UNIVERSIDADE FEDERAL DE PERNAMBUCO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

Models for Quantifying Risk and Reliability Metrics via Metaheuristics and Support Vector Machines

Isis Didier Lins

Orientador: Enrique Andrés López Droguett, PhD

Recife, 2013

UNIVERSIDADE FEDERAL DE PERNAMBUCO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

Models for Quantifying Risk and Reliability Metrics via Metaheuristics and Support Vector Machines

A thesis

by

Isis Didier Lins

Orientador: Enrique Andrés López Droguett, PhD

Recife, February/2013

UNIVERSIDADE FEDERAL DE PERNAMBUCO

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

Models for Quantifying Risk and Reliability Metrics via Metaheuristics and Support Vector Machines

A thesis presented to the Universidade Federal de Pernambuco in partial fulfillment of the requirements for the degree of Doutor

by

Isis Didier Lins

Orientador: Enrique Andrés López Droguett, PhD

Recife, February/2013

Catalogação na fonte
Bibliotecária: Rosineide Mesquita Gonçalves Luz / CRB4-1361 (BCTG)

L757m Lins, Isis Didier.

Models for quantifying risk and reliability metrics via metaheuristics and support vector machines / Isis Didier Lins – Recife: O Autor, 2013.
xii, 97f., il., figs., gráfs., tabs.

Orientador: Prof. Enrique Andrés López Droguett, PhD.

Tese (Doutorado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia de Produção, 2013.
Inclui Referências e Apêndice.

1. Engenharia de Produção. 2. Risco e Confiabilidade nas Indústrias de Petróleo e Gás. 3. Alocação de Redundâncias. 4. Planos de Inspeção. 5. Previsão de Falhas. 6. Algoritmos Genéticos Multiobjetivo. 7. Support Vector Machines. I. Droguett, Enrique Andrés López (Orientador). II. Título.

658.5 CDD (22.ed.) UFPE/BCTG-2013/050

UNIVERSIDADE FEDERAL DE PERNAMBUCO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

PARECER DA COMISSÃO EXAMINADORA

DE DEFESA DE TESE DE

DOUTORADO DE

ISIS DIDIER LINS

"MODELS FOR QUANTIFYING RISK AND RELIABILITY METRICS VIA METAHEURISTICS AND SUPPORT VECTOR MACHINES"

ÁREA DE CONCENTRAÇÃO: **PESQUISA OPERACIONAL**

A comissão examinadora, composta pelos professores abaixo, sob a presidência do(a) primeiro(a), considera a candidata **ISIS DIDIER LINS APROVADA.**

Recife, 27 de fevereiro de 2013.

Prof. ENRIQUE ANDRÉS LÓPEZ DROGUETT, PhD (UFPE)

Prof. FERNANDO MENEZES CAMPELLO DE SOUZA, PhD (UFPE)

Profa. ANA PAULA CABRAL SEIXAS COSTA, Doutor (UFPE)

Prof. PAULO FERNANDO FERREIRA FRUTUOSO E MELO, Doutor (COPPE/UFRJ)

Prof. MARCELO RAMOS MARTINS, Doutor (USP)

To my parents Bernardete and Sóstenes
To my husband Vicente

ACKNOWLEDGEMENTS

Finishing the Doctorate marks the end of a cycle in my life: almost 10 years have gone by dedicated to university as a student. I owe the success in Undergraduation, Master and Doctorate primarily to God. I feel He speaks to me by means of a positive, peaceful and serene interior force that made me persist in various moments of doubt.

I would like to thank:

- My mother Bernardete and my father Sóstenes for the fondness, support and incentive. In this work, my mother helped me with the English and my father helped me with enumeration techniques.
- My husband Vicente, for the love, companionship and comprehension. In these years, several leisure moments were sacrificed for study activities. Vicente also taught me the basic concepts of Interior Point methods.
- My brother Lauro that lives far away, but is always present.
- My grandparents Myriam and Lauro that wherever they are, they are certainly happy with this important step and also my grandparents Lourdes and Luiz.
- Tia Adelaide, tia Ló, tia Dulce, tia Eneida, Nadja, Niedja, Câmara, tio Alcimar, Marco Antônio, Jorginho, Adriana, Joana, João, Pedrinho, Arthur, Fernandinha, Mariana, Sofia, D. Sonia, Sr. Vicente, Ester e Giovanna for the enjoyable moments we have passed together.
- My advisor Professor Enrique López Droguett, who believed in me, for the opportunity of being part of CEERMA – UFPE.
- Márcio for some of the interesting ideas present in this work and Paulo for clarifying some issues related to Binary Decision Diagrams.
- Professor Francisco Cribari for the discussions about bootstrap methods.
- Professors Ana Paula Cabral, Fernando Campello, Marcelo Martins and Paulo Frutuoso for the valuable comments that enhanced the quality of the work.
- Alexandra, Alessandra, Thalles and Marcella for the long hours dedicated to study, works and preparation for the Qualifying.
- Ana, Miriam and Joelma for the company and support.
- Bárbara and Juliane for helping me with the bureaucracy.
- CNPq for the financial support.

For you all, my sincere Thanks!

Isis Didier Lins
Recife, March 2013

ABSTRACT

This work develops models for quantifying risk and reliability-related metrics of systems in different phases of their life cycle. For systems in the design phase, a Multi-Objective Genetic Algorithm (MOGA) is coupled with Discrete Event Simulation (DES) to provide non-dominated configurations with respect to availability and cost. The proposed MOGA + DES incorporates a Generalized Renewal Process to account for imperfect repairs and it also indicates the optimal number of maintenance teams. For the operational phase, a hybridism between MOGA and Risk-Based Inspection is proposed for the elaboration of non-dominated inspection plans in terms of risk and cost that comply with local regulations. Regression via Support Vector Machines (SVR) is applied when the reliability-related metric (response variable) of an operational system is function of a number of environmental and operational variables with unknown analytical relationship. A Particle Swarm Optimization is combined to SVR for the selection of the most relevant variables along with the tuning of the SVR hyperparameters that appear in its training problem. In order to assess the uncertainty related to the response variable, bootstrap methods are coupled with SVR to construct confidence and prediction intervals. Numerical experiments and application examples in the context of oil industry are provided. The obtained results indicate that the proposed frameworks give valuable information for budget planning and for the implementation of proper actions to avoid undesired events.

Keywords: Risk and Reliability in Oil and Gas Industries, Redundancy Allocation, Inspection Plans, Failure Prediction, Multi-objective Genetic Algorithms, Support Vector Machines.

RESUMO

Nesse trabalho são desenvolvidos modelos de quantificação de métricas de risco e confiabilidade para sistemas em diferentes etapas do ciclo de vida. Para sistemas na fase de projeto, um Algoritmo Genético Multiobjetivo (MOGA) é combinado à Simulação Discreta de Eventos (DES) a fim de prover configurações não-dominadas com relação à disponibilidade e ao custo. O MOGA + DES proposto incorpora Processos de Renovação Generalizados para modelagem de reparos imperfeitos e também indica o número ótimo de equipes de manutenção. Para a fase operacional é proposto um hibridismo entre MOGA e Inspeção Baseada no Risco para elaboração de planos de inspeção não-dominados em termos de risco e custo que atendem às normas locais. Regressão via *Support Vector Machines* (SVR) é aplicada nos casos em que a métrica relacionada à confiabilidade (variável resposta) de um sistema operacional é função de variáveis ambientais e operacionais com expressão analítica desconhecida. Otimização via Nuvens de Partículas é combinada à SVR para a seleção simultânea das variáveis explicativas mais relevantes e dos valores dos hiperparâmetros que aparecem no problema de treinamento de SVR. Com o objetivo de avaliar a incerteza relacionada à variável resposta, métodos bootstrap são combinados à SVR para a obtenção de intervalos de confiança e de previsão. São realizados experimentos numéricos e são apresentados exemplos de aplicação no contexto da indústria do petróleo. Os resultados obtidos indicam que os modelos propostos fornecem informações importantes para o planejamento de custos e para a implementação de ações apropriadas a fim de evitar eventos indesejados.

Palavras-chave: Risco e Confiabilidade nas Indústrias de Petróleo e Gás, Alocação de Redundâncias, Planos de Inspeção, Previsão de Falhas, Algoritmos Genéticos Multiobjetivo, *Support Vector Machines*.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACO Ant Colony Optimization.

ANN Artificial Neural Network.

B-P Brown-Proschan.

BDD Binary Decision Diagram.

DES Discrete Event Simulation.

ERM Empirical Risk Minimization.

GA Genetic Algorithm.

GRP Generalized Renewal Process.

IP Interior Point.

KKT Karush-Kuhn-Tucker.

MOACO Multi-objective Ant Colony Optimization.

MOGA Multi-objective Genetic Algorithm.

MSE Mean Squared Error.

MTBF Mean Time Between Failures.

MTFF Mean Time to First Failure.

MTTF Mean Time To Failure.

MTTR Mean Time To Repair.

NHPP Non-Homogeneous Poisson Process.

NPGA Niched-Pareto Genetic Algorithm.

NRMSE Normalized Root Mean Square Error.

NSGA Nondominated Sorting Genetic Algorithm.

NSGA-II Nondominated Sorting Genetic Algorithm II.

PSO Particle Swarm Optimization.

RAP Redundancy Allocation Problem.

RBI Risk-Based Inspection.

ROI Return On Investment.

RP Renewal Process.

SMO Sequential Minimal Optimization.

SPEA Strength Pareto Evolutionary Algorithm.

SRM Structural Risk Minimization.

SVM Support Vector Machine.

SVR Support Vector Regression.

TBFs Times Between Failures.

TTFs Times To Failure.

TTRs Times To Repair.

VEGA Vector Evaluated Genetic Algorithm.

1 INTRODUCTION

Risk is a measure of the potential losses due to natural or human activities and is quantitatively defined by the interaction of an event that leads to natural or artificial hazard exposure, its likelihood or frequency of occurrence and its related consequences (MODARRES, 2006). Regarding industrial activities and their associated hazards, the estimation of frequencies depends greatly on the reliability of systems' components (MODARRES *et al.*, 1999). Hence, the consideration of reliability aspects of production processes are fundamental in the assurance of system, environmental and human safety.

The occurrence of failures can lead to system shutdowns, loss of production, equipment damage. In more complex systems, *e.g.* oil and gas industries and nuclear power plants, failures can incur accidents that degrade the environment and prejudice human integrity. All these undesired effects are translated into increased costs due to system recovery, legal penalties and organization's affected image in face of society. Failure prediction modeling of systems may be conducted during various phases of their life cycle, including the concept validation and definition, the design and operation. At any stage, obtained predictions serve the purpose of anticipating the reliability behavior of the systems so as to enable the implementation of appropriate actions for their maintaining and, possibly, improvement (ZIO *et al.*, 2008). Also, those predictions allow for a cost evaluation due to maintenances and inspections, which is a valuable information for properly planning the allocation of resources to these activities.

Systems in different stages of their life cycle may demand various approaches for reliability assessment. For example, in the design phase, simulation techniques can be applied to imitate components' and system's failure-operation processes, which depend on system's logic, number of redundancies, among other characteristics. Discrete Event Simulation (DES) is a powerful tool for systems' dynamic modeling, as realistic aspects can be introduced, *e.g.* effectiveness of repairs, availability of maintenance teams, among others. Besides, system's configurations can be determined with respect to various objectives such as cost and availability, by coupling the simulation method with an optimization procedure to define, for example, the types and the number of redundancies in each subsystem. As cost and availability are ususally conflicting objectives, a multi-objective approach can be adopted. In fact, this is a variation of a Redundancy Allocation Problem (RAP), which in its traditional single-objective formulation concerns the definition of the numbers of (equal) redundancies in each subsystem that maximize overall system reliability subject to cost constraints (KUO *et al.*, 2001).

In the operational phase of an equipment, inspection activities play an important role as an integrity control technique to track the real state of the equipment often exposed to a damage mechanism. Risk-Based Inspection (RBI) (API, 2008) has been used for guiding inspection activities mainly on equipments of the petrochemical industry. Based on a predefined risk level – risk target – and on the information provided by an inspection, the actual risk to which the equipment is exposed is updated by RBI, which indicates when the next inspection should occur. However, RBI requires the risk target *a priori* and it does not consider the expenditures due to the inspection performance. In this way, as minimum cost and risk are both desired but conflicting, a multi-objective optimization emerges as an alternative to overcome the drawbacks of RBI: a risk target is not required, as risk is an objective to be minimized, and the cost due to inspections is also considered as an objective.

In multi-objective optimization, a solution that optimizes all objectives concurrently is very difficult to be reached or it does not exist. In this way, instead of having a unique solution as in single objective cases, one may obtain a set with multiple solutions. These solutions, named nondominated solutions, present a compromise among objectives and usually do not yield an optimal value for either of them individually. Once this set is obtained, the decision maker can choose any of its elements based on her preferences and then implement the selected solution.

Probabilistic optimization methods, such as Genetic Algorithms (GAs) have interesting characteristics to handle multi-objective optimization problems (DEB, 1999): (i) they are population-based, that is, many potential solutions are simultaneously considered; (ii) they permit a separated treatment of the different objectives, thus not requiring any transformations of the multiple objectives into a unique function. GAs (GOLDBERG, 1989) attempt to computationally imitate natural evolution process in which the fittest individuals are more likely to remain in population. In the optimization context, an individual is a potential solution of the considered problem and a set of individuals is the population, which evolves according to some genetic-based operators, such as selection, crossover and mutation.

Frequently, the reliability behavior of an equipment during its operational phase is influenced by a number of factors usually interdependent and an analytical model of the reliability behavior of these systems becomes impractical. For these situations, in which the underlying process that maps input – regressors, influential variables – into output – response or target variable – is not known, empirical regression via Support Vector Machines (SVMs) – Support Vector Regression (SVR) – is an effective option. SVM is a supervised learning method whose foundations stem from the statistical learning theory (VAPNIK, 2000). The training step of SVMs involves a quadratic optimization problem for which the Karush-Kuhn-Tucker first order conditions for a global optimum are necessary and sufficient, differently from other learning techniques such as Artificial Neural Networks (ANNs) that can be trapped on local minima (SCHÖLKOPF & SMOLA, 2002). Also, the SVM training objective function embodies the Structural Risk Minimization (SRM) principle, which consists in the minimization of the errors computed in the training phase and also of the errors associated with the machine capacity in accurately predicting the response variable related to input observations not in the original training data set (the so-called generalization ability). On the other hand, ANNs entail the Empirical Risk Minimization (ERM) principle, which only considers the minimization of the training error and is suitable for handling large amounts of data (VAPNIK, 2000).

The performance of SVM is highly dependent on the values of some hyperparameters that appear in the associated training problem. The simultaneous adjustment of these hyperparameters by trial and error is time-consuming and does not guarantee the achievement of good values for them. Thus, structured methods such as grid and pattern search (MOMMA & BENNETT, 2002), gradient-based methods (CHAPELLE *et al.*, 2002; CHANG & LIN, 2005; ITO & NAKANO, 2003) and heuristics such as Particle Swarm Optimization (PSO) (LIN *et al.*, 2008; FEI *et al.*, 2009; LINS *et al.*, 2010a, 2010b) and GA (PAI, 2006; CHEN, 2007) have been used to tune SVMs hyperparameters. These procedures are usually driven by performance metrics based on validation data, which is a part of the available data set not used in actual training.

Furthermore, in practice, among the many factors that are supposed to influence the response variable, only a subset of them may be important to describe its behavior. Actually, some of these potential regressors may be redundant, noisy or even irrelevant for predicting the response variable. Thus, a variable selection procedure may be applied to identify such a subset of variables. According to Guyon & Elisseeff (2003), the objectives of variable selection are to improve the prediction performance of predictors (*e.g.* SVMs), to construct faster predictors and also to provide better understanding of the underlying process that might have generated the response variable. Given that the introduction or removal of input variables change the data set, a hyperparameter tuning may be performed in order to avoid an eventual decrease in the predictive ability of SVM.

As in any regression method, once the SVR regression function is estimated, prediction becomes a straightforward task: values of the input variables are applied to the regression formula and an estimated value of the response is obtained. In this way, if the same observed values are used in the estimated model, the point estimate will be exactly the same no matter how many calculations are performed. However, it is also important to evaluate the uncertainty related to the response variable so as to provide not only accurate point estimates, but also the more informed confidence and prediction intervals, which give an idea about the precision of the quantities under consideration. In order to preserve the interesting non-parametric properties of SVR, non-parametric bootstrap methods (EFRON, 1979; EFRON & TIBSHIRANI, 1993) can be used to perform uncertainty analyses concerning the response variable.

In this section, some limitations of the techniques for reliability modeling have been presented and this work aims at solving them in the following contexts: (i) design of systems submitted to imperfect repairs considering availability and cost; (ii) elaboration of efficient inspection plans in terms of risk and cost; (iii) prediction of Times Between Failures (TBFs) via SVR models with adjusted hyperparameters and most relevant subset of regressors; (iv) construction of confidence and prediction intervals concerning failure times by means of bootstrapped SVRs. In next section, the main motivations and justifications of the work are given for each of these contexts.

1.1 Motivation and Justification

By an organizational perspective, systems without considerations about risk and reliability may cost less at a first glance, however the undesired failures' consequences in its various forms – unsatisfied demand, components' replacement, system recovery, human losses, environmental accidents – are more prone to occur, which certainly increase costs and can definitely harm the companies' activities. These effects can be reduced or even avoided by means of well performed risk and reliability analyses, which are specially important when the consequences are severe.

On the other hand, by the viewpoint of stakeholders – employees, costumers, suppliers, investors, nearby population, society, – organizations committed to offer goods and/or services originated from safe and reliable production systems seem "healthier" to support and to invest in. Thus, investments in reliable and safe processes turn into competitive advantage to the company for maintaining and increasing its market share.

In addition to these general motivations, in the following, further details on the limitations of the techniques in the areas of interest previously mentioned are provided. Some of these limitations are tackled in this work by the use of metaheuristics (multi-objective GAs, PSO), SVR and simulation methods.

1.1.1 Multi-objective Redundancy Allocation Problems

According to (KUO & WAN, 2007), it is usually difficult for a single objective to adequately describe a real problem for which an optimal design is required, thus multi-objective approaches deserve attention. In this context, Multi-objective Genetic Algorithm (MOGA) emerges as an alternative optimization procedure to tackle multi-objective RAPs. Moreover, the authors comment the existence of many unsolved topics related to redundancy allocation, including optimal design of nonrenewable systems. Indeed, the majority of the works considering repairable systems involve components with constant failure rates, thus use an underlying Exponential distribution to model their failure processes (BUSACCA *et al.*, 2001; ELEGBEDE & ADJALLAH, 2003; CHIANG & CHEN, 2007; JUANG *et al.*, 2008).

However, the hypothesis of components with constant failure rates is often non-realistic, as it does not incorporate the effects of component degradation / improvement.

Therefore, such a supposition can mislead the evaluation of some characteristics of the complete system such as system reliability. Specifically in the context of software reliability, Littlewood (2008) points out the irrelevant results provided by non-realistic assumptions upon reliability models. Jones & Hayes (2001) state that, in the context of electronic systems, the hypothesis of constant failure rates are not met in real situations and may deceive the estimation of the entire system reliability. Also, Bowles (2002) discusses and illustrates the effects of assuming constant failure rates in modeling component reliability in system design with Weibull distributions with the same Mean Time To Failure (MTTF) and different shape parameters (β), including the special case when $\beta = 1$, *i.e.* an Exponential distribution. By considering a non-repairable parallel redundant system, it is shown that if components are supposed to have a constant failure-rate but actually have either increasing or decreasing hazard functions, the complete system reliability is either overestimated or underestimated, respectively.

The works of Cantoni *et al.* (2000) and Lins & Droguett (2009) present some sophistication regarding the reliability portion of the redundancy allocation problem. The former couples GA with Monte Carlo simulation in order to obtain an optimal plant design. They tackle a single objective redundancy allocation problem in which it is desired to obtain a combination of several repairable components to be placed in a series-parallel layout in order to maximize system profit. They consider that all standby components are cold (there are no failures in standby mode) and that components failure rates are constant from the time they return from a maintenance intervention to the time of the very next failure. This means that the system deterioration during operational time is not modeled. Moreover, the authors take into account a modified Brown-Proschan (B-P) model of imperfect repairs, which is in the group of failure intensity models (DOYEN & GAUDOIN, 2004). The modified B-P approach assumes that either a minimal or a deteriorating repair is performed with respective probabilities p and $1 - p$ (*i.e.* according to a Bernoulli distribution). If a minimal repair is performed, the system returns to operation with the same condition it had before the failure occurrence (failure-rate remains the same). Otherwise, if a deteriorating repair is executed, the system returns to operation worse than it was before the failure occurrence (failure-rate is increased by a given percentage).

The other work, from Lins & Droguett (2009), tackles RAPs for repairable systems via MOGA and DES. It does not consider constant failure rates, but involves the supposition of perfect repairs, which are often not met in practice. Also, Lins & Droguett (2008) propose a combination of a Multi-objective Ant Colony Optimization (MOACO) and DES for solving RAPs involving systems with imperfect repairs modeled by a Generalized Renewal Process (GRP), but do not optimize the number of maintenance teams.

In fact, any improvement in system reliability / availability demands resources, which in turn raise the associated costs. As a consequence, a more realistic treatment of RAPs in repairable systems should consider not only the choice of components and systems configurations, but also the number of available resources. Marseguerra *et al.* (2005), for example, aim to encounter the optimal number of spare parts, which are a sort of resource, but they do not aggregate it with finding optimal system configurations.

1.1.2 Multi-objective Inspection Plans

Planning inspections involve the definition of which techniques have to be adopted and when they have to be performed in full compliance with local and specific regulations. In the context of oil and petrochemical industries, RBI has been used to support the management and scheduling of inspections.

