

---

# Choosing the Appropriate Forecasting Model for Predictive Parameter Control

**Aldeida Aleti**

aldeida.aleti@monash.edu

Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia

**I. Moser**

imoser@swin.edu.au

Faculty of Information and Communication Technologies, Swinburne University of Technology, Hawthorn, VIC 3122, Australia

**Indika Meedeniya**

imeedeniya@swin.edu.au

Faculty of Information and Communication Technologies, Swinburne University of Technology, Hawthorn, VIC 3122, Australia

**Lars Grunske**

lars.grunske@informatik.uni-stuttgart.de

Institute of Software Technology, Stuttgart University, D-70569 Stuttgart, Germany

---

## Abstract

All commonly used stochastic optimisation algorithms have to be parameterised to perform effectively. Adaptive parameter control (APC) is an effective method used for this purpose. APC repeatedly adjusts parameter values during the optimisation process for optimal algorithm performance. The assignment of parameter values for a given iteration is based on previously measured performance. In recent research, time series prediction has been proposed as a method of projecting the probabilities to use for parameter value selection. In this work, we examine the suitability of a variety of prediction methods for the projection of future parameter performance based on previous data. All considered prediction methods have assumptions the time series data has to conform to for the prediction method to provide accurate projections.

Looking specifically at parameters of Evolutionary Algorithms (EA), we find that all standard EA parameters with the exception of population size conform largely to the assumptions made by the considered prediction methods. Evaluating the performance of these prediction methods, we find that linear regression provides the best results by a very small and statistically insignificant margin. Regardless of the prediction method, predictive parameter control outperforms state-of-the-art parameter control methods when the performance data adheres to the assumptions made by the prediction method. When a parameter's performance data does not adhere to the assumptions made by the forecasting method, the use of prediction does not have a notable adverse impact on the algorithm's performance.

## Keywords

Evolutionary Algorithms, adaptive parameter control, time-series analysis, forecasting

## 1 Introduction

Evolutionary Algorithms (EAs) have shown good performance and robustness in solving hard optimisation problems. In recent years, it has been acknowledged that the robustness of EAs can largely be attributed to the adjustability of the algorithm parameters [18, 40, 5], which enable the optimisation procedure to adapt to different search spaces. That said, it comes as no surprise that the parameterisation of EAs greatly affect the performance of the algorithm [40, 5, 37, 19, 2].

The parameterisation is usually the responsibility of the practitioner, who may not be an expert in EAs. Even after many years of research into EAs and other stochastic approaches, there are no straightforward parametrisation guidelines for interdisciplinary users of these methods [4]. General guidelines regarding the crossover rate recommend a divergent variety of values, such as 0.6 [14], 0.95 [25], or 'any value in the range [0.75, 0.95]' [43]. DeJong [14] recommends a mutation rate equal to 0.001, whereas Grefenstette [25] argues that the mutation rate should be less than 0.05.

Any one of these conflicting guidelines is the ideal configuration for a particular search space. As the search space of a newly arising problem is typically unknown, general principles about EA parameter configurations for universal applicability can hardly be formulated [41] and EA parameterisation presents itself as an optimisation problem in its own right [19]. Pressed for time, practitioners tend to choose parameter values based on few preliminary trials on the problem at hand. It has been demonstrated that different parameter settings are required also for different optimisation stages of the same problem instance [5, 9, 45]. The ‘optimal’ values for the mutation rate of an Evolutionary Algorithm were studied using a set of benchmark problems [9]. It became clear that these values depend on the state of the search space and any fixed mutation rate will produce suboptimal results. Time dependency has since been reported for several EA parameters - the mutation rate [13, 6, 1, 3], the crossover rate [13, 1, 3] and the population size [5, 37].

As prior tuning of an EA’s parameter values does not provide optimal performance, a great number of researchers have since proposed to re-set the parameter values of an EA repeatedly during the optimisation process [5, 16, 21, 15, 39], an approach known as parameter control. The intuitive motivation comes from the way the optimisation process unfolds from a more diffused global search, requiring parameter values responsible for the exploration of the search space, to a more focused local optimisation process, requiring parameter values which help with the convergence of the algorithm.

One of the most conspicuous ways of varying parameter values during the optimisation process is adaptive parameter control [46, 22, 11, 27, 29], in which properties of an EA optimisation process (typically the quality of the solutions produced) are monitored and the change in the properties is used as a signal to change the parameter values. The area of adaptive parameter control has been researched more actively in recent times [26, 22, 46, 1]. Parameter values are assessed based on past performance and subsequently adapted for the next iteration of the algorithm.

In essence, adaptive parameter control methods assess the potential future success of parameter values based on the estimated effect of the parameter values in the last iterations [28, 30, 21]. Quality attribution mechanisms use this quality measure gained from recent performance (time  $t$ ) to decide which value(s) to use in the immediate future. It is generally argued that the past effectiveness of parameter values provides the information necessary to project their chance of leading to good results at iteration  $t + 1$ . In previous work [1], we have suggested that using time series prediction to attribute a quality metric to the parameter values to choose from for the next iteration is beneficial to the performance of the algorithm. The predictive parameter control method used linear regression to forecast the quality of mutation and crossover rates, and outperformed best-performing approaches with the same functionality AP [46], PM [46] and DMAB [22].

However, time series prediction methods require the data series to have a number of statistical properties, such as linearity of the relationship between dependent and independent variables, normality of the error distribution, independence of the errors, homoscedasticity, and stationarity of the mean and standard deviation. Different methods have been made available to accommodate different types of data. Typical EA parameters whose adaptation is meaningful to automate using predictive parameter control are mutation rate, crossover rate, mutation and crossover operators, population and mating pool sizes. In the current work, we investigate the statistical properties of the quality of these parameter values recorded over time. The analysis of the statistical properties is employed to examine the suitability of a number of forecasting techniques for adaptive parameter value optimisation and provide recommendations on the adequacy of these methods for forecasting the quality to be expected from the use of certain parameter values. To demonstrate the benefits of predictive parameterisation, we present a comparison with state-of-the-art quality attribution strategies - Average Quality Attribution (AQA) and Extreme Quality Attribution (EQA).

## 2 Adaptive Parameter Control for Evolutionary Algorithms

Adaptive parameter control describes the application of separate meta-optimisation methods which use feedback from the optimisation process to evaluate the effect of parameter value choices and adjust the parameter values over the iterations. The steps of adaptive parameter control as applied to an EA are depicted in Figure 1.

The optimisation process of EAs begins with a set of initial solutions, which is evolved in itera-

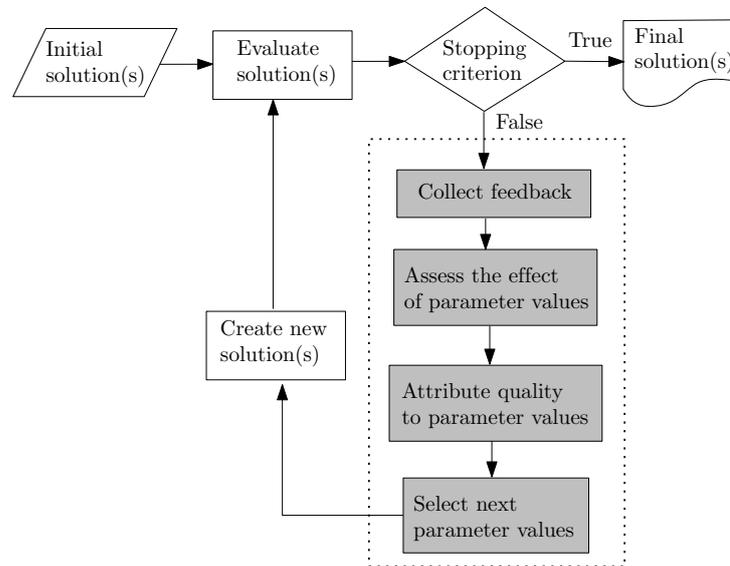


Figure 1: Steps of adaptive parameter control in Evolutionary Algorithms

tions until a stopping criterion is reached. The new generations of solutions are created with the help of crossover, mutation and selection operators. These operators, their probabilities, the size of the population and the number of new solutions produced every iteration, are examples of parameters that can be adjusted by a parameter control method. Every iteration, the solutions created are evaluated using the objective function(s) of the problem at hand. The results from this fitness evaluation process can be used by the adaptive parameter control methods to assess the effect of the parameter values on the performance of the optimisation algorithm. This feedback collection as a part of the parameter control method records the change in the properties of the population - usually fitness, occasionally also diversity - as the performance of the algorithm.

The feedback from the optimisation process is used by the parameter effect assessment strategy to estimate the effect the parameter values on the change in solution properties. The approximated effect of the parameter values is used to estimate a quality value which is attributed to the parameter value to project the potential of these values in the next iteration(s). These quality attributions provide the probability distributions used to select the parameter values to be employed to create the next generation of solutions. The selection mechanism subsequently configures the components of an Evolutionary Algorithm (variation, selection and replacement operators).

State-of-the-art adaptive parameter control methods can be classified into two main categories: (i) methods that use an Average Quality Attribution (AQA) strategy [30, 46], where the quality of parameter values is estimated as the average of the qualities measured in the past iterations, or (ii) Extreme Quality Attribution strategy (EQA) [46, 21], in which the best quality measured so far is used to estimate the overall quality of parameter values.

Usually, adaptive parameter control methods use a predefined time window  $w$  to approximate the quality of parameter values for the following iteration. Some methods use this time window to estimate the quality of parameter values as the average effect on algorithm performance [30]. Probability Matching (PM) technique [46] estimates the quality of a parameter value based on a running average of past rewards. Rewards are allocated on the basis of the outcome of the optimisation process the parameter value was used in. A minimum probability is enforced on values which do not receive rewards in order to maintain a non-zero probability. The motivation of a minimum value is the assumption that parameter values which do not perform well at present might be optimal in the future. PM has been criticised for the fact that the probability values resulting from the reward allocations poorly reflect the relative differences in algorithm performance when using the values. Values with vastly superior performance may only be differentiated by a marginal increase of the probability of being chosen in the next step.

Adaptive Pursuit (AP) [46] was conceived with the goal of improving the performance of PM by ensuring an appropriate difference in probabilities depending on experienced performance. After an iteration of the optimisation process, AP establishes the respective rewards for the parameter values used, but only applies the maximum award to the value of the best-performing algorithm instance. All other values have their probabilities of future use diminished. A nonzero probability is enforced as a minimum probability.

Fialho et al. [21, 20, 22] employ the maximum effect in  $w$  iterations to approximate the quality of parameter values in the preceding iteration  $t - 1$ . A normalisation scheme is used, which divides parameter effects by the best effect achieved so far by any parameter value. The Dynamic Multi-Armed Bandit (DMAB) [22] also employs a performance-based rewards approach, but here the probability of re-using a certain value is based on a tradeoff between the number of times the parameter value was used and the reward gained from its previous performance. Rather than using a rewards-based weighted adjustment of the probabilities, the DMAB completely recalculates the probabilities when a change in the rewards distribution is detected. DMAB uses the Page-Hinkley test to detect a change in the rewards distribution.

The most common way of attributing quality to the parameters is by using the improvement in specific properties of the generated solutions with respect to a reference solution  $s_{ref}$ . This reference point can be either the parents of the generated solutions [48, 30, 28, 29, 22], the best solution so far [12, 17], or a specified quantile, e.g. the median [32, 33]. For example, if a mutation rate 0.01 is applied to create a solution  $s'$  from its parent  $s$ , the effect of the mutation rate on the performance of the algorithm is equal to the fitness gain  $\Delta f = f(s') - f(s)$ .

