Expert Systems with Applications 38 (2011) 12826–12838

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Tuning metaheuristics: A data mining based approach for particle swarm optimization

Stefan Lessmann [a,*], Marco Caserta [a,1], Idel Montalvo Arango [b,2]

[a] Institute of Information Systems, University of Hamburg, Von-Melle-Park 5, D-20146 Hamburg, Germany
[b] Centro Multidisciplinar de Modelación de Fluidos, Polytechnic University of Valencia, Spain

ARTICLE INFO

Keywords:
Metaheuristics
Particle swarm optimization
Forecasting
Data mining

ABSTRACT

The paper is concerned with practices for tuning the parameters of metaheuristics. Settings such as, e.g., the cooling factor in simulated annealing, may greatly affect a metaheuristic's efficiency as well as effectiveness in solving a given decision problem. However, procedures for organizing parameter calibration are scarce and commonly limited to particular metaheuristics. We argue that the parameter selection task can appropriately be addressed by means of a data mining based approach. In particular, a hybrid system is devised, which employs regression models to learn suitable parameter values from past moves of a metaheuristic in an online fashion. In order to identify a suitable regression method and, more generally, to demonstrate the feasibility of the proposed approach, a case study of particle swarm optimization is conducted. Empirical results suggest that characteristics of the decision problem as well as search history data indeed embody information that allows suitable parameter values to be determined, and that this type of information can successfully be extracted by means of nonlinear regression models.

1. Introduction

The support of managerial decision making is a key area of interest in many scientific domains and several qualitative and quantitative planning methods have been developed to support various types of operational, tactical and strategic planning problems. Especially well-structured and repetitive decision tasks are commonly approached by translating the problem into a mathematical program whose optimal solution is then determined by an optimization procedure like, e.g., linear programming.

Over the past decades, heuristic procedures from the field of intelligent search have received much attention and nowadays represent a well-established approach towards solving complex optimization problems. Offering high-quality solutions in a timely manner, they complement classical mathematical programming methods, especially in large, combinatorial settings where optimal solutions are usually unattainable. Respective techniques have been developed in several domains (e.g., Operational Research, Engineering, Soft Computing, etc.) and include algorithms like simulated annealing (Kirkpatrick, Gelatt, & Vecchi, 1983), tabu-search (Glover & Laguna, 1997), cross-entropy (Boer, Kroese, Mannor, & Rubinstein, 2005) as well as nature-inspired methods like genetic algorithms (Goldberg, 1989; Holland, 1975), evolution strategies (Beyer & Schwefel, 2002) and, more recently, ant colony optimization (Dorigo & Stützle, 2005) and particle swarm optimization (PSO, Kennedy & Eberhart, 1995). The umbrella term *metaheuristics* is used throughout the paper to subsume respective approaches (see, e.g., Glover, 1986 for a formal definition of a metaheuristic).

A large body of literatures evidences the success of metaheuristics in various applications (see, e.g., Blum & Roli, 2003; Caserta & Voß, 2009; Glover & Kochenberger, 2003 for a survey). One key factor that seems to have a strong impact on the algorithmic performance is the fine tuning of the method; that is, the determination of the metaheuristic's parameters (e.g., the number of generations in genetic algorithms, the length of the tabu-list in tabu-search or the cooling factor in simulated annealing, etc.). According to Adenso-Diaz and Laguna (2006), there is evidence that 10% of the time required to develop a new metaheuristic is devoted to the actual development and that the remaining 90% is spent on fine tuning of algorithmic parameters. Consequently, it is of paramount importance to make a concerted effort in identifying and establishing a set of "standard" techniques to fine-tune a metaheuristic. One of the major achievements of such an effort would be to offset parameter specific issues in evaluating an algorithm. In addition, reproducibility of results would also be enhanced by such an approach, by making transparent the way in which parameter values should be set to tackle a given problem instance.

[*] Corresponding author. Tel.: +49 40 42838 4706; fax: +49 40 42838 5535.
E-mail addresses: lessmann@econ.uni-hamburg.de (S. Lessmann), caserta@econ.uni-hamburg.de (M. Caserta), imontalvo@gmmf.upv.es (I.M. Arango).
[1] Tel.: +49 40 42838 4294; fax: +49 40 42838 5535.
[2] Tel.: +34 963879890; fax: +34 963877981.

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.075

Some work has been carried out to develop elaborate parameterization approaches. For example, adaptations of individual metaheuristics have been proposed to tune parameters in a self-adaptive fashion (see, e.g., Battiti, Brunato, & Mascia, 2009; Beyer & Schwefel, 2002). Whereas the potential of such reactive metaheuristics is undoubted, the objective of this paper is to lay ground for an orthogonal approach towards parameter tuning. In particular, we examine whether a formal relationship between *effective* parameter values and characteristics of the underlying optimization problem exist and whether it is sufficiently strong to be exploited for an automated tuning approach.

Most metaheuristics operate in an iterative manner: Given a starting solution to an optimization problem, neighboring solutions are evaluated and used in some way to improve the present one, i.e., find better solutions to the problem. Consequently, data concerning the assessed candidate solutions and their appropriateness naturally becomes available during the execution of a metaheuristic. The particular way in which a solution is altered or a new solution is obtained on the basis of a present one, respectively, depends upon settings of the metaheuristic's parameters. Hence, we consider a parameter setting to be *effective*, if it facilitates improving a candidate solution's fitness. Employing the data produced, we aim at constructing a prediction model capable of estimating suitable parameter values for a subsequent iteration by means of regression. It is argued that if such a model shows high forecasting accuracy, it could be employed as a *tuning agent* to predict effective parameter settings on the basis of the metaheuristic's current status and problem-specific information. This differs from previous approaches to achieve self-adaptivity in the sense that it does not depend upon any particular search algorithm.

Therefore, the paper contributes to the literature by proposing a data mining based solution to the problem of tuning metaheuristics' parameters and providing empirical evidence of its efficacy. To that end, a case study is undertaken to systematically explore the feasibility of forecasting effective parameter settings for one particular metaheuristic, the PSO algorithm. The study comprises established as well as state-of-the-art regression methods to scrutinize the nature of the relationship between the employed independent variables and effective parameter values (e.g., linear versus nonlinear) and identifies promising candidate models. Since embedding a forecasting model as tuning agent into PSO – or any other metaheuristic – would involve processing batches of data that become available throughout successive PSO iterations, a *learning curve analysis* is conducted to explore the forecasting models' sensitivity towards data size and simulate *online learning*. This complements the assessment of different candidate models and lays ground for future research to develop and assess a hybrid metaheuristic with integrated tuning agent.

The paper is organized as follows: Related work concerning the tuning of metaheuristics as well as hybrid procedures at the interface of metaheuristics and data mining is reviewed in the following section, before the proposed tuning approach together with some background on PSO and regression modeling is elaborated in Section 3. Section 4 explains the design of the empirical study; respective results are reported in Section 5. Limitations and opportunities for future research are discussed in Section 6, before conclusions are drawn in Section 7.

2. Related work

The paper proposes employing data-driven regression models to determine effective parameter settings for a particular metaheuristic, the PSO algorithm. Therefore, using the taxonomy of (Meisel & Mattfeld, 2010), the work can be described as an application of "*Data Mining (DM) to increase the effectiveness of Operational Research (OR) by refinement*".

Previous approaches at the interface of OR and DM have mainly considered a reverse approach. That is, OR techniques have been employed to support DM tasks by, e.g., automating some parts of the DM process. In particular, the use of metaheuristics in conjunction with predictive regression or classification models is widespread. First, building a prediction model inevitably involves an optimization step to, e.g., minimize some loss function of the model's forecasts and true outcomes over a set of example cases. Consequently, metaheuristics can be employed to perform this optimization, i.e., construct the forecasting model from empirical data. Examples include, e.g., the construction of classification models by means of genetic or ant algorithms (Martens et al., 2007; Sharpe & Glover, 1999) or the training of support vector machine (SVM) or neural network classifiers with evolutionary algorithms (Pendharkar, 2001; Pendharkar & Rodger, 2004; Stoean, Preuss, Stoean, El-Darzi, & Dumitrescu, 2008).

Furthermore, the ability of metaheuristics to solve complex combinatorial optimization tasks has facilitated modifications and extensions of existing data mining techniques and the development of novel algorithms, respectively, to better reflect the requirements of real-world decision making tasks. For example, (Bhattacharyya, 1999) proposes a customer targeting model for direct marketing applications. Contrary to standard approaches that first estimate customers' likelihood of responding by means of classification and subsequently decide upon the number of prospects to be contacted, this model explicitly considers the available marketing budget (i.e., how many customers can be solicited) to optimize the response rate among those customers who will eventually be included in the campaign. As a result, the model directly optimizes campaign profit, rather than response rate, and is in this respect more aligned with marketers' objective. However, incorporating such additional information complicates the construction of the forecasting model. Therefore, (Bhattacharyya, 1999) develops a genetic algorithm to perform this task. In a similar fashion, (Orsenigo & Vercellis, 2004) consider the SVM classifier, and argue that the optimization necessary to construct a SVM model exhibits inaccuracies. For mathematical convenience, classification errors are taken into account in an approximate way, whereas a discrete counting of errors is more aligned with the principles of statistical learning theory. However, building a discrete SVM requires solving a complex mixed-integer problem, which is accomplished by means of a tabu-search metaheuristic.

