtered in Fourier space and a further slope parameter is specified; slopes flatter than -1 are called stationary in [20], whilst slopes between -1 and -3 are called nonstationary, with stationary increments. These nonstationary cases are at least continuous, but not differentiable. Slopes steeper than -3 are nonstationary and differentiable.

The approach in constructing multivariate test series with differentiated subseries is as follows. Firstly, for each of κ subseries as required, m “master” series of length $n \cdot T_k$, $k = 1 \dots \kappa$ and random parameters $p \in [0.25, 0.49]$ and $slope \in$

$[-3, -1]$ are generated, the random parameters recorded and log differences taken and the series “stitched” together so that the total length is $n \cdot T$. The parameter ranges are set so that the series are neither too peaked nor excessively smooth and are nonstationary. Next, for each subseries $k = 1 \dots \kappa, m$ FITS of length T_k are generated for each subseries, again each with independent uniformly random parameters but with the random seed reset to the same state s recorded before the first series was generated. Hence the random numbers used are the same for each FITS, but the parameters are different, such that the correlations between different FITS vary randomly. Log differences are taken and scaled down by a factor of $n^{1/\alpha}$, where α is a scaling parameter, and each t -th log difference from these series is added to the $t \cdot n$ -th log difference of each of the m FITS. The effect is to add an additional shock to the beginning of each period. This might be thought of for example in the context of financial returns series, as reflecting new information available at the beginning of each trading day, and is meant to simulate the phenomenon of price jumps at the beginning of the trading day which is commonly observed in financial data. Finally the entire system is normalized to be non-zero and have the same starting values.

It is not of course guaranteed that the stitched FITS generated by such a process will have subseries that are sufficiently distinct, in terms of the serial correlation of the realized covariances of the constructed coarse-grained subseries, to be useful in testing our EA. Therefore, suitable FITS are found by running a Monte Carlo simulation and selecting FITS with sufficiently distinct subseries. With small values of T , it is not hard to find series with significantly negatively correlated subseries, but with larger values of T it becomes difficult to find series with maximum correlations between successive subseries that are any lower than a small positive number (less than 0.1, say). This means that the task faced by our EA will not necessarily be easy, as the original cutpoints in the construction of the simulated data are hard to identify. The actual stitched FITS generated and used in the first study with just described below are illustrated in Figure (V.1), with actual observations plotted in the upper window, and log differences in the lower window.

B. Results from the EA for simulated data

In an initial experiment the generated FITS comprised $m = 8$ series with 2 central partitions and a further 2 incomplete subseries at the beginning, so that $\kappa = 4$, with a total length of $T = 512$ periods, each comprising $n = 32$ observations, for 16,384 high frequency observations in all. Because of the difficulties in constructing test data with significantly negative serial correlations between subseries, especially with larger numbers of periods, correlation coefficients observed in data used tend to be slightly negative and close to zero (typically in $[0, -0.1]$). The EA was run over 15 generations with a population size of 10,000 individuals for each of 11 subgroups. All testing was conducted on a 3.7GHz, 4 core system with 32GB of RAM. Earlier experimentation had shown that such population sizes could lead to relatively rapid convergence with reasonable results, whilst smaller population sizes tend to lead

to convergence after more generations but with poorer results, though possibly with lower overall runtime. The number of series in the multivariate system was chosen so that sufficient covariances were generated to make correlations meaningful but not so many that in particular the clarity of graphical representations is compromised. Computational complexity is also an issue but for reasons explained in Subsection IV-C4, the choice of m affects only the $\kappa - 1$ correlation calculations, generally in a sub-quadratic manner, and does not affect the underlying calculations of realized covariance at each generation. Hence this approach is potentially well suited to high-dimensional problems.

Two measures were considered to assess the success of the EA in finding suitable coarse-grained partitions. The first looks at the errors between the individuals with the best (i.e. lowest) values for the first objective f_1 , i.e. the solutions with the highest dissimilarity between regimes, as assessed by the following formula:

$$\theta_{i,r,g}^{(1)} = \frac{1}{T \cdot (\kappa - 1)} \sum_{k=1}^{\kappa-1} |c_{k,i,r,g}^{EA} - c_k^{FITS}|, \quad (V.1)$$

where $c_{k,i,r,g}^{EA}$ is the k -th cut point for a given subgroup, partition and generation, and c_k^{FITS} is the actual cut point used to generate the FITS; lower scores are better. The second is derived from our fitness metric (III.10) as follows:

$$\begin{aligned} \theta_{i,r,g}^{(2)} = & \left[\sqrt{\frac{1}{4} \left(\overline{\rho^{FITS}} + \sup \{ \rho^{FITS}(s, s-1) \} + 2 \right)} \right. \\ & \left. - \sqrt{\frac{1}{4} \left(\overline{\rho_{i,r,g}^{EA}} + \sup \{ \rho_{i,r,g}^{EA}(s, s-1) \} + 2 \right)} \right], \end{aligned} \quad (V.2)$$

that is, the difference between the coarse-grained metric values for the best individual generated by the EA and for the FITS cut points; again, lower scores are better. Note that these measures consider only the coarse-grained partitioning for a given solution.