Contemporary parameter control methods [11, 30, 21, 47] attribute the quality derived from recent performance directly to the parameter values as a basis for the choice to make in the next iteration (time  $t + 1$ ). One might argue that these approaches are 'one step behind', in that they represent the parameter value which is optimal for the previous iteration (time  $t$ ). Ideally, we would use a setting optimised for the beginning iteration (time  $t + 1$ ). It follows that an ideal parameter control method would attempt to predict successful parameter values for time  $t$  based on previous performance.

### 3 Predictive Quality Attribution

Given a set  $v_i, i = (1, \dots, n)$  of algorithm parameters with values  $v_{ij}, j = (1, \dots, m)$ , where  $m$  can be a discrete value or an interval of continuous values, the adaptive method has the task of deriving the optimal next value  $v_{ij}$  to optimise the influence of  $v_i$  on the performance of the algorithm. As an example, when the mutation rate  $v_1$  is dynamically adjusted using 4 intervals ( $m = 4$ ),  $v_{12}$  stands for the second interval. In the discrete case of optimising the type of mutation operator  $v_1$ ,  $v_{12}$  stands for the second operator type. The probability of choosing values from a particular interval depends on the previous performance of values sampled from this interval. Values from intervals which have not been used have a small initial probability of being chosen. When a value is used but does not produce any above-average outcome, its probability of being chosen reverts to the initial value.

To explore the parameter space,  $k$  independent algorithm instances are started in parallel. Each optimisation process optimises the same problem instance. The processes set their adaptive parameters for an iteration at a time, choosing the respective value probabilistically from the distribution obtained from the performance-based prediction. At the end of an iteration, the average solution fitness among all instances is established. Each value interval keeps a history list of success rates from all iterations. Algorithm instances which produce above-average population fitness gain are considered successful. For each parameter value or interval, the number of times the value has been used  $n(v_{ij})$  and the number of times the value led to a success  $n^s(v_{ij})$  (above-average increase in population fitness) are established. The effect of each parameter value is calculated using Equation 1.

$$e_t(v_{ij}) = \frac{n^s(v_{ij})}{n(v_{ij})} \tag{1}$$

Predictive Quality Attribution (PQA) applies forecasting techniques to predict the quality of parameter values to be applied in the parameter control strategy. The effects  $e$  of parameter intervals/types

$v_{ij}$  are recorded for each iteration, resulting in the time series  $e_1(v_{ij}), e_2(v_{ij}), \dots, e_t(v_{ij})$ , where  $t$  is the current iteration. The time series is used to forecast the effect of the parameter intervals/types for the next iterations. The purpose of predictive modelling is to estimate the expected effect of a parameter interval/type in the future iterations. A predictive model in adaptive parameter control has the general form:

$$q_t(v_{ij}) = f(e_1(v_{ij}), e_2(v_{ij}), \dots, e_t(v_{ij}), \epsilon) \quad (2)$$

where  $q_t(v_{ij})$  is the response,  $(e_1(v_{ij}), e_2(v_{ij}), \dots, e_t(v_{ij}))$  is a vector of predictors and  $\epsilon$  is a random error. The different predictive models vary in the type of the response, which can be discrete or continuous, and the function  $f$ . In this paper we consider predictive models with continuous response, where the response is the expected quality of parameter values/intervals. The probability of sampling from a range  $p(v_{ij})$  is the combined quality of its members  $q_t(v_{ij})$ , normalised by the summed qualities of all intervals.

Forecasting necessitates the use of analytical methods to identify the relationship between future (predicted) effects of parameter values and past effects. The forecasting techniques explored in this work are the most prominent methods available for time series prediction, ranging from simple models such as Linear Regression (LR), Simple Moving Average (SMA), and Exponentially-Weighted Moving Average (EWMA) to more sophisticated formulations such as the Autoregressive Integrated Moving Average (ARIMA).

LR fits a linear equation to the observed data, hence it makes the assumption that the data points have a linear relationship with time. If this is not the case, LR cannot be expected to provide accurate predictions and other alternatives have to be explored. The SMA model measures trends by smoothing the data using the mean of the data values. As a result, the trend in the data is extrapolated into a forecast. The assumption is that prior trend movements will continue. EWMA, on the other hand, focuses mostly on recent effects and captures shorter trends in the data. As a result, an EWMA is expected to reduce any lags in the simple moving average such that the fitting line is closer to the real values of the effects compared to a simple moving average. EWMA is vulnerable to sudden changes in the data, hence easily affected by noise. If this problem arises, it can be counteracted by increasing the length of the moving average term or using SMA instead. The ARIMA model is a combination of moving average models and a special case of the linear regression model. The ARIMA model is considered more powerful than LR, but also complex.

Using the most sophisticated model available does not guarantee better results, as all models make assumptions about the data used for projection. For an optimal outcome, the parameter performance data must conform to the assumptions made by the forecasting method. It is necessary to investigate the data with regards to the assumptions to make recommendations which forecasting methods are likely to produce well-parameterised algorithms. To this end, the analysis of the properties of the data is complemented with the performance of the algorithm when parameterised using the respective forecasting methods.

### 3.1 Linear Regression

Linear regression builds a linear mathematical model of past observations that can be used to forecast future values of a discrete time-series. Regression models are based on the notion of *regression to the mean*, i.e. if a time step is  $x$  standard deviations from the mean, the model predicts that the next value will be  $r \cdot x$  standard deviations from the mean of the time series, where  $r$  is the correlation coefficient between time and parameter effects. This is an indication of the extent to which the linear model can be used to predict the deviation of the effect of a parameter value  $e(v_{ij})$  from the mean when the deviation of time  $t$  from the mean is known.

The correlation coefficient measures the degree of linear relatedness between the data and time on a scale of  $-1$  to  $+1$ . The correlation coefficient is positive if both variables vary on the same side of the respective means, negative if they vary on opposite sides. If the effect varies independently of the number of iterations, the correlation coefficient is equal to zero. The linear model for predicting the quality of a parameter value  $q_t(v_{ij})$  from  $t$  is equal to:

$$\frac{q_t(v_{ij}) - \bar{e}_t(v_{ij})}{\sigma(e_t(v_{ij}))} = r \cdot \frac{t - \bar{t}}{\sigma(t)} \quad (3)$$

where  $t$  is the total number of observations,  $\bar{e}_t(v_{ij})$  is the mean of the effects in the last  $t$  iterations,  $\bar{t}$  is the mean of the time-steps and  $\sigma(e_t(v_{ij}))$  is its standard deviation at time  $t$ . The correlation coefficient  $r$  is equal to the average of the product of the standardized effect  $\mathcal{E}_{t-1}(v_{ij})$  and time  $\tau_t$  calculated as follows:

$$r = \overline{\mathcal{E}(v_{ij}) \cdot \tau} = \frac{\tau_1 \mathcal{E}_1(v_{ij}) + \tau_2 \mathcal{E}_2(v_{ij}) + \dots + \tau_t \mathcal{E}_t(v_{ij})}{t} \quad (4)$$

The current parameter effect and time are standardised by expressing them in units of standard deviations from their own mean. The best fit of the model is calculated by minimising the mean squared error.

### 3.2 Simple Moving Average

Due to the stochastic nature of approximate optimisation methods, the performance of the algorithm can display random variation with time. Therefore, a smoothing method could be useful to cancel, or at least reduce, the effect due to random variation. Averaging and exponential smoothing models, such as the SMA model, reduce the noise in the data. A simple moving average model uses equal weights for all the data points in the time series as follows:

$$q_t(v_{ij}) = \frac{e_{t-1}(v_{ij}) + \dots + e_{t-k}(v_{ij})}{k} \quad (5)$$

where  $k$  is the ‘width’ of the moving average and denotes the number of observations used to compute the average. The forecast  $q_t(v_{ij})$  is equal to the average of the last  $k$  data points in the time series, centred at  $t - \frac{k+1}{2}$ ; the forecast will thus be delayed by  $\frac{k+1}{2}$  time steps. The selection of the window width  $k$  depends on the type of prediction of interest, which can be short, intermediate, or long term. Short-term averages respond quickly to changes in the effect of the parameter values, while long-term averages are slow to react.

### 3.3 Exponentially Weighted Moving Average

In the LR and SMA models, past observations are weighted equally. EWMA gives exponentially decreasing weights to past observations, i.e. feedback taken from the current iteration is given more importance in forecasting the next parameter values than the earlier observations. The forecast of the future effect of parameter values is derived from the weighted sum of all observations. Mathematically, EWMA is the convolution of the data points with an exponentially smoothing weighting function defined as:

$$q_t(v_{ij}) = \alpha e_{t-1}(v_{ij}) + \alpha(1 - \alpha)e_{t-2}(v_{ij}) + \dots + \alpha(1 - \alpha)^k e_{t-(k+1)}(v_{ij}) \quad (6)$$

where  $\alpha$  is the smoothing factor in the range (0,1). It can be observed that the multiplicands of the data series decrease exponentially, approaching zero without reaching it. The smoothing factor  $\alpha$  controls the closeness of the interpolated value to the most recent observation. Large values of  $\alpha$  reduce the level of smoothing. To determine the value of  $\alpha$  we use the least-squares method, for which the mean of squared errors is minimised.

### 3.4 Autoregressive Integrated Moving Average

The use of ARIMA models is appropriate when the time series data shows a stable trend with time and does not have many outliers. ARIMA can perform better than exponential smoothing techniques given a sufficient number of observations and a stable correlation between past observations. The ARIMA model is composed of three parts, which are calculated individually: the Integrated (I) part, the Autoregressive part (AR) and the Moving Average part (MA).

The integrated part  $I(d)$  is used if the data shows evidence of non-stationarity, i.e. the time series is not stable with time and exhibits a trend, and  $d$  is the number of non-seasonal differences. The stationarity of the time series is measured using the stationarity in mean and variance. The first step in the identification of the appropriate ARIMA model is the identification of a value for  $d$ , i.e. the order(s) of differencing needed to make the time series stationary. The process of differencing subtracts each observation at time  $t$  from the observation at time  $t - 1$ .

The  $AR(p)$  model is used to describe the correlation between the current value of the time series and  $p$  past values. The order  $p$  of the  $AR$  model describes the number of lags (past values) included in the model prediction, i.e. the number of the autoregressive terms. In adaptive parameter control,  $AR(1)$  describes the correlation between the prediction of the effect  $q_t(v_{ij})$  at time  $t$  and the immediately preceding value in the time series, equal to  $q_t(v_{ij}) = \phi_1 e_{t-1}(v_{ij}) + \epsilon_t$ , where  $\phi_1$  is the first-lag (lag 1) autoregressive parameter, and  $\epsilon_t$  the error.

The Moving Average ( $MA(q)$ ) part of the ARIMA model considers the time series as an unevenly weighted moving average of random errors  $\epsilon_t$ , capturing the deviation of the current effect as a finite weighted sum of  $q$  previous deviations of the effects, plus a random error. In the first order  $MA(1)$ , the error of the current value is directly correlated with the error of the immediate past observation, equal to  $q_t(v_{ij}) = \epsilon_t - \theta_1 \epsilon_{t-1}$ , where  $\epsilon_t, \epsilon_{t-1}$  are the errors at times  $t$  and  $t - 1$ , and  $\theta_1$  is the first order moving average coefficient. Both the AR model and the MA model can be extended to a higher order model in order to include more information than that of the immediate past observation.