The most popular application of metaheuristics in predictive modeling is concerned with determining a model's *hyperparameters* (e.g., the number of hidden layers in a neural network) to adapt them to a particular forecasting task. This approach has received particular attention within in scope of SVM based classification (Chen & Hsiao, 2008; Friedrichs & Igel, 2005; Guo, Yang, Wu, Wang, & Liang, 2008; Hou & Li, 2009) and regression models (Fei, Liu, & Miao, 2009; Gijsberts, Metta, & Rothkrantz, 2010; Tang, Zhuang, & Jiang, 2009; Üstün, Melssen, Oudenhuijzen, & Buydens, 2005). Commonly, the model selection problem (i.e., hyperparameter tuning) is considered alongside the task of feature selection, which is concerned with discarding less informative independent variables to ease prediction. Due to the potential of metaheuristics to effectively identify high quality solutions in very large search spaces, they facilitate a joint optimization of model parameters and features. Respective approaches can be found in, e.g., Huang, Chen, and Wang (2006), Huang and Wang (2006), Lin, Shiue, Chen, and Cheng (2009), Min, Lee, and Han (2006), Wu, Tzeng, Goo, and Fang (2007). Dedicated feature selection algorithms that ground on the principles of heuristic search are developed in, e.g., Kim, Street, Russell, and Menczer (2005), Meiri and Zahavi (2006), Pacheco, Casado, and Núñez (2009), Tosun, Turhan, and Bener (2009).

More recently, the construction of ensemble models has become an active research topic in data mining. Aiming at improving forecasting accuracy, this approach stipulates constructing several *base* models (i.e., individual forecasting models) and aggregating their predictions to form a final forecast. The selection of appropriate base models for the ensemble as well as their weighting within the committee represents a challenging optimization problem and some encouraging results have been obtained with approaches that make use of metaheuristics (Johansson, König, & Niklasson, 2010; Kim & Oh, 2008; Nanni & Lumini, 2009).

Contrary to the large body of research on the use of metaheuristics to aid predictive DM, approaches that employ data-driven DM techniques within the scope of heuristic search are scarce. It appears that the task of determining the parameters of a DM model is considered imperative for its success, whereas many applications of metaheuristics devote less attention towards tuning the algorithm's parameters. The papers that do address the problem of determining metaheuristics' parameters commonly make use of well-established statistical designs to examine the response surface of a particular method with respect to parameter settings. The information gained through this analysis may subsequently be employed to define a tuning procedure. For example, in Adenso-Diaz and Laguna (2006), a tool called CALIBRA is proposed as a technique that finds good values for up to five algorithmic parameters. They exploit Taguchi fractional factorial design to reduce the overall number of trials required to train the model. The authors assess CALIBRA on a set of six different algorithms and use hypothesis testing to assert whether the parameter values suggested by the tool allow finding solutions which are significantly better than those proposed by the original authors of the algorithms.

In Coy, Golden, Rungen, and Wasil (2001), a four-step procedure is proposed, in which a two-level factorial design is coupled with linear regression to find a linear approximation of the response surface of the set of parameters. Subsequently, a gradient descent technique is employed to find a "good" value for each parameter. Finally, the method is applied on a set of benchmark problems, and results are collected to show that a good parameter setting leads to improvements in the objective function value. A similar technique has been employed in Caserta and Rico (2009), where a circumscribed central composite design is used to generate observations and a higher degree polynomial is then used to approximate the response surface of a set of parameters. Once the response surface is known, a global optimization method is employed to find the optimal parameter value with respect to the surface. The model is built on a set of training instances and then validated on a different set of testing instances of a class of lot sizing problems.

In Park and Kim (1998), a nonlinear response surface is used to capture the impact of parameters on a SA algorithm. A good set of values of such parameters is then determined via a modified simplex method for nonlinear programming, that can be used to deal with bounds on parameter values. The proposed method is tested on three different combinatorial optimization problems and the collected results are then compared with those of other SA implementations, whose fine tuning was achieved via extensive experiments. The results show that there is no statistical difference in performance between the two types of SA algorithms and, consequently, that the proposed approach can be used to conduct an "automatic" fine tuning. A similar approach is followed in Parson and Johnson (1997), where a full-factorial design has been employed to define a response surface for up to four parameters of a GA. The fine-tuned algorithm is then tested on a pool of DNA sequence assembly problems. The results illustrate the effectiveness of the procedure.

A somehow different approach is proposed in Xu, Chiu, and Glover (1998), where two statistical tests are employed to fine-tune a set of algorithmic parameters of a TS algorithm and to validate the proposed statistical model. First, a Friedman test is used to detect significance of a specific parameter upon the algorithm performance and, via a series of pairwise comparisons, to identify a good parameter value; subsequently, a Wilcoxon test is employed to verify whether there is a statistically significant difference between any two runs of the algorithm. The TS algorithm is then used upon a set of problems drawn from the telecommunication network design field.

The tuning method proposed here draws inspiration from these previous approaches to determine a metaheuristic's parameters on the basis of an estimated response surface. However, instead of employing an offline tuning mechanism, i.e., a mechanism that takes as input some characteristics of the instance at hand and provides a "good" set of parameters values before the optimization process starts, in this paper we suggest organizing tuning in an online fashion. That is, we would like to *learn* which parameters work well for a given instance of a problem during the optimization phase and exploit this knowledge to increase the metaheuristics efficiency. Specifically, the benefits of a well-designed tuning mechanism concern either the computational time, by reducing the total running time, or the solution quality, by increasing the final solution quality. The conceptual advantage of our approach is that the online tuning procedure has access to more detailed information, i.e., about the particular problem instance, rather than a group of problems (e.g., the group of lot sizing problems).

Moreover, many of the previous tuning approaches employ techniques from the field of parametric statistics. On the one hand, these procedures rely upon distributional assumptions that may well be violated in the focal application. Another concern relates to the prominent use of linear regression. Since this technique is unable to capture complex, nonlinear relationships in data, previous studies may have failed to identify the full potential of regression-based parameter tuning. Therefore, this papers aims at exploring the true potential of regression-based tuning. This is achieved by (1) employing not only linear but also powerful nonparametric procedures, capable of approximating any existing relationship in data and (2) by collecting information during the search process and exploiting it in an online fashion within a self-adapting mechanism to update parameters values.

3. Methodology

3.1. Particle swarm optimization

PSO is an evolutionary computation technique that was first developed by Kennedy and Eberhart (1995) and is inspired by the behavior of a flock of birds trying to reach, when flying, a destination not completely known. PSO is a population-based metaheuristic since, at each iteration, a population is used to explore the solution space. In PSO, each particle (bird of the flock) contains a *position vector* in the solution space that represents a potential solution of the problem; communication and information exchange among particles is done while they are moving. Birds, besides having an individual behavior, also develop some social behavior and coordinate their movement towards a destination (Montalvo, Izquierdo, Perez, & Tung, 2008).

The performance of each particle is measured using a predefined fitness function, according to the problem at hand. The position vector of a particle is evaluated using the fitness function and the particle with the highest fitness value will be considered as the leader of the swarm.

The initial position vector of every particle is generated randomly. Later on, particles change their position iteratively. Besides the position vector, particles also have two other associated vectors: the *flying velocity vector* and the *best position vector* ever reached by the particle. Subsequently, the combination of these three vectors is referred to as a particle's *signature*. It should be noted that every vector associated with particles has as many components as there are decision variables in the underlying optimization problem. Therefore, the signature embodies information about the decision problem, the present solution represented by the particle, the overall best solution found so far and information concerning the search history of PSO.

Movements of individual particles involve the use of their own three vectors as represented in the following equations:

$$X_i^{new} = X_i + V_i^{new} \tag{1}$$

$$V_i^{new} = \omega \cdot V_i + c_1 \cdot rnd() \cdot (P_i - X_i) + c_2 \cdot rnd() \cdot (P_g - X_i). \tag{2}$$

Eq. (1) shows that for calculating a new position vector for the *i*th particle, X_i^{new}, it is only necessary to calculate a new velocity vector, V_i^{new} and add this new velocity vector to the current position of the particle. The new velocity vector is calculated by (2), whereby c_1 and c_2 are two positive constants called *learning rates,* the *rnd*() functions generate two (independent) random numbers between (0, 1) and ω is a factor of inertia that has been suggested by Shi and Eberhart (1998) to control the impact of velocity history on the new velocity. This factor may vary from one cycle to the next. As it permits to balance global and local searches, it has been suggested to let it decrease linearly with time, so as to first emphasize global search and then, with each cycle of the iteration, prioritize local search. P_i and P_g are the best position ever reached by the *i*th particle and the position of the leader of the swarm, respectively.

On each dimension *j*, particle velocities are adjusted to minimum and maximum velocities, to control excessive roaming of particles outside the search space.

$$\begin{aligned} &Vmin_j = -Vmax_j \\ &Vmax_j = Vmax_{factor} \cdot (maxValue_j - minValue_j) \\ &Vmin_j \leqslant Vj \leqslant Vmax_j \end{aligned} \tag{3}$$

$maxValue_j$ and $minValue_j$ correspond to the extreme values of the range of the *j*th variable. These values are considered known in the problem but $Vmax_{factor}$ should be set up at first. Parameter $Vmax_{factor}$ will have a direct influence on the exploration and exploitation capacities of particles. If V_j is too big, particles might fly through good solutions. If V_j is too small, on the other hand, particles may not explore sufficiently beyond locally good regions and could easily be trapped in local optima and unable to move far enough to reach a better position in the problem space.