The best results in terms of the first measure had a maximum error of 5 periods, i.e. an “error” of less than 1%, for any cut point (there were several individuals, distributed across different subgroups, which met this standard). The results demonstrate that the EA can find the “correct” cut points with a good degree of accuracy, although larger T and κ will lead to slower convergence or equivalently, lower accuracy for a given computational budget. One solution is illustrated in Figure V.1, where the solid lines indicate the cutpoints in the original data and the dashed lines the coarse-grained cutpoints found by the algorithm.

A second simulated FITS dataset was then generated for the main experiment with $m = 8$ and $n = 32$ as before but with a much larger number of periods, $T = 4096$, or 131,072 observations in all, and $\kappa = 6$, so 4 internal partitions instead of 2; constructing the test data was significantly more computationally expensive, and this time the correlations of successive subseries tend to be close to zero but slightly

Figure V.1. A typical close fit to the original subseries cut points

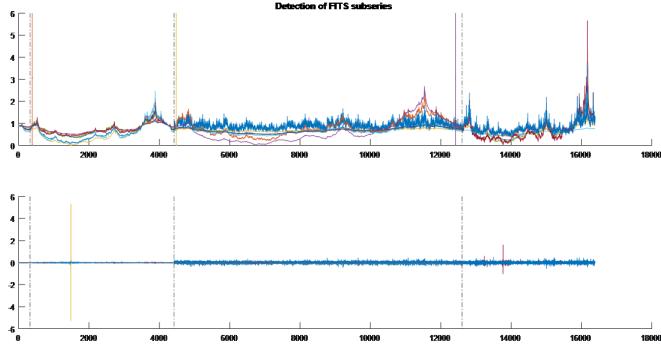
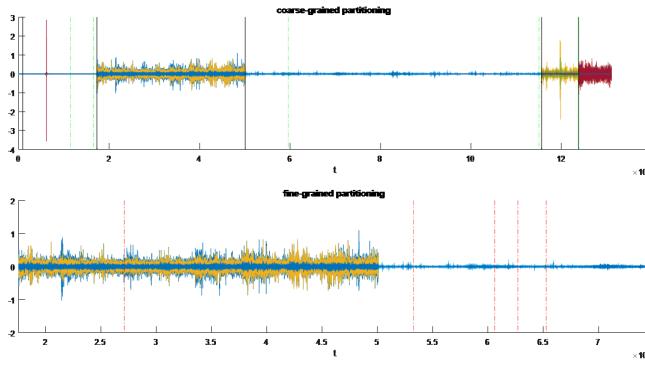


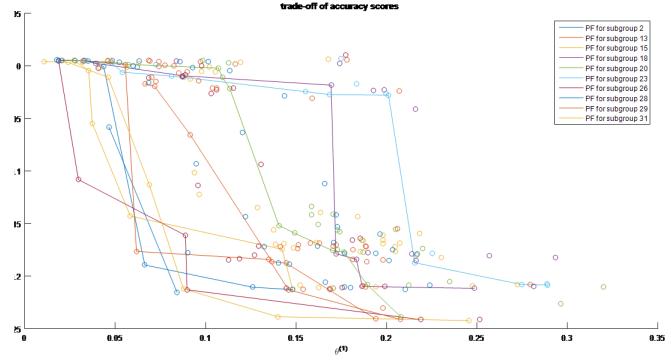
Figure V.2. Coarse-grained and fine-grained partitionings



positive, typically in $[0, 0.1]$. The EA was then run for 100 generations with a total of 32 subgroups, each with a population of 10,000, and a total of 30 runs, and the results were aggregated by subgroup across the runs. Typical running times for this data were around 22 seconds per generation but this dropped to below 7 seconds when parallelization of subgroups across 4 cores was implemented, showing that parallelization is highly effective in reducing computation time.

Figure V.2 shows an example result. In the upper plot, the solid black lines represent the coarse-grained cutpoints used to set up the actual dataset, whilst the dashed lines represent the cutpoints found in the EA solution. The plots of the data use log differences rather than the untransformed data, and this makes the distinctions between the partitions much clearer. In the lower plot of Figure V.2, only a zoomed-in section of the data is shown, and the dashed lines represent the cutpoints of the fine-grained partitioning found in this particular solution. Note that the relative lengths of the subseries in the fine-grained partitioning are quite different to those in the coarse-

Figure V.3. trade-off of theta scores



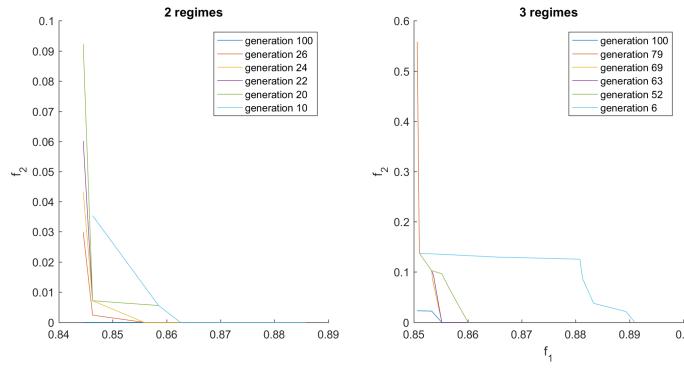
grained partitioning. It is also quite difficult to detect by eye, even with a system with just 8 different series, that the subseries in the fine-grained partitions correlate quite closely with those in the coarse-grained partitions, even though this is in fact the case; for a higher-dimensional system, visual interpretation becomes impossible, but for the EA, only a relatively modest increase in computational complexity is involved.