#### 4 Statistical Properties of Time Series Data

The field of statistics defines a number of properties which a data series has to have to be modelled by time series prediction methods. Different methods have been made available to accommodate different types of data. The relevant statistical properties are

- linearity of the relationship between dependent and independent variables,
- normality of the error (residual) distribution,
- independence of the errors (residuals),
- homoscedasticity, which defines the variance of the errors as constant with time and
- stationarity of the mean and standard deviation.

Linear regression requires the first four properties; SMA and EWMA assume only the last two. It is generally accepted that ARIMA produces accurate predictions only after a threshold of 50 data points. In addition, ARIMA requires the data to meet all but the first assumption.

##### 4.1 Testing for the Statistical Properties

For each of the five properties we perform statistical tests which reveal to what degree the property exists in the given data series.

###### 4.1.1 Linearity

Linearity can be determined visually by plotting the data. A more reliable method is to check for a statistical significance of the linearity by fitting a linear model and reporting the  $p$ -value which indicates the significance of the regression. A  $p$ -value  $< 0.05$  indicates that the coefficient belonging to the linear term in the model is non-zero, which implies that the time series is linear with all data points having equal influence on the fitted line. The hypothesis is formulated as follows:

- $H_0$ : The correlation coefficient is equal to zero.
- $H_1$ : The correlation coefficient is not zero.

###### 4.1.2 Normality

The calculation of the confidence intervals for the model coefficients in linear regression is based on the assumptions of normally distributed errors. In terms of predictive quality attribution, normality of errors implies that most of the effects of the parameter values are expected to fall close to the regression line and very few to fall far from the line. Since the estimation of model coefficients is based on the minimisation of the squared error, the presence of extreme data points can exert a disproportionate influence on the estimates. To check for the normality assumption, we use the Kolmogorov-Smirnov

(KS) test [24] which tests the null hypothesis that the errors of the effects are normally distributed. The null hypothesis is rejected if the  $p$  – value is smaller than 0.05, which indicates that the data cannot be considered normally distributed. The hypothesis is formulated as follows:

- $H_0$ : The data follows a normal distribution
- $H_1$ : The data does not follow a normal distribution

#### 4.1.3 Independence

Linear regression models make the assumption that the error terms are independent. One example of the violation of this assumption is autocorrelation, which occurs if each error term in the time series is related to its immediate predecessor. The Durbin-Watson (DW) statistical test measures the correlation between the error terms and their immediate predecessors. The  $D$  statistic tests the null hypothesis that the errors from a least-squares regression are not autocorrelated. The values of  $D$  vary in the interval  $[0, 4]$ . For independent error terms, the value of  $D$  is expected to be close to 2. The value of  $D$  tends to be smaller than 2 for positive autocorrelation between the terms, and greater than 2 for negative autocorrelation.

- $H_0$ : The autocorrelation of errors is zero.
  - $H_1$ : The autocorrelation of errors is less than zero.
- The null hypothesis is rejected if  $p$  – value is less than 0.05.

#### 4.1.4 Homoscedasticity

Homoscedasticity refers to the assumption that the effects of parameter values exhibit a similar variance with time. If homoscedasticity is violated, it is difficult to measure the true standard deviation of the residuals, which can result in confidence intervals that are too wide or too narrow. In particular, if the variance of the errors increases with time, confidence intervals for predictions will tend to be unrealistically narrow. Heteroscedasticity may also have the effect of placing too much weight on a small subset of the data, i.e. the subset where the error variance is the largest when estimating coefficients.

Residuals are tested for homoscedasticity using the Breusch-Pagan test, which regresses square residuals to independent variables. The null hypothesis for the Breusch-Pagan test is homoscedasticity. The alternative hypothesis is that the error variance varies with a set of regressors.

- $H_0$ : The data exhibit a similar variance with time.
- $H_1$ : The error variance varies with a set of regressors.

If the  $p$  – value is less than 0.05, the null hypothesis is rejected, which means that the data is not homoscedastic.

#### 4.1.5 Stationarity

A time-series is stationary if the mean and standard deviation remain the same with time. Formally, the effect of parameter values are said to be stationary if the distribution of the time-series measured during the time segment  $[1, t]$  is the same as the distribution of the effects shifted by some lag  $k$ , i.e. in the time segment  $[1 + k, t + k]$ .

In the application of the predictive quality attribution, this means that it does not matter when the effects are observed, since the distribution of the parameter effects does not depend on time. In the case of the mutation rate, the assumption is that the 'diffuse' part of the search with large values is followed by a more focussed part, which favours lower mutation rate values. Hence a decreasing trend in the time series is expected.

To check for the stationarity of the effects of parameter values we use the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) [36] test, which tests the null hypothesis that the data is stationary, formulated as follows:

- $H_0$ : The data is stationary.
- $H_1$ : The data is not stationary.

If the  $p$  – value is less than 0.05, the null hypothesis is rejected, which means that the data is not stationary.

## 5 Design of Experiments

We are interested in investigating statistical properties of the time-series data obtained from the effects of parameter values over time. Typical EA parameters whose adaptation is meaningful to automate using parameter control are the mutation rate, crossover rate, the mutation and crossover operators, the population and mating pool sizes. More specialised EA implementations may have additional parameters. In this work we limit the investigation to the parameters EAs most commonly use.

The real-valued parameters mutation rate, crossover rate, population size and mating pool size are discretised into intervals. The quality attribution targets the intervals. The selection strategy samples the new parameter values from the intervals chosen according to the quality attribution mechanism. The definition of the interval bounds is likely to affect the performance of the algorithm and potentially the properties of the performance data to analyse. Therefore, we perform the experiments twice: with a total of two range intervals over the whole ‘meaningful’ parameter range and a second time with intervals that divide the complete ‘meaningful’ range into four intervals. All intervals are evenly spread within the range.

It is obvious that the practitioner is still left with the task of defining the range bounds for ‘meaningful’ values of tunable parameters. For most well-known parameters, bounds are given in the literature. When practitioners customise algorithms and in the process define new parameters, the definition of range bounds is an unavoidable task even when using adaptive parameter control methods. Inexperienced practitioners best avoid this dilemma by adhering to well-known EAs which permit the application of known bounds as listed in Table 1.

Table 1: Ranges/values of parameters analysed.

Parameter	Ranges/values
Mutation rate - 2 ranges	[0.001,0.249], [0.25,0.49]
Mutation rate - 4 ranges	[0.001,0.1249], [0.125,0.249], [0.25,0.3749], [0.375,0.49]
Crossover rate - 2 ranges	[0.6,0.79], [0.8,0.99]
Crossover rate - 4 ranges	[0.6,0.69], [0.7,0.79], [0.8,0.89], [0.9,0.99]
Population size - 2 ranges	[20,60], [61,100]
Population size - 4 ranges	[20,40], [41,60], [61,80], [81,100]
Mating pool size - 2 ranges	[0.1,0.39], [0.4,0.69]
Mating pool size - 4 ranges	[0.1,0.249], [0.25,0.39], [0.4,0.549], [0.55,0.69]
Mutation operator	Single-point, Uniform
Crossover operator	Single-point, Uniform

### 5.1 Benchmark Problems

The current work on parameterising EAs is necessarily experimental in nature. A set of complex benchmark problems ensures the results are representative of typical combinatorial optimisation processes in which EAs are often used. The Quadratic Assignment Problem (QAP) is a well-known and very complex combinatorial optimisation problem. In the current work, we use the QAP in its single- and multiobjective formulations. Another difficult and equally well-known problem tailored to the optimisation behaviour of EAs is the Royal Road (RR) function.

#### 5.1.1 Quadratic Assignment Problem

In the Quadratic Assignment Problem (QAP) [35],  $n$  facilities have to be allocated to  $n$  locations, such that the total cost is minimised and every location has only one assigned facility. The total cost is calculated as the flow between the facilities multiplied by the costs for placing the facilities at their respective locations. More formally, the problem is modelled with two  $n \times n$ , representing the cost and the flow. The aim is to assign  $n$  utilities to  $n$  locations with minimal cost. The candidate assignments are evaluated according to the following cost function:

$$\sum_{i,j,k,l}^n D_{ij} * C_{kl} * u_{ik} * u_{jl} \tag{7}$$

where

- $n$  is the number of facilities and locations,
- $D_{ij}$  is the distance between location  $i$  and location  $j$ ,
- $C_{kl}$  is the cost of the flow between neighbouring utilities  $k$  and  $l$ ,
- $u_{ik}$  is 1 if utility  $i$  is assigned to location  $k$ , 0 otherwise.

QAP does not allow multiple assignments to the same location, hence solutions are subject to the following constraints:

$$\sum_{j=1}^n u_{ij} = 1, \quad i = \{1, 2, \dots, n\} \tag{8}$$

$$\sum_{i=1}^n u_{ij} = 1, \quad j = \{1, 2, \dots, n\} \tag{9}$$

$$u_{ij} \in \{0, 1\}, \quad i, j = \{1, 2, \dots, n\} \tag{10}$$

**Singleobjective Formulation** For the purpose of this experiment a well-known instance (TAI30A) from the QAPlib [8] was used as a benchmark for the optimisation process whose parameter values are adaptively set. The scales of the fitness values vary considerably over different QAP instances. For ease of comparison, we normalise the fitness values according to Equation 11.

$$f_{norm}(s) = \frac{f_{max} - f(s)}{f_{max} - f_{min}} \tag{11}$$

where  $f_{max}$  is the maximum value and  $f_{min}$  is the minimum value that the fitness function  $f$  can take. Since QAP is a minimisation problem, the normalisation of the results converts it into a maximisation problem.

**Multiobjective Formulation** The multiobjective model of QAP (mQAP) concerns itself with the flow of more than one type of item. A practical application may be a hospital layout problem, where the goal is to optimise the distances doctors, visitors and patients have to travel each day. The cost of assigning all facilities to locations is evaluated according to Equation 7 for each of the objectives. The task is to optimise all  $m$  objectives according to Equation 12.

$$\text{minimise : } \vec{C} = [C_1, C_2, \dots, C_m] \tag{12}$$

The instances used for the current work have been published by Knowles and Corne [34]. File names ending in ‘rl’ denote instances designed to be ‘real-like’ with rugged fitness landscapes.

### 5.1.2 The Royal Road Problem

The Royal Road functions are parametrized landscapes, first introduced by Mitchell, Forrest, and Holland [31]. They are especially devised to demonstrate that there exist problems which are easier to solve using a Genetic Algorithm than a hill climbing algorithm.

The function of the form  $F : \{0, 1\}^l \rightarrow \mathbb{R}$  is used to define a search task in which one wants to locate strings that produce high fitness values. The string is composed of  $2^k$  non-overlapping contiguous sections each of length  $b + g$ , where  $b$  is known as the block and  $g$  is known as the gap. These sections form the first-level schemata, composed of short, low-order, highly fit schemata or building blocks. A schema  $s_i$  is a combination of ones, zeros and the special character ‘\*’ which is interpreted as ‘do not care’. Lower-level schemata are used to form intermediate-level higher-fitness schemata, which in turn are combined to create even higher-fitness schemata.