The previously described approach can be considered the standard PSO algorithm, which is applicable to continuous systems and cannot be used directly for discrete problems. It could also have difficulties to keep good levels of population diversity. In this paper, to deal with discrete variables, only the integer part of the discrete velocity components will be used. To enrich diversity, particles will be send to a random position in the solution space, if they get to exactly the same position of the leader of the swarm (Montalvo, Izquierdo, Pérez, & Iglesias, 2008).

A very important step is to properly set up the parameters used by the algorithm. For example, individual parameter values could be determined for each particle and these could be updated after every cycle. Whereas this opportunity is commonly ignored in applications of PSO, one may speculate that such a fine-tuning has the potential to improve the overall efficiency of the algorithm. Therefore, the paper's objective is to explore whether information contained in particles' signatures (i.e., position vectors, flying velocity vectors and best position vectors) can be used to determine suitable parameter settings in a self-adaptive fashion. In particular, an automated tuning of the parameters c_1, c_2, and $Vmax_{factor}$ is considered whereas the following heuristic is employed to determine ω (Jin, Cheng, Yan, & Zhang, 2007):

$$\omega = 0.5 + \frac{1}{2(\ln(k) + 1)} \tag{4}$$

with *k* denoting the iteration number.

3.2. Regression analysis

The objective of regression is to estimate a functional relationship between a continuous target variable $y \in R$ and a set of attributes[3] $x_1, \ldots, x_M \in R$. The dependency between attributes and target is assumed to exists but unknown and needs thus to be estimated (*learned*) from a dataset of past observations $D = \{\mathbf{x}_i, y_i\}_{i=1}^N \in R^{M+1}$. Subsequently, the terms *pattern* and *example* are used to refer to a single element of the dataset *D* and a vector of attributes, $\mathbf{x}_i \in R^M$, respectively. A regression model can be characterized as a functional mapping from examples to targets: $f(\mathbf{x}_i)$: $R^M \mapsto R$.

For example, the classical multiple linear regression (MLR) assumes an additive, linear relationship of the form:

$$\hat{y} = f(\mathbf{x}) = \beta_0 + \sum_{j=i}^{M} \beta_j x_j, \tag{5}$$

whereby $\hat{y}$ denotes the model-based estimate of *y*, β_j an attribute's coefficient and β_0 the intercept of the regression function, respectively. These model parameters are determined by means of minimizing the squared error loss of the model over *D*:

$$\beta^* = \min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=i}^{M} \beta_j x_j \right)^2. \tag{6}$$

The task of solving the optimization problem underlying the determination of $\boldsymbol{\beta}$, and a regression model's free parameters in general, is subsequently referred to as *model building.*

A large number of modifications and extensions of the standard linear regression model have been proposed in the literature and several alternative techniques for solving the regression problem have been developed. The reader is referred to statistical or data mining textbooks like, e.g., Hastie, Tibshirani, and Friedman (2009), Izenman (2008) for an elaborate discussion of respective approaches.

3.3. Forecasting PSO parameters by means of regression

Within the scope of PSO, candidate solutions to an optimization problem are represented by particles (birds), which, in turn, are characterized by their signature. The particular way in which the *positions* of these particles, and thus the decision variables and the problem's solution, respectively, are updated during PSO depends upon the setting of PSO parameters c_1, c_2, and $Vmax_{factor}$ (see (2) and (3)). Whether an update, or *move* in metaheuristics terminology, is successful can be determined ex post by appraising the change in a particle's fitness. It may be speculated that the appropriateness of parameter settings depends upon the particle's signature. In other words, it seems reasonable that the signature of a particle contains valuable information how updates of this particle should be performed (i.e., how parameter values should be set) to improve its fitness in the next move. If this hypothesis can be confirmed, it may be possible to exploit this type of information for constructing an intelligent tuning agent that determines

[3] These are usually referred to as independent variables in the statistics literature.

suitable parameter settings for individual particles in a fully-automatic manner on the basis of their signatures.

This paper employs regression analysis to formally test the existence and strength of such a relationship between signatures and effective parameter values. In particular, individual regression models are built for the PSO parameters c_1, c_2, and $Vmax_{factor}$. That is, these parameters represent individual dependent variables and particles' signatures are taken as covariates. Since regression models are estimated from a training dataset, a sample of patterns is needed. This requirement can be fulfilled in the following way: the population-based nature of PSO ensures that a set of updated particles (with new signatures) becomes available after every cycle. Therefore, some default approach to set parameters (i.e., the ones in use today) can be employed in the first few cycles until a sufficiently large datasets of parameter values and signatures has been gathered. Then, a regression agent could replace the default tuning approach and determine parameter settings for subsequent cycles. In fact, this leads to an *online learning* (Cesa-Bianchi, Conconi, & Gentile, 2002; Shalev-Shwartz & Singer, 2007) approach, where the regression model is updated after every PSO cycle.

It has to be emphasized that the datasets to be gathered during PSO cycles should only contain those particles that have been improved during their last update. That is, the success of an update indicates that the corresponding parameter values have been suitable. More specifically, we define an *effective* parameter setting to be a setting that improves the particle's fitness in the next update. Consequently, employing a dataset of all successful updates during PSO iterations, a regression model can be trained to approximate a functional relationship between particle signatures and parameter values. The model may then be employed to estimate *effective* parameter settings for subsequent PSO iterations, i.e., tune the algorithm's parameters in a fully self-adaptive manner. The architecture of the overall system is depicted in Fig. 1 whereas information about the data employed in this study is provided in the following section.

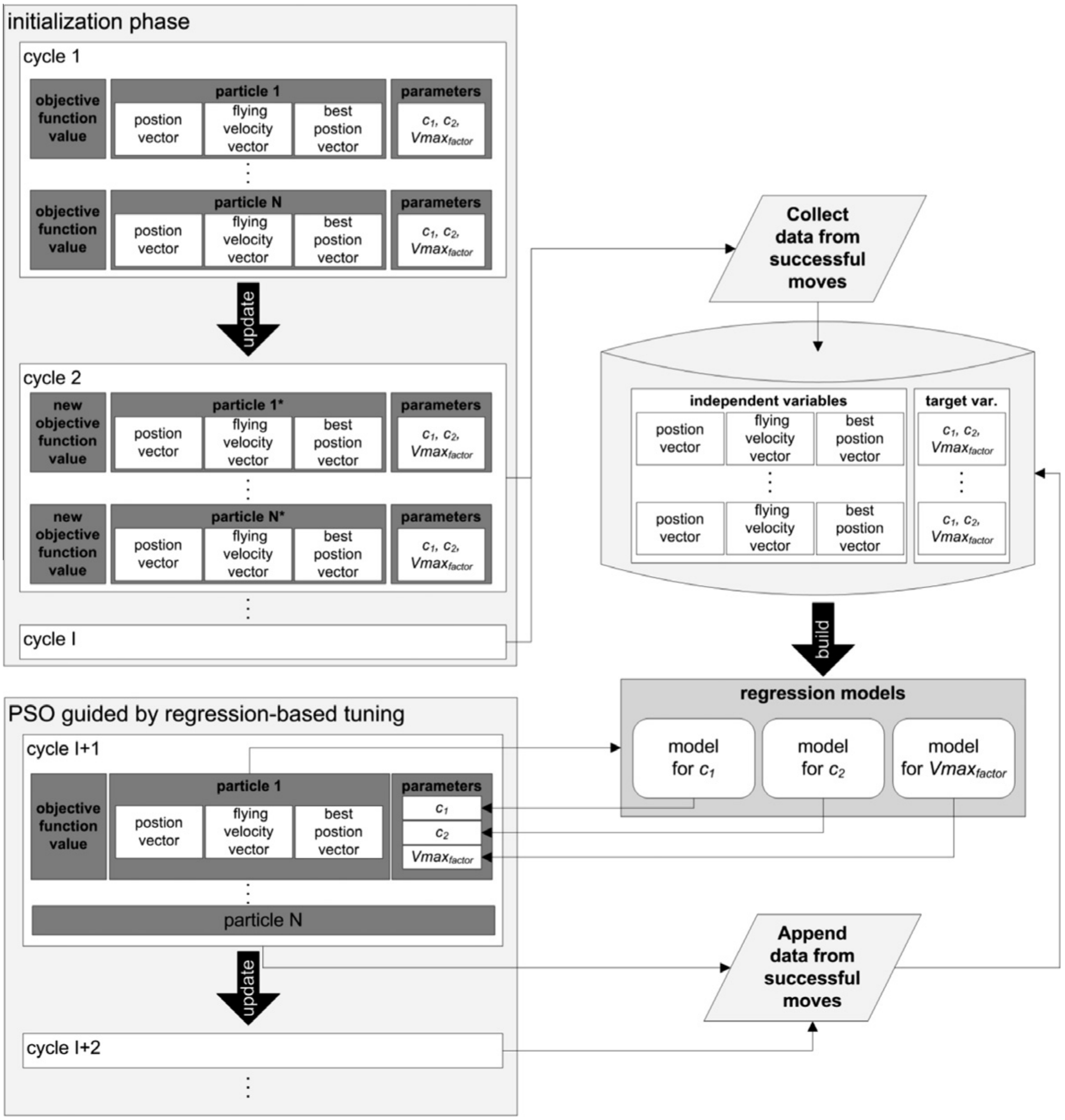

Fig. 1. Architecture of the proposed system to determine PSO parameters.