Figure V.3 shows scatter plots of solutions for selected subgroups in terms of their values for $\theta_{i,r,g}^{(1)}$ and $\theta_{i,r,g}^{(2)}$, the one on the left after the first generation, the one on the right after the last. The lines represent Pareto fronts for each of the subgroups shown; these PFs are not to be confused with the ones constructed during the running of the EA, which use metrics (III.10) and (III.11) instead of (V.1) and (V.2), and so have no knowledge of the partitioning used to set up the simulated data. The best results from the EA show an error (as measured by $\theta^{(1)}$) of less than 2%, and manage to find reasonable fine-grained partitionings as well.

There is a clear trade-off between $\theta^{(1)}$ and $\theta^{(2)}$, but notice also that most individuals have a negative value for $\theta^{(2)}$, indicating that in terms of metric (III.10), the EA finds coarse-grained partitions of subseries that are actually better differentiated than those used in the original construction of the test data. Also note that, although the EA has no knowledge of the partitioning used to set up the simulated data but which is used in the ex post calculation of $\theta_{i,r,g}^{(1)}$ and $\theta_{i,r,g}^{(2)}$ after the EA has run, at the last generation the EA has in fact found many more solutions close to the bottom left extremes of the plot, that is, solutions with, in particular, better solutions in terms of $\theta^{(1)}$, than was the case at the first generation. Note also that no single subgroup's PF completely dominates that of all others, though some are completely dominated; indeed, just 6 of the original 32 subgroups (only 10 of which in total are shown in the plots, for clarity) have PFs containing points which are non-dominated with respect to all other points from all subgroups.

It would be simple to extract from the several subgroup PFs

Figure V.4. Progress of PFs by generation



for any given partition into κ subseries a single global PF. However, each subgroup PF is a valid subset of solutions on its own, and if not completely dominated by another subgroup PF, has valuable information about the solution space, given that each subgroup is optimizing over a different location of the fine-grained partition and potentially also a different scale. Similarly, with real data where the actual number of partitions is unknown, for each partition size $\kappa = 3 \dots \kappa_{MAX}$, that solution subset is also valid in its own right, and in many cases we cannot say that one number of subseries κ forms a superior partition to another number; it is again operating at a different scale. Taken together, all the subgroup PFs for all the partition sizes κ may be thought of as forming a single solution set over a number of different scalings and fine-grained subseries locations; in other words an optimized sample of a much larger optimal solution set.

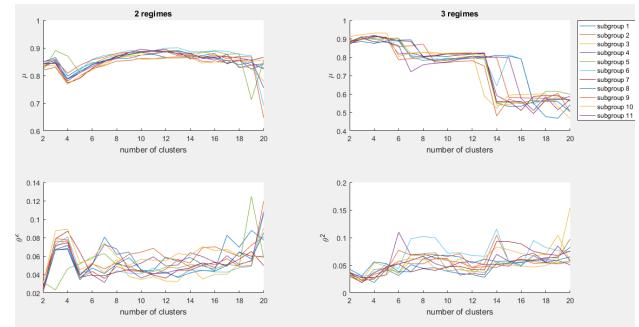
C. Testing with real data

We used the public access CRSP data¹ for second-by-second calculations of 9 capitalization and market indices calculated using intraday prices of US stocks over a 12-month period ending March 2016. As examples, CRSPSCT is an index of the total return of smaller stocks, whilst CRSPTMT is an index of total market return. To ensure greater stability of the realized covariance matrices and reduce noise, the second-by-second measurements were first aggregated into 5-minute bars and the daily covariance matrices were then calculated from these. The system is highly correlated, with correlations between series over the period varying between 0.0072 and 0.9985 but averaging 0.7718.

D. Results from the EA for real data

A total of 100 runs were effected for the data and the results, i.e. the sets of coarse-grained and fine-grained cutpoints with objective values on the PFs produced after 100 generations for each run, were recorded for each subgroup. This produced a

Figure V.5. Means and variances of silhouette numbers



large number of partitionings, all of which are potentially valid (as they are non-dominated). The EA was used to investigate partitionings with 3 and with 4 cutpoints (i.e. 2 or 3 regimes), and was run with a smaller population per subgroup (1000) and a smaller number of subgroups (11) than used in the experiment with simulated data, and 100 generations per run as before. The progress of the combined PFs for the whole population, aggregated from all runs and subgroups, is illustrated in Figure V.4. Note that in later generations the f_2 values are extremely small (and so close to the axis) and this phenomenon is seen more quickly for the runs with 3 regimes; this provides strong evidence of the self-affinity of the real data set.

We used k-means to cluster the sets of coarse-grained cutpoints obtained from all the PFs obtained from the EA, with $k = \{2, 3, 4, 5\}$, running the algorithm 1000 times for each k and retaining the results with the lowest sums of in-cluster distances from centroids. We then calculated silhouette numbers [61] for each subgroup; V.5 shows the means and variances for each subgroup, for both 2 and 3 regimes. The subgroups generally show similar patterns. For 2 regimes, the silhouette means do not vary greatly but variances increase with the number of clusters, indicating more low silhouette values. For 3 regimes, means decrease significantly for more than 4 clusters and variances also increase with the number of clusters. This indicates that a low number of clusters is most supported by the data, with in all likelihood, just 2 clusters being best of all in the case of 2 regimes, whilst for 3 regimes, results are very similar for 2-4 clusters. Overall this implies that after 100 generations, results were already tightly grouped into a small number of clusters. In particular, the number of clusters is much smaller than the number of subgroups, indicating some convergence of results between subgroups. The silhouettes for different numbers of clusters for 2 regimes and for 3 regimes are shown in Figures V.6 and V.7; note that whatever the number of clusters, with 2 regimes there are 2 clusters of results that contain most of the results, whilst with 3 regimes a single cluster contains most of the results.