However, in RBI methodology, the risk target is usually user defined and not an objective to be optimized. Thus, there is no guarantee about the efficiency of inspections plans elaborated based on such a risk level. Also, it does not take into account the costs associated with the inspection activities and does not suggest how resources should be allocated. For example, it does not indicate if it is better to perform a number of low cost / low-effective inspections or fewer inspections with higher effectiveness.

Since techniques of high effectiveness are usually more expensive, it is not interesting to only adopt them instead of using simpler but cheaper inspections. Additionally, regulations often provide a maximum permitted period between inspections using different techniques. In this context, a multi-objective approach emerges as an alternative to handle the conflicting objectives of risk and cost so as to create efficient inspection plans that comply with regulation standards concerning several inspection techniques. Given a planning horizon, the idea is to find the optimal compromise between risk and cost by answering two questions: (i) in each period, an inspection should or should not be performed? and (ii) if the answer of (i) is affirmative, which techniques should be used?

Depending on the number of periods considered within the planning horizon and on the quantity of inspection techniques, the number of possible inspection plans can be prohibitively large for an exhaustive evaluation of their performance. In this way, a probabilistic approach such as MOGA can be adopted for the quest of inspection plans representing the optimal compromise between risk and cost.

1.1.3 Variable Selection and Adjustment of Support Vector Regression Hyperparameters

In the SVM context, Rakotomamonjy (2003) performs variable selection for classification based on relevance criteria originated from SVM theory (weight vector and upper bound of the generalization error). The author uses a backward elimination strategy, in which the initial model involves all variables and the least promising ones are progressively eliminated (GUYON & ELISSEEFF, 2003). Search for SVM hyperparameters is not performed but the author states that more work should be devoted to the problem of hyperparameter selection in conjunction to that of variable selection.

Probabilistic heuristics such as PSO (KENNEDY & EBERHART, 1995) and GAs (GOLDBERG, 1989) are flexible to permit the simultaneous resolution of variable and SVM hyperparameter selection problems. For SVM classification, (FRÖHLICH *et al.*, 2003) and (LIN *et al.*, 2008) apply, respectively, GA and PSO methods to tackle both problems at the same time. However, GAs often require more computational effort than PSO, given the necessity of various genetic operators to mimic the evolutionary process, *e.g.* mutation, crossover, among others. The main idea of PSO is derived from the motion of groups of organisms, *e.g.* schools of fishes and flocks of birds. The swarm evolution is mainly governed by a couple of update equations concerning particles' velocities and positions in the search space. Indeed, besides the computational advantage of PSO over GA, (BABAOGLU *et al.*, 2010) report the better accuracy of PSO-based feature selection for SVM classification problems.

For SVR, Yang & Ong (2010) use a grid search method to tune SVR hyperparameters previous to variable selection by a recursive feature elimination scheme, which is an instance of backward feature elimination (GUYON *et al.*, 2002). At each step of the algorithm, after the specification of the variables to be included in the regression model, an SVR training takes place without the re-tuning of the hyperparameters. However, their most suitable values are dependent on the data set used to guide their search. Thus, the obtained hyperparameters' values over a set with all regressors do not guarantee a satisfactory SVR performance over an adjusted training data.

Wu & Wang (2009), in turn, present successive PSO algorithms for SVR hyperparameter tuning and feature selection. Firstly, a PSO involving a project pursuit technique (FRIEDMAN & TUCKEY, 1974) is performed for feature selection. Afterwards, another PSO is applied to hyperparameter selection. Thus, the feature selection is essentially a preprocessing step and the obtained results are used to feed the SVR algorithm coupled with the PSO related to hyperparameter search. In this way, the SVR prediction accuracy is not used to guide the quest for the most relevant subset of features.

Thus, given the importance of selecting the most relevant variables and of tuning of the SVR hyperparameters for its predictive ability (usually dependent on the available data set), a method for the concurrent performance of both tasks is required. In this work, a PSO is combined to SVR to tackle such a problem.

1.1.4 Uncertainty assessment via Bootstrap and Support Vector Regression

After the SVR training step, in correspondence of a new observation of the input vector x, henceforth called x_+, the estimate $\hat{y}_+$ of the true mean response $\mu_Y(\mathbf{x}_+)$ can be obtained via the adjusted regression function (*i.e.* the estimator). Besides point estimates, confidence intervals for $\mu_Y(\mathbf{x}_+)$ are also needed to account for the variability of the estimator. Furthermore, it is important to assess the uncertainty on the prediction of the response variable Y_+ itself by means of prediction intervals (MONTGOMERY *et al.*, 2006).

Given that SVR does not require any hypothesis about the distribution of the error term, the central limit theorem enables the approximation of confidence and prediction intervals when large data sets are available (BRABANTER *et al.*, 2011). On the other hand, for small numbers of data points, the intervals based on bootstrap (EFRON, 1979; EFRON & TIBSHIRANI, 1993) tend to be more accurate, given that they do not rely on asymptotic results but on the construction of the limit distribution from the available data.

The main idea of bootstrap methods is to estimate probability distributions for statistics of interest obtained from the available data. They are widely used in (generalized) linear, non-linear and nonparametric regression (DAVISON & HINKLEY, 1997). For example, in linear regression, Cribari-Neto (2004) and Cribari-Neto & Lima (2009) use bootstraped hypothesis testing and intervals tailored to account for heteroskedasticity with an estimator of the covariance matrix that considers the effects of leverage points in the design matrix. In non-parametric regression, Zio (2006), Cadini *et al.* (2008), Secchi *et al.* (2008) and Zio *et al.* (2010) analyze by bootstrap the uncertainty of ANNs predictions of nuclear process parameters. In the specific context of SVM, bootstrap approaches have been mainly applied to classification problems (ANGUITA *et al.*, 2000; TSUJITANI & TANAKA, 2011). For SVR, Lin & Weng (2004) and Yang & Ong (2010) have proposed probabilistic outputs, but assuming a probability distribution for the response variable.

Indeed, Brabanter *et al.* (2011) compares the proposed approximation intervals with bootstrap intervals, but these were only based on a residuals sampling. In this work, bootstrap methods – both pairs and residuals schemes – are combined to SVR for the construction of confidence and prediction intervals. The proposed bootstrapped SVRs are first tested on a simulated example and then applied to a real case study for the prediction of scale growth rate on metal surfaces of an equipment used in offshore oil wells.

Given the abovementioned limitations and the general ideas to overcome some of them in each context, next section gives the main and specific objectives of the thesis.

1.2 A Comprehensive Framework

In spite of the different contexts to which the proposed models are applied, the used techniques can be combined to form a comprehensive framework to evaluate risk, reliability and cost of production systems. The outcomes can guide, for example, decisions concerning the allocation of resources related to components acquisition and to inspection and maintenance activities.

In Figure 1.1, a connection among the different problems, contexts and techniques is illustrated. Imagine that, during the operational phase of a component, data related to failure, maintenance and inspection as well as data concerning operational and environmental conditions can be observed and gathered. In a first moment, these data can feed a learning method, such as SVR, in order to produce accurate reliability prediction functions with the most relevant influential variables, proper hyperparameters and uncertainty analysis. These adjusted non-parametric models can substitute the probabilistic distributions concerning components' Times To Failure (TTFs) in DES, which is used to mimic the system operational behavior in the design step.

The collected data can also enable the calculation of risk – *e.g.* via RBI if the component is from oil or petrochemical industry – and cost so as a multi-objective optimization (*e.g.* MOGA) can be used for the quest of nondominated inspection plans. The selected inspection plan provided by a post-optimization procedure can be implemented and give new inspection observations and can also feed the simulation block.

In the case of a critical component, for which failures are extremely undesired, the failure times predicted by SVR models enable the implementation of preventive actions that can restore the components to an intermediate state between "new" and "as bad as before intervention" (*e.g.* imperfect maintenance). Preventive maintenance can also be guided by inspection activities. Otherwise, if the component is not critical so that it can fail, corrective maintenance based on imperfect repairs can be adopted.

A simulation method comprising all these features replicated for several components and for different combinations among them permits the computation of availability and cost associated with various system configurations. By coupling simulation with multi-objective optimization a set of nondominated system configurations can be obtained and one of them can be chosen to be implemented. When operational, the system provides observations of each of its components and this completes the cycle.

Therefore, the proposed methods to evaluate risk and reliability during the components' operational phase can be transformed into valuable information for the design step of systems to be implemented in the future. However, the combination of all mentioned aspects into a unique model involving observation, estimation, simulation and (post-)optimization is a complex task. In this way, instead of directly handling the big problem, it is wise to divide it into parts and tackle each of them in a separate way so as to understand their particularities. Often, even the subproblems involve a great amount of complexity. Afterwards, once the methods used in each of the parts are developed and stabilized, they can be joined in order to tackle the more general problem.

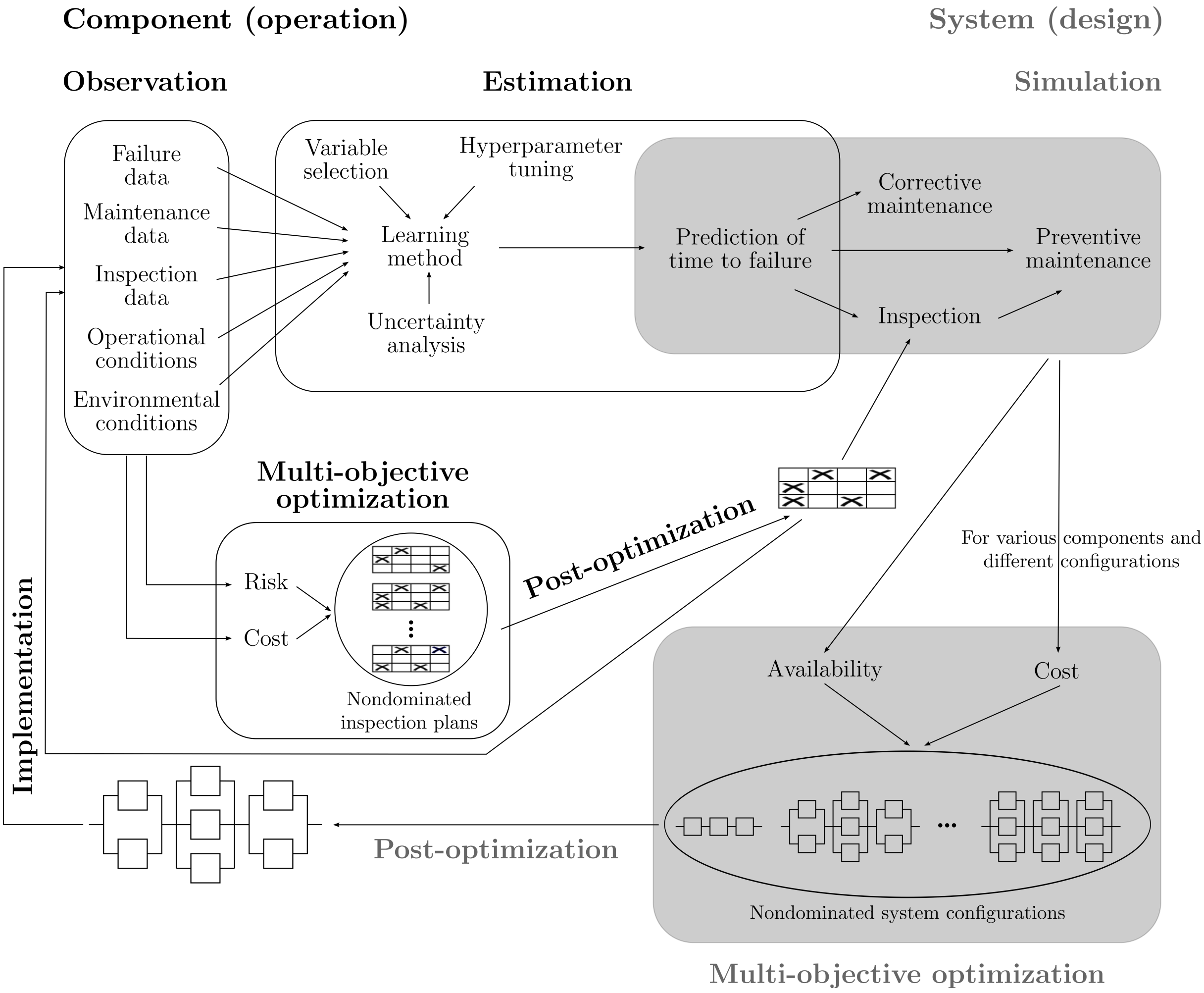


Figure 1.1: Relationships among analyzed problems

1.3 Objectives

1.3.1 Main Objective

This thesis proposes quantitative models based on metaheuristics and SVMs to assess risk and reliability-related metrics of systems in two different life cycle phases:

- Design:
  - Development of a MOGA + DES method for the indication of efficient system designs subjected to imperfect repairs in terms of both availability and cost.
- Operation:
  - Development of genetic operators for the MOGA to be combined with the RBI methodology in order to provide inspection plans representing the optimal compromise between risk and cost.

  - Development of a PSO + SVR for obtaining the subset of the most important input variables influencing the response along with the proper values for the SVR hyperparameters, so as to improve the quality of the SVR predictions.
  - Combination of SVR with bootstrap techniques for uncertainty handling in the prediction of equipment failures in order to support maintenance-related decisions.

1.3.2 Specific Objectives

The following specific objectives are defined in order to attain the main goals of the thesis:

- Implementation of the metaheuristics MOGA and PSO with the necessary adaptations for each problem.
- Implementation of SVR using Interior Point (IP) methods.
- Comparison of results obtained by the proposed MOGA for system design with the ones provided by other metaheuristics, such as MOACO, and also with real Pareto fronts.
- Performance comparison between the model considering all available input variables as regressors and the reduced model from the proposed PSO + SVR.
- Assessment of coverage properties of the different bootstrap techniques (pairs and residuals sampling) when combined to SVR.
- Application of the proposed methods for the operation phase to examples and case studies in the context of oil and gas industries.

1.4 Outline of the Thesis

Besides this introductory chapter, this thesis contains 7 additional chapters, whose contents are described as follows:

- **Chapter 2** concerns the theory underlying the two subsequent chapters involving MOGA developments. It includes: a description of multi-objective optimization; an introduction to MOGA; the main concepts of the techniques to be coupled with MOGA – GRP, DES and RBI.
- **Chapter 3** refers to the development of a MOGA and its coupling with DES for solving multi-objective RAPs with cost and availability as objective functions. Two

performance metrics for comparison between simulated and real Pareto fronts are presented and used in two validation examples of the MOGA. The proposed MOGA + DES is applied to an illustrative example with components submitted to imperfect repairs.

- **Chapter 4** is related to the development of genetic operators specific for the construction of feasible inspection plans and to the coupling of MOGA and RBI for obtaining inspection plans representing the optimal compromise between risk and cost. The proposed MOGA + RBI is applied to example involving an oil and gas separator.

- **Chapter 5** contains the theoretical background of the SVM-related chapters: an overview of SVM; a detailed description of SVR; an introduction to IP methods used to solve the SVR training problem; comments on the model and variable selection problems and a description of the PSO method used to tackle them; the general ideas concerning bootstrap methods.

- **Chapter 6** presents the combination of PSO and SVR for the simultaneous variable selection and SVR hyperparameters adjustment. The proposed PSO + SVR is applied to an example in the context of onshore oil wells.

- **Chapter 7** details the development of bootstrapped SVRs, which are validated in an artificial example and applied to a case study involving an equipment used in offshore oil wells.

- **Chapter 8** presents the main contributions and limitations of the work along with suggestions for future research.

2 THEORETICAL BACKGROUND – PART 1

This chapter contains the theoretical background used in the MOGA developments that are present in Chapters 3 and 4. The essential concepts associated with multi-objective optimization and an introduction of MOGA are provided. Also, the techniques coupled with the proposed MOGAs – GRP, DES and RBI – are described.

2.1 Multi-objective Optimization

Practical situations often require the achievement of many objectives simultaneously. In risk and reliability fields, usual goals are to minimize risk, to maximize reliability/availability, to minimize costs. The general formulation of a multi-objective optimization problem is:

$$\max_{x} \quad z = [f_1(x), \ldots, f_k(x)] \tag{2.1}$$

$$\text{s.t.} \quad g_j(x) = 0, \quad j = 1, \ldots, p, \tag{2.2}$$

$$h_j(x) \leq 0, \quad j = p+1, \ldots, q, \tag{2.3}$$

in which z is a vector formed by the objective functions $f_i(x)$, $i = 1, \ldots, k$, x is the vector of decision variables, p is the number of equality constraints $g_j(x)$ and $q-p$ is the number of inequality constraints $h_j(x)$.

Given the conflicting relation among objectives, a unique solution that optimizes all elements of z at the same time is very difficult to be found or such a solution does not even exist. In this way, instead of a unique optimal solution as in the single-objective situation, the resolution of a multi-objective problem may provide a set of solutions representing the compromise among objectives, all "optimal" according to the multi-objective viewpoint. This set, formed by *nondominated* solutions, is known as optimal Pareto set due to the Italian economist Vilfredo Pareto, who generalized the optimality concept for the multi-objective context (COELLO *et al.*, 2002).

A solution is nondominated if, for all objectives, it has a performance at least as good as the performance of the other solutions and, at least for one of the objectives, its performance overcomes the performance of the others. On the other hand, the dominance relation is mathematically defined as follows:

$$x_1 \succ x_2 \Leftrightarrow f_i(x_1) \geq f_i(x_2), \forall\, i \text{ and } f_i(x_1) > f_i(x_2) \text{ for some } i, \tag{2.4}$$

where $\succ$ denotes dominance, x_1 is a nondominated solution for a maximization problem and x_2 is a dominated solution for the same problem; f_h denotes the hth objective function. If a minimization problem is considered, the signs $\geq$ and $>$ in Equation (2.4) are replaced by $\leq$ and $<$, respectively. If one of the conditions in the right side of Equation (2.4) is not satisfied, x_1 is said to be nondominated in relation to x_2 and vice-versa. That is, for a number of objectives, x_1 overcomes the performance of x_2 and, for the remaining objectives, x_2 overcomes the performance of x_1. The nondominance relation is also observed when $x_1 = x_2$.

The concepts of local and global optimality in single-objective optimization are replaced by local optimal Pareto set ($\underline{P}$) and global optimal Pareto set ($\overline{P}$), respectively, in the multi-objective case (DEB, 1999; ZITZLER, 1999). To define these concepts, let X be the set of all x that satisfies the constraints in Equations (2.2) and (2.3), *i.e.*, the feasible set:

- Local optimal Pareto set ($\underline{P}$): $\forall x \in \underline{P}, \quad \nexists \;\; x' \in X$ satisfying $||x' - x|| < \varepsilon$ which dominates any member of $\underline{P}$ ($|| \cdot ||$ is the Euclidean distance between two points and $\varepsilon > 0$).

- Global optimal Pareto set ($\overline{P}$): $\forall x \in \overline{P}, \quad \nexists \;\; x' \in X$ such that $x' \succ x$.

If the solutions in $\underline{P}$ are substituted in the objective functions, the local Pareto front ($F\underline{P}$) is obtained (Equation (2.5)); following the same reasoning for the solutions of $\overline{P}$, the global Pareto front $F\overline{P}$ is found (Equation (2.6)):

$$F\underline{P} = \{f_1(x), \ldots, f_k(x)), \forall\, x \in \underline{P}\}, \tag{2.5}$$

$$F\overline{P} = \{f_1(x), \ldots, f_k(x)), \forall\, x \in \overline{P}\}. \tag{2.6}$$

Figure 2.1 illustrates the mapping of solutions for a problem with two maximization objectives, $f_1(x)$ and $f_2(x)$, and two decision variables, *i.e.* $x = (x_1, x_2)$. In the search space, the light gray ellipse and the elements within it represent the feasible set X. The black circle with locally nondominated pairs (x_1,x_2) in the same space is the local optimal Pareto set ($\underline{P}$) that is mapped into the local Pareto front (black line) in the objective space. Similarly, the dark gray circle ($\overline{P}$) containing the globally nondominated pairs (x_1,x_2) in the search space is mapped into the global optimal Pareto front (dark gray line) in the objective space. Note that the dominated pairs in the search space produces dominated Pareto fronts in the objective space (in light gray).

As multiple solutions are available, the multi-objective approach involves two distinct phases: (i) the search for potential solutions and (ii) the decision making associated to the selection of a single solution representing the *adequate* compromise among objectives. The manner these two phases are combined determines the classification of the multi-objective methods into three categories (ZITZLER, 1999):

- Decision making before search: the objectives are aggregated into a unique objective function that implicitly involves information about the decision maker preferences.

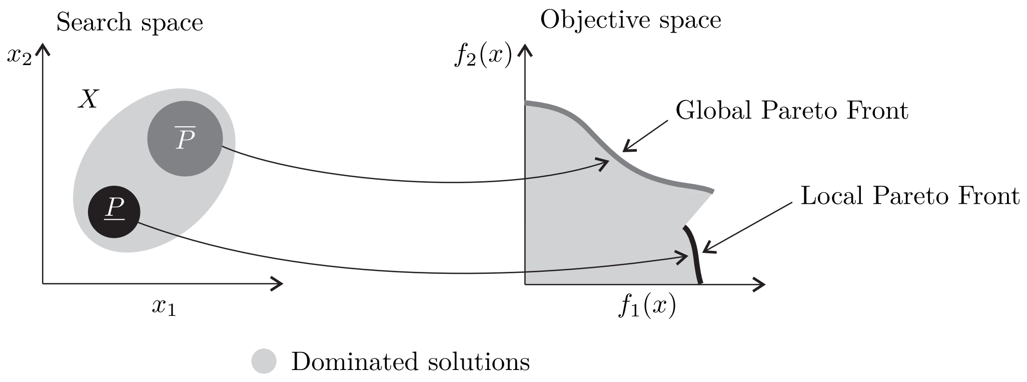


Figure 2.1: Mapping of solutions from search space to objective space: local and global Pareto sets into local and global Pareto fronts. (Adapted from Zitzler (1999), p. 9)

The transformation of a multi-objective problem into a single-objective one enables the use of techniques devised for the resolution of single-objective optimization problems. However, the incorporation of the decision maker preferences, usually in the form of weights, is a relevant step and, in general, demands elicitation methods.