4. Experimental design

4.1. Data and variables

In order to evidence the viability of the proposed approach, a problem from the domain of water supply network planning has been selected as case study. Roughly speaking, the objective in such problems is to determine a cost-effective system design, which guarantees a satisfactory delivery of water in the amount, and of the quality, required by all the connected users. Moreover, the design may need to fulfill additional constraint associated with reliability in case of system failures (Izquierdo, Pérez, & Iglesias, 2004). The variables in the problem are the diameters pertaining to new pipes of the network or those of additional pipelines in case of network extensions, respectively. The problem can be characterized as a mixed-integer problem since feasible values for decision variables have to be selected from a discrete set of commercially available pipe diameters.

PSO has been found to work well for solving water supply network planning problems in previous work, which motivates the selection of this particular problem case. The reader is referred to Montalvo, Izquierdo, Pérez, and Iglesias (2008), Montalvo, Izquierdo, Perez, and Tung (2008)) for an elaborate description of the optimization problem and explanation how it can be approached by means of PSO.

We apply PSO to a water supply network planning problem with 34 decision variables using a swarm of 100 particles and collect data for 100 PSO cycles. Consequently, a dataset of 10,000 cases is obtained, each of which is characterized by 102 attributes (34 decision variables for each of the three vectors constituting a particle's signature; see Section 3.1). Out of these 10,000 particle updates, 5284 eventually increased fitness and can therefore be considered successful. Therefore, the maximal number of cases available for regression analysis is 5284. In fact, the motivation for considering 100 cycles has been to produce a dataset of 'reasonable' size with regard to the computational effort of constructing many forecasting models. The results of Section 5.3 will show that the present data size is sufficient to facilitate stable learning.

In order to simulate online learning where batches of data become available throughout successive PSO cycles, regression models are built from datasets of increasing size. For example, the first regression analysis employs only the first 100 patterns of the full dataset and strives to estimate a functional relationship between signatures and PSO parameters. Subsequently, 100 additional patterns are appended to the datasets and the experiment is repeated. Overall, 22 experiments are conducted with datasets of $100, 200, \ldots, 1000, 1200, \ldots, 2200, 2500, 3000, \ldots, 5000$ patterns.

4.2. Learning methods for regression

In order to solve the forecasting task associated with predicting candidate settings for PSO parameters, any regression method could be employed. Whereas MLR (6) seems an obvious technique to consider, it has to be noted that, depending upon the decision problem PSO is applied to, the number of attributes may be very large. Remember that the dataset employed in this study contains 102 covariates. It has been shown that any regression method that focuses solely on minimizing the empirical error over D is vulnerable to over-fitting in high-dimensional settings (Vapnik, 1998). However, regression analysis is commonly employed in conjunction with a feature selection schema that identifies and dismisses non-informative attributes and constructs a parsimonious model. Specifically, a stepwise linear regression model (SWMLR) will be considered alongside traditional MLR in the empirical evaluation, whereby individual attributes are assessed in terms of their ability to improve the coefficient of determination, R^2, of the overall model.

Whereas attribute selection is expected to overcome problems concerned with over-fitting, another limitation of (SW)MLR is associated with the assumption of an additive and linear relationship between attributes and targets. Therefore, the empirical study comprises two state-of-the-art learning methods that are robust towards a large number of attributes and are capable of approximating nonlinear relationship in data.

The least-square support vector machine (LSSVM, Suykens, Van Gestel, De Brabanter, DeMoor, & Vandewalle, 2002; Suykens & Vandewalle, 1999) augments the objective of MLR (6) by incorporating a ridge-penalty to prevent over-fitting. Specifically, the squared sum of attributes' coefficients is minimized alongside empirical errors to balance the conflicting goals of building accurate and parsimonious models. That is, the modified objective (7) creates an incentive to set the coefficients of less informative attributes to small values and thereby limit such attributes' influence on the model. Note that the extreme case of setting a coefficient to zero equates to discarding the corresponding attribute. In this sense, LSSVM shares similarities with SWMLR, but does not require an iterative model building and evaluation to decide upon an appropriate treatment of attributes. Furthermore, LSSVM can accommodate *kernel functions* that facilitate nonlinear regression models to be built. In particular, a kernel function performs an implicit mapping of the data into a feature space of higher dimension. A linear relationship between (transformed) attributes and target is then constructed in the feature space, which is equivalent to a nonlinear regression in the original input space (Vapnik, 1995). The LSSVM can be depicted as follows (Suykens et al., 2002):

$$\hat{y} = f(\mathbf{x}) = \beta_0^* + \beta^* \varphi(\mathbf{x}_i), \tag{7}$$

$$\begin{aligned} &\beta^* = \min_{\beta} \frac{1}{2} \sum_{i=1}^{N} e_i^2 + \gamma \frac{1}{2} \|\beta\|_2^2 \\ &s.t.: \\ &y_i(\beta_0 + \beta\varphi(\mathbf{x}_i)) = 1 - e_i, \quad \forall i = 1, \ldots, N, \end{aligned} \tag{8}$$

whereby $\|\cdot\|_2$ denotes the *Euclidian* norm. The model parameter γ allows users controlling the trade-off between accuracy and simplicity (i.e., emphasizing minimization of the empirical error or the ridge-penalty, respectively) The kernel function, ϕ, has to fulfill *Mercer's* conditions (see, e.g., Cristianini & Shawe-Taylor, 2000) and the *Gaussian radial basis function* (Vapnik, 1995) kernel is selected for the empirical evaluation because of its popularity in previous research (see, e.g., Lessmann & Voß, 2009). However, it should be noted that, by employing the linear kernel function, LSSVM facilitates constructing a linear regression model. In this case, (7) and (8) differ from MLR only by incorporating the regularization term to penalize complex models. Consequently, comparing the accuracy of such a linear LSSVM (L-LSSVM) model with MLR and SWMLR allows conclusions regarding the importance of limiting the effect of less informative attribute, and the effectiveness of the respective approaches implemented in LSSVM and SWMLR, respectively, to be drawn. Consequently, L-LSSVM is also considered within the empirical evaluation.[4]

[4] The selection of LSSVM instead of standard support vector regression (see, e.g., Smola & Schölkopf, 2004) is motivated by the fact that the latter incorporates a special loss function that requires an additional parameter, the width of the -tube, to be set. We are not aware of any theoretical or empirical evidence that would suggest this loss function to provide higher forecasting accuracy in general. However, the need to tune an additional model parameter is highly detrimental in the application considered here because an effective tuning of PSO parameters dictates that using the agent (i.e., the forecasting model) should be as simple as possible. This important issue is discussed in more detail in Section 6.

A different approach towards building robust and accurate forecasting models is taken in random regression forests (REGFOR, Breiman, 2001). This type of model constructs a large number of *regression trees* (Breiman, Friedman, Olshen, & Stone, 1984) and averages their predictions to form a forecast. Each tree is constructed from a *bootstrap sample* of size *N*, drawn from *D* with replacement. These random samples differ from each other (i.e., contain different patterns) and therefore increase the diversity among tree models. Further diversification is achieved by modifying the tree growing procedure, key to any decision tree type model. Instead of selecting an attribute among all available attributes when dividing a tree-node, a random sample of attributes is drawn and the best split is determined among these attributes (Breiman, 2001). Random forest forecasting models have provided promising results in previous work, especially for predicting discrete target variables (see, e.g., Buckinx & Van den Poel, 2005; Buckinx, Verstraeten, & Van den Poel, 2007; Burez & Van den Poel, 2007; Burez & Van den Poel, 2009; Coussement & Poel, 2009; Coussement & Van den Poel, 2008). Fewer studies have employed REGFOR regression models but the available findings suggest that this type of model represents a highly competitive forecasting approach (see, e.g., Lariviere & Van den Poel, 2005), so that its inclusion into the study seems well justified.

Finally, the study incorporates a naïve approach that does not tune parameters but always considers an identical setting for each parameter. In particular, the means of c_1, c_2, and $Vmax_{factor}$, respectively (computed over the training data) are employed. It is argued that the naïve approach resembles the way metaheuristics are commonly used today, i.e., without changing and tuning their parameters (see also Section 2). Therefore, it may be seen as a benchmark more sophisticated approaches have to beat.

4.3. Evaluation of forecasting accuracy

Assessing a forecasting model requires determining a procedure for evaluating its performance on hold-out data and selecting an appropriate measure of forecasting accuracy. Model evaluation is organized by means of a nested four-fold cross-validation in this study. That is, an available dataset is randomly partitioned into four disjoint subsets, three of which are used to build a forecasting model, whereas the remaining set facilitates an unbiased evaluation of the model on 'fresh' data (see, e.g., Kohavi, 1995). This is repeated four times, each time using a different evaluation set. Forecasting accuracy is computed as the average over the four individual model assessments.

In addition, LSSVM and REGFOR require some user parameters to be determined (e.g., the parameter γ in (8) or the number of trees in REGFOR). These are determined by means of a *grid-search* approach. Specifically, a set of candidate settings is determined from the literature for each parameter and all possible combinations are examined empirically within a second (nested) cross-validation loop. The efficacy of such a fully automatic tuning has been confirmed in several empirical benchmarks (see, e.g., Baesens et al., 2003; Lessmann, Baesens, Mues, & Pietsch, 2008; Lessmann & Voß, 2010; Van Gestel et al., 2004).