Figures V.8 and V.9 for partitionings with 3 and 4 cutpoints respectively show the indices together with vertical lines

¹ Available at: <https://wrds-web.wharton.upenn.edu/wrds/about/index.cfm>

Figure V.6. Silhouette plots for clusters of results with 2 regimes

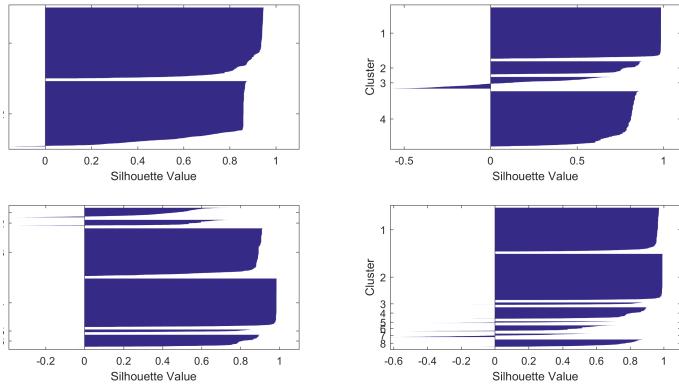


Figure V.7. Silhouette plots for clusters of results with 3 regimes

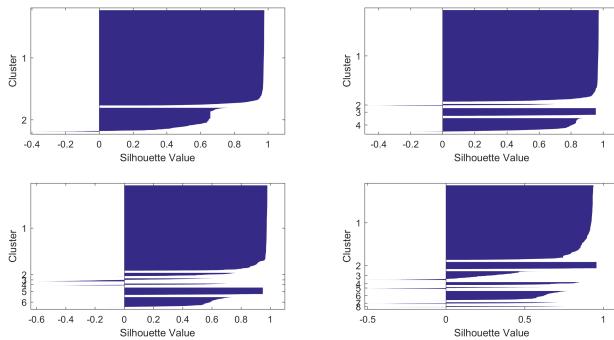
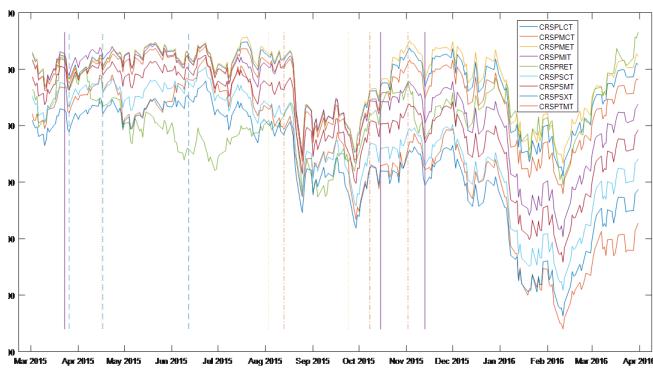


Figure V.8. CRSP data with representative partitionings: 3 cutpoints



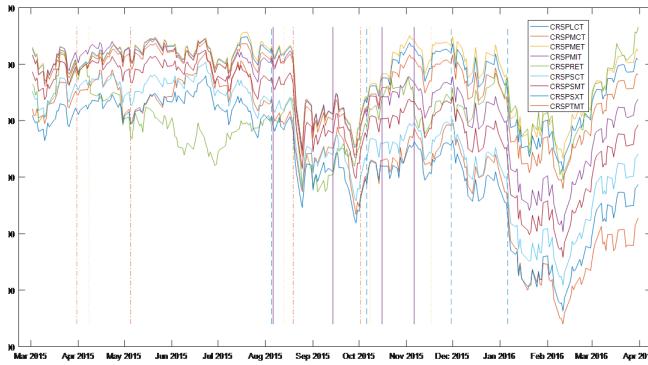
indicating the centroids of the 4 clusters; each cluster has 3 cutpoints with the line colour and line type the same, which may be taken as representative of a particular type of solution with two defined partitions (which have corresponding fine-grained partitions) plus “incomplete” partitions before the first and after the last cutpoint.

With real data, we do not know the “true” partitionings although with small systems we might be guided by visual cues, standard multivariate statistical techniques or some other known facts in guessing one or more valid partitionings; with large systems, such guesses may be impossible. If we look at the results as represented in the two figures without any prior knowledge, several observations can nonetheless be made regarding the representative partitionings. Firstly, the “incomplete” final partition of the data lying after each final cutpoint is in all cases quite large, with almost all final cutpoints indicated before the end of 2015 and some much earlier. This tells us either that later data the system is considerably different from earlier sections in terms of our coarse-grained metric III.10 or that it is easier to find similar fine-grained partitions in terms of our fine-grained metric III.11 for earlier data, or both. With this relatively low-dimension and highly correlated dataset, it is perhaps possible to note visually that there is a change in the system in late 2015, but it is not feasible to check the similarity of coarse-grained and fine-grained partitions visually even with such a relatively small system. Secondly, the partitionings vary considerably in spread, i.e. the distance between the first and last cutpoints, so some solutions involve much larger “incomplete” first and last partitions than others, or looked at differently, the partitionings in effect operate on different time scales to one another; this is arguably a desirable feature in the context of our assumption that the system has scalable, fractal properties and is likely maintained by the island approach to segregating the EA population. Finally, the greatest concentration of cutpoints is in the August-November period, this being particularly noticeable in the second figure, and this is perhaps indicative of a greater change in the system covariances during that period.