- Decision making during search: the decision maker provides information about her preferences during the search procedure; at each step they are used in the determination of new potential solutions. The methods within this category are called "interactive"; they require the decision maker availability for direct participation in the search process and that she is familiar with the used elicitation and optimization procedures as well.
- Decision making after search: the optimization process is performed without any preference information. As a result, a set of solutions, – ideally Pareto optimal – is obtained and then the decision maker finally chooses one of them according to her own preferences. Sometimes, the obtained set of solutions is very large, which turns the selection of a unique solution into a difficult task. However, the decision-making after search avoid eventual problems with decision makers non-familiarized with elicitation and optimization processes required in the previous categories.

The outcome of the methods of the third category is a set of nondominated solutions. Then another problem arises: given that all solutions are similar from a multi-objective perspective, which of them the decision-maker must choose? She may select it according to her own preferences, but such a task often becomes a challenge in practical situations, specially when they involve budget and safety-related issues. In order to support decision-making for a given Pareto set, Taboada *et al.* (2007) suggested data clustering techniques in order to reduce the number of solutions of the Pareto set: the main idea is to form groups internally homogeneous externally heterogeneous, to take a representative solution of each group and to select the most relevant one – such a solution is likely to be related to the “knee” region of the Pareto front, in which small deterioration in one objective, *e.g.* cost, turns into high improvements on the other *e.g.* reliability.

Santos *et al.* (2010), Lins *et al.* (2011a, 2013) present an interesting game theory approach to support decisions related to the design of security systems subjected to purposeful attacks. Given the Pareto solutions, the equilibria of a sequential game involving preferences of both agents (defender and attacker) are the selected solutions. In this work, however, as only deterioration processes are taken into account, Return On Investment (ROI) analyses are performed as in Lins & Droguett (2009). The main objective is to provide the decision-maker with information about how investements (higher costs) are translated into gains (reduction) in avaialability (risk). ROI analyses are presented in Sections 3.6.2.1 and 4.5.1.

The two main traditional approaches for multi-objective optimization are the Weighted Sum and the ε-Perturbation methods (DEB, 1999; COELLO *et al.*, 2002). Both of them transform the multi-objective problem into a single-objective one. The Weighted Sum Method makes use of weights to aggregate all the objective functions into a single objective-function. The weights do not represent the relative importance among objectives: they are only factors that may be altered to locate different points in the optimal Pareto set. One of the main drawbacks of the Weighted Sum Method is the fact that it cannot find solutions in non-convex regions of the Pareto front if the Pareto front is non-convex (MESSAC *et al.*, 2000).

In the ε-Perturbation Method, an objective is arbitrarily chosen (or the one considered as the most important) and optimized as the others become constraints that must satisfy acceptable levels previously defined. Different Pareto optimal solutions are found by varying the acceptable levels. The choice of the acceptable levels demands a preliminary analysis since inappropriate values can result in empty feasible sets (DEB, 1999).

In addition, both traditional methods not only require multiple runs in order to obtain different optimal Pareto solutions (it is expected that they are indeed optimal Pareto solutions). Alternatively, nature-based algorithms such as GAs and Ant Colony Optimization (ACO) can be used to resolve multi-objective problems. These algorithms overcome some of the drawbacks of the above-mentioned traditional methods: they handle many potential solutions simultaneously, which allow the achievement of different Pareto solutions in a single run of the algorithm and do not impose any requirements regarding the convexity of the Pareto front.

2.1.1 Multi-objective Genetic Algorithms

GAs attempt to computationally mimic the natural evolution process by the use of genetic operators such as selection, crossover and mutation. They are mainly applied in optimization problems that have some characteristics not accepted by traditional methods of mathematical programming and are within the group of probabilistic optimization methods. In addition to the fact that more than one objective is taken into account, the main difference between the single objective GA and the multi-objective GA is the selection phase. In the multi-objective case, the concept of dominance is directly or indirectly incorporated in that step. Table 2.1 lists the main multi-objective methodologies involving evolutionary algorithms. For further details in single and/or multi-objective GA, see Goldberg (1989), Michalewicz (1996), Deb (1999), Messac *et al.* (2000) and Coello *et al.* (2002).

Table 2.1: Main methods of multi-objective optimization via evolutionary algorithms

Method	**Author(s) and date**
Vector Evaluated Genetic Algorithm (VEGA)	Schaffer (1985)
MOGA	Fonseca & Fleming (1993)
Niched-Pareto Genetic Algorithm (NPGA)	Horn *et al.* (1994)
Nondominated Sorting Genetic Algorithm (NSGA)	Srinivas & Deb (1994)
Strength Pareto Evolutionary Algorithm (SPEA)	Zitzler (1999)
Nondominated Sorting Genetic Algorithm II (NSGA-II)	Deb *et al.* (2002)

In this work, integer-coded MOGAs are adopted and they do not make use of elaborated fitness metrics; hence each individual has an associated fitness vector with size equal to the number of the considered objectives. One MOGA is coupled with DES to solve a multi-objective RAP, which is detailed in Chapter 3. Another MOGA is combined to RBI in order to provide optimal inspection plans in terms of risk and cost. The latter methodology is described in Chapter 4.

2.2 Generalized Renewal Processes

The usual stochastic processes to model the failure-repair process of repairable components (or systems) are the Renewal Processs (RPs) and the Non-Homogeneous Poisson Processs (NHPPs). Both of them are counting processes and the Times To Repair (TTRs) are negligible if compared to the component operational time. If RP is chosen, the TBFs are independent and identically distributed (i.i.d.) with an arbitrary probability distribution. Besides, one assumes that the component, after a failure, is subjected to a perfect repair and returns to operation with a condition it presented when new ("as good as new"). On the other hand, using NHPP, the TBFs are neither independent nor identically distributed. In addition, it is supposed that the maintenance crew makes a minimal repair in the failed component, that is, it returns to an operational state with the same condition it had just before the failure occurrence ("as bad as old").

Nevertheless, the assumption of minimal or perfect repairs required to utilize either NHPP or RP, respectively, is often not realistic. In practical situations, corrective maintenance actions are likely to be imperfect repairs, *i.e.*, they are intermediate actions between minimal and perfect repairs and the equipment returns to operation with a condition better than old and worse than new. In this way, GRPs can be used to model failure-repair processes of components subject to imperfect repairs. In GRP, a parameter q (rejuvenation parameter) is introduced in the model and the value it assumes is related to the maintenance action efficacy (see Table 2.2). The common values of q are in $[0, 1]$, but $q < 0$ and $q > 1$ are also possible and represent the improved and the worse repair, in this order. However, an improved repair might require some project modifications of the component and this is not a procedure used in ordinary repair actions (MOURA *et al.*, 2007). Also, a corrective maintenance action that returns the equipment worse than immediately before the failure occurrence is not common in real situations. In this work, the rejuvenation parameter is considered to be in $[0, 1]$.

Table 2.2: Repair classification according to parameter q

q **value**	**Repair type**
$q < 0$	Improved
$q = 0$	Perfect
$0 < q < 1$	Imperfect
$q = 1$	Minimal
$q > 1$	Worse

The parameter q is used in the calculation of the component virtual age (V_n), which is defined as follows:

$$V_n = V_{n-1} + qX_n, \tag{2.7}$$

where X_n is the time between the $(n-1)$th and the nth failure. By definition, $V_0 = 0$. Thus the expansion of Equation (2.7) yields:

$$V_n = q\sum_{i=1}^{n} X_i, \tag{2.8}$$

where $\sum_{i=1}^{n} X_i$ is the real component age.

The definition of virtual age presented in Equations (2.7) and (2.8) is in accordance with Kijima Type I model, which assumes that the repair actuates just on the very last failure and compensates only the damage accumulated in the interval between the $(n-1)$th and nth failures. In this way, only the additional age X_n is reduced.

The distribution of the nth failure time (T_n) can be calculated by a probability distribution function conditioned to the $(n-1)$th component virtual age, as follows:

$$F(x_n|v_{n-1}) \;=\; P(X \leq x_n|V_n > v_{n-1}) = P(T_n \leq v_{n-1} + x_n|V_n > v_{n-1}) \tag{2.9}$$

$$= \frac{F(x_n + v_{n-1}) - F(v_{n-1})}{1 - F(v_{n-1})}, \tag{2.10}$$

If the TBFs are distributed according to a Weibull distribution with α and β as scale and shape parameters, respectively, (2.10) becomes as follows:

$$F(x_n|v_{n-1}) = 1 - \exp\left[\left(\frac{x_n}{\alpha}\right)^{\beta} - \left(\frac{v_{n-1} + x_n}{\alpha}\right)^{\beta}\right] \tag{2.11}$$

If the TTRs are not small when compared to the component operational time, they might be taken into account in the system failure-repair process evaluation. In this way, two different GRP counting processes can be considered: one related to failure occurrences and the other to repair actions. The superimposing of these stochastic processes yields an alternating process that characterizes the component state, *i.e.*, if it is either operational or under a maintenance action.

Suppose that a system is made of many components and that the failure-repair process of each one of them follows the two above-mentioned GRP. Thus, the failure-repair process of the entire system is a superimposition of all components failure-repair processes, which, in turn, are alternating processes. The analytical handling of the failure-repair process of the full system is not possible and, as an alternative, DES can be adopted. A detailed discussion about RP and NHPP is found in Rigdon & Basu (2000) and Rausand & Hoyland (2004). For more information on GRP, see Kijima & Sumita (1986), Yañes *et al.* (2002) and Moura *et al.* (2007).

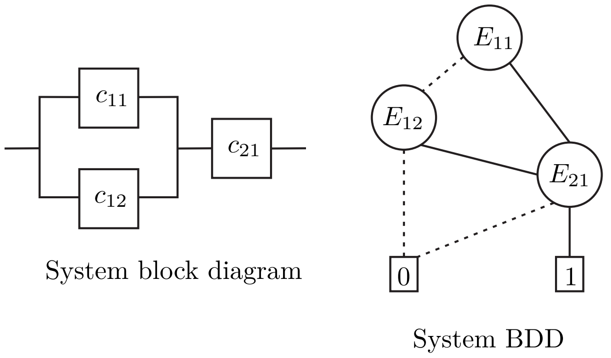
2.3 Discrete Event Simulation

The DES framework is specially useful when the system characterization involves a complex logical structure among its components. The underlying idea of DES is to stochastically generate "events" of interest over time in order to obtain quantities necessary for evaluating system performance. The basic elements of a DES are variables and events. The variables are usually categorized into: (i) time variable (t), which is the amount of simulated time; (ii) counter variable that refers to the number of times an event have occurred by time t; (iii) system state variable that concerns the state of the system on time t (ROSS, 2002). Whenever an event occurs, these variables are updated and the dynamics of the system can be observed. In the end of the simulation period, an overview of the system behavior during simulation time can be provided along with a performance assessment.

In the case of the system configurations related to RAPs, failure and repair of each component are the fundamental events of interest. By means of the states of the various components at a given time and of the system logic, the state of the entire system can be determined. Indeed, the system logic is often represented by reliability block diagrams (MODARRES *et al.*, 1999) and computationally assessed by Binary Decision Diagram (BDD) (RAUZY, 2001).

2.3.1 Binary Decision Diagrams

A BDD represents a Boolean function as a rooted, directed acyclic graph (BRYANT, 1992). As an illustration, suppose the series-parallel system represented by the block diagram in Figure 2.2 and its related BDD. Each node is characterized by the success event associated with a component (c_{ij}). The solid line corresponds to the realization of the success event – c_{ij} is operational, whereas the dashed line is associated with the occurrence of the failure event – c_{ij} is unavailable. The BDD is responsible for evaluating the system state on a given DES time step, in which the state of each component is already defined: c_{ij} is either operational or failed. Thus the probabilities associated with the success event is either 0 or 1, as shown in Figure 2.2. All paths leading to either 0 (failed system) or 1 (available system) are mutually excluding. Furthermore, paths toward 1 represent the minimal paths of the related block diagram (MODARRES *et al.*, 1999). For more on BDD, see Bryant (1992) and Rauzy (2001).



System block diagram

System BDD

If c_{ij} is operational, then $P(E_{ij}) = 1$ and $1 - P(E_{ij}) = 0$; otherwise $P(E_{ij}) = 0$ and $1 - P(E_{ij}) = 1$.

$$\text{System logic function: } P(E_{11})P(E_{21}) + [1 - P(E_{11})]P(E_{12})P(E_{21}) = \begin{cases} 0, & \text{if system is down.} \\ 1, & \text{if system is up.} \end{cases}$$

Figure 2.2: System BDD

2.4 Risk-Based Inspection

In risk analysis, usually a qualitative assessment is primarily performed in order to obtain a categorized risk level for the considered equipments. Probabilities of failures and consequences are divided into categories and then combined so as to find an appropriate category for the risk (*e.g.* high, medium high, medium and low). The equipments with (medium) high risk level are often submitted to a more detailed quantitative risk analysis

(MODARRES, 2006).

RBI emerges as a quantitative methodology to support inspection decision making in the context of pressurized equipment often used in petrochemical industry. The risk assessment relies on the evaluation of likelihood or frequencies or of equipment malfunction due to changes in material and structure (*e.g.* cracks, holes, ruptures) and of the related consequences such as leakage of dangerous substances, fires and explosions.

For RBI (API, 2008) the probability of failure is given by:

$$P_f(k) = gff \cdot D_f(k) \cdot F_{MS}, \tag{2.12}$$

in which k is a given time period, gff is the generic failure frequency obtained from a representative failure database; $D_f(k)$ is the damage factor related to the applicable damage mechanisms (*e.g.* corrosion) acting on the equipment and it modifies the gff to make it specific to the equipment under evaluation; F_{MS} is the management systems factor that accounts for the influence of the facility's management system on the mechanical integrity of the plant equipment and is often obtained by the application of a questionnaire.

In RBI, the consequences are determined using consequence analysis techniques and are expressed in affected area or financial terms. In this work, only financial consequences (FC) are considered. The underlying idea of a consequence analysis is to estimate the consequences of releases of hazardous fluids for different hole sizes, which depend on the phase of the fluid. For a pressurized equipment from petrochemical industry, the steps of a consequence analysis are (API, 2008):

- Determine representative fluid.
- Select a set of release hole sizes.
- Calculate theoretical release rate/mass.
- Estimate fluid inventory.
- Establish release type.
- Estimate impact of detection and isolation systems.
- Calculate adjusted release rate / mass.
- Determine flammable / explosive consequences.
- Determine toxic consequences.
- Determine non-flammable non-toxic consequences.
- Determine component damage and personnel injury consequence areas.

- Determine FC.

The probability of failure $P_f(k)$ and the FCs can be calculated for different types of damage mechanisms (*e.g.* internal or external corrosion) and for a number of different hole sizes, $h = 1/4”, 1”, 4”, 16”$, that are related to small, medium, large and rupture, respectively. For each damage mechanism, there are several costs taken into account: (i) costs of repair and replacement, (ii) costs of damage to surrounding equipment in affected areas, (iii) costs associated with production losses and business interruption as a result of downtime to repair or replace damaged equipment, (iv) costs due to potential injuries associated with a failure, (v) environmental cleanup costs (API, 2008). All of these costs are aggregated for the different types of holes and an overall financial consequence FC, per damage mechanism, is obtained. The total risk value to which the equipment is exposed for a given damage mechanism is as follows:

$$R(k) = \sum_h P_f^h(k) \cdot FC. \tag{2.13}$$

If two or more damage mechanisms are present, then the total risk is the sum of the risks related to each of them.

Inspection is an integrity control technique that permits the identification of damage mechanisms responsible for the equipment deterioration process. Although it does not necessarily reduce the involved risk, the knowledge about the presence of damage mechanisms reduces the uncertainty about the actual deterioration state of the equipment. The RBI approach basically consists in the determination of inspection plans with risks that do not exceed a predefined acceptable risk level (risk target). Thus, the calculation of risk should be updated whenever an inspection is performed in order to represent the equipment condition over time.

Actually, as gff, F_{MS} and FC are obtained only once, *i.e.* they become constant, only the damage factor $D_f(k)$ needs to be computed after an inspection and it is responsible for modifying the risk level in time. Also the uncertainty about the $D_f(k)$ depends on the efficacy of the inspection technique adopted. For example, the application of an inspection of low (high) efficacy can result in high (low) uncertainty level about $D_f(k)$, even if applied in the beginning (end) of the considered period. This is contrary to the intuition of an increasing damage factor, as no interventions are performed to delay such a mechanism. The uncertainty about the $D_f(k)$, partially due to the efficacy of the inspection techniques, reflects on the calculated risk values over time.

This chapter presented the theoretical foundations used in the subsequent two chapters, which tackle multi-objective problems with cost and availability / risk as conflicting objectives in the design of systems with imperfect repairs and in the construction of inspection plans.

3 DESIGN OF SYSTEMS SUBMITTED TO IMPERFECT REPAIRS BY MULTI-OBJECTIVE GENETIC ALGORITHMS AND DISCRETE EVENT SIMULATION

In this chapter, the multi-objective RAP considered in this work is formally stated and a solution framework combining MOGA and DES is developed. Two examples with known exact Pareto fronts are used to validate the proposed MOGA by means of two performance metrics. Also, MOGA + DES is applied to a more realistic example involving systems subjected to imperfect repairs, which are incorporated to DES by a GRP model. Indeed, the DES portion of the proposed methodology enables the evaluation of the system's failure-repair process and of the related costs during its lifetime cycle and not only at the acquisition moment. Also in the more realisitic application, the maintenance teams are taken into account as a decision variable itself. As a result, nondominated system configurations representing the compromise between availability and cost are obtained and the number teams necessary to maintain the system along its mission time is also indicated. The main aspects and findings of the proposed methodology are in Lins & Droguett (2011).

3.1 Problem Statement and Formulation

Suppose that one has to design a system formed by a predefined number of subsystems in series (s), which, in turn, may have several components in parallel. Also, assume that the designer objectives are to maximize system availability (A) and to minimize system cost (C). There are several components available in market, with different reliability and cost characteristics: suppose that for each subsystem j ($j = 1, \ldots, s$) there are ct_j different component types that perform the same function. Moreover each subsystem may have a minimum and a maximum number of allowed components ($n_{j,\min}$, $n_{j,\max}$, respectively). Thus, the designer may obtain a set of nondominated solutions (*i.e.* system configurations) that represent the compromise between availability and cost. The general mathematical formulation of this problem is as follows:

$$\max_{x} \quad A(x) \tag{3.1}$$

$$\min_{x} \quad C(x) \tag{3.2}$$

$$\text{s.t.} \quad n_{j,\min} \leq \sum_{k=1}^{ct_j} x_{jk} \leq n_{j,\max}, \tag{3.3}$$

$$x_{jk} \in \{0, 1, \ldots, n_{j,\max}\}, \tag{3.4}$$

in which $k = 1, \ldots, ct_j$ and x_{jk} is the number of components of type k in the jth subsystem. The argument of both objective functions reflects the dependence of system performance on the performance of its components. In this way, input data regarding the components' availability and cost features are required. As a result, a set of nondominated solutions of the form

$$x = (x_{11}, \ldots, x_{1k}, \ldots, x_{1,ct_1}; \ldots; x_{j1}, \ldots, x_{jk}, \ldots, x_{j,ct_j}; \ldots; x_{s1}, \ldots, x_{sk}, \ldots, x_{s,ct_s}) \tag{3.5}$$

is found.

The definition of the objective functions in Equations (3.1) and (3.2) depends on the considered reliability approach during the design phase of the system, whereas the constraints given by Equations (3.3) and (3.4) remain unchanged. If a static analysis is adopted, the components availabilities can be interpreted as steady-state availabilities (constant values) and the system acquisition cost can be computed. Then Equations (3.1) and (3.2) become Equations (3.6) and (3.7), respectively:

$$\max_x \quad A(x) = \prod_{j=1}^{s} \left[1 - \prod_{k=1}^{ct_j} (1 - A_{jk})^{x_{jk}} \right], \tag{3.6}$$

$$\min_x \quad C_a(x) = \sum_{j=1}^{s} \sum_{k=1}^{ct_j} c_{jk}^{a} x_{jk}, \tag{3.7}$$

where A_{jk} and c_{jk}^{a} are, respectively, the components' steady-state availabilities and acquisition costs. When the number of possible configurations allows the exhaustive calculation of the Pareto front, then one has an exact solution for the redundancy allocation problem under consideration. This is specially useful for validating heuristic algorithms (MOGAs, for example), since simulated and exact Pareto fronts can be compared.