Several indicators of forecasting accuracy have been proposed to assess the predictive performance of regression models. In general, the characteristics of an indicator should match the requirements of the particular application. For example, a squared error-measure emphasizes large deviations between forecasts and true values. Since a novel forecasting application is considered in this work, no a priori knowledge concerning an accuracy indicator's suitability is available. This uncertainty is taken into account in two ways: First, multiple accuracy indicators with different characteristics (e.g., squared and absolute measures) are considered in the study. Furthermore, an integrated measure of model performance (IMP) is constructed from the base measures to obtain a more robust assessment. In particular, each model is assessed in terms of the coefficient of determination, R^2, mean squared error (MSE), root-mean squared error (RMSE), and mean absolute error (MAE). In order to compute R^2 for all models, the following definition is considered, whereby $\bar{y}$ denotes the target's mean value:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}. \tag{9}$$

It should be noted that this version of R^2 can be negative if predictions are worse than always forecasting the mean.

The IMP is constructed as follows: the scores of all forecasting models for one indicator (e.g., MSE) are normalized to the interval (0, 1), ensuring that one and zero consistently represent best and worst performance across all indicators. Consequently, the normalized values can be interpreted as the percent of 'performance' one model achieved relative to the best model. This relationship facilitates averaging across the different (base) accuracy indicators and the resulting mean represents the integrated performance measure. The feasibility of computing meaningful averages is also important for the statistical analysis (see Section 5.2).

5. Empirical results

5.1. Overview

In order to shed some light upon the main research question whether a relationship between particles' signatures and effective parameter settings exists and is sufficiently strong to facilitate forecasting parameters from signatures, Table 1 provides a high-level overview of the results obtained throughout the empirical comparison of different forecasting methods. In particular, R^2 is considered as singular measure of predictive accuracy because it represents the amount of variation of the dependent variable explained by the model. In this sense, it may be taken as an indicator of *forecasting feasibility*. Furthermore, results are based on four-fold cross-validation on the *full* dataset. This resembles a situation, where the forecasting component is not invoked until a relatively large amount of data is available, which, in turn, requires that PSO has already been executed for some time (i.e., 100 cycles in the present case). Consequently, results are expected to be optimistic. However, it is justified to begin with this setting to explore whether forecasting is feasible at all.

Table 1 indicates that linear forecasting models are very inappropriate for the considered application. Neither of the representatives explains more than 15% of the variation of the dependent variable. This is interesting since the results of LSSVM and REGFOR reveal that substantially higher levels of accuracy are achievable. In

Table 1
Top-level overview of empirical results contrasting alternative forecasting methods in terms of R^2 for predicting PSO parameters using all available data.

Forecasting method	PSO parameter			Avg. rank[b]
	c_1	c_2	$Vmax_{factor}$	
NAIVE[a]	−0.002	−0.001	−0.004	5.667
MLR	−0.018	0.107	0.134	5.333
MLR-SW	0.032	0.127	0.139	3.667
LSSVM-linear	0.031	0.127	0.147	3.333
LSSVM-radial	0.266	0.553	0.558	1.667
REGFOR	0.309	0.510	0.591	1.333

[a] Note the results differ from zero because the mean is estimated from training data, whereas performance is measured on a test dataset.

[b] The last column of Table 1 gives a method's average rank across all three target variables. That is, all methods are ordered according to R^2 in descending order and the resulting ranks (i.e., one for the best method, two for the follow-up, etc.) are averaged over the three target variables.

particular, R^2 values around 0.5, as observed for c_2 and $Vmax_{factor}$ for nonlinear regression models, evidence a semi-strong relationship between effective parameter settings and particles' signatures. Therefore, regression-based tuning appears to be a viable approach towards PSO parameterization.

However, the high-level overview does not clarify whether differences in model performance are significant in a statistical sense and whether differences in the predictability of different PSO parameters exist.

5.2. Statistical analysis

To complement previous results, a statistical analysis is carried out to scrutinize whether observed differences across regression models, PSO parameters (dependent variables) and dataset sizes are statistically significant. In particular, a mixed-model analysis of covariance (ANCOVA) is considered, with forecasting accuracy in terms of IMP as dependent variable, PSO parameter as between-subject factor (three levels), forecasting model as within-subject factor (six levels) and sample size as covariate to account for the fact that forecasting accuracy as well as model performance may depend upon the available amount of data. Note that a more detailed analysis of the sample size's effect will be presented in Section 5.3.

The results of ANCOVA are reported in Table 2.

Table 2 confirms the particular importance of the type of forecasting model for the application considered here. That is, forecasting accuracy varies significantly across different forecasting models (*F*-value: 12.945, *p*-value: 0.000) and the strength of this effect, measured in terms of partial η^2, is greater than for any other factor. Having confirmed the significance of the main effect of the factor regression model, a multiple comparison analysis may be performed to examine which models excel. In particular, the hypothesis of two regression models' mean IMP (across PSO parameters and sample sizes) being identical is tested for all possible pairs of models, using *Bonferroni's* correction to protect against elevation of Type I errors in multiple comparisons (Zar, 2007). The experiment reveals that all pairwise differences between models are highly significant (*p*-value < 0.000). For example, MLR is found to be the weakest model, whereas REGFOR is significantly better than all other techniques. Overall, these results confirm the view that selecting the 'right' forecasting model is imperative for this application of regression.

In addition to its main effect, the factor regression model shows a highly significant interaction (*p*-value $\leqslant$ 0.01) with the factor PSO parameter, whereas the very high *p*-value of the latter's main effect indicates that variations of IMP across the three PSO parameters c_1, c_2, and $Vmax_{factor}$ are most likely due to chance.

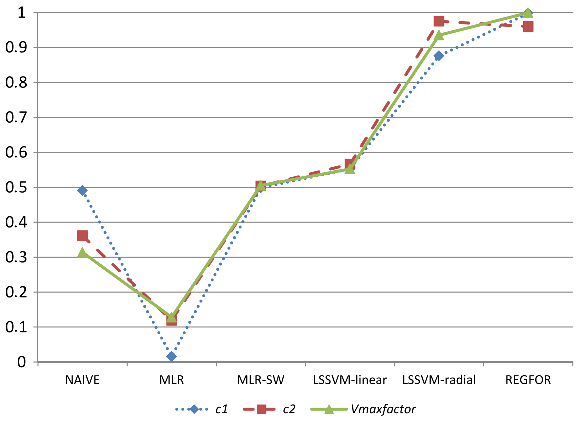

Fig. 2. Profile plot of marginal means of forecasting performance in terms of IMP across PSO parameters and sample sizes.

Contrary to Table 1, the latter result indicates that, given the independent variables considered in the models, all three PSO parameters are predictable to similar degree in general. This issue is emphasized in Fig. 2, which depicts the development of the marginal means of the two factors regression model and PSO parameter and clarifies upon their interaction. In particular, the insignificance of the factor PSO parameter is indicated by the lack of substantial vertical variation between the marginal means of c_1, c_2, and $Vmax_{factor}$ across different regression models, whereas the line's slope emphasizes that large performance differences between models exists. Finally, the fact that all three lines' development across models is relatively alike, indicates that the interaction between regression model and PSO parameter, although statistically significant (see Table 2), is not particularly pronounced (also expressed by a small partial η^2 of 0.166). However, minor deviations are detectable, e.g., in case of c_2 where the highest forecasting accuracy is achieved by LSSVM, whereas REGFOR provides the best performance for the other two PSO parameters.

Finally, ANCOVA reveals that the factor interaction between the two factors sample size and regression model is highly significant and that the sample size also possess a significant main effect. In comparisons, both of these effects are moderately strong (partial $\eta^2 = 0.680$, cf. 0.624). The importance of the sample size (i.e., the amount of data available to build a forecasting model) is intuitive in the sense that constructing models from larger datasets is well known to improve forecasting accuracy. More specifically, the significant factor interaction reveals that the models considered in the study differ in terms of their dependence upon data (i.e., need more or less data to provide accurate forecasts). As explained in more detail below, a model's robustness/sensitivity towards dataset size is of imperative importance within the considered application. Therefore, the effect of sample size is explored in detail in the following section.

5.3. Learning behavior of forecasting models

Previous results have confirmed the feasibility of employing forecasting models to determine parameters of PSO. However, this approach requires a dataset of covariates (e.g., characteristics of

Table 2
Results of mixed-model ANCOVA on forecasting accuracy in terms of IMP with factors PSO parameter, sample size and regression model.

Factor	Sum of squares	Degrees of freedom	Mean squares	F[a]	*p*-Value	Partial η^2
Regression model (RM)	16.970	1.311	12.945	311.632	0.000	0.834
Sample size (SS)[b]	5.411	1	5.411	102.689	0.000	0.624
PSO parameter (PP)	0.007	2	0.003	0.063	0.939	0.002
RM * SS	7.175	1.311	5.473	131.759	0.000	0.680
RM * PP	0.670	2.622	0.255	6.151	0.001	0.166

[a] Degrees of freedom have been adapted by means of a *Greenhouse-Geisser* correction to account for the fact that the assumption of *sphericity* is violated for the results of this study, which has been detected by employing *Mauchly's* test (Zar, 2007).