CONCLUSION

We have seen that the problem of how to best partition a multivariate time series into subseries with different distributional properties is an important one with many potential uses in different areas of research, but is also at root a difficult combinatorial problem with high computational complexity. The problem becomes more complex still if we assume self-affinity in the underlying DGP; yet many real data types, including but by no means limited to financial data, display this attribute. Identifying valid partitions on this basis may not only be the best way to identify time-varying features of the data but may also open the door to prediction of future states, or at least identification of the current state on some scale. However, statistical techniques currently available are not well-suited to high-dimensional multivariate analysis of time series showing time-varying, self-affine distributional attributes. Furthermore, all rely on assumptions about the underlying DGP and often on large numbers of model parameters.

Figure V.9. CRSP data with representative partitionings: 4 cutpoints



The novel approach investigated in this research makes only the simplest of assumptions about the DGP and uses only 3 input parameters, 2 of which, the population size N and the maximum number of generations G , relate to the computational structure of the EA rather than being model parameters as such, with the third, the minimum partition size t_{min} , in practice being set to the effective minimum value 2 in all experiments. The price paid is that the starting computational complexity is very high. To address this, a highly parallelizable population-based evolutionary algorithm was developed, which reduces the problem to a biobjective one using objective functions based on the correlation of realized covariances for successive coarse-grained subseries and for fine-grained sub-subseries with the coarse-grained subseries.

The problem is formulated so as to minimize the similarity between successive coarse-grained subseries and maximize the similarity between this coarse-grained partitioning and some fine-grained partitioning at a smaller time scale, using functions that summarize each objective. This summarizing approach significantly simplifies the problem, yet still yields a set of solutions for analysis *a posteriori* rather than a single solution based on *a priori* objective weightings, say, which would yield much less in terms of insights into the fitness landscape.

The population is split into subgroups specialized to examining fine-grained partitions in differing sections of the time series, and all possible partition sizes can be investigated. The algorithm uses biobjective tournament selection and a crossover method that in effect applies affine transformations where necessary to fit together elements of each parent's representation.

Testing was conducted using both simulated data and real stock market data. The simulated data was constructed using generation processes known to be self-affine and designed to have as clear a partitioning as possible in terms of the main metric used by the EA to assess differentiation of successive subseries. Initial results have indicated that the EA may be

able to come close to the partitioning used in the simulated data whilst simultaneously finding reasonable self-affinity, and indeed in limited testing was able to find partitions with better differentiation than the ones used to set up the test data. It was also observed that there is a clear trade-off between closeness to the original partition and the measured power of the differentiation between successive coarse-grained subseries, but that the overall solution set improved with successive generations. For the testing using real data, although it is not possible to comment directly on the accuracy of results as the “true” partitions are unknown, we were able to make several useful observations regarding the operation of the EA on the specific data set.

To progress further with the work, it is anticipated that it will be necessary to develop new techniques and extensions to the EA framework to test whether the partitionings produced are useful for example in minimizing variance of a portfolio of assets in out-of-sample testing when the current state of the asset market is unknown. It will then be possible to address the feasibility of probabilistically assessing the current state. It would also be useful to develop suitable metrics for quality of the PF, with attention to the particular nature of the problem. Although the motivation for this research comes from financial applications, it can be seen more widely as the first steps on the road to a broader framework for threat analysis, detection and mitigation, and it would be useful to extend testing to other types of data. For example, the work presented in this paper could be used to detect abnormal patterns temporarily present at different time scales in multivariate data of many types. Future work might allow optimization to mitigate the adverse impact of such patterns, and even detect the onset of such patterns in real time, allowing re-optimization.

ACKNOWLEDGEMENT

We are grateful for the financial support provided by the ESRC funded BLG-DRC project hosted by the University of Essex.

REFERENCES

- [1] Patrice Abry and Gustavo Didier. Wavelet estimation of operator fractional Brownian motions. *Bernoulli*. URL <http://www.e-publications.org/ims/submission/BEJ/user/submissionFile/21616?confirm=cffb2b8c>.
- [2] Torben G Andersen, Tim Bollerslev, Francis X Diebold, and Paul Labys. Realized Volatility and Correlation. 1999.
- [3] Torben G Andersen, Tim Bollerslev, Francis X Diebold, and Paul Labys. The Distribution of Realized Exchange Rate Volatility. *Journal of the American Statistical Association*, 96(453):42–55, 2001. ISSN 0162-1459. doi: 10.1198/016214501750332965. URL <http://www.nber.org/papers/w6961>.
- [4] Ole E. Barndorff-Nielsen and Neil Shephard. Estimating quadratic variation using realized variance. *Journal of Applied Econometrics*, 17(5):457–477, 2002. ISSN 08837252. doi: 10.1002/jae.691.