The cost function of Equation (3.7) can be slightly modified to incorporate the steady-state unavailability cost:

$$C(x) = C_a(x) + c_S^u \cdot t_n \cdot [1 - A(x)], \tag{3.8}$$

in which c_S^u is the cost per time unit related to system unavailability and t_n is the mission time. Examples of works that incorporate a static approach to the reliability / availability are Taboada *et al.* (2007) and Taboada *et al.* (2008). Nevertheless, the static formulation by no means incorporates the dynamic behavior that the different configurations can have during their lifetime. Hence, a more realistic formulation may be implemented and can be combined with DES to compute system availability and also to provide some metrics for the cost calculation. Additionally, besides optimal system configurations in accordance with a multi-objective perspective, the number of maintenance teams related to a system design can be determined, given that the availability of maintenance teams may be considered by DES when a component failure takes place and has to be repaired. Then the general solution presented in Equation (3.5) has an additional decision variable y at the end, regarding the number of maintenance teams, *i.e.*,

$$x = (x_{11}, \ldots, x_{1k}, \ldots, x_{1,ct_1}; \ldots; x_{j1}, \ldots, x_{jk}, \ldots, x_{j,ct_j}; \ldots; x_{s1}, \ldots, x_{sk}, \ldots, x_{s,ct_s}; y). \tag{3.9}$$

For the more realistic formulation involving DES, the considered objective functions are:

$$\max_x \quad A(x) = \text{Operational time/Mission time}, \tag{3.10}$$

$$\min_x \quad C(x) = C_a(x) + C_o(x) + C_{cm}(x) + C_{mt}(x) + C_u(x), \tag{3.11}$$

where $C_a(x)$ is the acquisition cost calculated as in Equation (3.7);

$$C_o(x) = \sum_{j=1}^{s} \sum_{k=1}^{ct_j} \sum_{l=0}^{x_{jk}} c_{jk}^o t_{jkl}^o \tag{3.12}$$

is the operational cost, in which c_{jk}^o is the cost per unit time of operating a component of type k in jth subsystem and t_{jkl}^o is the operational time of the lth copy of that component;

$$C_{cm}(x) = \sum_{j=1}^{s} \sum_{k=1}^{ct_j} \sum_{l=0}^{x_{jk}} c_{jk}^{cm} n_{jkl} \tag{3.13}$$

is the corrective maintenance cost, where c_{jk}^{cm} is the corrective maintenance cost of a component of type k in subsystem j and n_{jkl} is the number of repairs that the lth component undergoes during mission time;

$$C_{mt}(x) = c^{mt} \cdot y \tag{3.14}$$

is the cost associated with maintenance teams and c^{mt} is the cost of a single maintenance team for the period of mission time;

$$C_u(x) = c_S^u \cdot t_S^u \tag{3.15}$$

is the cost of system unavailability, where c_S^u is the cost of system unavailability per time unit and t_S^u is the time in which the system remains unavailable during mission time. This cost can be interpreted as a penalty to the system due to its unavailability for a certain amount of time. The quantities t_{jkl}^o, n_{jkl}, t_S^u and the objective function concerning system availability (Equation (3.10)) are all obtained via DES. The failure-repair processes of each component are modeled via alternating GRP introduced in Section 2.2.

3.2 Proposed Multi-objective Genetic Algorithm

Genetic operators that provide only feasible individuals as outcomes significantly reduces the search space, which is formed by all possible solutions disregarding the physical constraints in Equation (3.3). For the RAP discussed in the previous section, the percentage of the search space concerning feasible solutions is given by:

$$\frac{\prod_{j=1}^{s}\sum_{k=1}^{ct_j}\binom{n_{j,\max}+k-1}{n_{j,\max}-1}}{\prod_{j=1}^{s}(n_{j,\max}+1)^{ct_j}}\cdot 100\%, \tag{3.16}$$

where the numerator concerns the number of feasible solutions, whereas the denominator is related to the number of total solutions in the entire search space.

As only the feasible search space is explored, the MOGA can not be “lost” in some unfeasible location and also the use of penalty functions due to an eventual unfeasibility is not required. Additionally, the fitnesses’ evaluation step can demand an increased computational cost (*e.g.* DES), which is only used to assess actual potential solutions – otherwise, such an effort would be unnecessarily applied to the evaluation of unfeasible individuals.

In the following, let N be the fixed size of population P, P_i the ith individual of P that represents a system configuration, P_{aux} the auxiliary population that stores nondominated individuals and that is updated at each MOGA iteration. Also, let x denote an individual phenotype. The individual representation and the genetic operators devised to create feasible system configurations with respect to constraints in Equation (3.3) are presented in the next subsections.

3.2.1 Individual Representation

The redundancy allocation problems tackled in this work have only integer-valued decision variables. Therefore, an integer representation of individuals is used. As an illustration, suppose that a system is formed by two subsystems in series and that each one of them can have at least 1 and at most 4 components in parallel ($n_{j,\min} = 1$ and $n_{j,\max} = 4$, $j = 1, 2$). Moreover, consider that the first subsystem has 4 component options ($ct_1 = 4$), whereas the second subsystem has only 2 component options ($ct_2 = 2$). Each x_{jk} value ($x_{jk} \in \{0, 1, \ldots, 4\}$, $j = 1, 2$, $k = 1, \ldots, ct_j$) represents the number of every component type that may be put in parallel in its respective subsystem. In addition, y is the decision variable associated with the number of maintenance teams that can be hired and that $y \in \{0, 1, \ldots, 10\}$. The individual phenotype as well as the system design it represents are depicted in Figure 3.1. It is important to note that y can be interpreted as an additional subsystem having only one available type of component and with minimum and maximum number of redundant components respectively equal to the minimum and maximum maintenance teams that can be used. Such interpretation is very useful, since the developed genetic operators do not make any difference among variables' nature. Hence, in the example just described and if at least one maintenance team is required, $s = 3$, $ct_3 = 1$, $n_{3,\min} = 1$ and $n_{3,\max} = 10$.

Subsystem	1				2		3
Variable	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	y
System configuration	2	0	1	1	3	0	4

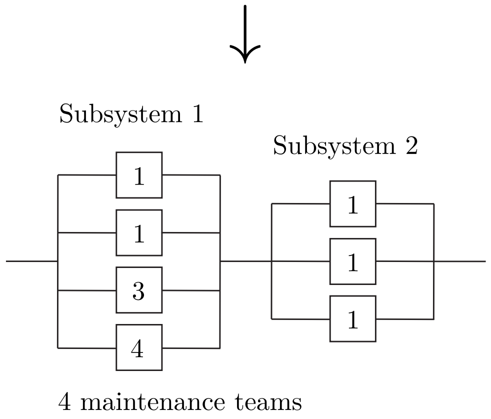


Figure 3.1: Individual representation for MOGA (system configuration)

3.2.2 Generation of Initial Population

Each one of the N individuals of the initial population is generated in accordance with Algorithm 1, whose main idea is to create one subsystem at a time and to generate random integers for the different component types within the considered subsystem. A vector pos containing a permutation of $1, \ldots, ct_j$ is used for the generation of the variables' values so that one component type has no advantages over the other ones. The parameters $n_{j,\min}$ and $n_{j,\max}$ along with the number of components already allocated in subsystem j enables the generation of feasible values for x_{j,pos_k}. Whenever a subsystem attains the maximum number of components $n_{j,\max}$, the other positions are set to 0. A feasible individual represented by x is the result of Algorithm 1.

3.2.3 Selection and Update of the Auxiliary Population

In the selection step, the dominance relation among individuals is evaluated according to their fitness values (fitness evaluation step is described in Section 3.3). Firstly, dominance relation is assessed between pairs of individuals within the current population. Dominated individuals are eliminated and the nondominated ones are then candidate solutions to become part of the auxiliary population P_{aux}. Secondly, P_{aux} update takes place

```
1. For $j = 1, \ldots, s$:
   a. Set $S = 0$.
   b. While $S < n_{j,\min}$:
         I. S = 0.
        II. $pos \leftarrow$ random permutation of $1, \ldots, ct_j$
       III. For $k = 1, \ldots, ct_j$:
              i. If $n_{j,\max} - S > 0$, generate $x_{j,pos_k}$ from $0, \ldots, n_{j,\max} - S$ and $S \leftarrow S + x_{j,pos_k}$.
             ii. Else if $n_{j,\max} - S = 0$, set $x_{j,pos_o} = 0$ for $o = k+1, \ldots, ct_j$ and $k \leftarrow ct_j + 1$.
2. Return $x$.
```

Algorithm 1: Generation of a feasible individual

in accordance with the following rules:

- If a candidate solution is dominated by some individual in P_{aux}, it is discarded.
- If a candidate solution dominates individuals in P_{aux}, all dominated solutions are deleted from P_{aux} and a copy of the candidate solution is stored in P_{aux}.
- If a candidate solution neither dominates nor is dominated by individuals in P_{aux}, it is inserted into P_{aux}.

Since the dominated individuals are eliminated, the size of P is reduced to N_r ($N_r \leq N$). In order to maintain the population with N individuals, $N - N_r$ solutions are randomly selected from P_{aux} and inserted into P. Algorithm 2 summarizes the selection and the update of P_{aux}; I represents the set of indexes related to dominated solutions in the current population P.

```
1. For $i = 1, \ldots, N$:
   a. If $P_i$ is nondominated in $P$:
         I. If $P_i$ is nondominated in $P_{\text{aux}}$, insert a copy of $P_i$ into $P_{\text{aux}}$.
        II. Else if $P_i$ dominates individuals in $P_{\text{aux}}$, eliminate dominated individuals from $P_{\text{aux}}$ and
            insert a copy of $P_i$ into $P_{\text{aux}}$.
       III. Else if $P_i$ is dominated by any solution in $P_{\text{aux}}$, ignore $P_i$.
   b. Else if $P_i$ is dominated in $P$, insert $i$ into $I$.
2. Eliminate $P_i$ from $P$ for all $i \in I$.
3. Randomly select $N - N_r$ individuals from $P_{\text{aux}}$ and insert them into $P$.
4. Return $P$ and $P_{\text{aux}}$.
```

Algorithm 2: Selection and update of P_{aux}

With the elimination of dominated individuals from P, the update of P_{aux} and the posterior random choice of its solutions to complete P, the MOGA tends to maintain solutions of higher expected quality, whose information can be exchanged and modified in the crossover and mutation steps, respectively. One may wonder if this approach would harm the MOGA ability in exploring the search space, as it could lead the algorithm to local Pareto sets. However, it is important to emphasize that the solutions that are nondominated in P but dominated by individuals in P_{aux} remain in P. Such individuals can have information of non-explored parts of the search space and provide the required variability among potential solutions.

Another possibility would be not updating P_{aux} until the stop criterion is reached. Some early experiments indicated that such a strategy have better performance in real-valued problems than in integer-valued ones. The obtained Pareto fronts in the former type of problems were much better with the non-updating approach and the opposite was observed for the latter problems, *i.e* the Pareto fronts of the integer-valued problems were enhanced by the updating strategy. Also, there were some problems concerning the size of the auxiliary population, which sometimes became prohibitively large. Since only integer-valued problems are taken into account in this work, the update of P_{aux} at every iteration is adopted.

3.2.4 Crossover and Replacement

After selection, a random number in $[0, 1]$ is generated for each individual in P. If this number is less than the crossover probability p_c for a certain P_i, this individual will participate in the crossover. In this work, the proposed crossover operator handles integer variables and avoids the creation of unfeasible individuals.

For each pair of individuals participating in crossover (parents), m different positions are defined in their phenotypes (ignoring subsystems' boundaries) by the generation of random numbers in $[1, n]$, where n is the phenotype length or the total number of variables. As long as these positions are chosen, the subsystem to which they belong is verified and their contents are exchanged between parents in order to generate children.

In a first moment, these children can be unfeasible, *i.e.* they can present subsystems with forbidden quantities of components. With the aim of rendering an eventual unfeasible subsystem j into a feasible one, one position at a time is randomly chosen among the m selected in the preceding step and its content can be:

- Added by one unit, if j has less than $n_{j,\min}$ components or
- Diminished by one unit if it is greater than the minimum value the related variable can assume and if j contains more than $n_{j,\max}$ components.

Such a procedure is repeated until the number of components in j is in $[n_{j,\min}, n_{j,\max}]$. Since only one unit is incremented / decremented at a time, the algorithm tendency in providing a lower / greater number of components in subsystem j is not severely modified.

The crossover step for a given couple of parents x^1 and x^2 is detailed in Algorithm 3. Note that the replacement strategy "children replace parents" is already incorporated in the crossover, as Algorithm 3 returns the modified x^1 and x^2 (children), which automatically replace the original parents x^1 and x^2.

```
1. For a given pair of parents $x^1$ and $x^2$, randomly select $m$ positions and exchange their contents.
2. For $j = 1, \dots, s$ and for $o = 1, 2$:
   a. $S^o = \sum_{k=1}^{ct_j} x^o_{jk}$.
   b. Select a random position $w$ in subsystem $j$ whose content was exchanged.
   c. While $S^o < n_{j,\min}$, $x^o_{jw} \leftarrow x^o_{jw} + 1$ and $S^o \leftarrow S^o + 1$.
   d. While $S^o > n_{j,\max}$:
         I. If $x^o_{jw} > 0$, $x^o_{jw} \leftarrow x^o_{jw} - 1$ and $S^o \leftarrow S^o - 1$.
        II. Else if $x^o_{jw} = 0$, eliminate $w$ from the possible positions to be selected.
3. Return $x^1$ and $x^2$.
```

Algorithm 3: Crossover and replacement return feasible individuals

3.2.5 Mutation

The mutation step consists in changing the content of a position in an individual phenotype. If a position is selected in accordance with the predefined mutation probability p_m, its content is substituted by a random integer uniformly generated in an interval that does not violate the individual's feasibility. If position k from the jth subsystem is selected, then the number of components of other types in the current subsystem (S_{-k}), $n_{j,\min}$ and $n_{j,\max}$ are taken into account to generate a new value for x_{jk}, as described in Algorithm 4.

```
1. For $j = 1, \dots, s$:
   a. For $k = 1, \dots, ct_j$:
         I. Generate $u$ from $\mathcal{U}(0, 1)$.
        II. If $u \leq p_m$:
              i. $S_{-k} = x_{j1} + \cdots + x_{j,k-1} + x_{j,k+1} + \cdots + x_{j,ct_j}$.
             ii. If $S_{-k} \geq n_{j,\min}$, generate $x_{jk}$ from $0, \dots, n_{j,\max} - S_{-k}$.
            iii. Else if $S_{-k} < n_{j,\min}$, generate $x_{jk}$ from $n_{j,\min} - S_{-k}, \dots, n_{j,\max} - S_{-k}$.
   b. Return $x$.
```

Algorithm 4: Mutation returns a feasible individual

3.3 Assessment of Fitnesses by Discrete Event Simulation

The MOGA summarized in the previous section is coupled with Monte Carlo simulation for a more realistic representation of the dynamic behavior of systems. In particular, DES is used to analyze the behavior of a system, which consists in the generation of random discrete events during simulation time with the aim of creating a "typical" scenario for a system so as to allow for the evaluation of some of its features that are of interest for the calculation of the objective functions of an individual P_i previously defined.

As depicted in Figure 3.2, the coupling takes place at the fitness evaluation step. For a given generation in the genetic algorithm evolution process, the availability objective function is estimated for every individual $P_i \in P$, $i = 1, \dots, N$, via the DES algorithm. For a predetermined mission time, t_n, the ith individual (a candidate system) undergoes stochastic transitions between its possible states which, for the problem of interest, correspond to available and unavailable states. The system availability estimation is based on its components, *i.e.*, the system evolves through states of availability and unavailability depending on the components states: available or unavailable.

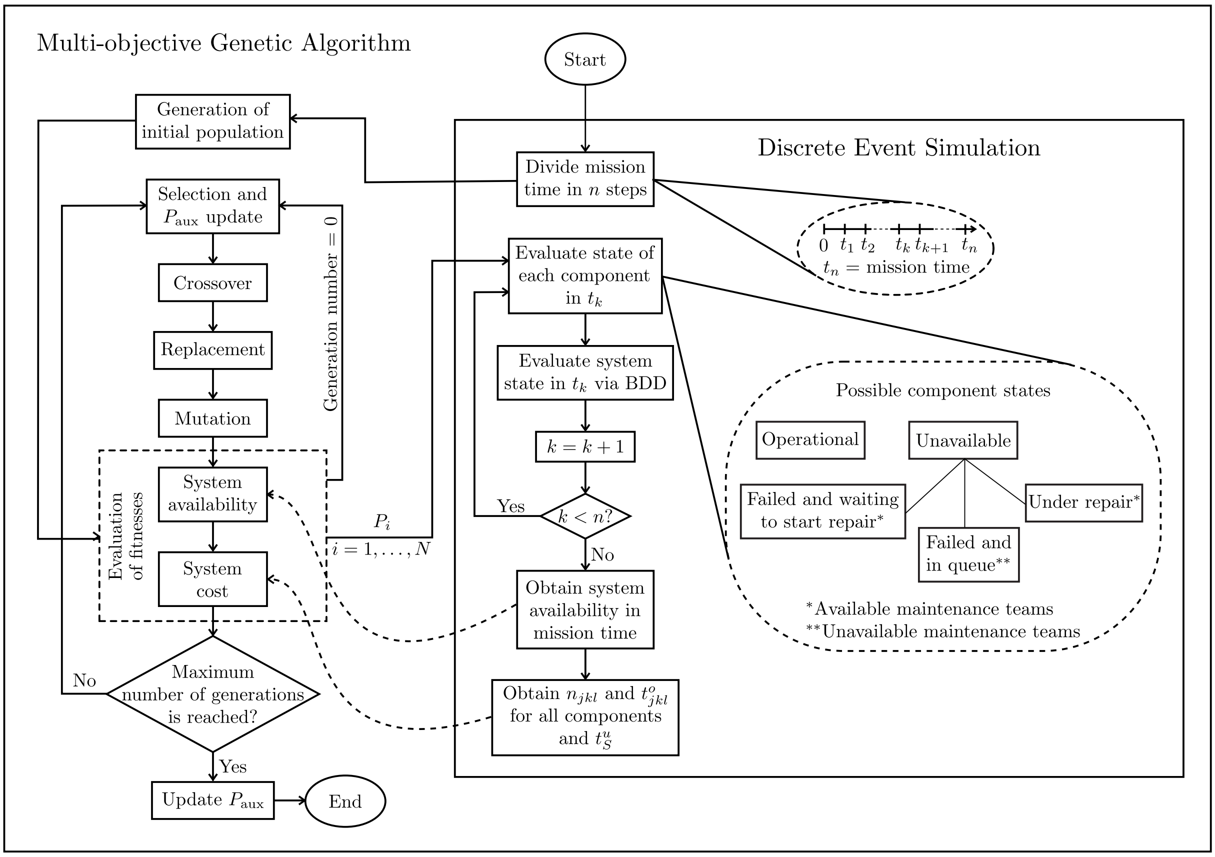

Multi-objective Genetic Algorithm
Start
Generation of initial population
Divide mission time in n steps
Discrete Event Simulation
Selection and P_{aux} update
Generation number = 0
0 t_1 t_2 t_k t_{k+1} t_n
t_n = mission time
Crossover
Evaluate state of each component in t_k
Replacement
Evaluate system state in t_k via BDD
Possible component states
Mutation
Operational
Unavailable
$k = k + 1$
Evaluation of fitnesses
System availability
Yes
$k < n$?
Failed and waiting to start repair*
Under repair*
P_i
$i = 1, \dots, N$
No
Failed and in queue**
System cost
Obtain system availability in mission time
*Available maintenance teams
**Unavailable maintenance teams
No
Maximum number of generations is reached?
Obtain n_{jkl} and t^o_{jkl} for all components and t^u_S
Yes
Update P_{aux}
End

Figure 3.2: MOGA + DES

The failed components along with the available maintenance teams form a queueing system (ROSS, 2010), in which the former are the “customers” and the latter are the “servers”. When a component fails, it is necessary to evaluate if there is any maintenance team available to perform the repair action. If there is an available maintenance team, then the component has to wait until it becomes ready to initiate the repair. If there is not, the component is added to a queue (in this work, the modeled queue is without priority, since all components are supposed to have the same relevance). Under this circumstance, the component must wait until all components that were already in queue to be repaired. Moreover, when the considered component becomes the very next to be repaired, it may have to wait the delay previously commented. Therefore, a component is said to be unavailable if it is:

- Failed and in queue: this occurs when all maintenance teams are occupied.
- Failed and waiting to initiate repair: the component is the very next to be repaired, however there is a delay from the time a maintenance team becomes available to

the time it is ready to perform the repair action.

- Under repair.

Indeed, the availability quantification for the ith individual starts by subdividing the mission time t_n into n time steps. For a given iteration, say t_k, with $k = 1, \ldots, n$, the current state for each component is evaluated, as shown in Figure 3.2-b. In order to determine the system state from its components states at t_k, one needs to construct the relationship characterizing the system logic as a function of the components. In the proposed approach, this is achieved by means of BDDs, briefly described in Section 2.3.1. Thus, in order to keep track of the simultaneous evolutions of all components of the ith individual as well as of the system itself during the mission time, the observed realizations of the following random variables are recorded: number of failures and repairs of each component, number of times the system fails, the time intervals for which each component is under repair (corrective maintenance) and the time intervals during which the system is in the available state. The DES is replicated several times such that independent realizations of these random variables are obtained and then used to estimate the point availability, $A(t)$, for the ith system over the mission time. Then, one obtains estimates for the two objective functions, namely the system average availability, and the system total cost, where both metrics are estimated over the mission time. These results are then fed back to the MOGA, as illustrated in Figure 3.2.

The greater the number of steps and replications, the better the DES estimates. However, given that the redundancy allocation problem may involve a considerable number of components, each with a particular failure-repair process, DES may demand great computational and time efforts. Both MOGA and DES were implemented in C++.

3.4 Overview

The proposed MOGA described in Section 3.2 is summarized in Algorithm 5. Note that the coupling between MOGA and the DES detailed in Section 3.3 takes place at the fitness evaluation step. The algorithm is repeated until the maximum number of generations N_{gen} is attained. At the end, the algorithm returns P_{aux} with the feasible and overall nondominated individuals, which represent efficient system configurations in terms of mean availability and cost. In addition, Figure 3.2 depicts the flowchart of the proposed MOGA + DES with further details of the DES portion of the methodology.

3.5 Metrics for Comparing Real and Simulated Pareto Fronts

As mentioned earlier, the static availability approach of a multi-objective redundancy allocation problem is useful to assess the performance of MOGA by comparing the sim-

```
1. Set $g = 0$ and generate initial population.
2. For $i = 1, \ldots, N$:
   a. Compute $A(x^i)$ via DES.
   b. Obtain metrics from DES and calculate $C(x^i)$.
3. Perform selection and update of $P_{\text{aux}}$.
4. If $g = N_{\text{gen}}$, go to step 7; else go to step 5.
5. Perform crossover and replacement.
6. Perform mutation, $g \leftarrow g + 1$, go to step 2.
7. Return $P_{\text{aux}}$.
```

Algorithm 5: Proposed MOGA + DES

ulated Pareto fronts with the exact one. Such a comparison requires the use of metrics that can, even if heuristically, represent the convergence of the simulated Pareto front towards the exact one.