[b] The single degree of freedom is explained by the fact that results are averaged over all sample sizes in the ANCOVA analysis.

the underlying optimization problem) and appropriate parameter settings (i.e., dependent variables to be forecasted). Since such data has to be gathered on-the-fly during initial iterations of the PSO algorithm, the sensitivity of a forecasting model towards dataset size is of pivotal importance: the less data is needed, the earlier the model can be employed to govern PSO parameterization.

The objective of the following experiment is therefore to appraise the influence of dataset size on forecasting accuracy. To that end, a learning curve analysis (Perlich, Provost, Simonoff, & Cohen, 2003; Weiss & Tian, 2008) is undertaken that iteratively adds additional records to the dataset and examines the accuracy of forecasting models derived from this data by means of cross-validation. Thus, the learning curve analysis simulates a real-world application of the proposed approach where blocks of new data are made available over time through PSO iterations and facilitate recalibrating the forecasting model.

Fig. 3 depicts the learning curves for c_1, c_2, and $Vmax_{factor}$ in terms of IMP. It should be remembered that the IMP statistic measures the relative performance of a forecasting model compared to the best model. In this sense, Fig. 3 reveals that REGFOR is indeed the most suitable model for the considered application: It outperforms all other methods across all dataset sizes and PSO parameters by several orders of magnitude. This model not only provides highest forecasting accuracy on average but is also least sensitive towards dataset size.

The IMP measure is suitable for comparing the relative performance of alternative regression models. However, it does not facilitate any conclusions regarding the achievable (absolute) level of forecasting accuracy to be drawn. That is, although REGFOR is identified as (relatively) best model, further experimentation is needed to clarify whether its forecasts appear sufficiently accurate. Clearly, the question what degree of accuracy is deemed *sufficient* is difficult to answer. A forecasting error is a deviation between a dependent variable's true value and its estimate as computed by the regression model. In the present case, a forecasting error represents a situation where the regression-based tuning approach would set a particle's parameters to values that differ from those known to eventually increase its fitness within the next update. Therefore, it seems reasonable to measure the magnitude of an error in relation to the value range of the dependent variable. Specifically, it can be expected that setting a PSO parameter to, e.g., 4 if the true value is 5 has a more severe effect on the next particle update than setting it to 99 if the true value is 100, although the respective (absolute) forecasting error is one in both cases. Consequently, a relative error measure should be employed when examining forecasting models' accuracy.

To shed some light upon the important question whether REGFOR (and other models) deliver *sufficient* accuracy, the previous learning curve analysis is repeated in terms of the mean absolute percentage error (MAPE) and respective results are shown in Fig. 4. Note that MLR has been excluded since its performance has already been shown to be strongly inferior to alternative approaches and using fewer models improves the readability of the figure. Remember that Fig. 4, in contrast to Fig. 3, depicts an error measure, so that lower values indicate higher forecasting accuracy.

Fig. 4 amends the previous analysis by providing a clearer view on the absolute importance of dataset size. That is, the oscillations on the left-hand side of the learning curves indicate that all forecasting models suffer from the yet limited amount of data, i.e., show somewhat unstable performance. For example, approximately 1,750 examples are needed for forecasts of c_1 to stabilize, whereas predictions of c_2 exhibit variation even if dataset sizes are increased beyond 2,000. On the contrary, estimates of $Vmax_{factor}$ show high variation only for small datasets of less than ~600 examples.

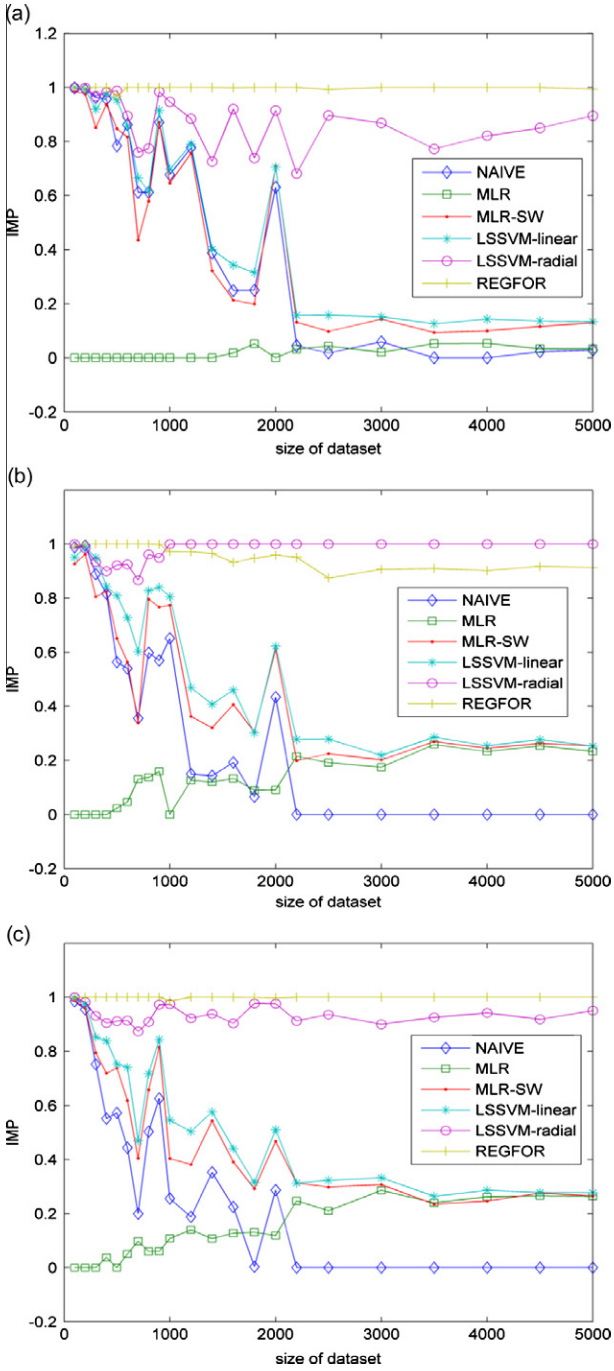

Fig. 3. Forecasting accuracy in terms of IMP for candidate regression models across increasing dataset sizes for PSO parameters c_1 (a), c_2 (b) and $Vmax_{factor}$ (c).

Fig. 4 once more confirms that REGFOR and LSSVM perform consistently better than their linear counterparts (see also Fig. 3). However, another insight of this analysis is that these models also benefit from larger datasets to a higher degree. That is, forecasting errors of REGFOR and LSSVM continue to decrease when increasing dataset size beyond, e.g., 3000, whereas the linear models remain on their level of accuracy. Consequently, one may speculate that a limited amount of data suffices to capture the linear relationships between particles' signatures and effective parameter values. In addition, more complex, nonlinear relationships can be distilled from the data if the regression models possess sufficient expressive power.

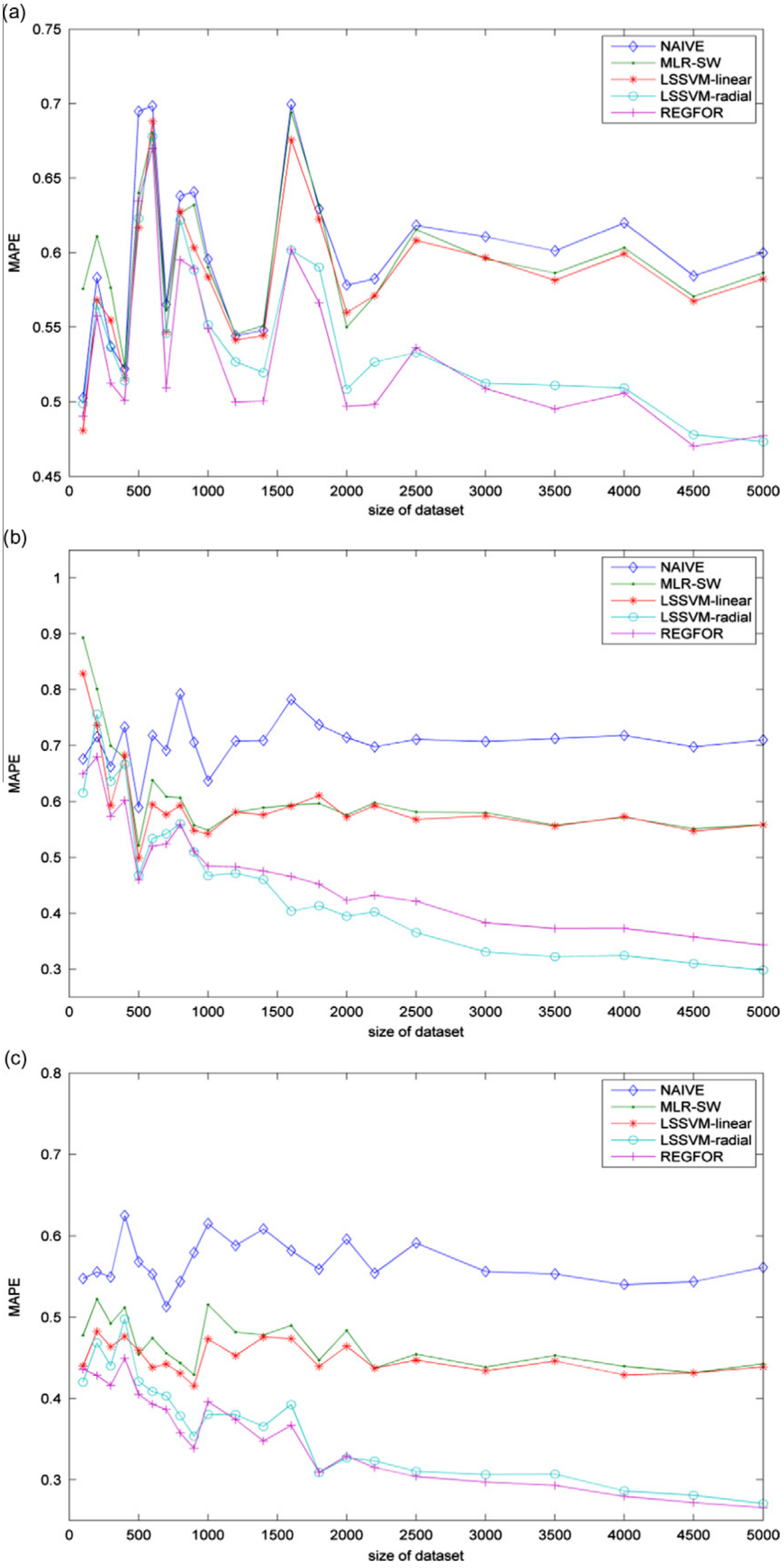

Fig. 4. Forecasting accuracy in terms of MAPE for candidate regression models across increasing dataset sizes for PSO parameters c_1 (a), c_2 (b) and $Vmax_{factor}$ (c).