- [5] Ole E Barndorff-Nielsen and Neil Shephard. Econometric analysis of realised covariation: high frequency covariance, regression and correlation in financial economics. *Econometrica*, 72(3):885–925, 2004. ISSN 1556-5068. doi: 10.2139/ssrn.305583.
- [6] Jozef Barunik, Tomaso Aste, T. Di Matteo, and Ruipeng Liu. Understanding the source of multifractality in financial markets. *Physica A: Statistical Mechanics and its Applications*, 391(17):4234–4251, sep 2012. ISSN 03784371. doi: 10.1016/j.physa.2012.03.037. URL <http://linkinghub.elsevier.com/retrieve/pii/S0378437112002890>.
- [7] Barbara Bedowska-Sojka and Agata Kliber. Realized Volatility versus GARCH and Stochastic Volatility Models. The Evidence From The WIG20 Index and the EUR/PLN Foreign Exchange Market. *Przeglad Statystyczny*, 2010. URL http://keii.ue.wroc.pl/przeglad/Rok2010/Zeszyt4/2010_{ }57_{ }4{ }105-127.pdf.
- [8] Fischer Black. Studies in Stock Price Volatility Changes. In *Proceedings of the 1976 Business and Economic Studies Section*, pages 177–181. American Statistical Association, Alexandria, Va., 1976.
- [9] Tobias Bickle and Lothar Thiele. A Mathematical Analysis of Tournament Selection. *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 9–16, 1995. doi: 10.1.1.52.9907.
- [10] Tim Bollerslev. Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986. ISSN 03044076. doi: 10.1016/0304-4076(86)90063-1.
- [11] Laurent Calvet and Adlai Fisher. Forecasting multifractal volatility. *Journal of Econometrics*, 105(1):27–58, 2001. ISSN 03044076. doi: 10.1016/S0304-4076(01)00069-0.
- [12] Laurent E. Calvet and Adlai J. Fisher. How to Forecast Long-Run Volatility: Regime Switching and the Estimation of Multifractal Processes. *Journal of Financial Econometrics*, 2(1):49–83, 2004. ISSN 1479-8409. doi: 10.1093/jfec/nbh003. URL <http://jfec.oupjournals.org/cgi/doi/10.1093/jfec/nbh003>.
- [13] Laurent E. Calvet and Adlai J. Fisher. *Multifractal Volatility: Theory, Forecasting, and Pricing*. Academic Press, 2008. ISBN 0080559964. URL <http://books.google.com/books?id=ARuiiaog0IgC&pgis=1>.
- [14] Laurent E. Calvet, Adlai J. Fisher, and Samuel B. Thompson. Volatility comovement: A multifrequency approach. *Journal of Econometrics*, 131(1-2):179–215, 2006. ISSN 03044076. doi: 10.1016/j.jeconom.2005.01.008.
- [15] Gagari Chakrabarti and Chitrakalpa Sen. *Anatomy of Global Stock Market Crashes*. Number September 1837. Springer India, India, 2012. ISBN 978-81-322-0462-6. doi: 10.1007/978-81-322-0463-3. URL <http://link.springer.com/10.1007/978-81-322-0463-3>.
- [16] Fei Chen, Francis X. Diebold, and Frank Schorfheide. A Markov-switching multifractal inter-trade duration model, with application to US equities. *Journal of Econometrics*, 177(2):320–342, 2013. ISSN 03044076. doi: 10.1016/j.jeconom.2013.04.016. URL <http://dx.doi.org/10.1016/j.jeconom.2013.04.016>.
- [17] Carlos A Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, second edition, 2007. ISBN 9780387310299. URL <http://books.google.com/books?id=rXIuAMw3lGAC&pgis=1>.
- [18] Adriana S. Cordis and Chris Kirby. Discrete stochastic autoregressive volatility. *Journal of Banking and Finance*, 43(1):160–178, 2014. ISSN 03784266. doi: 10.1016/j.jbankfin.2014.03.020.
- [19] M.C. Cowgill, R.J. Harvey, and L.T. Watson. A genetic algorithm approach to cluster analysis. *Computers & Mathematics with Applications*, 37(7):99–108, 1999. ISSN 08981221. doi: 10.1016/S0898-1221(99)00090-5. URL <http://www.sciencedirect.com/science/article/pii/S0898122199000905>.
- [20] Anthony Davis, Alexander Marshak, Robert Cahalan, and Warren Wiscombe. The Landsat Scale Break in Stratocumulus as a Three-Dimensional Radiative Transfer Effect: Implications for Cloud Remote Sensing. *Journal of the Atmospheric Sciences*, 54(2):241–260, 1997. ISSN 0022-4928. doi: 10.1175/1520-0469(1997)054<0241:TLSBIS>2.0.CO;2.
- [21] Kalyanmoy Deb and Amrit Pratap. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. URL http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=996017.
- [22] Gustavo Didier and Vladas Pipiras. Integral representations and properties of operator fractional Brownian motions. *Bernoulli*, 17(1):1–33, 2011. ISSN 1350-7265. doi: 10.3150/10-BEJ259. URL <http://projecteuclid.org/euclid.bj/1297173831>.
- [23] Ralph Nelson Elliott. *Nature's Law*. Snowball Publishing, 2010.
- [24] Ralph Nelson Elliott. *The Wave Principle*. Snowball Publishing, 2012.
- [25] Robert F Engle. Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of UK Inflation. *Econometrica: Journal of the Econometric Society*, 50(4):987–1007, 1982.
- [26] Eugene F. Fama. Mandelbrot and the Stable Paretian Hypothesis. *The Journal of Business*, 36(4):420–429, 1963. ISSN 00219398; 15375374. URL <http://www.jstor.org/stable/2350971>.
- [27] P J Flynn, A K Jain, M N Murty, P J Flynn, Azriel Rosenfeld, K Bowyer, N Ahuja, and A K Jain. Data Clustering: A Review. 2000. ISSN 03600300. doi: 10.1145/331499.331504.
- [28] Rene Garcia and Pierre Perron. An analysis of the real interest rate under regime shifts. *The Review of Economics and Statistics*, 78(1):111–125, 1996. URL <http://www.jstor.org/stable/2109851>.
- [29] Weiyu Guo and Mark E. Wohar. Identifying Regime Changes in Market Volatility. *Journal of Financial Research*, 29(1):79–93, mar 2006. ISSN 0270-2592. doi: 10.1111/j.1475-6803.2006.00167.x. URL <http://doi.wiley.com/10.1111/j.1475-6803.2006.00167.x>.