System cost and system steady-state availability belong to different scales. Therefore, in order to avoid scale problems, the values of the objectives are scaled in $[0, 1]$. The scaling factors, *i.e.* the minimum and maximum observed cost and availability, are provided by the exact Pareto front, which can be obtained by an exhaustive algorithm. Two distance metrics to be applied in the objective space are considered and then described.

3.5.1 Point-to-Point Distance

For each point in the obtained front, the minimum Euclidean distance from it to one of the points in the real front is computed. In order to find a mean distance representing an entire front ($\bar{d}_i$, $i = 1, \ldots, nf$), all minimum distances (d, see Figure 3.3) are summed and divided by ns_i (number of obtained nondominated solutions in the ith simulated front). Then, the following weighted mean is calculated:

$$D = \frac{\sum_{i=1}^{nf} \bar{d}_i \cdot ns_i}{\sum_{i=1}^{nf} ns_i} \tag{3.17}$$

as an attempt to summarize the convergence of the obtained fronts in one single number. However, note that in this procedure there is loss of information since the solution is the entire front and not only one single point.

3.5.2 Coordinate Distance

As a consequence of the dominance / nondominance relation, of the nature of the handled objectives and also of the objective space presentation (*e.g.* cost as the horizontal axis and steady-state availability as the vertical one), a monotonically increasing Pareto front is obtained. The coordinate distance metric would be very natural if the real Pareto front were continuous. However, the considered problem is discrete and a suitable approximation consists in supposing the steady-state availability as a piecewise

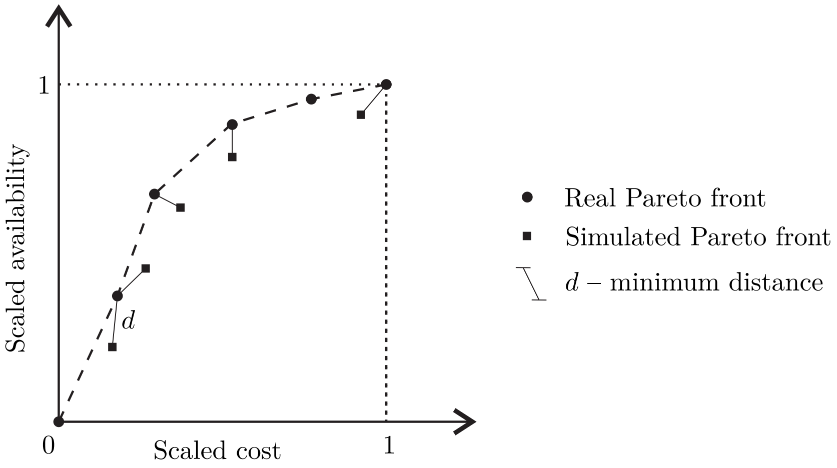


Figure 3.3: Point to point distance

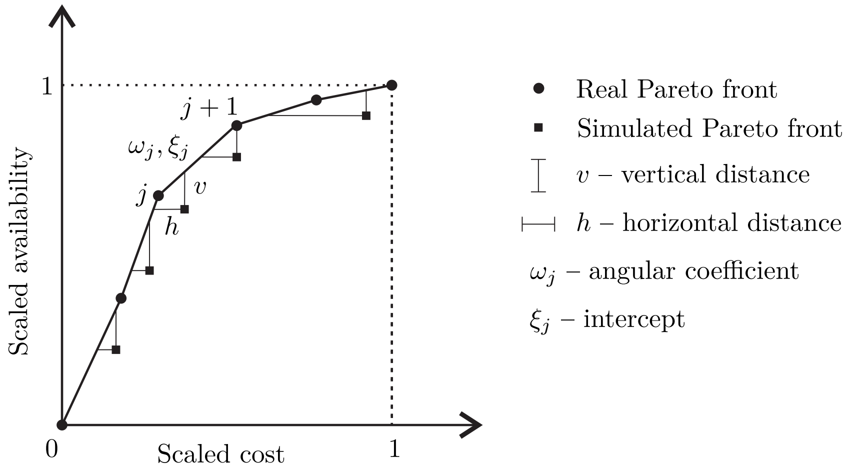


Figure 3.4: Coordinate distance

linear function of the system cost. Hence each two consecutive points j and $j+1$ from the exact Pareto front are connected by a straight line segment with angular coefficient ω_j and intercept ξ_j, that is, $A_j(C) = \omega_j C + \xi_j$, $C \in [C_j, C_{j+1}]$. Thus, each point $(C_{\text{sim}}, A_{\text{sim}})$ from the simulated front has two associated distances (see Figure 3.4): a horizontal distance h, shown in Equation (3.18) and a vertical distance v, given in Equation (3.19):

$$h = C_{\text{sim}} - \frac{(A_{\text{sim}} - \xi_j)}{\omega_j}, \quad C_{\text{sim}} \in [C_j, C_{j+1}], \tag{3.18}$$

$$v = \omega_j C_{\text{sim}} + \xi_j - A_{\text{sim}}, \quad C_{\text{sim}} \in [C_j, C_{j+1}]. \tag{3.19}$$

Then, for every simulated Pareto front, a mean horizontal distance $(\bar{h}_i)$ and a mean vertical distance $(\bar{v}_i)$ are calculated: all horizontal distances (h, Figure 3.4) are summed and divided by ns_i and all vertical distances (v, Figure 3.4) are summed and divided by ns_i. Similar to the case of the point-to-point distance, weighted means H and V regarding horizontal and vertical distances, respectively, are calculated:

$$H = \frac{\sum_{i=1}^{nf} \bar{h}_i \cdot ns_i}{\sum_{i=1}^{nf} ns_i}, \tag{3.20}$$

$$V = \frac{\sum_{i=1}^{nf} \bar{v}_i \cdot ns_i}{\sum_{i=1}^{nf} ns_i}. \tag{3.21}$$

These metrics are used as an approximation for the convergence of the obtained fronts toward the real one and small values for them are desired. For further details on metrics for comparing Pareto fronts, see Knowles & Corne (2002) and Yan *et al.* (2007).

3.6 Numerical Experiments

All experiments described in this section were executed in a PC with Windows operating system, 2.0 GHz processor and 2 GB of RAM.

3.6.1 Validation Examples

In order to validate the proposed MOGA, two analytical examples are devised so as to allow for the comparison between the exact Pareto fronts and the results obtained via MOGA. For both of them suppose a series system composed by 3 subsystems ($s = 3$), which, in turn, may have several components in parallel. The aim is to maximize system steady-state availability and minimize system cost. Table 3.1 lists some configuration features of each subsystem. Components steady-state availabilities (A_{jk}) and acquisition costs (c^a_{jk}) are shown in Table 3.2. These examples have each 1,394,525 feasible components combinations, which represent 0.0135% of the search space (see Equation (3.16)), and were resolved by means of an exhaustive recursive method in order to obtain the exact sets of Pareto nondominated solutions.

Table 3.1: Subsystems' characteristics

j	$n_{j,\min}$	$n_{j,\max}$	ct_j
1	1	3	5
2	1	5	6
3	1	5	3

In the first validation example, Example 1, system cost is defined only by components acquisition cost. The multi-objective formulation is given by Equations (3.6), (3.7), (3.3) and (3.4). The associated real Pareto front is formed by $nr = 144$ nondominated system designs. The second example, Example 2, takes into account the cost due to system unavailability along with the acquisition cost, thus the cost objective is calculated by

Table 3.2: Example 1 – Components' characteristics

Component	**Subsystem 1**		**Subsystem 2**		**Subsystem 3**	
	A_{jk}	c^a_{jk}	A_{jk}	c^a_{jk}	A_{jk}	c^a_{jk}
1	0.9861	10,500	0.9346	6,400	0.9272	7,500
2	0.9699	9,100	0.9664	8,000	0.9175	6,900
3	0.9684	8,300	0.9512	7,600	0.9343	6,200
4	0.9777	8,700	0.9672	9,300	–	–
5	0.9769	8,000	0.9568	7,000	–	–
6	–	–	0.9570	8,800	–	–

Equation (3.8). The parameter c_u is set to 500 monetary units per time unit and t_n is set to 730 time units. A set of $nr = 119$ nondominated system designs outlines the exact Pareto front.

Afterwards, $nf = 100$ trials of the MOGA were executed for each example and the distance metrics of Equations (3.17)-(3.21) were used to provide an idea of the proximity of simulated and real Pareto fronts. The scaling factors obtained from the exact Pareto front are presented in Table 3.3, whereas Table 3.4 shows the parameters used to feed the MOGA.

Table 3.3: Scaling factors

Example	**Cost**		**Steady-state availability**	
	Min.	**Max.**	**Min.**	**Max.**
1	20,600.00	109,000.00	0.853026	0.999996
2	43,859.94	109,001.44	0.977644	0.999996

Table 3.4: MOGA parameters

Parameter	**Value**
Population size (N)	100
Number of generations (N_{gen})	200
Probability of crossover (p_c)	0.95
Number of variables for crossover	7
Probability of mutation (p_m)	0.01

Table 3.5 summarizes results from Examples 1 and 2. It presents D, H, V and some descriptive statistics (minimum, maximum, mean and standard deviation) regarding the number of solutions in each simulated Pareto front as well as the number of obtained exact Pareto solutions. Both examples were also solved by a MOACO algorithm (100 replications), whose essence is described in (LINS & DROGUETT, 2008). The used parameters for the MOACO were: 100 ants, 1000 cycles, $\alpha = 1$, $\beta = 1$, $q = 0.5$ and

$Q = 1$. The related results are presented in the last two columns of Table 3.5. It can be observed that the proposed MOGA is, in general, more accurate than the MOACO. Indeed, the maximum number of solutions obtained by the MOACO is lower than the minimum number of solutions provided by the MOGA for Examples 1 and 2. The same reasoning can be applied to the number of exact solutions.

Table 3.5: Results of validation Examples 1 and 2 – MOGA × MOACO

		MOGA		**MOACO**	
		Example 1	**Example 2**	**Example 1**	**Example 2**
Distance metrics	D	$4.71 \cdot 10^{-4}$	$1.05 \cdot 10^{-3}$	$7.39 \cdot 10^{-4}$	$1.87 \cdot 10^{-3}$
	H	$3.10 \cdot 10^{-3}$	$5.48 \cdot 10^{-3}$	$3.88 \cdot 10^{-3}$	$7.13 \cdot 10^{-3}$
	V	$2.08 \cdot 10^{-4}$	$7.80 \cdot 10^{-4}$	$2.29 \cdot 10^{-4}$	$1.80 \cdot 10^{-3}$
Number of solutions	Min.	107	84	74	51
	Max.	143	114	89	67
	Mean	130.6	101.2	81.6	58.9
	Std. dev.	6.06	5.47	3.19	3.48
Exact solutions	Min.	73	45	50	29
	Max.	132	96	65	41
	Mean	106.1	77.7	56.8	34.8
	Std. dev.	12.74	11.97	2.79	2.47

Figures 3.5 and 3.6 depict the exact and two simulated Pareto fronts for both Examples 1 and 2, respectively. Note that the Pareto front from Example 1 is more scattered than the one associated with Example 2, which is concentrated in higher values of the steady-state availability. Since Example 2 incorporates the unavailability cost, the solutions may reflect this fact, that is, designs with greater steady-state availabilities may incur in higher acquisition costs but lower unavailability costs. The simulated fronts in those figures correspond to the ones with minimum and maximum exact Pareto solutions. It can be noticed that the MOGA is able to find solutions on or near the real Pareto front.

3.6.2 Application Example

In this section, an application example is discussed. As in the previous examples, the system is formed by 3 subsystems, whose characteristics are shown in Table 3.1. The objectives taken into account are the maximization of availability and minimization of system total cost. System availability, as well as several parameters used in the calculation of system total cost, is estimated by means of the DES portion of the methodology. The aim is to find the configurations, including the number of maintenance teams, that are compromise solutions between system availability and system total cost. The formulation of the redundancy allocation problem involves Equations (3.3), (3.4), (3.10) and (3.11).

The available components of each subsystem have diverse reliability and cost charac-

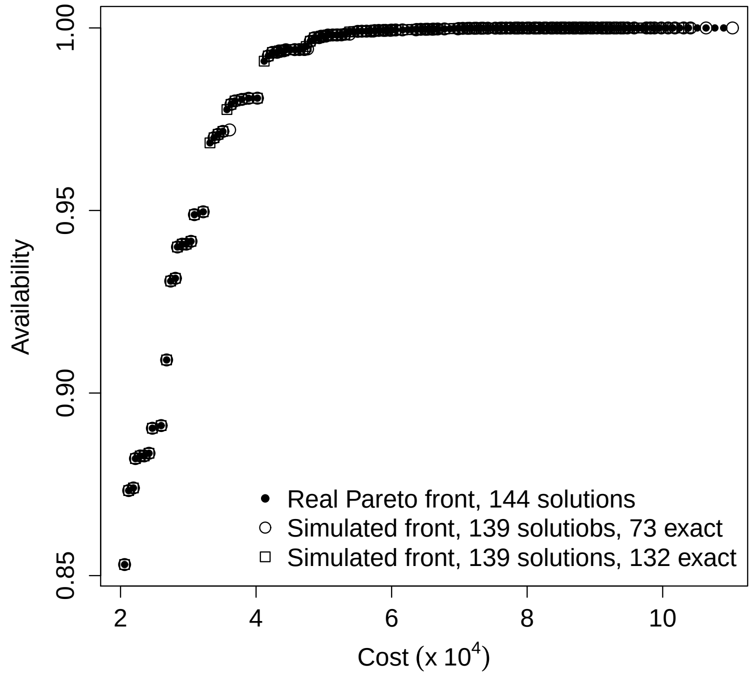


Figure 3.5: Validation example 1 – Exact and simulated Pareto fronts

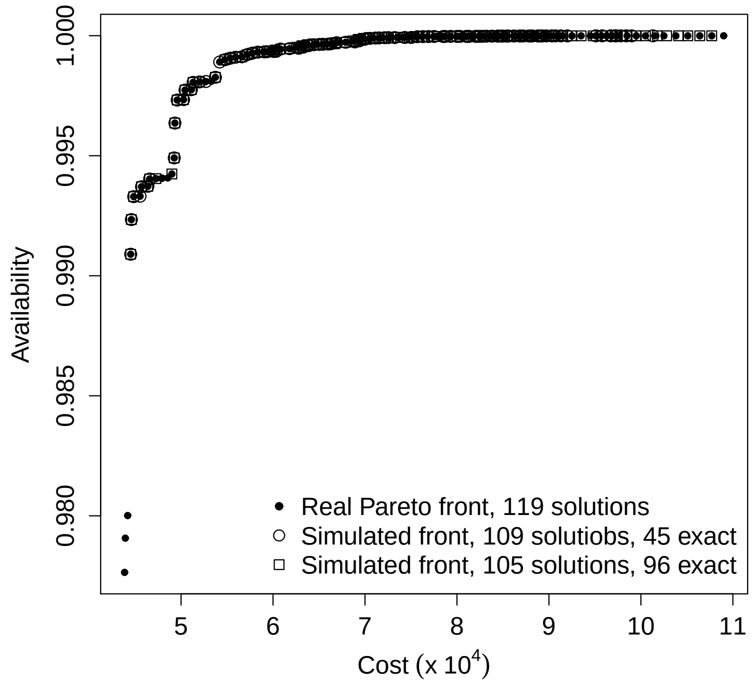


Figure 3.6: Validation example 2 – Exact and simulated Pareto fronts

teristics, as presented in Table 3.6. The TTFs – X – are supposed to follow Weibull distributions with different scale (α) and shape (β) parameters and with the same rejuvenation parameter (q). The TTRs – D – in turn, are supposed to be exponentially distributed with different parameters and the expected value associated to each D, Mean Time To Repair (MTTR), is also given. Notice that other probabilistic distributions, *e.g.* Log-normal, could also be used. Often, in practice, the adoption of a distribution depends on components' historical data related to TTFs and TTRs, so as parameter estimation and goodness-of-fit evaluation can be performed via statistical inference methods. Also, as mentioned in Section 1.2 and illustrated in Figure 1.1, instead of using probabilistic distributions, one can adopt previously adjusted SVR models to provide components' failure times.

In addition, $c^{mt} = 1200$; $c^u_S = 500$; the number of maintenance teams (y) is in $\{0, 1, \ldots, 7\}$, but at least one is required and the time since a maintenance crew is or becomes available up to the beginning of repair is supposed to be exponentially distributed with parameter 1. The addition of the number of maintenance teams as a decision variable increased the number of feasible solutions to 9,761,765 that represents 0.0118% of the search space (see Equation (3.16)).

The Mean Time to First Failure (MTFF) is also presented in Table 3.6 in order to provide an idea of the components reliability features. If perfect repairs are assumed the MTFF is also equal to all the Mean Time Between Failures (MTBF), since components are delivered to operation as if they were new. On the other hand, in minimal and imperfect repairs approaches, MTBF are smaller than the mean time to first failure, given that component deterioration processes are taken into account.

Table 3.6: Application example – components' characteristics

j	**Component**	$f_X(x)$	**MTFF**	$f_D(d)$	**MTTR**	c^a_{jk}	c^o_{jk}	c^{cm}_{jk}
1	1	Weibull(40; 1.9; q)	35.4945	Exp(2.0)	0.5000	10,500	210	1,050
	2	Weibull(36; 1.6; q)	32.2767	Exp(1.0)	1.0000	9,100	182	910
	3	Weibull(34; 1.5; q)	30.6933	Exp(1.0)	1.0000	8,300	166	830
	4	Weibull(32; 1.4; q)	29.1655	Exp(1.5)	0.6667	8,700	174	870
	5	Weibull(30; 1.2; q)	28.2197	Exp(1.5)	0.6667	8,000	160	800
2	1	Weibull(19; 1.2; q)	17.8725	Exp(0.8)	1.2500	6,400	128	640
	2	Weibull(29; 1.5; q)	26.1796	Exp(1.1)	0.9091	8,000	160	800
	3	Weibull(23; 1.2; q)	21.6351	Exp(0.9)	1.1111	7,600	152	760
	4	Weibull(27; 1.4; q)	24.6084	Exp(1.2)	0.8333	9,300	186	930
	5	Weibull(24; 1.3; q)	22.1658	Exp(1.0)	1.0000	7,000	140	700
	6	Weibull(31; 1.6; q)	27.7938	Exp(0.8)	1.2500	8,800	176	880
3	1	Weibull(22; 1.1; q)	21.2281	Exp(0.6)	1.6667	7,500	150	750
	2	Weibull(25; 1.7; q)	22.2322	Exp(0.5)	2.0000	6,900	138	690
	3	Weibull(26; 1.8; q)	23.6970	Exp(0.6)	1.6667	6,200	124	620

In order to analyze the impacts caused by different types of corrective maintenance actions (minimal, imperfect and perfect repairs) in the multi-objective optimization of system designs, the same application example was solved three times, each one considering a specific repair policy. Hence, the rejuvenation parameter (q) was equal to 1 when components were subjected to minimal repairs and equal to 0 when perfect repairs were executed. In the situation in which maintenance teams performed imperfect repairs, all q were set to 0.5. As in the validation examples, the considered mission time was $t_n = 730$ time units. For the MOGA, $N = 75$, $N_{\text{gen}} = 150$ and all the other parameters' values are the same as the ones shown in Table 3.4. Steps and replications of the DES portion were set to 30 and 100, respectively.

The number of globally nondominated solutions obtained from MOGA + DES for the minimal, imperfect and perfect repairs were, respectively, 78, 89 and 106, which are depicted in Figure 3.7. It can be observed that the points regarding perfect repairs are above the solutions related to imperfect repairs, which in turn, are over the compromise solutions associated with minimal repairs. Indeed, solutions of the perfect repair approach, in most cases dominate solutions concerning imperfect repairs and these latter results, in general, dominate minimal repair solutions. This is an expected behavior since the system with components subject to imperfect repairs may have an intermediate performance between systems with components that undergo either minimal or perfect repairs.

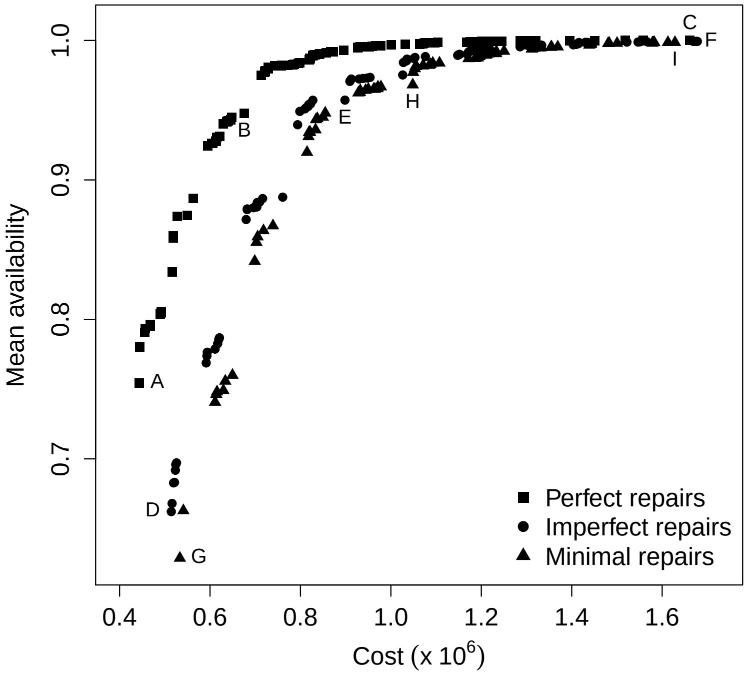


Figure 3.7: Application example MOGA + DES – Obtained Pareto fronts for each type of repair.