The fitness calculation of a string  $x$  proceeds in two steps, the part calculation and the bonus calculation. The overall fitness assigned to that string is the sum of these two calculations. The *part calculation*

considers each block individually and assigns a fitness according to the number of 1's. The block fitnesses are later summed to form the total fitness of the part calculation. The fitness function  $RRP(x)$  for the part calculation of a string  $x$  is defined as:

$$RRP(x) = \sum_i c_i \delta_i(x) \tag{13}$$

where

$$\delta_i(x) = \begin{cases} 1 & \text{if } x \in s_i \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

The aim of the *bonus calculation* is to reward complete blocks and some combinations of complete blocks. Holland's approach assigns rewards for attaining a certain level. At the lowest level, rewards are given for complete blocks. If such a block exists, it receives a fitness equal to  $u^*$ . Any additional complete blocks receive a fitness of  $u$ .

### 5.2 Data Collection

The data series we wish to analyse is the performance feedback data for the typical EA parameters identified. The data gathered is the success rate (Equation 1) of the particular parameter or interval. The success rate is calculated as a ratio of the times the value or interval was used *successfully* to the number of times the value or interval was used. Successful usage is defined as the optimisation outcome producing above-average fitness values, where the average fitness is calculated across all new population members produced in the iteration under scrutiny.

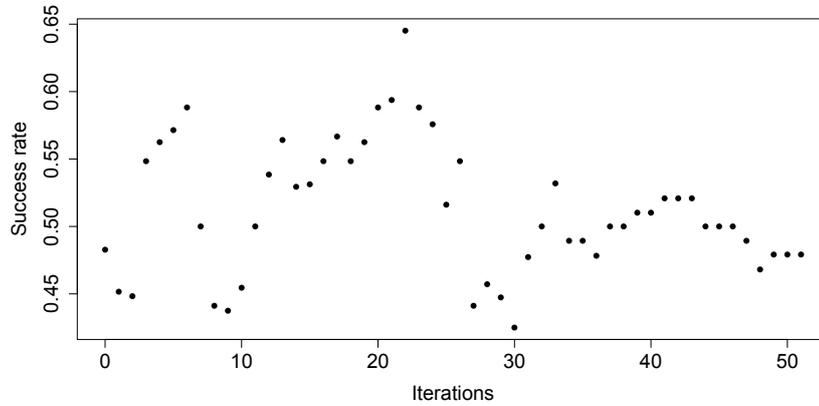
To investigate the characteristics of the data, we ran as many parallel trials as we had discrete parameter values/intervals to investigate. In each trial, one of the parameters was the design factor and all others were set to fixed values which were identical in all parallel trials. It is clear that this procedure cannot eliminate possible interactions between certain parameter values. However, we can expect to reduce these interactions to a possible minimum. Every instance was granted 15000 function evaluations. Every 150 function evaluations, the effect of the parameter intervals was recorded. In total, each experiment produces 100 data points for each parameter range. Approximate algorithms are not expected to deliver exact and repeatable results. Therefore, for the current data analysis, all trials were repeated 30 times for each parameter value.

An example of two time series recording the success rate of mutation rate in the ranges [0.001,0.249] and (0.25,0.49] are shown in Figure 2. The scatter plots of the data (with time on the x-axis) are useful for establishing trends in the time series. It can be observed that both time series are non-linear. In fact, neither of them seems to exhibit any particular trend. Furthermore, the value- or interval-specific time series data from the optimisation process are tested for linearity, normality, independence, homoscedasticity, and stationarity using the statistical tests described in the previous sections. The results from the statistical tests that evaluate the linearity, normality of the residuals (KS test), independence of the residuals (DW test), homoscedasticity of the time series (BP test), and stationarity of the data (KPSS test) are shown in Table 2.

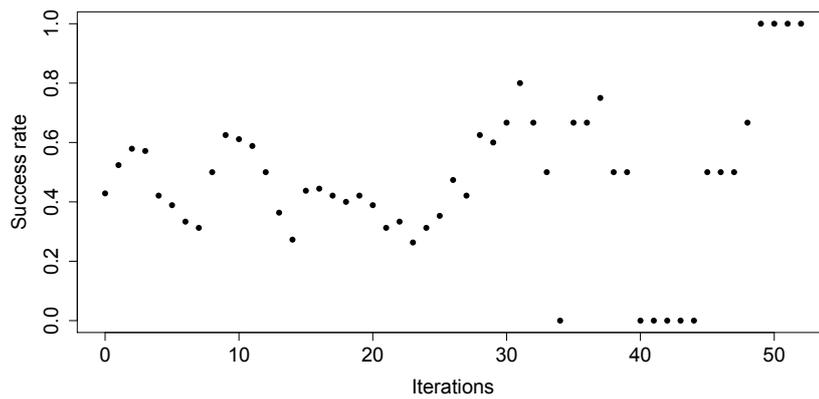
Table 2:  $p$  – value of statistical tests.

Parameter Range	Linearity test	KS test	DW test	BP test	KPSS test
Mutation rate [0.001,0.249]	0.1435	0.9684	1	< 0.01	0.04
Mutation rate (0.25,0.49]	0.2366	0.4445	1	< 0.01	0.1

The  $p$  – value from fitting a linear model to the data helps decide whether there is a significant relationship between the variables in the linear regression model of the time series at a 0.05 significance level. The  $p$  – values are greater than 0.05 for both ranges of the mutation rate, hence the null hypothesis is not rejected and we can say that there is no significant relationship between the variables of the regression model. It can be concluded that the time series in these two examples are not linear.



(a) Mutation rate values in the range (0.001,0.249).



(b) Mutation rate values in the range (0.25,0.49).

Figure 2: Scatter-plot of the time series recording the success rate of mutation rate.

The KS test is employed to test the assumption of normally distributed errors. The  $p$  – values for both time series are greater than 0.05, which means that the null hypothesis stating that the data follows a normal distribution is not rejected. To test for the independence of error terms we record the  $p$  – value from the DW test, which checks if the errors from a least-squares regression are autocorrelated. The null hypothesis stating that the autocorrelation of errors is zero is rejected if the  $p$  – value is less than 0.05. The  $p$  – values from the DW test for the two time series are not less than 0.05, hence the null hypothesis is not rejected. This indicates that the error terms are not autocorrelated. Residuals are tested for homoscedasticity using the Breusch-Pagan test, with a null hypothesis formulated as the data exhibit a similar variance with time, which is rejected with a  $p$  – value less than 0.05 for both time-series. Finally, the stationarity of the time series is analysed using the KPSS test. The  $p$  – value from testing the time series of the mutation rate in the range [0.001, 0.249) is less than 0.05, which indicates that this time-series is not stationary. On the contrary, the time series of the mutation rate in the range (0.25, 0.49] is stationary, with a  $p$  – value equal to 0.1.

## 6 Evaluation of Statistical Properties

Table 3 summarises the results from the statistical tests when the continuous values have been divided into two ranges, Table 4 lists the same outcomes in the case where four intervals were used. The discrete parameters are listed in Table 5. The percentages represent the runs among the 30 trials in which  $H_0$  was not rejected. We use these results to derive conclusions regarding the appropriateness of using forecasting techniques to predict the effects of the parameter values.

Table 3: Characteristics of the effects of the continuous parameter values when the ranges are divided into two intervals. The percentages represent the percentage of times (of a total 30)  $H_0$  was not rejected.

Parameter	Problem	Linear	Normal	Independent	Homoscedastic	Stationary
Mutation rate [0.0010, 0.25]	QAP	97%	100%	80%	90%	97%
	MQAP	97%	100%	93%	93%	93%
	Royal Road	93%	90%	87%	80%	80%
Mutation rate [0.2505, 0.5]	QAP	100%	100%	70%	97%	100%
	MQAP	100%	100%	73%	93%	100%
	Royal Road	93%	93%	93%	87%	100%
Crossover rate [0.6, 0.799]	QAP	100%	100%	83%	90%	93%
	MQAP	100%	100%	93%	90%	90%
	Royal Road	93%	93%	93%	87%	87%
Crossover rate [0.8, 1.0]	QAP	100%	100%	83%	97%	93%
	MQAP	100%	100%	83%	90%	90%
	Royal Road	100%	97%	80%	87%	90%
Population size [20, 60]	QAP	100%	100%	100%	53%	13%
	MQAP	100%	100%	90%	50%	13%
	Royal Road	90%	97%	80%	43%	10%
Population size [61, 100]	QAP	33%	73%	100%	77%	13%
	MQAP	33%	80%	83%	73%	17%
	Royal Road	30%	97%	80%	87%	13%
Mating pool [0.1, 0.4]	QAP	100%	100%	77%	93%	97%
	MQAP	100%	100%	80%	93%	97%
	Royal Road	100%	97%	73%	87%	90%
Mating pool [0.41, 0.7]	QAP	100%	100%	80%	87%	100%
	MQAP	100%	100%	80%	90%	100%
	Royal Road	100%	97%	80%	87%	100%

In general, the time series data is linear with normally distributed errors. The lowest number of normally distributed residuals resulted from the experiment with the population size in the interval [61,100], in which the null hypothesis was rejected in 27% of the cases (c.f. Table 3). The time series belonging to the population values in this range is also the only time series that was not linear in the majority of the cases (the null hypothesis was rejected in 70% of the runs).

The Durbin-Watson (DW) test used to verify the assumption of no correlation between the error terms of the time series shows that performance data of parameter values are independent for the majority of the cases, with the lowest percentage scored by the mutation rate in the interval [0.2505,0.5] (70% of the runs have independent errors).

The Breusch-Pagan (BP) test was used to measure the homoscedasticity of the time series data. In general, homoscedasticity holds for the majority of the time series data, although the null hypothesis can only be confirmed in 80-90% of the runs for all parameter intervals. The lowest percentages of homoscedastic runs are in the experiments with the ranges [20,40] and [41,60] of the population size, in which only 40% of the runs are homoscedastic. Similar results were observed in the other ranges of the population size.

The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test verifies the null hypothesis that the data is stationary. It can be observed that stationarity is a common feature in the effects of the single-point mutation, uniform mutation, single-point crossover and uniform crossover operators. Similarly, in some of the intervals of the mutation rate ([0.0010,0.25] and [0.2505, 0.5]), crossover rate ([0.6, 0.799], [0.8, 1.0]

Table 4: Characteristics of the effects of continuous parameter values with four ranges. The percentages represent the percentage of times (of a total 30)  $H_0$  was not rejected.