Regarding the level of accuracy, the best forecasts deviate from true values by roughly 30% on average. However, in view of the fact that neither REGFOR nor LSSVM have reached their saturation level (i.e., the learning curve shows a negative slope at dataset size of 5000), further reductions of MAPE seem attainable. Moreover, all models perform much better than the naïve benchmark of using the same setting for a PSO parameter in all iterations, which may be taken as evidence for the appropriateness of the regression-based approach.

5.4. Summary and discussion

The empirical study confirms that the effectiveness of parameter settings to improve a particle's fitness is connected to its signature; i.e., characteristics of the optimization problem to be solved. This relationship embodies potentially valuable information that is ignored when PSO parameters are set to fixed values, which are not changed throughout the course of the algorithm. Moreover, it has been shown that it is well feasible to extract this information by means of (nonlinear) regression. Therefore, it seems feasible to improve PSO (e.g., accelerate the search, achieve better solutions, etc.) if updates of particles take into account information concerning their present status. This is what the regression-based tuning approach strives to achieve.

Concerning different candidate models, strong evidence for the need of nonlinear models has been provided. In particular, REGFOR appears most suitable and has outperformed alternative techniques under a variety of conditions. As expected, standard linear regression has been found to be infeasible. In particular, the inability of MLR to model the relationship between effective parameter values and signatures can be attributed to the large number of independent variables that characterize this type of application. Regularized regression models like SWMLR or L-LSSVM can overcome this obstacle. The concrete approach to cope with high dimensionality (i.e., performing variable selection as in SWMLR or penalizing complex models as in L-LSSVM) does not seem to be of much importance since both methods achieve comparable forecasting accuracy. However, their performance remains clearly inferior to nonlinear technique with more expressive power.

Finally, the results suggest – to some extent – that certain parameters are more difficult to forecast than others. Although this could not be confirmed within the statistical analysis in general, Fig. 4 indicates that estimates of $Vmax_{factor}$ require the least amount of data. With respect to the considered application, this feature represents a major advantage.

6. Limitations and further research

A critical reflection of obtained results, their external validity and an experiment's possible limitations is a critical part of empirical research. This study has limited its attention to one particular application of PSO and examined one respective dataset. Therefore, future research is needed to clarify whether the viability of estimating effective parameter values from characteristics of the optimization problem generalizes to other decision problems and other metaheuristics. Since metaheuristics generally solve a specific optimization problem iteratively, it is always possible to collect data concerning past iterations of the algorithm and also problem characteristics. In this sense, the proposed approach is generic. However, empirical evidence obtained in conjunction with other heuristics/problems is desirable to replicate and confirm previous findings.

The need of a datasets of covariates (characteristics of the underlying optimization problem) and effective parameter settings (dependent variables to be forecasted), may be seen as a general limitation of the regression-based tuning approach. However, our view is that the online learning setting – simulated by means of a learning curve analysis in this study – is well suited for coping with this requirement. The fact that metaheuristics are commonly employed without any formal tuning approach today and that this has not hindered their widespread success in various applications indicates that the need for an initialization period does not represent a major obstacle. The responsibility of calibrating parameters can be transferred to forecasting agents once an initial dataset has been collected during the early cycles of the algorithm. Previous results have demonstrated that the regression models considered here are moderate in terms of their data requirements. Consequently, the hope that predictive modeling enables a more structured and effective approach towards tuning metaheuristics' parameters – at least for later iterations/cycles – seems well justified.

This study has taken a forecasting perspective and contrasted the competitive performance of different candidate models to appraise forecasting feasibility. However, it is crucial to integrate a regression method into PSO – or other metaheuristics – to confirm that high predictive accuracy truly converts into better parameter settings. That is, a prototype of the proposed system needs to be developed to verify that parameters set through regression agents are indeed better than values determined by some conventional approach. The present study supports this task by providing strong evidence for the appropriateness of REGFOR.

Finally, an important assumption of this experiment is that parameter settings which lead to an improvement of a particle's fitness can be considered *effective*. It has to be remembered that the study embraces three PSO parameters, which have been examined independently. Consequently, it cannot be guaranteed that a fitness increase can be attributed to all three parameters. That is, the move's success might have been due to only one or two parameters being set to suitable values, whereas the other setting has been less appropriate. Such cases would impede regression and bias results. On the one hand, this risk once more emphasizes the importance of developing and examining a prototype of the proposed system in future research to gain a deeper understanding of the effects of regression-based parameters on PSO, the complex nature between signatures and parameters, and fully understand which parameter settings are effective and why. On the other hand, one could attempt a joint estimation of multiple dependent variables (e.g., the three PSO parameters c_1, c_2, and $Vmax_{factor}$) instead of considering them independently. This would require the prediction of a *structured output* (i.e., an independent 'variable' that is not scalar) and respective algorithms have recently been developed in the machine learning community (see, e.g., Joachims, Finley, & Yu, 2009; Tsochantaridis, Joachims, & Hofmann, 2005). Exploring their potential for tuning metaheuristics appears to be another fruitful avenue for future research.

7. Conclusions

The paper offers a novel perspective on the task of tuning parameter settings in metaheuristics. Employing information concerning the particular decision task and the heuristic's past search operations, an automated tuning approach on the basis of regression has been proposed. The approach is generic in principle and has been evaluated empirically in conjunction with the PSO algorithm and one particular problem instance. The empirical results indicate that the efficacy of parameter settings can indeed be related to problem-specific information (i.e., particle's signatures), so that a regression-based tuning approach appears to be promising in general. An important result of the experiment is that this particular application of regression requires nonlinear forecasting techniques, whereas linear regression, considered in previous work on metaheuristic tuning, is less suitable. In particular, the REGFOR procedure has been identified as most promising approach, so that subsequent work to integrate PSO with a regression-based tuning agent should initially consider this type of model.

References

Adenso-Diaz, B., & Laguna, M. (2006). Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research, 54*, 99–114.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*(6), 627–635.

Battiti, R., Brunato, M., & Mascia, F. (2009). *Reactive search and intelligent optimization*. New York: Wiley.

Beyer, H.-G., & Schwefel, H.-P. (2002). Evolution strategies – A comprehensive introduction. *Natural Computing, 1*(1), 3–52.

Bhattacharyya, S. (1999). Direct marketing performance modeling using genetic algorithms. *INFORMS Journal on Computing, 11*(3), 248–257.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys, 35*, 268–308.

Boer, P. D., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research, 134*, 19–67.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Belmont: Wadsworth.

Buckinx, W., & Van den Poel, D. (2005). Customer base analysis: Partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting. *European Journal of Operational Research, 164*(1), 252–268.

Buckinx, W., Verstraeten, G., & Van den Poel, D. (2007). Predicting customer loyalty using the internal transactional database. *Expert Systems with Applications, 32*(1), 125–134.

Burez, J., & Van den Poel, D. (2007). CRM at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services. *Expert Systems with Applications, 32*(2), 277–288.

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications, 36*(3, Part 1), 4626–4636.

Caserta, M., & Rico, E. Q. O. (2009). A cross entropy-Lagrangian hybrid algorithm for the multi-item capacitated lot sizing problem with setup times. *Computers and Operations Research, 36*(2), 530–548.

Caserta, M., & Voß, S. (2009). Metaheuristics: Intelligent problem solving. In V. Maniezzo, T. Stützle, & S. Voß (Eds.), *Matheuristics: Hybridizing metaheuristics and mathematical programming* (pp. 1–38). Berlin: Springer.

Cesa-Bianchi, N., Conconi, A., & Gentile, C. (2002). On the generalization ability of on-line learning algorithms. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.). *Advances in neural information processing systems* (Vol. 14, pp. 59–366). Cambridge: MIT Press.

Chen, L.-H., & Hsiao, H.-D. (2008). Feature selection to diagnose a business crisis by using a real GA-based support vector machine: An empirical study. *Expert Systems with Applications, 35*(3), 1145–1155.

Coussement, K., & Poel, D. V. d. (2009). Improving customer attrition prediction by integrating emotions from client/company interaction emails and evaluating multiple classifiers. *Expert Systems with Applications, 36*(3, Part 2), 6127–6134.