If a repairable system with perfect repairs is considered, one assumes that maintenance

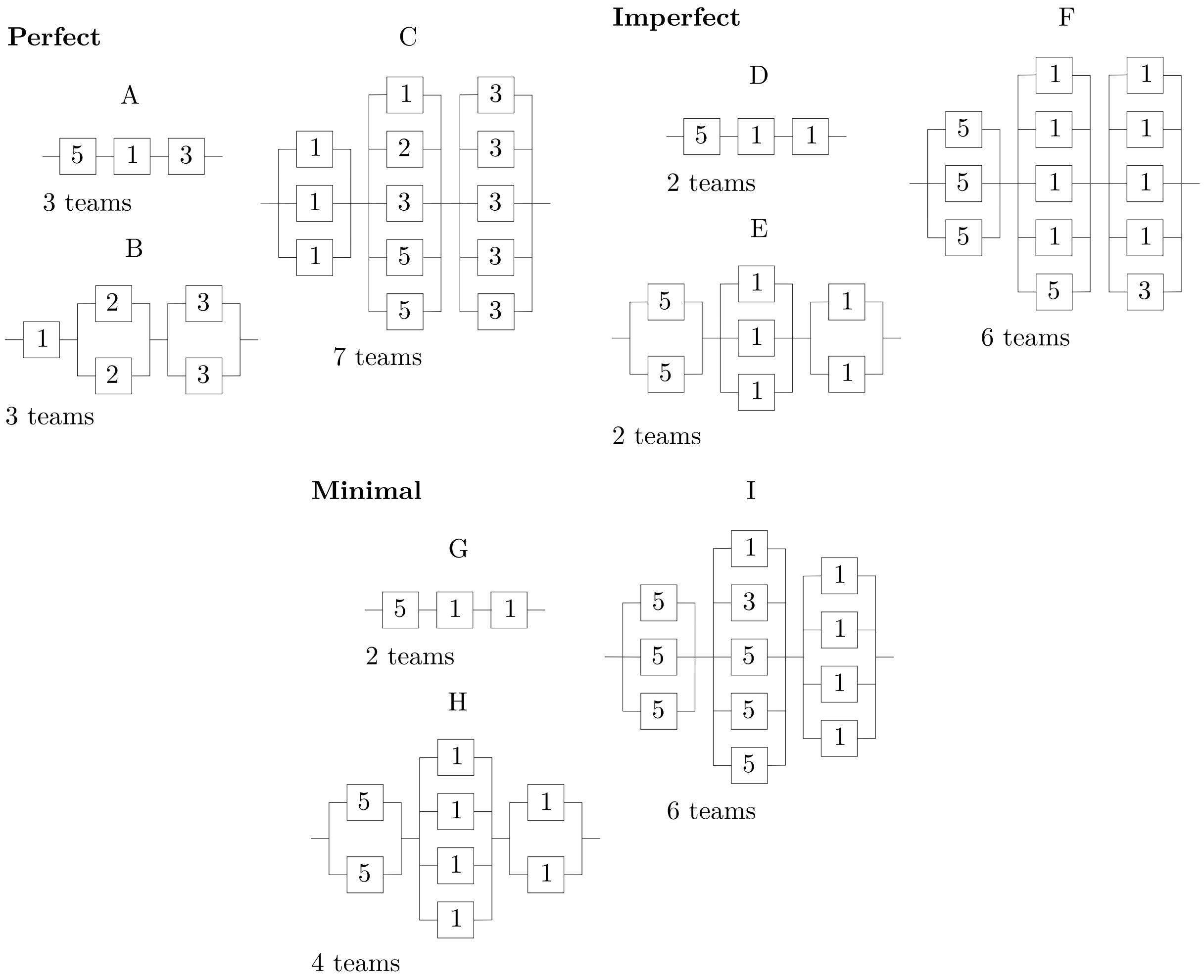


Figure 3.8: Application example MOGA + DES – selected solutions related to perfect, imperfect and minimal repairs

teams are so efficient that they can bring failed components to a condition "as good as new". If otherwise minimal repairs are supposed, maintenance teams can only repair a failed component and deliver it to operation in an "as bad as old" condition. However, in practice, imperfect repairs are often performed. Thus, in such real situations, if perfect repairs are assumed, system mean availability may be overestimated, whereas if minimal repairs are considered, it may be underestimated: note that points regarding perfect repairs in Figure 3.7 have the highest mean availabilities and the ones related to minimal repairs have the lowest mean availabilities. Figure 3.8 illustrates the system configurations related to some selected solutions from the three Pareto fronts.

3.6.2.1 Return on Invesment Analysis

Since all the Pareto solutions are optimal in a multi-objective perspective, a suggestion to the decision maker is to evaluate the gain in availability due to an investment in system design between pairs of solutions in the Pareto front, that is, to perform a ROI analysis.

Mathematically:

$$\mathrm{ROI} = \frac{A_i - A_j}{C_i - C_j}, \quad i \neq j, \tag{3.22}$$

in which A_i and A_j are the mean availabilities of configurations i and j, in this order, and C_i and C_j are their respective costs. As an example, consider the solutions indicated in Figure 3.7. It can be inferred from Table 3.7 that the investment in system design to gain about 0.3 in mean availability is 384,123.83 (from solution A to solution B). On the other hand, it is necessary to invest 770,855.97 to obtain an addition of about 0.04 in system mean availability (solution B to solution C). Thus, high investments on system design do not necessarily result in great gains on system mean availability. The same analysis can be done for the minimal and perfect repair cases and the ROIs of selected solutions related to each repair type are also shown in Table 3.7.

Table 3.7: ROI of selected solutions related to minimal, imperfect and perfect repairs.

Repair	Solution	Mean availability	Total system cost	ROI
Minimal	A	0.6288	533,345.37	6.5899×10^{-7}
	B	0.9682	1,048,371.86	5.2418×10^{-8}
	C	0.9986	1,628,322.45	
Imperfect	D	0.6622	514,312.39	7.6798×10^{-7}
	E	0.9572	898,436.22	5.4615×10^{-8}
	F	0.9993	1,669,292.19	
Perfect	G	0.7545	443,753.28	8.3226×10^{-7}
	H	0.9476	675,771.58	5.3070×10^{-8}
	I	0.9999	1,661,268.77	

3.7 Summary and Discussion

This chapter presented a procedure based on the coupling of a MOGA with DES to find compromise solutions for redundancy allocation problems related to systems subject to imperfect repairs. The genetic operators developed to handle only feasible solutions significantly shrink the search space to be explored by MOGA and the evaluation of fitnesses does not require penalty functions to tackle unfeasibility. The two considered objectives were the system mean availability and the system total cost. The second objective comprised not only the acquisition cost, but also other costs often incurred in practical situations such as operating and maintenance costs and the penalties due to system unavailability. Moreover, the number of maintenance teams was considered as a decision variable and the (un)availability of that resource during simulation time was also taken into account.

In order to tackle the unrealistic assumptions regarding the reliability behavior of systems that are imposed by the majority of the procedures to handle redundancy allocation problems, the system failure-repair process was modeled according to GRP providing a more realistic treatment of the problem. In this context, the proposed approach is able to circumvent the perfect or minimal repairs assumptions underpinning other approaches such as NHPP or RP when components are in fact subjected to imperfect repairs.

As it was demonstrated by means of the validation examples, the proposed MOGA was able to provide solutions on or very near the exact Pareto fronts. With the purpose of measuring the difference between real and simulated Pareto fronts, two distance metrics were presented. Their achieved values were satisfactory, all of them smaller than 10^{-2}. Moreover, the glsmoga provided more exact solutions of the real Pareto fronts than the MOACO, as well as better values for the presented distance metrics.

The application example was solved three times, each one considering a specific type of maintenance action. As expected, the solutions related to perfect repairs completely dominated the ones associated with either imperfect or minimal repairs. After obtaining the nondominated solutions, a ROI analysis was suggested to aid the decision maker in choosing a specific design and the number of maintenance teams to be hired. It was observed that considerable investments in system design do not necessarily result in great gains on system mean availability.

4 ELABORATION OF INSPECTION PLANS BY MULTI-OBJECTIVE GENETIC ALGORITHMS AND RISK-BASED INSPECTION

The multi-objective problem concerning the elaboration of inspection plans based on risk and cost is mathematically formalized in this chapter. The proposed solution framework combining MOGA and RBI is provided along with the description of the genetic operators adapted to return only feasible individuals (*i.e.* feasible inspection plans). The performance of the MOGA + RBI is evaluated by means of a validation example for which the true Pareto front is known. Also, the proposed methodology is applied to an example involving a separator vessel of oil and gas. The main results of this chapter can be found in Furtado *et al.* (2012).

4.1 Problem Statement and Formulation

An inspection plan x is represented by the following vector:

$$x = (x_{11}, \ldots, x_{1k}, \ldots, x_{1m}; \ldots; x_{j1}, \ldots, x_{jk}, \ldots, x_{jm}; \ldots; x_{n1}, \ldots, x_{nk}, \ldots, x_{nm}), \tag{4.1}$$

in which n is the number of available inspection techniques and m is the number of time steps considered in the planning horizon. Each element x_{jk} of x is either 0 or 1, $j = 1, \ldots, n$ and $k = 1, \ldots, m$. If $x_{jk} = 1$, then an inspection involving technique j is performed at period k. Otherwise, if $x_{jk} = 0$, then there is no inspection of type j at period k. In this way, at most one inspection using a given technique is allowed in the same period.

Given a planning horizon with m periods and a number n of inspection techniques, the multi-objective optimization problem consists in the selection of inspection plans x that represent the compromise between risk and cost and that satisfy the maximum allowed intervals between inspections ($t_{j,\max}$, $j = 1, \ldots, n$) for each considered technique. The problem is mathematically defined as follows:

$$\min_x \quad C(x) = \sum_{j=1}^{n} \left(c_j \cdot \sum_{k=1}^{m} x_{jk} \right) + c_p \cdot \sum_{j=1}^{n} \sum_{k=1}^{m} x_{jk} + c_d \cdot \sum_{j \in J'} \sum_{k=1}^{m} x_{jk} \tag{4.2}$$

$$\min_x \quad R(x) = \sum_{k=1}^{m} R(k) \tag{4.3}$$

$$\text{s.t.} \quad x_{j,\ell+1} + \cdots + x_{j,\ell+t_{j,\max}+1} \geq 1, \ \forall j, \tag{4.4}$$

where ℓ represents a period with an inspection ($x_{j\ell} = 1$) and can possibly assume any of the values $1, \ldots, m - t_{j,\max} - 1$. Thus, the constraints in Equation (4.4) concern the summations of inspections over groups of $t_{j,\max} + 1$ periods, which have to be at least 1, that is, at least one inspection using technique j have to be performed in such an interval. These constraints are usually related to inspection regulations devised for a specific type of equipments.

In the cost objective function (Equation (4.2)), c_j is the cost of performing technique j, c_p is the cost of qualified personnel per inspection, c_d is the downtime cost per period and J' is the set of techniques requiring equipment interruption to be executed. The risk objective function $R(x)$ (Equation (4.3)) concerns the total risk associated to an inspection plan x within the planning horizon m; it is the sum of the risks calculated at every period k, which are defined in Equation (2.13).

It is important to emphasize that the financial consequences that are part of the risk calculation (Equation (2.13)) and the cost objective function (Equation (4.2)) are of different nature. The former is related to expenditures due to failure occurrence, *e.g.* equipment and system repairs, environmental cleanup, injuries to personnel and nearby communities, legal penalties, among others. The latter, in turn, is associated to the inspection activity.

4.2 Proposed Multi-objective Genetic Algorithm

As in the case of the proposed MOGA for RAPs described in Chapter 3, the genetic operators are devised to allow only for feasible individuals, *i.e.* inspection plans. As previously commented, such an approach have the following advantages: (i) it reduces the search space to be explored by the MOGA; (ii) it prevents the MOGA from being lost in an unfeasible part of the search space; (iii) it avoids unnecessary fitness evaluations of unfeasible individuals; (iv) the use of penalty functions due to unfeasibility is not required. For the problem characterized in the previous section, the total number of solutions in the search space, when the constraints in Equation (4.4) are not taken into account, is given by $2^{m \cdot n}$. In turn, the number of feasible inspection plans for a given technique j is $a_{m,p_j} = [x^m]\left(f_{p_j}(x)\right)$, defined as the coefficient of x^m in the x-power series expansion of $f_{p_j}(x)$, where

$$f_{p_j}(x) = \frac{2(x + x^2 + \cdots + x^{p_j}) - x^{p_j}}{1 - x - x^2 - \cdots - x^{p_j}}, \tag{4.5}$$

for $p_j = t_{j,\max}+1$. Hence, the percentage of the search space concerning feasible inspection plans is given by

$$\frac{a_{m,p_1} \cdot a_{m,p_2} \cdot \cdots \cdot a_{m,p_n}}{2^{m \cdot n}} \cdot 100\%. \tag{4.6}$$

As an illustration, suppose $n = 1$, $m = 20$, $p = 3$ (thus, $t_{1,\max} = 2$). The x-power series expansion of $f_{p_1}(x)$ is

$$2x+4x^2+7x^3+13x^4+24x^5+44x^6+81x^7+149x^8+274x^9+504x^{10}+927x^{11}+1705x^{12}+3136x^{13}+5768x^{14}+10609x^{15}+19513x^{16}+35890x^{17}+66012x^{18}+121415x^{19}+223317x^{20}+\dots$$

Therefore the number of feasible inspection plans is $a_{20,3} = 223,317$ and it represents about 21.3% of the search space. For further details on enumeration techniques, the reader is referred to Lins (1981).

Next sections detail the individual representation along with the proposed genetic operators for the generation of initial population, crossover and mutation. Differently from the genetic operators presented in Chapter 3 for the multi-objective RAP, recursive algorithms are developed for the elaboration of inspection plans, since feasibility evaluation of $x_{j,\ell+1},\dots,x_{j,\ell+t_{j,\max}+1}$ is required whenever an inspection takes place (*i.e.* $x_{j\,\ell} = 1$) and for every technique $j = 1,\dots,n$. The selection and update of the auxiliary population P_{aux} is the same as presented in Section 3.2.3.

4.2.1 Individual Representation

An individual is represented by the vector in Equation (4.1), whose entries are either 1 or 0 if either an inspection using the related technique is or not performed in the associated period. As in the case of MOGA for RAPs, an integer representation of individuals are adopted. For the sake of illustration, suppose $n = 3$, $m = 6$, $t_{1,\max} = 2$, $t_{2,\max} = 3$, $t_{3,\max} = 4$ and the inspection plan in Table 4.1. It indicates that inspections using: (i) the first technique should be performed at periods 2 and 5; (ii) the second technique should be performed at periods 1, 3 and 4; (iii) the third technique should be performed at periods 1 and 6.

Table 4.1: Individual representation for MOGA (inspection plan)

	Technique 1						Technique 2						Technique 3					
Period	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
Inspection plan	0	1	0	0	1	0	1	0	1	1	0	0	1	0	0	0	0	1

4.2.2 Generation of Initial Population

Each of the N individuals of the initial population are randomly generated according to a discrete uniform distribution with additional features to handle unfeasibility, which are presented in Algorithm 6. The underlying idea of the algorithm is that, for a given technique j, every time an inspection is established ($x_{j,\ell} = 1$), a new group of values for the future periods $x_{j,\,\ell+1}, \ldots, x_{j,\,\ell+t_{j,\max}+1}$ is generated for feasibility investigation. In this way, an inspection to be carried out at period ℓ requires the restart of the algorithm from $\ell + 1$.

Indeed, groups of $t_{j,\max+1}$ periods, starting from $\ell + 1$, are considered one at a time. The values $x_{jr}, r = \ell+1, \ldots, \ell+t_{j,\max+1}$, are randomly set either to 0 or 1 and, whenever $x_{jr} = 1$, a recursion of the algorithm starting from $r+1$ is required. If all x_{jr} are equal to 0, then no inspection is performed within the considered period ($S = 0$) and the constraint in Equation (4.4) is violated. In order to tackle the unfeasibility, a position p among the ones taken into account is selected and the corresponding value is set to 1. Then, a recursion with p as argument is called. These steps are repeated until the final period m is reached. Once this reasoning is applied for each of the n considered techniques, a feasible individual is generated.

```
GenerateFeasiblePlan($j, t_{j,\max}, \ell, m, x$)
1. If $\ell \leq m - 1$:
   a. $S = 0$.
   b. For $r = \ell + 1, \ldots, \ell + t_{j,\max} + 1$ and $r \leq m$:
        I. Generate $x_{jr}$ from $\mathcal{U}(\{0, 1\})$.
       II. If $x_{jr} = 1$:
              i. $S \leftarrow S + x_{jr}$.
             ii. GenerateFeasiblePlan($j, t_{j,\max}, r, m, x$).
            iii. $r = m + 1$.
   c. If $S = 0$ and $\ell \leq m - t_{j,\max}$:
        I. $p \leftarrow \mathcal{U}(\{\ell + 1, \ldots, \ell + t_{j,\max} + 1\})$.
       II. $x_{ip} \leftarrow 1$
      III. $S \leftarrow S + x_{ip}$.
       IV. GenerateFeasiblePlan($j, t_{j,\max}, p, m, x$).
2. Return $x$.
```

Algorithm 6: Generation of a feasible plan for a given technique

4.2.3 Crossover and Replacement

Since the values of x_{jk} are either 0 or 1, the usual binary crossover (MICHALEWICZ, 1996) is performed between two individuals (parents, *e.g.* x^1 and x^2). The parents' positions are interchanged at randomly chosen cut points (c) so as to generate two new individuals: child 1 and child 2 that are respectively the modified x^1 and x^2, since the replacement is automatically performed as in Section 3.2.4. Figure 4.1-a depicts the crossover between two individuals when $n = 3$, $m = 6$, $c = 4$, $t_{1,\max} = 2$, $t_{2,\max} = 3$ and $t_{3,\max} = 4$. Then the investigation and handling of an eventual unfeasibility for each new individual, per technique, takes place. The algorithm used to perform these tasks is essentially the same as Algorithm 6; the only exception is step 1(b)I, given that in the crossover the values x_{jr} are not created. Notice that for the illustrated example, the crossover procedure generated an unfeasible offspring (child 2) that violated the maximum number of periods without an inspection using technique 1. In Figure 4.1-b, the unfeasibility is identified and a possible solution is given. As an outcome, child 2 becomes feasible.

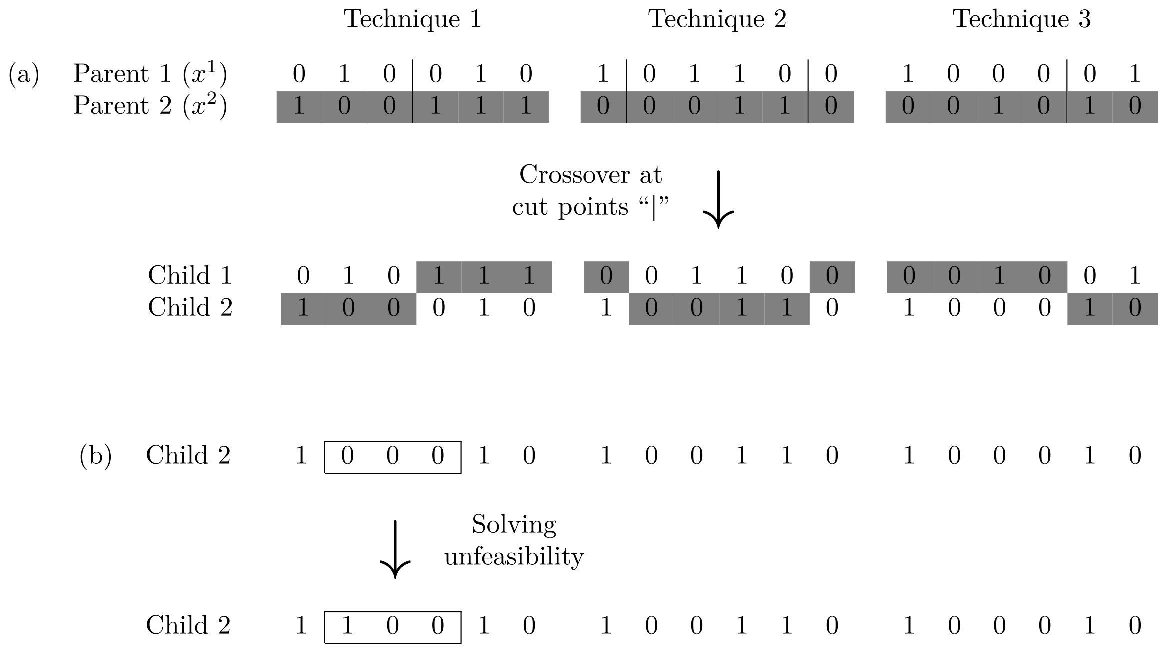

Technique 1
Technique 2
Technique 3
(a)
Parent 1 (x^1) 0 1 0 | 0 1 0   1 | 0 1 1 0 | 0   1 0 0 0 | 0 1
Parent 2 (x^2) 1 0 0 | 1 1 1   0 | 0 0 1 1 | 0   0 0 1 0 | 1 0
Crossover at cut points "|"
Child 1 0 1 0 1 1 1   0 0 1 1 0 0   0 0 1 0 0 1
Child 2 1 0 0 0 1 0   1 0 0 1 1 0   1 0 0 0 1 0
(b)
Child 2 1 0 0 0 1 0   1 0 0 1 1 0   1 0 0 0 1 0
Solving unfeasibility
Child 2 1 1 0 0 1 0   1 0 0 1 1 0   1 0 0 0 1 0

Figure 4.1: Example of binary crossover procedure (a); solving unfeasibility of child 2 (b)

4.2.4 Mutation

As in the case of the crossover, given that either $x_{jk} = 0$ or $x_{jk} = 1$ for all j and k, the traditional binary mutation (MICHALEWICZ, 1996) is applied. For every position a uniform random number $u \in [0, 1]$ is generated; given a position jk , if u is less or equal than the mutation probability (p_m), the value of x_{jk} is changed either (i) from 0 to 1 or (ii) from 1 to 0; otherwise, if $u > p_m$, x_{jk} remains the same.