- [30] Donatien Hainaut. A fractal version of the Hull-White interest rate model. *Economic Modelling*, 31(1989):1–15, 2010. ISSN 02649993. doi: 10.1016/j.econmod.2012.11.041. URL <http://dx.doi.org/10.1016/j.econmod.2012.11.041>.
- [31] James D. Hamilton. Rational-Expectations Econometric Analysis of Changes in Regime. *Journal of Economic Dynamics and Control*, 12(2/3):385–423, 1988. ISSN 0165-1889.
- [32] JD Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, 57(2):357–384, 1989. URL <http://www.jstor.org/stable/1912559>.
- [33] JD Hamilton. Analysis of time series subject to changes in regime. *Journal of Econometrics*, 45:39–70, 1990. URL <http://www.sciencedirect.com/science/article/pii/0304407690900939>.
- [34] Julia Handl and Joshua Knowles. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1):56–76, 2007. ISSN 1089778X. doi: 10.1109/TEVC.2006.877146.
- [35] Harold Edwin Hurst. Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers*, 116:770–808, 1951.
- [36] Julien Idier. (Re)correlation: a Markov switching multifractal model with time varying correlations. 2009. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1580075.
- [37] Julien Idier. short-term comovements in stock markets: the use of Markov-switching multifractal models. *The European Journal of Finance*, 17(1):27–48, 2011. ISSN 1351-847X. doi: 10.1080/13518470903448440.
- [38] Carla Inclan and GC Tiao. Use of cumulative sums of squares for retrospective detection of changes of variance. *Journal of the American Statistical Association*, 89(427):913–923, 1994. URL <http://www.tandfonline.com/doi/abs/10.1080/01621459.1994.10476824>.
- [39] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. *Proceedings 2001 IEEE International Conference on Data Mining*, pages 289–296, 2001. ISSN 15504786. doi: 10.1109/ICDM.2001.989531. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=989531>.
- [40] M. Laumanns, L. Thiele, and E. Zitzler. Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. *IEEE Transactions on Evolutionary Computation*, 8(2):170–182, 2004. ISSN 1089-778X. doi: 10.1109/TEVC.2004.823470.
- [41] T. Warren Liao. Clustering of time series data - a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.
- [42] Ruipeng Liu. *Multivariate Multifractal Models : Estimation of Parameters and Applications to Risk Management*. PhD thesis, 2008.
- [43] Ruipeng Liu and Thomas Lux. Higher Dimensional Multifractal Processes : A GMM Approach. In *ICCEF 2010 book of abstracts*. Society for Computational Economics, 2012.
- [44] Ruipeng Liu and Thomas Lux. Non-homogeneous volatility correlations in the bivariate multifractal model. *The European Journal of Finance*, (May 2015):1–21, 2014. ISSN 1351-847X. doi: 10.1080/1351847X.2014.897960. URL <http://www.tandfonline.com/doi/abs/10.1080/1351847X.2014.897960>.
- [45] Thomas Lux. The Markov-switching multifractal model of asset returns: GMM estimation and linear forecasting of volatility. *Journal of business & economic statistics*, 26(2):194–210, 2008. ISSN 07350015. doi: 10.1198/073500107000000403. URL <http://wrap.warwick.ac.uk/1749/>.
- [46] Thomas Lux. Flexible and Robust Modelling of Volatility Comovements : A Comparison of Two Multifractal Models. 2010.
- [47] Thomas Lux and Taisei Kaizoji. Forecasting volatility and volume in the Tokyo Stock Market: Long memory, fractality and regime switching. *Journal of Economic Dynamics and Control*, 31(6):1808–1843, 2007. ISSN 01651889. doi: 10.1016/j.jedc.2007.01.010.
- [48] B Mandelbrot. How long is the coast of Britain? Statistical self-similarity and fractional dimension. *Science*, 156(3775):636–638, 1967. ISSN 0036-8075. doi: 10.1126/science.156.3775.636.
- [49] Benoit Mandelbrot. The Variation of Certain Speculative Prices. *The Journal of Business*, 36(4):394–419, 1963. ISSN 00219398; 15375374. URL http://link.springer.com/chapter/10.1007/978-1-4757-2763-0_14.
- [50] Benoit Mandelbrot. The variation of some other speculative prices. *Journal of Business*, 40(4):393–413, 1967. URL <http://www.jstor.org/stable/2351623>.
- [51] Benoit Mandelbrot, Adlai Fisher, and Laurent Calvet. A Multifractal Model of Asset Returns. 1997. URL <http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=8752064&site=bsi-live&scope=site>.
- [52] Benoit B. Mandelbrot and John W. Van Ness. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, 10(4):422–437, 1968. ISSN 0036-1445. doi: 10.1137/1010093.
- [53] RN Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B*, 11:193–197, 1999. URL <http://link.springer.com/article/10.1007/s100510050929>.
- [54] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Genetic algorithm-based clustering technique. *Pattern Recognition*, 33:1455–1465, 2000. ISSN 00313203. doi: 10.1016/S0031-3203(99)00137-5.
- [55] Angelo Melino and Stuart M. Turnbull. Pricing Foreign Currency Options with Stochastic Volatility. *Journal of Econometrics*. Jul/Aug90, 45(1/2): 239–265. 27p. 7 Charts, 1990. ISSN 0304-4076. URL <http://www.sciencedirect.com/science/article/pii/0304407690901008>.
- [56] Lars Black Mogensen. *Models of Changing Volatility : A Multifractal Approach*. PhD thesis, Aarhus School of Business, 2011.
- [57] Raffaello Morales, T Di Matteo, and Tomaso Aste. Dependency structure and scaling properties of financial