Coussement, K., & Van den Poel, D. (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications, 34*(1), 313–327.

Coy, S. P., Golden, B. L., Rungen, G. C., & Wasil, E. A. (2001). Using experimental design to find effective parameter settings for heuristics. *Journal of Heuristics, 7*, 77–97.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press.

Dorigo, M., & Stützle, T. (2005). *Ant colony optimization*. Cambridge: MIT Press.

Fei, S.-W., Liu, C.-L., & Miao, Y.-B. (2009). Support vector machine with genetic algorithm for forecasting of key-gas ratios in oil-immersed transformer. *Expert Systems with Applications, 36*(3), 6326–6331.

Friedrichs, F., & Igel, C. (2005). Evolutionary tuning of multiple SVM parameters. *Neurocomputing, 64*, 107–117.

Gijsberts, A., Metta, G., & Rothkrantz, L. (2010). Evolutionary optimization of least-squares support vector machines. In R. Stahlbock, S. F. Crone, & S. Lessmann (Eds.), *Data mining* (pp. 277–297). Berlin: Springer.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research, 13*(5), 533–549.

Glover, F., & Laguna, M. (1997). *Tabu search*. Boston: Kluwer.

Glover, F. W., & Kochenberger, G. A. (2003). *Handbook of metaheuristics*. Boston: Kluwer.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading: Addison-Wesley.

Guo, X. C., Yang, J. H., Wu, C. G., Wang, C. Y., & Liang, Y. C. (2008). A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. *Neurocomputing, 71*(16–18), 3211–3215.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning* (2nd ed.). New York: Wiley.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.

Hou, S., & Li, Y. (2009). Short-term fault prediction based on support vector machines with parameter optimization by evolution strategy. *Expert Systems with Applications, 36*(10), 12383–12391.

Huang, C.-L., Chen, M.-C., & Wang, C.-J. (2006). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*(4), 847–856.

Huang, C.-L., & Wang, C.-J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications, 31*(2), 231–240.

Izenman, A. J. (2008). *Modern multivariate statistical techniques*. Berlin: Springer.

Izquierdo, J., Pérez, R., & Iglesias, P. L. (2004). Mathematical models and methods in the water industry. *Mathematical and Computer Modelling, 39*(11–12), 1353–1374.

Jin, Y. X., Cheng, H. Z., Yan, J. Y., & Zhang, L. (2007). New discrete method for particle swarm optimization and its application in transmission network expansion planning. *Electric Power Systems Research, 77*(3–4), 227–233.

Joachims, T., Finley, T., & Yu, C.-N. (2009). Cutting-plane training of structural SVMs. *Machine Learning, 77*(1), 27–59.

Johansson, U., König, R., & Niklasson, L> (2010). Genetically Evolved kNN Ensembles. In R. Stahlbock, S. F. Crone, & S. Lessmann (Eds.), *Data mining* (pp. 299–313). Berlin: Springer.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the international conference on neural networks* (pp. 1942–1948). IEEE Computer Society.

Kim, Y.-W., & Oh, I.-S. (2008). Classifier ensemble selection using hybrid genetic algorithms. *Pattern Recognition Letters, 29*(6), 796–802.

Kim, Y. S., Street, W. N., Russell, G. J., & Menczer, F. (2005). Customer targeting: A neural network approach guided by genetic algorithms. *Management Science, 51*(2), 264–276.

Kirkpatrick, S., Jr., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In C. S. Mellish (Ed.), *Proceedings of the 14th international joint conference on artificial intelligence* (pp. 1137–1143). San Francisco: Morgan Kaufmann.

Lariviere, B., & Van den Poel, D. (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications, 29*(2), 472–484.

Lessmann, S., Baesens, B., Mues, C., & Pietsch, S. (2008). Benchmarking classification models for software defect prediction: A proposed framework and novel findings. *IEEE Transactions on Software Engineering, 34*(4), 485–496.

Lessmann, S., & Voß, S. (2009). A reference model for customer-centric data mining with support vector machines. *European Journal of Operational Research, 199*(2), 520–530.

Lessmann, S., & Voß, S. (2010). Customer-centric decision support: A benchmarking study of novel versus established classification models. *Business and Information Systems Engineering, 2*(2), S. 79–S. 93.

Lin, S.-W., Shiue, Y.-R., Chen, S.-C., & Cheng, H.-M. (2009). Applying enhanced data mining approaches in predicting bank performance. A case of Taiwanese commercial banks. *Expert Systems with Applications, 36*(9), 11543–11551.

Martens, D., De Backer, M., Haesen, R., Vanthienen, J. A. V. J., Snoeck, M. A. S. M., & Baesens, B. A. B. B. (2007). Classification with ant colony optimization. *IEEE Transactions on Evolutionary Computation, 11*(5), 651–665.

Meiri, R., & Zahavi, J. (2006). Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research, 171*(3), 842–858.

Meisel, S., & Mattfeld, D. (2010). Synergies of operations research and data mining. *European Journal of Operational Research, 20*(1), S. 1–S. 10.

Min, S.-H., Lee, J., & Han, I. (2006). Hybrid genetic algorithms and support vector machines for bankruptcy prediction. *Expert Systems with Applications, 31*(3), 652–660.

Montalvo, I., Izquierdo, J., Pérez, R., & Iglesias, P. L. (2008). A diversity-enriched variant of discrete PSO applied to the design of water distribution networks. *Engineering Optimization, 40*(7), 655–668.

Montalvo, I., Izquierdo, J., Perez, R., & Tung, M. M. (2008). Particle swarm optimization applied to the design of water supply systems. *Computers and Mathematics with Applications, 56*(3), 769–776.

Nanni, L., & Lumini, A. (2009). Particle swarm optimization for ensembling generation for evidential k-nearest-neighbour classifier. *Neural Computing and Applications, 18*(2), 105–108.

Orsenigo, C., & Vercellis, C. (2004). Discrete support vector decision trees via tabu search. *Computational Statistics and Data Analysis, 47*(2), 311–322.

Pacheco, J., Casado, S., & Núñez, L. (2009). A variable selection method based on Tabu Search for logistic regression models. *European Journal of Operational Research, 199*(2), 506–511.

Park, M. W., & Kim, Y. D. (1998). A systematic procedure for setting parameters in simulated annealing algorithms. *Computers and Operations Research, 25*(3), 207–217.

Parson, R., & Johnson, M. E. (1997). A case study in experimental design applied to genetic algorithms with applications to DNA sequence assembly. *American Journal of Mathematical and Management Sciences, 17*(3), 369–396.

Pendharkar, P. C. (2001). An empirical study of design and testing of hybrid evolutionary-neural approach for classification. *Omega, 29*(4), 361–374.

Pendharkar, P. C., & Rodger, J. A. (2004). An empirical study of impact of crossover operators on the performance of non-binary genetic algorithm based neural approaches for classification. *Computers and Operations Research, 31*(4), 481–498.

Perlich, C., Provost, F., Simonoff, J. S., & Cohen, W. W. (2003). Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research, 4*(2), 211–255.

Shalev-Shwartz, S., & Singer, Y. (2007). A primal–dual perspective of online learning algorithms. *Machine Learning, 69*(2), 115–142.

Sharpe, P. K., & Glover, R. P. (1999). Efficient GA based techniques for classification. *Applied Intelligence, 11*(3), 277–284.

Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. In *Proceedings of the international conference on evolutionary computation* (pp. 69–73). Piscataway: IEEE Computer Society.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199–222.

Stoean, R., Preuss, M., Stoean, C., El-Darzi, E., & Dumitrescu, D. (2008). Support vector machine learning with an evolutionary engine. *Journal of the Operational Research Society, 60*(8), 1116–1122.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., DeMoor, B., & Vandewalle, J. (2002). *Least squares support vector machines*. Singapore: World Scientific.

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters, 9*(3), 293–300.

Tang, X., Zhuang, L., & Jiang, C. (2009). Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization. *Expert Systems with Applications, 36*(9), 11853–11857.

Tosun, A., Turhan, B., & Bener, A. B. (2009). Feature weighting heuristics for analogy-based effort estimation models. *Expert Systems with Applications, 36*(7), 10325–10333.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research, 6*, 1453–1484.

Üstün, B., Melssen, W. J., Oudenhuijzen, M., & Buydens, L. M. C. (2005). Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization. *Analytica Chimica Acta, 544*(1–2), 292–305.

Van Gestel, T., Suykens, J. A. K., Baesens, B., Viaene, S., Vanthienen, J., & Dedene, G. (2004). Benchmarking least squares support vector machine classifiers. *Machine Learning, 54*(1), 5–32.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Wiley.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

Weiss, G., & Tian, Y. (2008). Maximizing classifier utility when there are data acquisition and modeling costs. *Data Mining and Knowledge Discovery, 17*(2), 253–282.

Wu, C.-H., Tzeng, G.-H., Goo, Y.-J., & Fang, W.-C. (2007). A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert Systems with Applications, 32*(2), 397–408.

Xu, J., Chiu, S. Y., & Glover, F. (1998). Fine-tuning a tabu search algorithm with statistical tests. *International Transactions in Operational Research, 5*(3), 233–244.

Zar, J. H. (2007). *Biostatistical analysis* (5th ed.). Upper Saddle River: Prentice-Hall.