Mutations that can render an individual unfeasible are only of type (ii), since additional inspections due to type (i) mutations by no means harm the individual's feasibility. In this way, whenever a mutation of type (ii) occurs, the related technique and period are respectively stored in vectors it and ip. Once all positions of an individual have been submitted to the binary mutation procedure, it and ip are of the same length ($|it| = |ip|$). If it and ip are both empty, which means no mutations of type (ii) have happened, the related individual is still feasible.

On the other hand, if $|it| = |ip| > 0$, it is necessary to investigate eventual unfeasibility arisen due to type (ii) mutations. If technique j and period k are, respectively, at the same positions of it and ip and the corresponding mutation resulted in an unfeasibility (a greater number of periods without inspections using technique j than permitted), the

x_{jk} value is restored to 1 as if no mutation had taken place at position jk. Otherwise, if a mutation of type (ii) has occurred but has not generated an unfeasibility, the product of the mutation remains unchanged, *i.e.* $x_{jk} = 0$. The idea is to modify individuals as least as possible after mutation operator has been applied in order to preserve the MOGA evolution trend.

Algorithm 7 summarizes the investigation procedure for unfeasibility over an individual for a given technique as well as their associated treatment in order to render the inspection plan feasible. Notice that steps until (iii) are basically the same as in Algorithm 6. These steps are necessary because of alterations in feasibility analysis due to eventual 1's provided by mutations of type (i) that also demand the recursion of the algorithm starting from the immediate subsequent position. The elements of it and ip that are eliminated in steps D and E are those involving already solved unfeasibility. Thus, at the end, if $|it| = |ip| > 0$, the remaining elements refer to positions that have been submitted to type (ii) mutations but have not generated unfeasibility.

```
RenderPlanFeasible($j, t_{j,\max}, \ell, m, x, it, ip$)
1. If $\ell \leq m - 1$:
   a. $S = 0$.
   b. For $r = \ell + 1, \ldots, \ell + t_{j,\max} + 1$ and $r \leq m$:
        I. If $x_{jr} = 1$:
             i. $S \leftarrow S + x_{jr}$.
            ii. RenderPlanFeasible($j, t_{j,\max}, r, m, x, it, ip$).
           iii. $r = m + 1$.
   c. If $S = 0$ and $\ell \leq m - t_{j,\max}$:
        I. $flag = 1$.
       II. For $q = 1, \ldots, |ip|$ and $flag$:
             i. If $\ell < ip[q]$:
                  A. $p \leftarrow ip[q]$.
                  B. $x_{j,ip[q]} = 1$.
                  C. $S \leftarrow S + x_{j,ip[q]}$.
                  D. $it \leftarrow it[-q]$.
                  E. $ip \leftarrow ip[-q]$.
                  F. If $|ip| > 0$, RenderPlanFeasible($j, t_{j,\max}, p, m, x, it, ip$).
                  G. $flag \leftarrow 0$.
2. Return $x$.
```

Algorithm 7: Evaluation and solution of eventual unfeasibility after mutation

4.3 Evaluation of Risk via Risk Based Inspection

The calculation of the cost objective is straighforward as Equation (4.2) is directly applied to the considered inspection plan. The risk objective, in turn, is computed by an RBI methodology using API (2008) recomendations. The financial consequences FCs via consequence analysis along with the gff and the F_{MS} are obtained only once at the beginning of optimization procedure. The damage factor $D_f(k)$, in turn, is updated according to the specific inspection plan provided by MOGA for every technique and period. In this way, the overall risk $R(x)$ is obtained. For further details in the risk evaluation by RBI, the interested reader is referenced to API (2008).

4.4 Overview

The proposed methodology couples the RBI methodology to an optimization procedure – MOGA – that entails constraints to comply with regulations. An overview of the MOGA + RBI is provided in Algorithm 8. Notice that most of the steps in Algorithm 8 are the same as in the Algorithm 5, however the used initial sampling, crossover and mutation are the ones detailed in Sections 4.2.2, 4.2.3 and 4.2.4 respectively.

```
1. Obtain $FC$, $gff$ and $F_{MS}$ via RBI.
2. Set $g = 0$ and generate initial population.
3. For $i = 1, \dots, N$:
   a. Compute $R(x^i)$ via RBI.
   b. Calculate $C(x^i)$.
4. Perform selection and update of $P_{\text{aux}}$.
5. If $g = N_{\text{gen}}$, go to step 8; else go to step 6.
6. Perform crossover and replacement.
7. Perform mutation, $g \leftarrow g + 1$, go to step 3.
8. Return $P_{\text{aux}}$.
```

Algorithm 8: Proposed MOGA + RBI

4.5 Application Example

In this section, the proposed MOGA + RBI is applied to obtain non-dominated inspection plans for a separator vessel of oil and gas by considering 3 inspection techniques and a horizon of 20 years. Internal and external corrosion are damage mechanisms to which the vessel is exposed. For such an equipment, the recommended times between inspections, according to the Brazilian Regulation Standard (NR-13), are 3, 7 and 15 years, respectively (MARANGONE & FREIRE, 2005). The fincancial consequences were obtained from the consequence analysis performed by Furtado *et al.* (2012). The RBI features are summarized in Table 4.2 and MOGA parameters are presented in Table 4.3. For this problem, the number of feasible inspection plans is in the order of 10^{17} and represents 50.77% of the entire search space.

The MOGA + RBI provided 15 non-dominated inspection plans comprising the Pareto set, which enabled the construction of the Pareto front of Figure 4.2. In this graph, three points A, B and C are identified, whose cost and risk values along with the associated schedule of inspections are shown in Figure 4.3. The gray cells represent that an inspection is to be performed in the associated period (column) using the related technique (row). On the other hand, the white cells indicate no inspection is required.

Table 4.2: RBI parameters

Technique	$t_{i,max}$ (years)	c_i
1	3	1,000.00
2	7	5,000.00
3	15	10,000.00
c_p	300.00	
c_d	$1,000.00$	
FC	$6,743,238.37$	

Table 4.3: MOGA parameters

Parameter	**Value**
Population size (N)	50
Number of generations (N_{gen})	100
Probability of crossover (p_c)	0.95
Number of cut points	9
Probability of mutation (p_m)	0.01

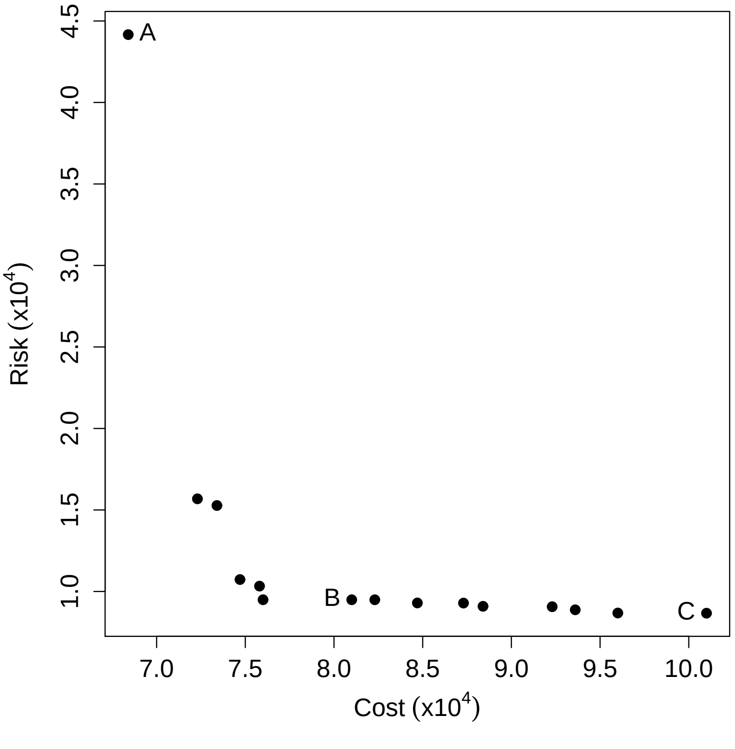


Figure 4.2: Application example MOGA + RBI – Obtained Pareto front

4.5.1 Return on Investment Analysis

The obtained non-dominated solutions can be submitted to ROI analysis so as to support inspection-related decision making. The ROI is given by:

Solution	Cost	Risk	Technique	Periods 1-20																			
A	68,400	44,163	1			▨	▨	▨		▨		▨		▨		▨		▨	▨			▨	▨
			2					▨		▨				▨	▨							▨	
			3						▨										▨				
B	81,000	9,492	1			▨	▨	▨	▨	▨			▨			▨		▨	▨		▨	▨	▨
			2			▨						▨		▨	▨			▨					
			3								▨									▨	▨		
C	101,000	8,667	1			▨	▨	▨		▨	▨			▨		▨			▨	▨			▨
			2	▨	▨		▨					▨					▨						
			3			▨														▨	▨	▨	▨

Figure 4.3: Application example MOGA + RBI – Selected inspection plans

$$\text{ROI} = \frac{R_i - R_j}{C_i - C_j}, \quad i \neq j, \tag{4.7}$$

where R_i and R_j are the risks related to inspection plans i and j and C_i and C_j are their respective costs. The ROI of solution A to B is about -2.75 monetary units, that is, the risk is reduced by 2.75 monetary units for every unit invested in inspection. On the other hand, the ROI is about -0.04 from solution B to C, which means that the reduction in risk for each monetary unit invested in inspection is only about 0.04. Thus, high investments in inspection are not necessarily translated into significant reduction in risk.

4.6 Summary and Discussion

In this chapter, a combination of MOGA and RBI was developed to provide efficient inspection plans in terms of both cost and risk. The genetic operators of MOGA were adapted for the creation of only feasible inspection plans in compliance with the maximum allowed time between inspections. In this way, there is a significant reduction in the space to be explored by MOGA and there is no need to handle unfeasible inspection plans during MOGA steps.

The RBI was used to assess the risk related to the inspection plans provided by MOGA. As the risk was considered as an objective to be optimized, the user defined risk target was not required. Additionally, each inspection plan was submitted to a cost evaluation, with which expenditures related to the inspection activity become known. This is not possible if only RBI methodology is adopted.

The proposed MOGA + RBI was applied to an example involving an oil and gas separator vessel. A ROI analysis was illustrated on the obtained nondominated inspection plans. It could be inferred that high investments on inspection do not necessarily yield a great reduction in risk. The results suggest that the MOGA + RBI with the post-optimization ROI analysis is an effective tool to support decisions related to equipment integrity.

5 THEORETICAL BACKGROUND – PART 2

This chapter provides the underlying theory related to the coupling of SVR with PSO and bootstrap methods presented in Chapters 6 and 7, respectively. An overview of SVM and a detailed description of SVR are given. A formulation of the SVR training problem to be solved by IP methods is also provided. Additionally, this chapter includes an introduction to the variable and SVR model selection problems that are tackled via PSO. Finally, the general ideas of the bootstrap techniques are presented.

5.1 Support Vector Machines

SVMs are well suited when the underlying process that maps a set of influential variables, represented by an input vector $x \in \mathbb{R}^m$, into the response variable of interest, denoted by the scalar output y, is not known or when analytical formulations relating them are difficult to be established. This feature renders SVM as a model-free or non-parametric approach.

In its classical formulation, SVM is a supervised learning method, given that the available set of observations D comprises not only the multidimensional inputs x, but also the associated outputs y, *i.e.* $D = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ (VAPNIK, 2000; SCHÖLKOPF & SMOLA, 2002). The set D is the so-called *training set* as it is used in the SVM learning (or training) phase.

Indeed, the SVM learning step concerns the resolution of a mathematical programming problem, whose objective function embodies the SRM principle that aims at balancing model's complexity and model's training accuracy in order to avoid underfitting and overfitting situations. The former case is characterized by models with low complexity that have both inaccurate training and generalization performances. The latter involves very complex models so specialized in the training examples that poorly generalize to unseen data (KECMAN, 2005). The behavior of training and generalization errors with respect to model complexity is illustrated in Figure 5.1.

Besides, the SRM principle was proved to be useful when dealing with small data sets. This is an advantage of SVM over other learning techniques such as ANNs that are suited for large training sets as they only involve the minimization of the training error by means of the ERM principle (VAPNIK, 1999).

The SVM training problem is a quadratic, thus convex, mathematical program, for which the Karush-Kuhn-Tucker (KKT) first order optimality conditions are not only necessary but also sufficient for a unique global optimum. In this way, SVM is not trapped in local optima as ANNs are (BOYD & VANDENBERGHE, 2004; SCHÖLKOPF & SMOLA, 2002).

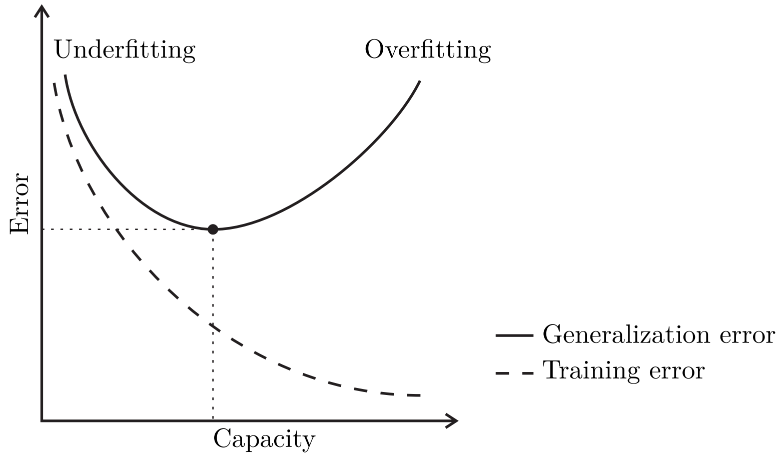


Figure 5.1: Relation between model complexity and errors

The nature of the scalar output defines the type of problem to be handled. If y assumes discrete values representing categories, then the problem is of classification. On the other hand, for real-valued y, one has a regression problem. In this work, only SVM for regression is taken into account. For an introduction to SVM classification, the interested reader is referred to Burges (1998), Schölkopf & Smola (2002) and Kecman (2005).

5.1.1 Regression via Support Vector Machines

Non-parametric regression can be formalized considering the response as a random variable Y generated by the model:

$$Y = \mu_Y(x) + \epsilon(x), \tag{5.1}$$

where $\mu_Y(x)$ is the deterministic but unknown expected value of Y and $\epsilon(x)$ is a random error term with zero mean and variance $\sigma_\epsilon^2(x) > 0$. SVR aims at estimating $\mu_Y(x)$ using the training set D for the adjustment of a regression expression of the form:

$$f(x) = w^T\phi(x) + b, \tag{5.2}$$

in which both w, the ℓ-dimensional weight vector, and b, the linear coefficient, are unknown. Also, in Equation (5.2), the operator ϕ maps x into a higher dimensional space $\mathcal{F}$ in order to account for possible nonlinearities between the input vector and the response variable. The underlying idea of the mapping ϕ is to translate a non-linear relationship between x and y in the input space into a linear association between $\phi(x)$ and y in $\mathcal{F}$, as illustrated in Figure 5.2 for the case of unidimensional x.

The SVR learning problem is as follows:

$$
\begin{aligned}
\min_{w,b,\xi,\xi^*} \quad & \frac{1}{2}w^T w + c \cdot \sum_{l=1}^{\ell} \xi_l + \xi_l^* && (5.3)\\
\text{s.t.} \quad & y_l - w^T\phi(x_l) - b \leq \varepsilon + \xi_l, \quad \forall\, l, && (5.4)\\
& w^T\phi(x_l) + b - y_l \leq \varepsilon + \xi_l^*, \quad \forall\, l, && (5.5)\\
& \xi_l \geq 0, \quad \forall\, l, && (5.6)\\
& \xi_l^* \geq 0, \quad \forall\, l, && (5.7)
\end{aligned}
$$

where the first part of the objective function (5.3) relates to the SVR model complexity and its ability in predicting data not in D and the second part is associated with training errors. The parameter c controls the compromise between these two parts. The slack variables ξ_l and ξ_l^* originate from Vapnik's ε-insensitivity loss function which forms a "tube" around the y values, such that errors are computed only for points lying outside it (VAPNIK, 2000; KECMAN, 2005):

$$
\begin{aligned}
|y_l - f(x_l)| - \varepsilon = \xi_l, \quad & \text{if observation } l \text{ is "above" the } \varepsilon\text{-"tube"}, && (5.8)\\
|y_l - f(x_l)| - \varepsilon = \xi_l^*, \quad & \text{if observation } l \text{ is "below" the } \varepsilon\text{-"tube"}, && (5.9)
\end{aligned}
$$

for all l. The parameter ε represents the "tube" width, *i.e.* the accepted deviation of $f(x_l)$ from y_l. Figure 5.2 contains a graphical visualization of the ε-"tube" formed by H_- and H_+ in $\mathcal{F}$ and of the ε-insensitivity loss function. Also, in the same Figure, pairs (x, y) associated to the circled dots lying on or beyond H_- and H_+ are named support vectors.

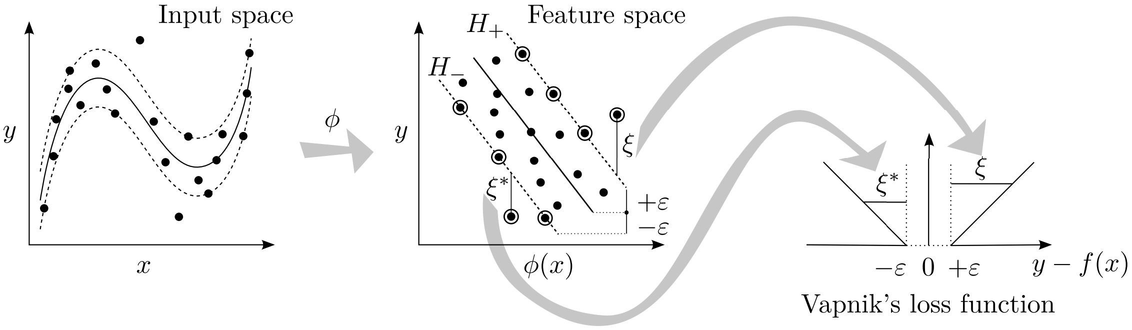


Figure 5.2: The role of mapping ϕ and Vapnik's ε-insensitivity loss function

Once the Lagrangian function related to the primal problem (5.3)-(5.7) is obtained, one may apply the KKT first order optimality conditions for a stationary point, so as to construct the dual form of the SVR training problem (see Appendix 1):

$$
\max_{\alpha,\alpha^*} \quad -\frac{1}{2}\sum_{l=1}^{\ell}\sum_{o=1}^{\ell}(\alpha_l - \alpha_l^*)(\alpha_o - \alpha_o^*)\phi(x_l)^T\phi(x_o) - \sum_{l=1}^{\ell}[\varepsilon(\alpha_l + \alpha_l^*) + y_l(\alpha_l - \alpha_l^*)] \quad (5.10)
$$

$$\text{s.t.} \quad \sum_{l=1}^{\ell}(\alpha_l - \alpha_l^*) = 0, \tag{5.11}$$

$$0 \leq \alpha_l \leq c, \quad \forall\, l, \tag{5.12}$$

$$0 \leq \alpha_l^* \leq c, \quad \forall\, l, \tag{5.13}$$

where α and α^* are ℓ-dimensional vectors comprising the Lagrange multipliers related to primal constraints (5.4) and (5.5), respectively. The resolution of the dual training problem provides the adjusted regression function:

$$f_0(x) = w_0^T\phi(x) + b_0 = \sum_{l=1}^{\ell}(\alpha_{l0} - \alpha_{l0}^*)\phi(x_l)^T\phi(x) + b_0, \tag{5.14}$$

in which the index 0 represents optimality. In fact, the optimal values assumed by α_l and α_l^* provide the classification of the training example (x_l, y_l) as: (i) *free support vector*, when $0 < \alpha_{l0} < c$ or $0 < \alpha_{l0}^* < c$; (ii) *bounded support vector*, if $\alpha_{l0} = c$ or $\alpha_{l0}^* = c$; (iii) *non-support vector*, if both α_{l0} and α_{l0}^* are equal to zero. Note that non-support vectors have no contribution in the regression function (5.14) as the support vectors are supposed to summarize all relevant information of the training set.

In practice, an appropriate mapping ϕ is often difficult to be determined and the calculation of the dot products that appear in Equations (5.10) and (5.14) may be computationally expensive. Fortunately SVR allows the use of kernel functions $K(x_l, x_o) = \phi(x_l)^T\phi(x_o)$, which are defined in the original space and can be used to implicitly compute the dot products (SCHÖLKOPF & SMOLA, 2002).

The Gaussian radial basis function (RBF)

$$K_{lo} = K(x_l, x_o) = \exp\left(-\gamma\, ||x_l - x_o||^2\right), \tag{5.15}$$

is the most widely used kernel function. One of the advantages of Gaussian RBFs is the fact that they are dependent only on the parameter γ to be tuned based on the available data. This is in line with the principle of parsimony according to which, provided that the achieved predictive ability is satisfactory, simpler models with less parameters are preferred. Also, Gaussian RBFs present relatively few numerical difficulties as compared, for example, with polynomial kernels whose values may go to infinity or zero as the degree of the polynomial increases or with the sigmoidal kernel, which may lose mathematical validity for certain values of the related parameters (HSU *et al.*, 2003; LIN & LIN, 2003). For further details on kernel functions and on the conditions they must satisfy, see Burges (1998), Cristiniani & Shawe-Taylor (2000), Vapnik (2000), Kecman (2001) and Schölkopf & Smola (2002).