- time series are related. *Scientific reports*, 4:4589, jan 2014. ISSN 2045-2322. doi: 10.1038/srep04589. URL <http://www.ncbi.nlm.nih.gov/articlerender.fcgi?artid=3980463&tool=pmcentrez&rendertype=abstract>.
- [58] Noemi Nava, T Di Matteo, and Tomaso Aste. Anomalous volatility scaling in high frequency financial data. 2015.
- [59] Jon D. Pelletier and Donald L. Turcotte. Self-affine time series: II. Applications and models. In *Advances in Geophysics Vol. 40*, pages 91–166. Academic Press, 1997. URL <http://arxiv.org/abs/physics/9705038>.
- [60] Lewis Fry Richardson. The problem of contiguity : An appendix to statistics of deadly quarrels. *General systems: Yearbook of the Society for the Advancement of General Systems Theory*, 6:139–187, 1961.
- [61] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, nov 1987. ISSN 03770427. doi: 10.1016/0377-0427(87)90125-7. URL <http://linkinghub.elsevier.com/retrieve/pii/0377042787901257>.
- [62] Martin Rypdal and Ola Løvsletten. Multifractal modeling of short-term interest rates. 2011. URL <http://arxiv.org/abs/1111.5265>.
- [63] A Sansó, V Aragó, and JL Carrion. Testing for changes in the unconditional variance of financial time series. *Revista de Economía Financiera*, 4:32–53, 2004. URL <http://www.aefin.es/AEFin{ }data/articulos/pdf/A4-2{ }443809.pdf>.
- [64] Won Min Song, T. Di Matteo, and Tomaso Aste. Hierarchical information clustering by means of topologically embedded graphs. *PLOS ONE*, 7(3):1–14, 2012. ISSN 19326203. doi: 10.1371/journal.pone.0031929.
- [65] N Srinivas and K Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994. URL <http://www.mitpressjournals.org/doi/abs/10.1162/evco.1994.2.3.221>.
- [66] Michele Tumminello, Fabrizio Lillo, and RN Mantegna. Correlation, hierarchies, and networks in financial markets. *Journal of Economic Behavior & Organization*, 75(1):40–58, 2010. URL <http://www.sciencedirect.com/science/article/pii/S0167268110000077>.
- [67] Victor Venema. p-model multifractal time series, 2006. URL <http://www2.meteo.uni-bonn.de/staff/venema/themes/surrogates/pmodel/>.
- [68] Darrell Whitley, Soraya Rana, and Robert B Heckendorn. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–47, 1999. doi: 10.1.1.36.7225. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.36.7225&rep=rep1&type=pdf>.
- [69] Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007. URL <http://ieeexplore.ieee.org/xpls/abs{ }all.jsp?arnumber=4358754>.
- [70] Jian Zhong and Xin Zhao. Modeling Complicated Behavior of Stock Prices Using Discrete Self-Excited Multifractal Process. *Systems Engineering Procedia*, 3 (2011):110–118, 2012. ISSN 22113819. doi: 10.1016/j.sepro.2011.11.015.



Christopher M. Taylor (M'16) received the

M.Sc. degree in financial economics from the University of Oxford, U.K. in 2006 and the M.Sc. degree in operational research from the University of Essex, U.K. in 2014. He is currently pursuing the Ph.D. degree in operational research from the University of Essex.

He has over 20 years' experience working as a finance professional in fund management and investment banking in Asia, U.K. and U.S.A.



Abdellah Salhi is Professor of Operational Research at the University of Essex, U.K. with interests

in the design, analysis and application of optimisation algorithms and heuristics. He was educated to degree level at the University of Constantine, in Algeria. He obtained his PhD from Aston University in Birmingham, UK. He acted as Research Fellow on a number of projects in OR and HPC at the universities of Leeds, Edinburgh and Southampton. He recently led projects such as Workforce Scheduling in Container Ports, the Beeswax project on Optimum Placement of Beehives for Optimum Pollination, and the Smart Town Centre. He has published over 70 articles in journals such as Annals of OR, Information Processing Letters and Applied Soft Computing. He has recently introduced the Plant Propagation Algorithm for global optimisation, a heuristic inspired by the way strawberry plants propagate.