Parameter	Problem	Linear	Normal	Independent	Homoscedastic	Stationary
Mutation rate [0.001, 0.124]	QAP	97%	97%	97%	90%	80%
	MQAP	97%	97%	97%	90%	80%
	Royal Road	90%	90%	87%	80%	77%
Mutation rate [0.125, 0.249]	QAP	100%	100%	100%	80%	63%
	MQAP	97%	93%	97%	83%	63%
	Royal Road	100%	97%	97%	80%	60%
Mutation rate [0.25, 0.3749]	QAP	100%	100%	97%	90%	87%
	MQAP	100%	97%	97%	90%	80%
	Royal Road	97%	100%	100%	93%	83%
Mutation rate [0.375, 0.49]	QAP	100%	90%	97%	87%	80%
	MQAP	100%	90%	93%	87%	83%
	Royal Road	97%	77%	93%	83%	80%
Crossover rate [0.6, 0.69]	QAP	100%	100%	93%	83%	80%
	MQAP	97%	100%	97%	87%	83%
	Royal Road	100%	93%	90%	83%	83%
Crossover rate [0.7, 0.79]	QAP	100%	100%	97%	90%	73%
	MQAP	100%	100%	97%	93%	73%
	Royal Road	100%	97%	93%	80%	70%
Crossover rate [0.8, 0.89]	QAP	100%	100%	97%	87%	80%
	MQAP	100%	100%	93%	90%	83%
	Royal Road	100%	90%	93%	80%	77%
Crossover rate [0.9, 0.99]	QAP	100%	100%	100%	83%	93%
	MQAP	100%	97%	97%	90%	90%
	Royal Road	97%	90%	97%	77%	83%
Population size [20, 40]	QAP	100%	97%	100%	40%	10%
	MQAP	100%	100%	97%	50%	13%
	Royal Road	97%	80%	97%	47%	10%
Population size [41, 60]	QAP	100%	100%	100%	40%	3%
	MQAP	97%	97%	100%	53%	7%
	Royal Road	97%	100%	97%	40%	10%
Population size [61, 80]	QAP	100%	100%	100%	57%	17%
	MQAP	100%	97%	97%	50%	13%
	Royal Road	93%	93%	97%	53%	10%
Population size [81, 100]	QAP	100%	100%	100%	53%	47%
	MQAP	100%	97%	100%	47%	50%
	Royal Road	97%	100%	90%	50%	43%
Mating pool [0.1, 0.249]	QAP	100%	100%	90%	80%	63%
	MQAP	100%	97%	93%	83%	70%
	Royal Road	97%	93%	93%	80%	70%
Mating pool [0.25, 0.39]	QAP	100%	100%	97%	83%	97%
	MQAP	100%	97%	97%	90%	90%
	Royal Road	93%	93%	97%	80%	90%
Mating pool [0.4, 0.549]	QAP	100%	100%	100%	80%	70%
	MQAP	100%	97%	100%	83%	83%
	Royal Road	97%	100%	97%	87%	83%
Mating pool [0.55, 0.69]	QAP	100%	100%	97%	80%	77%
	MQAP	100%	97%	97%	80%	77%
	Royal Road	90%	93%	93%	83%	73%

and [0.9, 0.99]) and mating pool size ([0.1, 0.4], [0.41, 0.7] and [0.25, 0.39]) the null hypothesis was not rejected in more than 90% of the runs with the mating pool size in the interval [0.41, 0.7] and mutation rate in the range [0.2505, 0.5] being the most stationary processes (100% of the runs are stationary).

However, the crossover rate in the interval [0.7,0.79], the mutation rate in the interval [0.125,0.249]

Table 5: Characteristics of the effects of the discrete parameter values. The percentages represent the percentage of times (of a total 30)  $H_0$  was not rejected.

Parameter	Problem	Linear	Normal	Independent	Homoscedastic	Stationary
Single-point mutation	QAP	100%	100%	70%	93%	97%
	MQAP	100%	97%	70%	83%	83%
	Royal Road	90%	90%	73%	87%	87%
Uniform mutation	QAP	100%	100%	77%	93%	93%
	MQAP	100%	97%	80%	83%	90%
	Royal Road	97%	87%	73%	87%	90%
Single-point crossover	QAP	100%	100%	77%	87%	93%
	MQAP	100%	97%	80%	83%	90%
	Royal Road	97%	93%	73%	87%	87%
Uniform crossover	QAP	100%	100%	77%	97%	93%
	MQAP	100%	97%	77%	83%	93%
	Royal Road	100%	90%	73%	87%	83%

and the mating pool size in the intervals [0.1,0.249], [0.4,0.549] and [0.55,0.69] have a moderately low number of trials which confirm stationarity. Furthermore, less than 20% of the runs in all ranges of the population size confirm stationarity. The time series belonging to the population size in the interval [41,60] has the lowest percentage of confirmation of the null hypothesis.

For the assessment of the suitability of forecasting methods LR, SMA, EWMA and ARIMA based on statistical properties it is necessary to reach a conclusion regarding the parameters themselves instead of the individual value options. For the comparison, the assumptions made by the considered forecasting models have been summarised in Table 6.

Table 6: Assumptions of the forecasting models.

	Linearity	Normality	Independence	Homoscedasticity	Stationarity
LR	+	+	+	+	-
SMA	-	-	-	+	+
EWMA	-	-	-	+	+
ARIMA	-	+	+	+	+

Fortunately, the characteristics of parameter value effects are reasonably consistent over the values or intervals of the same parameter. We also find that dividing the parameter value ranges into two versus four intervals causes no significant differences in the behaviour of the statistical properties.

The results flag stationarity as the least satisfied condition, where the null hypothesis can be confirmed for some intervals in only around 70% of the trials. Excluding population size for its problematic outcomes on many assumptions, most parameters' interval-specific results are mixed regarding stationarity. This would suggest that the best choice for all of these is LR, alternatively ARIMA with an appropriate differencing method. The condition of independence of errors seems threatened only when larger intervals are used (Table 3). Larger intervals seem to complicate the attribution of causality of parameter effects. If we assume smaller intervals, independence of errors seems unproblematic.

Homoscedasticity has low values only in the case of the intervals of the mating pool sizes. A decision has to be made whether a homoscedastic result in around 80% of the cases is sufficient to regard the mutation pool size effect data as homoscedastic. Since homoscedasticity is a prerequisite for the use of all considered models, we can only accept these low scores and investigate whether the use of the forecasting methods on the mating pool size leads to good overall results.

The parameter of population size scores poorly on both the test for homoscedasticity and stationarity. As all suggested forecasting models require homoscedasticity, we can only decide that none of these methods can be expected to provide accurate performance predictions on population size values. Perhaps the least unsuitable method for this parameter is LR for the reason that LR does not require stationarity.

Table 7: Recommendations on the suitability of forecasting methods for parameters.

Parameter	LR	SMA	EWMA	ARIMA
Mutation rate	+	+	+	+
Crossover rate	+	+	+	+
Population size	~	~	~	~
Mating pool size	+	+	+	+
Mutation operator	+	+	+	+
Crossover operator	+	+	+	+

Consequently, the resulting summary Table 7, which shows the suitability of forecasting methods for parameters, displays a ‘~’ sign against population size and all discussed forecasting methods. This indicates that the time series data of this parameter only partially follows the assumptions made by the forecasting model, which cannot be recommended for use without reserve. All other parameters have been marked as ‘+’ for all forecasting methods considered. This decision has been made notwithstanding the limitations discussed above.

## 7 Performance Comparison of the Prediction Methods

The parameters mutation rate, crossover rate, mating pool size, mutation and crossover operators are all similar in their adherence to the expectations made by the considered prediction models. From the outcome of the conformity tests we cannot definitively conclude which prediction model is clearly best suited for each of them. Therefore, in this section, we apply all considered forecasting methods to each of the parameters except the population size, which conforms too poorly to several of the assumptions.

To avoid over-fitting of the forecasting models, we employ regularization techniques in which a part of the data is reserved to test whether the model generalises well. We randomly split the data into training and testing set, and fit the model to the training set by minimising the squared error. The model is then evaluated in the testing set, by measuring the squared error and comparing it to the training set.

### 7.1 Experimental Settings

In the first instance, mutation rate, crossover rate, mating pool size, crossover operator, and mutation operator are set adaptively using predictive parameter control. The population size is not discussed in this section due to its problematic statistical properties. The parameter values or intervals used for these trials are listed in Table 1.

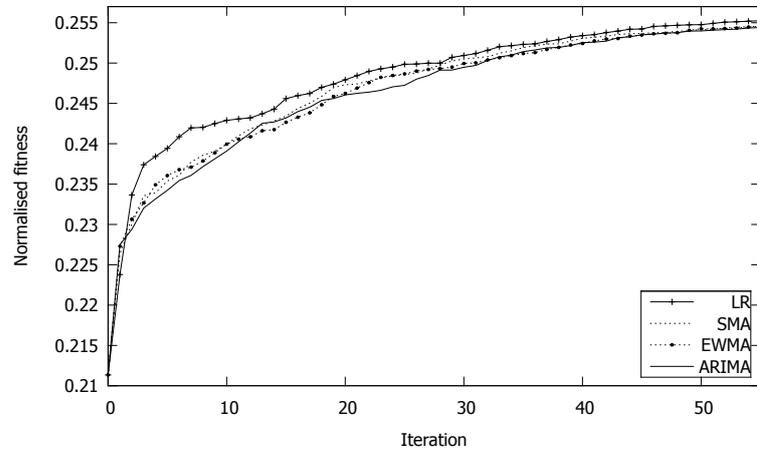
We performed 30 runs for each optimisation scheme with different quality attribution strategies. The parameter value effect is measured based on the fitness improvement over the previous iteration. In the case of a multiobjective setting we use the hypervolume indicator, which has been identified as the best unary measure to reflect the fitness of a population in a multiobjective setting [49]. The results are presented as the mean of 30 runs over 55 iteration. The problem instance used for QAP is BUR26E, for mQAP we used KC30-3fl-2rl. The Bur\* problems’ first matrices are average typists’ typing times whereas the second are the frequencies of combinations of two letters. The multiobjective problem KC30-3fl-2rl has 30 facilities/locations, 3 objectives and is the 2nd ‘real-like’ instance of this size created by Knowles [34].

### 7.2 Results

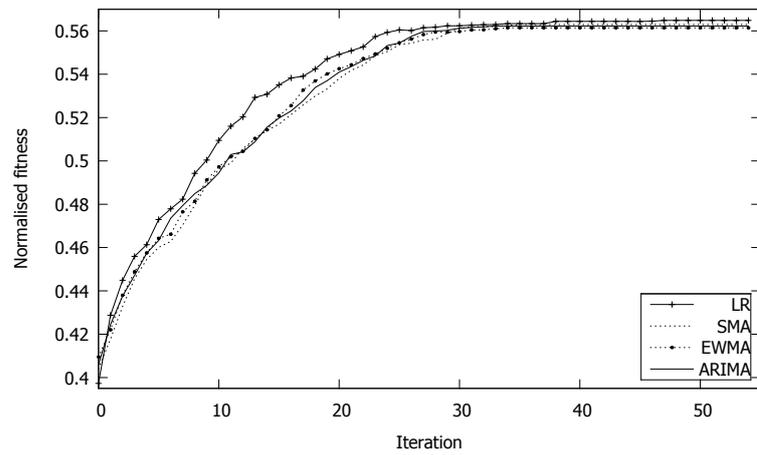
The development of the population fitness averaged over 30 trials, plotted in Figure 3, shows that using LR for the parameter optimisation helps find better solutions at an earlier stage of the optimisation process. In the case of the QAP problems, single- and multiobjective, the quality achieved by LR is not attained by the other methods. The performances of all predictive models ‘converge’ to the same quality on the RR problem. The optimisation scheme using ARIMA as a quality attribution scheme does not perform very well during the initial iterations, as could be expected given the requirement of a minimum of 50 data points.

On the other hand, certain ranges of mutation rate, crossover rate and mating pool size meet the stationarity assumption, which is a prerequisite of SMA, EWMA and ARIMA (see Table 7), only in a

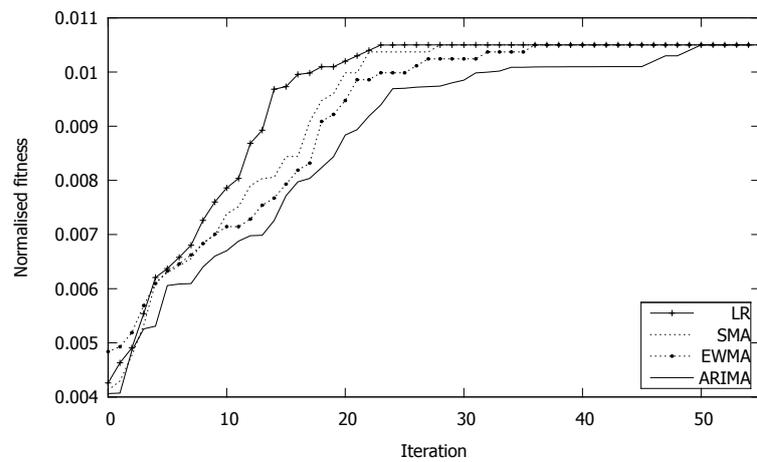
### Choosing the Appropriate Forecasting Model for Predictive Parameter Control



(a) BUR26E



(b) KC30-3fl-2rl



(c) Royal Road

Figure 3: Performance development of the EA with different forecasting techniques.

moderate number of runs. As a result, these models may not fit perfectly the time series data we are controlling. This explains the weaker performance of SMA, EWMA and ARIMA models when compared with LR. As the differences in performance between the models are not statistically significant, the results suggest that LR can be regarded as representative of all considered prediction models.

## 8 Experimental Evaluation

The experimental validation of predictive parameter control using the considered techniques is carried out on all parameters regardless of adherence to statistical properties as established in a previous section. Practitioners will not, in general, have the leisure to perform tests on statistical properties of parameters. We are therefore obliged to provide recommendations how to proceed when assumptions are not met or the adherence to statistical assumptions is unknown. We provide a validation for all parameters using all forecasting techniques to establish the potential of adverse effects on the performance of the EA.

Having investigated the appropriate forecasting method to use for PQA, we investigate if using prediction as part of adaptive parameter control is beneficial to the result quality. The PQA method using LR is compared to the average and extreme quality attribution schemes, which are currently the best-performing quality attribution strategies.

### 8.1 Benchmark Quality Attribution Strategies

The Predictive Quality Attribution strategy is compared to two conspicuous methods, an Average Quality Attribution (AQA) and an Extreme Quality Attribution (EQA) strategy. AQA is a relatively simple algorithm, which uses a time window  $W$  to approximate the quality of parameter values as the average effect on algorithm performance. The Average Quality Attribution (AQA) is given in Equation 15.

$$q_{t-1}(v_{ij}) = \frac{1}{N_w^{v_{ij}}(S)} \sum_{s=t-w-1}^{t-1} e_s(v_{ij}) \quad (15)$$

where  $e_s(v_{ij})$  is the effect of parameter value  $v_{ij}$ , assessed as the fitness difference between the solution produced by applying parameter value  $v_{ij}$  and its parent. The sum of the effect values of  $v_{ij}$  in the last  $w$  iterations (i.e.  $\sum_{s=t-w-1}^{t-1} e_s(v_{ij})$ ) is divided by the total number of solutions created using that parameter value ( $N_w^{v_{ij}}(S)$ ) normalising the quality of  $v_{ij}$ .

The credit attribution strategy shown in Equation 15 is susceptible to the scale of the fitness function. The Extreme Quality Attribution (EQA) strategy, proposed by Fialho et al. [23], known as the Area Under the Curve (AUC) addresses this problem by using a ranked-based strategy. The method ranks the effects of parameter values in the recent  $w$  applications in a descending order. Each parameter effect is assigned a rank-value  $w - r$ , where  $r$  refers to the rank-position of the effect value. The higher the rank-position, the lower is the rank-value. The final rank-value for each parameter effect is calculated as:

$$e_r(v_{ij}) = D^r(w - r) \quad (16)$$

where  $D \in [0, 1]$  is the decay factor, which emphasizes the influence of the top-ranked effect values on the overall quality of that parameter value, hence it belongs to the class of Extreme Quality Attribution Strategies. Next, the area under the curve is calculated by going down the sorted list, and drawing the Receiving Operator Curve starting from the origin [23]. Since AUC uses only the ranks of the parameter effects, this method is quite robust with respect to hyper-parameter and fitness transformations, as shown in the experimental evaluation by Fialho et al. [23].

### 8.2 Experimental Settings

For the benefit of the comparison between PQA, AQA and EQA, we control the mutation rate, the crossover rate, the mutation operator, the crossover operator, the population size and the mating pool size with values/ranges depicted in Table 1.

The experimental study in Section 4 demonstrated that the effect of the population size during the optimisation time-steps does not adhere to the assumptions made by the forecasting techniques. Hence

we conduct a separate set of experiments for the population size and another for the mutation rate, the crossover rate, the mutation operator, the crossover operator, and the mating pool size. The latter set of experiments is performed to examine the performance of PQA when used to predict the quality of parameter values which conform to the assumptions of the forecasting techniques. The population size is investigated for an insight into the problem whether the use of PQA can have adverse effects on the performance of the algorithm when the assumptions of the forecasting models are clearly not met.

Approximate algorithms are not expected to deliver exact and repeatable results, but to provide good approximate solutions where exact approaches cannot be devised. Hence, results concerning the performance of approximate algorithms such as EAs, are usually reported as mean values over repeated trials. To obtain a fair comparison, the generally accepted approach is to allow the same number of function evaluations for each trial. Every trial instance is granted 90000 function evaluations. To obtain a fair comparison among the different quality attribution schemes we repeat each run 30 times. To check for a statistical difference of the results, the different effect attribution schemes of the optimisation methods are validated using the Kolmogorov-Smirnov (KS) non-parametric test [42]. The final solutions of each optimisation scheme are recorded and the results are compared by using the normalised fitness for the singleobjective problems and the hypervolume indicator for the multiobjective problem.

### 8.3 Hyper-Parameters

All adaptive algorithms involve hyper-parameters, which need to be tuned depending on the optimisation problem at hand. This defines another optimisation problem, which can become quite computationally expensive if we attempt an exhaustive exploration of the search space. However, the number of hyper-parameters is usually lower than the number of algorithm parameters that should be controlled.

The tuning of the hyper-parameter values was performed using a Sequential Parameter Optimisation (SPO) [7], which tests each hyper-parameter combination using several runs. To decrease the number of tests required, we employ a racing technique, which uses a variable number of runs depending on the performance of the hyper-parameter configuration. Hyper-parameter configurations are tested against the best configuration so far, using at least the same number of function evaluations as employed for the best configuration. The results from the tuning process are shown in Table 8.

Table 8: Hyper-parameters of the three credit attribution strategies.

Benchmark	Hyperparameter	Value	Description
EQA	$\mathcal{D}$	0.5	decay factor
PQA, EQA	$p_{min}$	0.1	minimum selection probability
EQA, AQA, PQA	$W$	100	window size

The only hyper-parameter for the PQA method is the minimum probability used for the exploration of new parameter values. For the minimum selection probability, we employ the same settings as AP and PM, as shown in Table 8.

### 8.4 Benchmark Problems

The optimisation schemes are employed to optimise the Royal Road Problem, described in Section 5.1.2, four well-known instances of the Quadratic Assignment Problem (CHR20A, BUR26E, TAI30A, and STE36B) and three instances of the multiobjective Quadratic Assignment Problem (KC10-2fl-5rl, KC30-3fl-1rl, and KC30-3fl-2rl). CHR20a is known to be among the hardest of the CHR\* problems [10]. The BUR\* problems' first matrices are average typists' typing times whereas the second are the frequencies of combinations of two letters. TAI\*a problems are considered to have 'rugged' fitness landscapes, whereas TAI\*b instances are structured [38]. STE36b is a variation of a computer component placement problem with the goal to minimise the lengths of the connecting wires [44]. The multiobjective problem KC10-2fl-5rl has 10 facilities/locations, 2 objectives and is the 5th 'real-like' instance of this size created by Knowles [34]. By analogy, KC30-3fl-1rl and KC30-3fl-2rl have 30 facilities and 3 objectives.

### 8.5 Results

Figure 4 shows the results of the 30 trials in which the mutation rate, crossover rate, mating pool size, mutation operator and crossover operator were controlled. The experimental results are clearly not normally distributed, but the mean and 25th percentile of the PQA method are consistently above the respective values of the benchmarks.

The means and standard deviations are listed in Table 9, which shows a significant difference between the result groups of AQA, EQA and PQA. The mean performance of PQA is consistently above the averages of the benchmarks. The performance of EQA is clearly worse than that of the other two quality attribution strategies.

Table 9: The means and standard deviations over 30 runs of each problem instance controlling all EA parameters except population size using different quality attribution schemes: Average Quality Attribution (AQA), Extreme Quality Attribution (EQA) and Predictive Quality Attribution (PQA).

Problem	Mean			Standard Deviation		
	AQA	EQA	PQA	AQA	EQA	PQA
CHR20A	0.4909	0.4845	<b>0.4972</b>	1.627E-02	<b>1.306E-02</b>	1.920E-02
BUR26E	0.2416	0.2395	<b>0.2422</b>	2.049E-03	<b>1.893E-03</b>	2.365E-03
TAI30A	0.4909	0.4845	<b>0.4982</b>	1.627E-02	<b>1.306E-02</b>	1.870E-02
STE36B	0.7978	0.7984	<b>0.8308</b>	<b>7.823E-03</b>	9.038E-03	8.704E-03
Royal Road	0.0064	0.0063	<b>0.0072</b>	1.537E-04	6.3723E-03	<b>1.473E-03</b>
KC10-2fl-5rl	0.8062	0.8061	<b>0.8067</b>	1.040E-03	8.284E-04	<b>9.555E-04</b>
KC30-3fl-1rl	0.1984	0.1961	<b>0.1995</b>	<b>3.255E-03</b>	3.974E-03	3.806E-03
KC30-3fl-2rl	0.5199	0.5175	<b>0.5223</b>	7.312E-03	5.439E-03	<b>4.821E-03</b>

Table 10: The Kolmogorov-Smirnov test values for the 30 runs of each problem instance controlling all EA parameters except population size using different quality attribution schemes: Average Quality Attribution (AQA), Extreme Quality Attribution (EQA) and Predictive Quality Attribution (PQA).

Problem	PQA vs. AQA		PQA vs. EQA	
	d	p	d	p
CHR20A	0.2333	0.042	0.3333	0.045
BUR26E	0.3000	0.048	0.3000	0.049
TAI30A	0.3294	0.046	0.3517	0.039
STE36B	0.9667	0.000	0.9667	0.000
Royal Road	0.6333	0.000	0.7333	0.000
KC10-2fl-5rl	0.3333	0.045	0.4000	0.011
KC30-3fl-1rl	0.2897	0.016	0.4333	0.005
KC30-3fl-2rl	0.3433	0.035	0.4333	0.005

The gap between result qualities widens in favour of PQA as the problem difficulty increases. The smallest instances of the QAP (CHR20A) and mQAP (KC10-2fl-5rl) can be assumed to be the least challenging, and there the results are not as clearly in favour of PQA. The Royal Road problem and the larger instances of QAP (STE36B) and mQAP (KC30-3fl-1rl, KC10-2fl-5rl) are evidently solved to better quality using PQA, as they are more complex than the smaller problem instances.

To check for a statistical significance of the results we used the Kolmogorov-Smirnov (KS) non-parametric test [42]. PQA was compared with AQA and EQA with a null hypothesis of no difference between the performances. The 30 values of the hypervolume indicators of the repeated trials for each

## Choosing the Appropriate Forecasting Model for Predictive Parameter Control

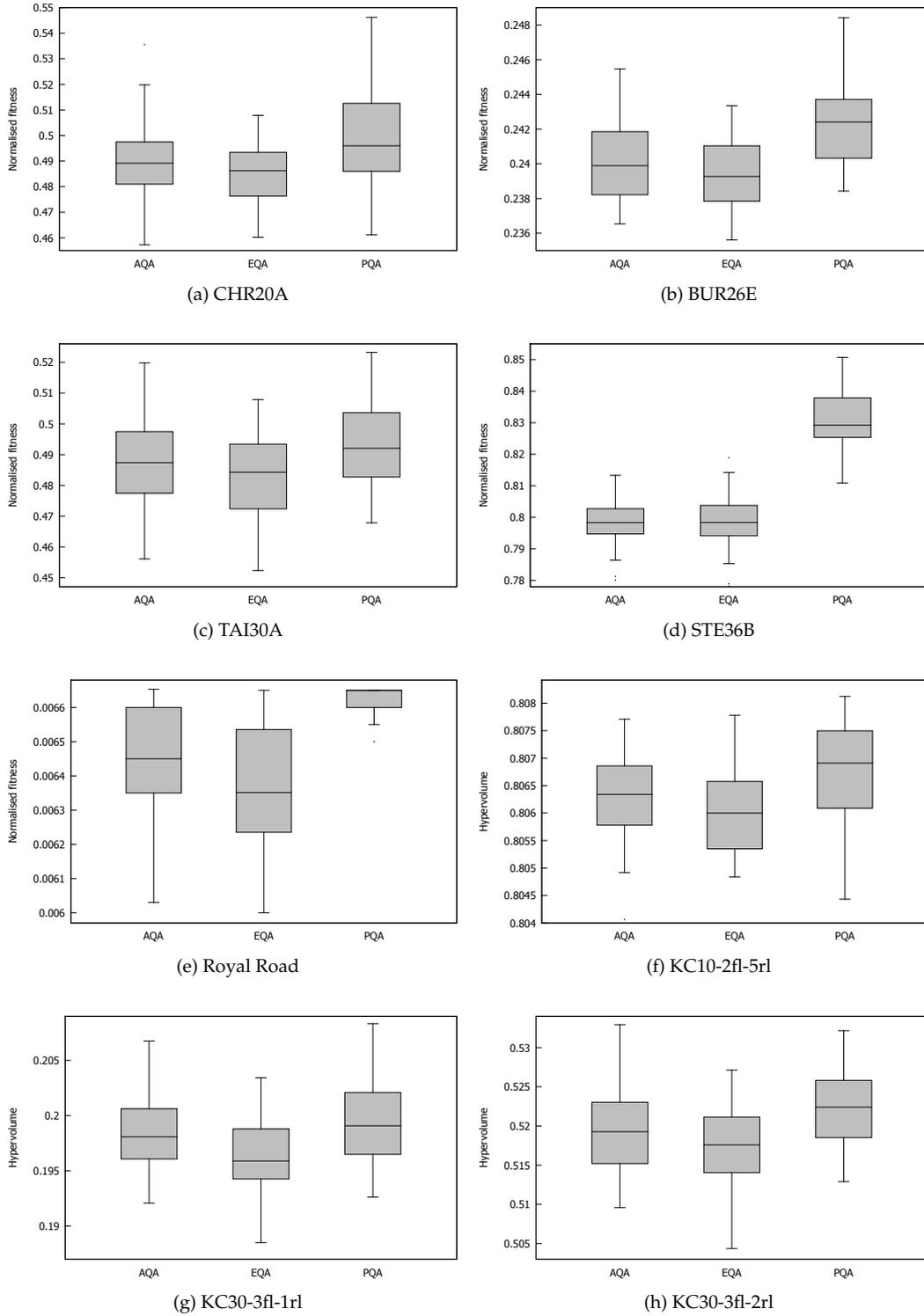


Figure 4: Boxplots of the 30 trials which control all parameters except population size. PQA is compared to the benchmark quality attribution schemes AQA and EQA.

problem instance were submitted to the KS analysis [42] and results are shown in Table 10. All KS tests, used for establishing differences between independent datasets under the assumption that they are not normally distributed, result in a rejection of the null hypothesis with a minimum  $d$ -value of 0.2333 at a 95% confidence level. Hence we conclude that the superior performance of the PQA method is statistically significant.

Given the outcomes of the tests for statistical properties, the population size was controlled individually. Figure 5 depicts the results of the 30 runs for each problem instance and quality attribution scheme. PQA is slightly outperformed by the EQA strategy in some of the problems. These results indicate that using forecasting techniques to predict the quality of the population choices for the future iterations may not always be beneficial.

Table 11: The means and standard deviations for the 30 runs of each problem instance using different quality attribution schemes when controlling the population size: Average Quality Attribution (AQA), Extreme Quality Attribution (EQA) and Predictive Quality Attribution (PQA).

Problem	Mean			Standard Deviation		
	AQA	EQA	PQA	AQA	EQA	PQA
CHR20A	0.4660	<b>0.4669</b>	0.4651	4.85E-03	3.51E-03	4.26E-03
BUR26E	0.2460	<b>0.2461</b>	0.2459	8.92E-05	6.88E-05	1.31E-04
TAI30A	0.4977	<b>0.4982</b>	0.4962	3.57E-03	3.23E-03	5.94E-03
STE36B	0.7934	<b>0.8164</b>	0.7945	3.29E-03	4.22E-02	3.64E-03
Royal Road	6.56E-03	<b>6.57E-03</b>	6.56E-03	2.14E-05	2.27E-05	1.82E-05
KC10-2fl-5rl	0.8065	0.8065	0.8065	2.74E-05	3.30E-06	2.74E-05
KC30-3fl-1rl	0.1936	<b>0.1993</b>	0.1974	5.68E-03	4.53E-03	6.74E-03
KC30-3fl-2rl	0.5754	<b>0.5785</b>	0.5745	4.92E-03	4.14E-03	3.67E-03

Table 12: The Kolmogorov-Smirnov test values for the 30 runs of each problem instance using different quality attribution schemes on experiments that control the population size: Average Quality Attribution (AQA), Extreme Quality Attribution (EQA) and Predictive Quality Attribution (PQA).

Problem	PQA vs. AQA		PQA vs. EQA	
	d	p	d	p
CHR20A	0.2000	0.537	0.2667	0.200
BUR26E	0.2000	0.537	0.3000	0.109
TAI30A	0.3750	0.162	0.3750	0.162
STE36B	0.2500	0.633	0.3750	0.162
Royal Road	0.2667	0.200	0.2667	0.200
KC10-2fl-5rl	0.3750	0.162	0.3750	0.162
KC30-3fl-1rl	0.3500	0.121	0.3542	0.113
KC30-3fl-2rl	0.2000	0.890	0.4000	0.136

The means and standard deviations of the three quality attribution schemes are shown in Table 11. It can be observed that the mean performance of PQA is usually not higher than that of the other two quality attribution schemes. To check for a statistical significance of the difference in the performances of the quality attribution schemes we submitted the results from the 30 runs to the Kolmogorov-Smirnov (KS) non-parametric test [42], with the null hypothesis that there is no significant difference between the result sets.

The null hypothesis was not rejected in any of the KS tests (see Table 12), which means that the superior performances of the Average Quality Attribution (AQA) and Extreme Quality Attribution (EQA) methods compared to the Predictive Quality Attribution (PQA) strategy are not statistically significant.

## Choosing the Appropriate Forecasting Model for Predictive Parameter Control

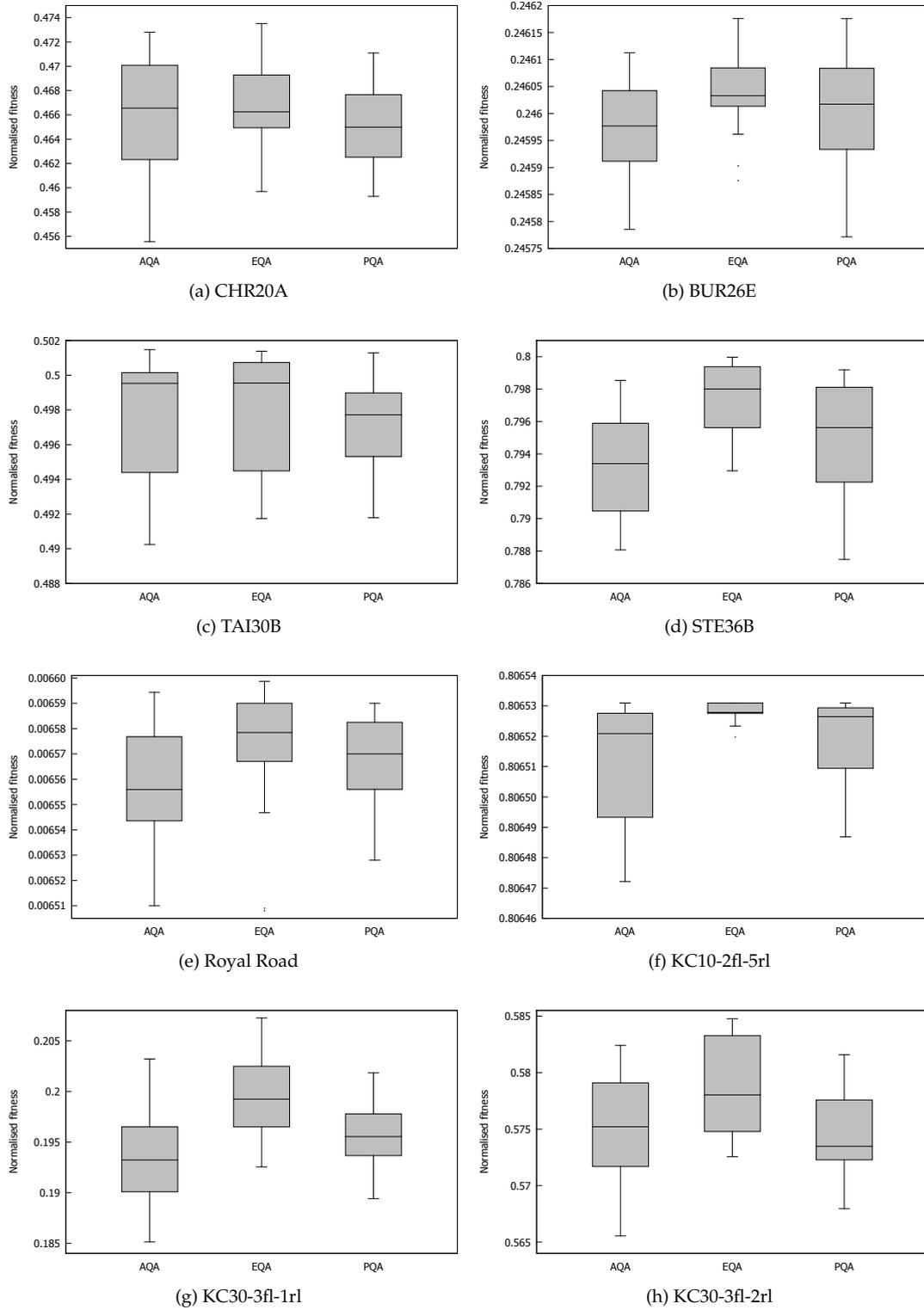


Figure 5: Performance improvement of the EA using different quality attribution strategies for controlling population size to solve the QAP, the mQAP and the Royal Road problem

This indicates that using PQA to forecast the effect of the population size does not have a negative effect on the performance of the algorithm, despite the fact that this parameter does not meet the assumptions made by the forecasting models. Hence, we expect that when new EA parameters are controlled, prediction can be used without a significant negative effect on the performance of the algorithm.

## 9 Conclusion

Controlling parameters adaptively relies on strategies to extract effects of parameter values from the optimisation process and extract information which is subsequently used to attribute a measure of quality to the parameter values used. In previous work [1], we have suggested that using prediction to attribute a quality metric to the parameter values to choose from for the next iteration is beneficial to the performance of the algorithm. Many forecasting techniques exist for time series prediction. Four of these were considered for use in PQA.

The suitability of each forecasting technique for predicting the effect of six EA parameters (mutation rate, crossover rate, population size, mating pool size, the type of crossover operator and the type of mutation operator) was assessed. It was established that all considered parameters, with the exception of the population size, adhered, within a certain tolerance, to the assumptions required for the use of each of the considered forecasting techniques. Hence all forecasting methods could reasonably be considered for the projection of future parameter quality in PQA.

Despite being one of the simplest forecasting strategies, LR has proven to be the most effective model to predict the effect of EA parameters, albeit by a negligible margin. In the experimental studies we have used the PQA method with LR to adjust the mutation rate, crossover rate, mating pool size, mutation operator and crossover operator throughout the optimisation process carried out using different problem instances. The trials have demonstrated that PQA outperforms the quality attribution methods currently considered most successful.

Although the population size was not found to conform to the assumptions made by the forecasting techniques to a reasonable tolerance, the results of the experiments have shown that using the PQA strategy to control the population size did not have a significant negative effect on the performance of the EA. Based on the experiments with the population size, we argue that if EA parameters are controlled without prior testing for statistical properties, the use of PQA has a potential benefit without the expectation of any negative effects on the performance of the algorithm. The exploration of the best possible prediction method is unlikely to be affected by the discretisation, given that we employ two different settings.

## References

- [1] Aldeida Aleti and Irene Moser. Predictive parameter control. In *Genetic and Evolutionary Computation Conference, Proceedings*, pages 561–568. ACM, 2011.
- [2] Aldeida Aleti and Irene Moser. Entropy-based adaptive range parameter control for evolutionary algorithms. In *Genetic and Evolutionary Computation Conference, Proceedings*, pages 1501–1508. ACM, 2013.
- [3] Aldeida Aleti, Irene Moser, and Sanaz Mostaghim. Adaptive range parameter control. In *Congress on Evolutionary Computation, Proceedings*, pages 1–8. IEEE, 2012.
- [4] Andrea Arcuri and Gordon Fraser. On parameter tuning in search based software engineering. In *Symposium on Search Based Software Engineering*, volume 6956 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2011.
- [5] Thomas Bäck, Ágoston Endre Eiben, and Nikolai A. L. van der Vaart. An empirical study on gas without parameters. In *Parallel Problem Solving from Nature*, volume 1917 of *Lecture Notes in Computer Science*, pages 315–324. Springer-Verlag, 2000.
- [6] Thomas Bäck and Martin Schütz. Intelligent mutation rate control in canonical genetic algorithms. In *International Symposium on Foundations of Intelligent Systems, Proceedings*, volume 1079 of *Lecture Notes in Artificial Intelligence*, pages 158–167, 1996.

- [7] Thomas Bartz-Beielstein, Christian Lasarczyk, and Mike Preuss. Sequential parameter optimization. In *IEEE Congress on Evolutionary Computation*, pages 773–780. IEEE, 2005.
- [8] Rainer E. Burkard, Stefan E. Karisch, and Franz Rendl. Qaplib - A quadratic assignment problem library. *European Journal of Operational Research*, 55(1):115–119, 1991.
- [9] Jorge Cervantes and Christopher R. Stephens. Limitations of existing mutation rate heuristics and how a rank GA overcomes them. *IEEE Transactions on Evolutionary Computation*, 13(2):369–397, 2009.
- [10] Nicos Christofides and Enrique Benavent. An exact algorithm for the quadratic assignment problem on a tree. *Operations Research*, 37(5):760–768, 1989.
- [11] David Corne, Martin J. Oates, and Douglas B. Kell. On fitness distributions and expected fitness gain of mutation rates in parallel evolutionary algorithms. In *Parallel Problem Solving from Nature*, volume 2439 of *Lecture Notes in Computer Science*, pages 132–141. Springer-Verlag, 2002.
- [12] Lawrence Davis. Adapting operator probabilities in genetic algorithms. In *International Conference on Genetic Algorithms, Proceedings*, pages 70–79. Morgan Kaufman, 1989.
- [13] Lawrence Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reingold, 1991.
- [14] Kenneth A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, 1995.
- [15] Kenneth A. De Jong. Parameter setting in EAs: a 30 year perspective. In *Parameter Setting in Evolutionary Algorithms*, volume 54 of *Studies in Computational Intelligence*, pages 1–18. Springer, 2007.
- [16] Ágoston Endre Eiben, Robert Hinterding, and Zbigniew Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 2007.
- [17] Ágoston Endre Eiben, Elena Marchiori, and V. A. Valkó. Evolutionary algorithms with on-the-fly population size adjustment. In *Parallel Problem Solving from Nature*, volume 3242 of *Lecture Notes in Computer Science*, pages 41–50. Springer-Verlag, 2004.
- [18] Ágoston Endre Eiben and Martijn C. Schut. New ways to calibrate evolutionary algorithms. In *Advances in Metaheuristics for Hard Optimization*, Natural Computing Series, pages 153–177. Springer, 2008.
- [19] Ágoston Endre Eiben and Selmar K. Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1(1):19–31, 2011.
- [20] Álvaro Fialho, Luís Da Costa, Marc Schoenauer, and Michèle Sebag. Extreme value based adaptive operator selection. In *Parallel Problem Solving from Nature*, volume 5199 of *Lecture Notes in Computer Science*, pages 175–184. Springer, 2008.
- [21] Álvaro Fialho, Luis Da Costa, Marc Schoenauer, and Michèle Sebag. Analyzing bandit-based adaptive operator selection mechanisms. *Annals of Mathematics and Artificial Intelligence - Special Issue on Learning and Intelligent Optimization*, 2010.
- [22] Álvaro Fialho, Marc Schoenauer, and Michèle Sebag. Analysis of adaptive operator selection techniques on the royal road and long k-path problems. In *Genetic and Evolutionary Computation Conference, Proceeding*, pages 779–786. ACM, 2009.
- [23] Álvaro Fialho, Marc Schoenauer, and Michèle Sebag. Toward comparison-based adaptive operator selection. In *Genetic and Evolutionary Computation Conference, Proceedings*, pages 767–774. ACM, 2010.

- [24] Jean Dickinson Gibbons and Subhabrata Chakraborti. *Nonparametric statistical inference*. CRC Press, 2003.
- [25] John J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-16(1):122–128, 1986.
- [26] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [27] Tzung-Pei Hong, Hong-Shung Wang, and Wei-Chou Chen. Simultaneously applying multiple mutation operators in genetic algorithms. *Journal of Heuristics*, 6(4):439–455, 2000.
- [28] Tzung-Pei Hong, Hong-Shung Wang, Wen-Yang Lin, and Wen-Yuan Lee. Evolution of appropriate crossover and mutation operators in a genetic process. *Applied Intelligence*, 16(1):7–17, 2002.
- [29] Christian Igel and Martin Kreutz. Operator adaptation in evolutionary computation and its application to structure optimization of neural networks. *Neurocomputing*, 55(1-2):347–361, 2003.
- [30] Christian Igel, Stefan Wiegand, and Frauke Friedrichs. Evolutionary optimization of neural systems: The use of strategy adaptation. In *Trends and Applications in Constructive Approximation*, volume 151 of *International Series of Numerical Mathematics*, pages 103–123. Birkhäuser Basel, 2005.
- [31] Terry Jones. A description of holland’s royal road function. *Evolutionary Computation*, 2(4):409–415, 1994.
- [32] Bryant A. Julstrom. What have you done for me lately? Adapting operator probabilities in a steady-state genetic algorithm. In *International Conference on Genetic Algorithms*, pages 81–87. Morgan Kaufmann, 1995.
- [33] Bryant A. Julstrom. Adaptive operator probabilities in a genetic algorithm that applies three operators. In *ACM Symposium on Applied Computing*, pages 233–238. ACM, 1997.
- [34] Joshua Knowles and David Corne. Instance generators and test suites for the multiobjective quadratic assignment problem. In *International Conference on Evolutionary Multi-Criterion Optimization, Proceedings*, volume 2632 of *Lecture Notes in Computer Science*, pages 295–310. Springer, 2003.
- [35] Tjalling C. Koopmans and Martin Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.
- [36] Denis Kwiatkowski, Peter C.B. Phillips Phillips, Peter Schmidt, and Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*, 54(1–3):159–178, 1992.
- [37] Fernando G. Lobo. Idealized dynamic population sizing for uniformly scaled problems. In *Genetic and Evolutionary Computation Conference, Proceedings*, pages 917–924. ACM, 2011.
- [38] Peter Merz and Bernd Freisleben. Fitness landscapes, memetic algorithms, and greedy operators for graph bipartitioning. *Evolutionary Computation*, 8:61–91, 2000.
- [39] Zbigniew Michalewicz and David B. Fogel. *How to solve it: modern heuristics*. Springer-Verlag, 2004.
- [40] Zbigniew Michalewicz and Martin Schmidt. Parameter control in practice. In *Parameter Setting in Evolutionary Algorithms*, volume 54 of *Studies in Computational Intelligence*, pages 277–294. Springer, 2007.
- [41] Melanie Mitchell. *An Introduction to Genetic Algorithms*. Complex Adaptive Systems. MIT-Press, Cambridge, 1996.
- [42] A. N. Pettitt and M. A. Stephens. The kolmogorov-smirnov goodness-of-fit statistic with discrete and grouped data. *Technometrics*, 19(2):205–210, 1977.

- [43] J. David Schaffer, Richard A. Caruana, Larry J. Eshelman, and Rajarshi Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *International Conference on Genetic Algorithms, Proceedings*, pages 51–60. Morgan Kaufman, 1989.
- [44] Leon Steinberg. The backboard wiring problem: a placement algorithm. *SIAM Review*, 3:37–50, 1961.
- [45] Dirk Thierens. Adaptive mutation rate control schemes in genetic algorithms. In *Congress on Evolutionary Computation, Proceedings*, pages 980–985. IEEE, 2002.
- [46] Dirk Thierens. An adaptive pursuit strategy for allocating operator probabilities. In *Genetic and Evolutionary Computation Conference, Proceedings*, pages 1539–1546. ACM, 2005.
- [47] James M. Whitacre, Tuan Q. Pham, and Ruhul A. Sarker. Use of statistical outlier detection method in adaptive evolutionary algorithms. *CoRR*, abs/0907.0595, 2009.
- [48] Yuk-Yin Wong, Kin-Hong Lee, Kwong-Sak Leung, and C.-W. Ho. A novel approach in parameter adaptation and diversity maintenance for genetic algorithms. *Soft Computing*, 7(8):506–515, 2003.
- [49] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and da V. G. Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7:117–132, 2003.