The substitution of the dot products in Equation (5.10) by $K(x_l, x_o)$ does not affect the general outline for solving the dual problem. Therefore, the dot products in the estimated regression expression may be replaced by the kernel function as well. Given α, α^* and the kernel, there is no need to explicitly determine w_0, remaining only b_0 to be defined.

Active-set methods have been extensively applied to SVM training problems. The main idea of these methods is to solve equality-constrained problems, which are in general simpler than nonlinear programs with inequality constraints. The procedure involves guesses of the set of active constraints at the solution. Then, the supposed active constraints are imposed as equalities and the innactive ones are simply ignored. If the guess is incorrect, the methods use Lagrange multiplier information to drop one index from the current working set and add a new one until optimality conditions are satisfied (NOCEDAL & WRIGHT, 2006). The decomposition strategy used by Joachims (1999) in the open source SVM library SVM^{light} and the Sequential Minimal Optimization (SMO) implemented in LIBSVM (CHANG & LIN, 2001) are examples of active-set methods for solving SVM training problems. With these methods, the estimation of b_0 can be performed by using the free support vectors (KECMAN, 2005).

IP methods can also be used to solve SVM training problems. Contrary to the active-set methods, IP algorithms are not based on estimates concerning the true active sets. Also, the adoption of IP methods to solve the SVR dual problem provides b_0 as a byproduct, with no requirements of further computations. However, IP methods have been indicated for small to moderately sized SVM (PLATT, 1998; SCHÖLKOPF & SMOLA, 2002) due to the storage and handling of large matrices during the optimization process. In spite of that, Woodsend (2009) and Woodsend & Gondzio (2011) explored the particularities of SVM training problems so as to efficiently apply IP algorithms in large-scale situations. The authors mainly tackle linear ℓ_1-norm binary classification, even though general outlines for non-linear SVM classification and regression are also given. A brief description of SVR training using IP methods is presented in next section.

5.1.2 Support Vector Regression via Interior Point Methods

In order to describe the IP method to solve the SVR dual training problem (5.10)-(5.13), consider its matrix formulation as a classical quadratic programming problem:

$$\min_{z} \quad \frac{1}{2} z^T Q z - d^T z \tag{5.16}$$

$$\text{s.t.} \quad a^T z = 0, \tag{5.17}$$

$$0 \le z \le ce, \tag{5.18}$$

in which

- $z = [\ \alpha \quad \alpha^*\ ]^T$, $d = \hat{I}^T y - \hat{\hat{I}}^T \varepsilon e$ and $a = \hat{I}^T e$ are $2\ell \times 1$ vectors;
- $y = [y_1 \ldots y_\ell]^T$ is the vector of observed values of the response variable;
- e is a column vector of ones with the appropriate dimension;
- $Q = \hat{I}^T K \hat{I}$ is a positive semidefinite $2\ell \times 2\ell$ matrix;
- K is the $\ell \times \ell$ matrix with the kernel values K_{lo} as elements, for $l, o = 1 \ldots, \ell$;
- $\hat{I}$ and $\hat{\hat{I}}$ are $\ell \times 2\ell$ matrices defined as $[\ I \quad -I\ ]$ and $[\ I \quad I\ ]$, respectively, *i.e.* concatenations by columns of identity matrices $I_{\ell \times \ell}$.

Equations (5.16)-(5.18) form the *dual* of the SVR training problem, which is the *primal* problem to be solved by the IP methods briefly described in next sections. Also, as Q is positive semidefinite, the KKT first order conditions are not only necessary but also sufficient for global optimality. Thus, the resolution of the optimization problem essentially consists in solving the system of equations resulted from the KKT first order conditions (NOCEDAL & WRIGHT, 2006).

5.1.2.1 Primal-Dual Interior Point Method

The primal-dual IP method solves a modified version of problem (5.16)-(5.18), in which the inequality constraints are transformed into equality constraints via the introduction of 2ℓ-dimensional non-negative slack vectors s and t. Also, in order to handle the non-negativity of s and t, a log-barrier function is incorporated to the objective function (5.16):

$$\begin{aligned} \min_{z,s,t} \quad & \frac{1}{2} z^T Q z - d^T z - \mu_k \sum_{i=1}^{2\ell} [\ln(s_i) + \ln(t_i)] && (5.19) \\ \text{s.t.} \quad & a^T z = 0, && (5.20) \\ & z - ce + s = 0, \quad s > 0, && (5.21) \\ & -z + t = 0, \quad t > 0, && (5.22) \end{aligned}$$

where μ_k is a positive barrier parameter that is forced to decrease to zero as $k \to \infty$. The conditions of strict positiveness on the slack variables ($s > 0$ and $t > 0$) are implicitly considered during the step update (NOCEDAL & WRIGHT, 2006). The Lagrangian function associated with problem (5.19)-(5.22) is defined as

$$\mathcal{L}(p, \mu_k) = \frac{1}{2} z^T Q z - d^T z - \mu_k \sum_{i=1}^{2\ell} [\ln(s_i) + \ln(t_i)] + \lambda a^T z + \pi^T (z - ce + s) + \upsilon^T (-z + t), \quad (5.23)$$

where $p = (s, t, \pi, \upsilon, \lambda, z)$; $\lambda \in \mathbb{R}$, $\pi \in \mathbb{R}^{2\ell}$ and $\upsilon \in \mathbb{R}^{2\ell}$ are the Lagrange multipliers, also known as dual variables. An optimal point p_0 for (5.19)-(5.22) must be a stationary point of $\mathcal{L}(p; \mu_k)$ and satisfy the KKT first order conditions, which form the following system of non-linear equations:

$$\begin{aligned} S\pi &= \mu_k e, && (5.24)\\ T\upsilon &= \mu_k e, && (5.25)\\ z - ce + s &= 0, && (5.26)\\ -z + t &= 0, && (5.27)\\ a^T z &= 0, && (5.28)\\ Qz - d + \lambda a + \pi - \upsilon &= 0, && (5.29) \end{aligned}$$

where S and T are diagonal matrices whose diagonals are given by the vectors s and t, respectively. Equations (5.24) and (5.25) are the μ-complementarity conditions. Note that, from Equation (5.27), $t = z$; thus Equation (5.25) becomes $Z\upsilon = \mu_k e$, with Z as a diagonal matrix obtained from z. In this way, Equation (5.27) can be eliminated, *i.e.* t needs no longer to be explicitly considered, $p = (s, \pi, \upsilon, \lambda, z)$ and the system (5.19)-(5.22) can be rewritten as:

$$\begin{aligned} S\pi &= \mu_k e, && (5.30)\\ Z\upsilon &= \mu_k e, && (5.31)\\ z - ce + s &= 0, && (5.32)\\ a^T z &= 0, && (5.33)\\ Qz - d + \lambda a + \pi - \upsilon &= 0. && (5.34) \end{aligned}$$

An iteration of the primal-dual IP method involves the following steps:

1. Apply one step of the Newton's method to find the roots of (5.30)-(5.34), which involves the resolution of the following sparse linear system:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 \\ 0 & Z & 0 & 0 & \Upsilon \\ I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & a^T \\ 0 & -I & I & a & Q \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta \upsilon \\ \Delta \pi \\ \Delta \lambda \\ \Delta z \end{bmatrix} = - \begin{bmatrix} S\pi - \mu_k e \\ Z\upsilon - \mu_k e \\ z - ce + s \\ a^T z \\ Qz - d + \lambda a + \pi - \upsilon \end{bmatrix} \qquad (5.35)$$

2. Calculate the step length α_k in Newton's direction;

3. Update variables: $p_{k+1} = p_k + \alpha_k \Delta p$;

4. Reduce the barrier parameter μ_k.

These steps are repeated until the optimality conditions are reached. For a complete description of the primal-dual IP method and their most successful variants the reader is referred to Wright (1997), Nocedal & Wright (2006).

5.2 Variable and Model Selection Problems

Variable selection procedures can be generally divided into wrappers and filters (KOHAVI & JOHN, 1997). Wrappers consider the learning machine to score subsets of variables in accordance with their predictive power. Filters select a subset of variables in a preprocessing step and are independent from the learning machine used. For example, the work of Yang & Ong (2010) presents a wrapper method, whereas Wu & Wang (2009) use a filter for feature selection.

A common filter method is to rank variables according to their coefficient of determination between each one of them and the response variable (Y) (GUYON & ELISSEEFF, 2003). Indeed, such a statistic indicates the percentage of the total variability around the mean of Y explained by a linear fit between each regressor and Y. In this way, the use of the coefficient of determination as a ranking criterion enforces a variable ordering according to goodness of *linear* fit of individual variables.

Wrappers often give superior results, since information provided by the considered learning machine governs the search for the most relevant set of variables. These superior results are obtained at the expense of increased computational effort, even though it is not always the case (GUYON & ELISSEEFF, 2003). Indeed, backward elimination and forward selection are common wrappers usually associated with computational advantages. These procedures are incremental and at each step the subset of considered variables is modified.

The predictive ability of SVR greatly depends on the values of its hyperparameters c, ε and γ. Since the quest of the most suitable values for the SVR hyperparameters, known as model selection problem, is usually based on the specific data set under analysis, it may be performed whenever variable elimination or incorporation takes place. This renders the usual incremental wrappers computationally prohibitive, in spite of their original efficiency. For instance, Yang & Ong (2010) asserts that, like other wrappers methods, SVR hyperparameters are not re-tuned before performing each SVR training required by their variable selection procedure due to the increased computational effort required.

Thus, in the present work, variable selection along with SVR hyperparameters' tuning are simultaneously performed by PSO. Variable selection procedures in learning machines are comprehensively discussed by Kohavi & John (1997) and Guyon & Elisseeff (2003). For more on SVR model selection problems, the reader is referred to Momma & Bennett (2002), Ito & Nakano (2003), Yan *et al.* (2004), Pai (2006), Fei *et al.* (2009), Lins *et al.* (2010a) and Lins *et al.* (2012a).

5.3 Particle Swarm Optimization

For PSO, the basic element is a particle i, $i = 1, \ldots, n_{\text{part}}$, which is characterized by its current position in the search space (s_i), the best position it has visited (p_i) and its velocity (v_i). Also, a fitness function, *i.e.* the objective function to be optimized, is used to evaluate the particle performance. The n_{part} particles comprising the swarm fly throughout the search space towards an optimum by using both individual and collective information. This process is governed by the following update equations:

$$
\begin{aligned}
v_{ij}(t+1) &= \chi\cdot\{v_{ij}(t) + c_1 \cdot u_1 \cdot [p_{ij}(t) - s_{ij}(t)] + +c_2 \cdot u_2 \cdot [p_{gj}(t) - s_{ij}(t)]\}, && (5.36)\\
s_{ij}(t+1) &= s_{ij}(t) + v_{ij}(t+1), && (5.37)
\end{aligned}
$$

in which j regards the jth dimension of the d-dimensional search space, t indicates the time step (*i.e.* PSO iteration), χ is the constriction factor used to avoid huge velocity values, c_1 and c_2 are constants, u_1 and u_2 are uniform random numbers in $[0, 1]$ generated whenever the update takes place and for each j, $p_g = (p_{g1}, \ldots p_{gd})$ is the position associated with the best neighbor of particle i. Indeed, the second part of Equation (5.36) concerns the particle's cognition ability, whereas the third part is related to its social capacity of learning from its neighbors.

The number of neighbors each particle has (n_{neigh}) characterizes the swarm communication network. In a global topology (*gbest*), a particle is able to communicate with all others (*i.e.* the entire swarm, $n_{\text{neigh}} = n_{\text{part}}$), whereas in a local topology (*lbest*), a particle exchanges information with *some* of the others ($n_{\text{neigh}} < n_{\text{part}}$). As the *lbest* approach is part of the standard PSO suggested by Bratton & Kennedy (2007), this is the neighborhood structure adopted in this work. Also, some experiments performed by Lins *et al.* (2010a) contend that the *lbest* method is prone to require less computational effort (time and number of fitness evaluations).

Equation (5.37) may render particles unfeasible, since it can yield positions outside the search space, which is bounded by the predefined variables' intervals. In these situations, Bratton & Kennedy (2007) recommend the "let particles fly" strategy, which consists of skipping the fitness evaluation phase so as to avoid unfeasible positions from becoming the best. Besides the constriction factor χ, particles' velocities may be in $[-v_j^{\max}, v_j^{\max}]$, where $v_j^{\max}$ is the maximum velocity allowed for the j^{th} dimension. This procedure has been suggested to avoid particles from going too far beyond the feasible space.

The update of velocities and positions, and fitness evaluation phases are repeated until a stop criterion is met. For further details in PSO, the interested reader can consult Kennedy *et al.* (2001).

5.4 Bootstrap

Bootstrap is a computer intensive method, whose main idea is to resample from the original data, either directly or via a fitted model, in order to create replicate data sets. These replicates enable the variability assessment of the quantities of interest (DAVISON & HINKLEY, 1997). Bootstrap is particularly useful when no probabilistic model apply to the data under analysis and / or when the amount of data is not sufficient to use the central limit theorem.

Suppose a data set x with n elements. The boostrap begins by generating a large number of independent bootstrap samples $x^1, x^2, \dots, x^B$, each of size n. Corresponding to each bootstrap sample is a bootstrap replication of the statistic of interest $s(x^b)$, $b = 1, \dots, B$ (*e.g.* mean, media, standard error). The set of B replications can be used to construct an empirical probability distribution for s as well as confidence intervals (EFRON & TIBSHIRANI, 1993).

In the case of regression models, the bootstrap samples can be obtained based on pairs or on residuals. In the first case, the original data pairs (x, y) are sampled with replacement from the original data set and each of them have the same constant probability $1/\ell$ of being selected. The residuals sampling, in turn, require a fitted regression model over the original data set and the computation of residuals. Thus, the fitted model along with the resampled residuals are used to construct the bootstrap samples. The bootstrap techniques used linear regression is detailed in Efron & Tibshirani (1993) and Davison & Hinkley (1997). The adaptation for SVR is formalized in Chapter 7.

This chapter provided the theory underlying the proposed methodologies PSO + SVR and bootstrapped SVR detailed in Chapters 6 and 7, respectively. The first one is used to solve the variable and SVR model selection problems simultaneously and the second gives not only point estimates but also the confidence and prediction intervals related to the response variable of interest.

6 PARTICLE SWARM OPTIMIZATION FOR VARIABLE SELECTION AND SUPPORT VECTOR REGRESSION HYPERPARAMETER TUNING

In this chapter, a PSO + SVR is developed to simultaneously tackle the variable and SVR model selection problems commented in 5.2. Indeed, the methodology used in this work is an extension of the one presented by Lins *et al.* (2010a), which only involves SVR hyperparameter tuning. The PSO + SVR is applied to an example in the context of onshore oil wells and the predictive ability of SVR is evaluated on a reduced model – in which only a subset of variables identified as important is taken into account – and on a full model involving all available input factors. This chapter is based on Lins *et al.* (2011b).

6.1 Coupling Particle Swarm Optimization and Support Vector Regression

For the quest for SVR hyperparameters and variable selection, the PSO search space is formed by $d = 3 + n$ dimensions, where the first three regard c, ε, γ, in this order, and the remaining n are the variables r_h associated with regressors x_h, $h = 1, \ldots, n$. The latter variables are defined in the range $[0, 1]$ and if $r_h \geq 0.5$, the hth regressor is included in the model, otherwise it is not considered. The intervals of definition of c and γ are arbitrarily chosen, whereas ε is in

$$[0.001, 0.15] \cdot \frac{1}{\ell} \sum_{l=1}^{\ell} y_l. \tag{6.1}$$

In this way, the ε range is based on the considered training+validation set.

The proposed PSO + SVR is summarized in the flowchart of Figure 6.1. The initial particles velocities and positions are uniformly initialized considering the definition intervals of the PSO decision variables (particles are initially feasible). In this work, $v_j^{\max} = s_{ij}^{\max} - s_{ij}^{\min}$, for all i, and the initial maximum velocity range is set to $0.1\,[-v_j^{\max}, v_j^{\max}]$ in order to prevent particles from having great velocities in the early stages of the algorithm. After the initialization phase, particles' neighborhoods are determined, followed by an initial fitness evaluation of particles, which includes the update of their best positions. Then, particles' best neighbors, the overall best, velocities and positions are updated and fitness assessment phase is again reached and only performed for feasible particles. This cycle repeats until one of the following stop criteria is met: (i) the maximum number of iterations (n_{iter}) is achieved; (ii) consecutive iterations representing 10% of n_{iter} have provided the same best fitness value; (iii) the difference between two consecutive best fitness values is less than a tolerance δ.

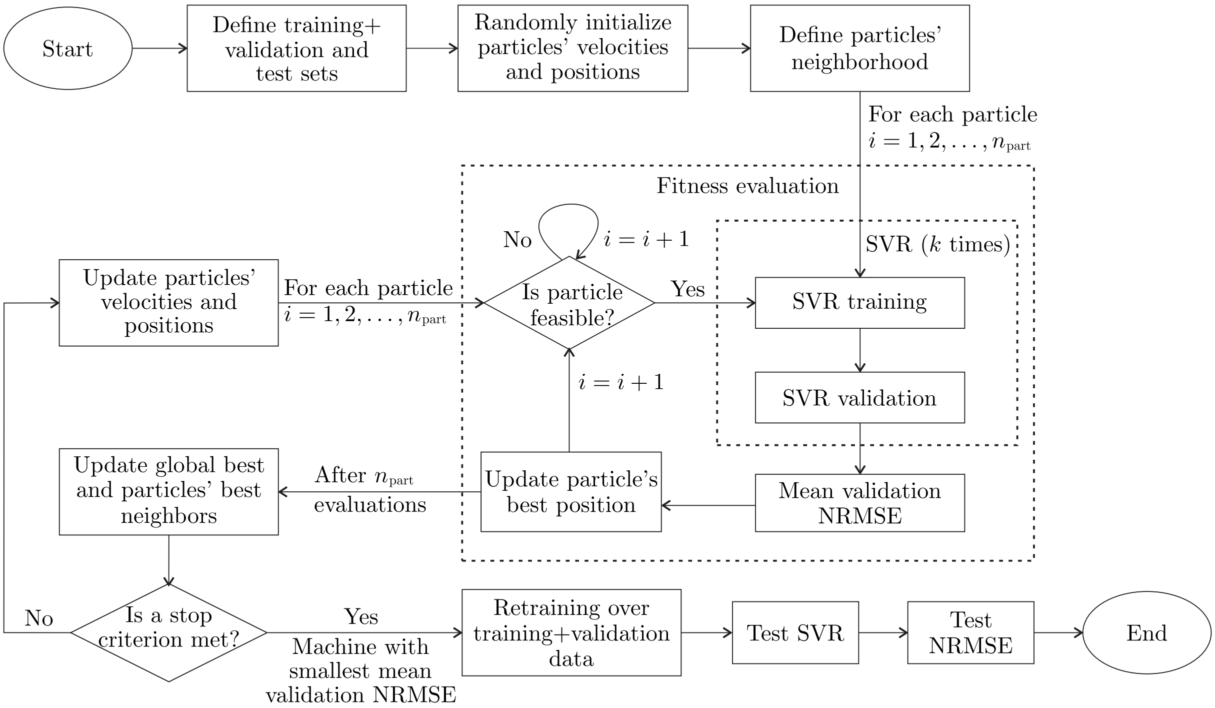

Start
Define training+ validation and test sets
Randomly initialize particles' velocities and positions
Define particles' neighborhood
For each particle $i = 1, 2, \ldots, n_{\text{part}}$
Fitness evaluation
No
$i = i + 1$
SVR (k times)
Update particles' velocities and positions
For each particle $i = 1, 2, \ldots, n_{\text{part}}$
Is particle feasible?
Yes
SVR training
$i = i + 1$
SVR validation
Update global best and particles' best neighbors
After n_{part} evaluations
Update particle's best position
Mean validation NRMSE
No
Is a stop criterion met?
Yes
Machine with smallest mean validation NRMSE
Retraining over training+validation data
Test SVR
Test NRMSE
End

Figure 6.1: PSO + SVR for variable selection and hyperparameter tuning

The adopted fitness function is the mean validation Normalized Root Mean Square Error (NRMSE) (Equation (6.3)), given that the present algorithm includes a cross-validation technique further detailed in Section 6.1.1. At the fitness evaluation step, the hybridism of the PSO and SVR takes place. For each set of hyperparameters' values and a subset of variables, *i.e.* for each particle position $s_i = (c_i, \varepsilon_i, \gamma_i, r_{i1}, \ldots, r_{ih}, \ldots, r_{in})$, the SVR portion performs k trainings and predictions so as to enable the computation of the mean validation NRMSE.

As a result, in relation to SVR accuracy in the validation phase, the PSO provides the optimal SVR hyperparameters values combined with the most relevant regressors to explain the variability of the response variable. Notice that the hyperparameters are tuned considering the selected set of variables and the problem of having inappropriate c, ε and γ for the group of chosen regressors is then avoided.

6.1.1 Cross-validation

In order to evaluate the prediction performance of SVR given the values of c, ε and γ and a set of variables, one may compute the NRMSE, defined by the following formula:

$$\text{NRMSE} = \sqrt{\frac{\sum_l (y_l - \hat{y}_l)^2}{\sum_l y_l^2}}, \tag{6.2}$$

where y_l is the observed value for the response variable and $\hat{y}_l$ is the related prediction provided by SVR.

In practice, the available data set is divided into two parts: (i) training + validation, formed by ℓ observations and (ii) test, with m cases. The k-fold cross-validation is performed over the training + validation part, which is further divided into k subsets. Each one of them, with ℓ/k elements, plays the role of a validation set, one at a time, and the remaining $k-1$ form the actual training set, upon which the SVR training is performed. This procedure is repeated k times until all subsets have been used as a validation set. Thus, k validation NRMSE values are obtained, which may be summarized as the mean validation NRMSE

$$\frac{\text{NRMSE}_1 + \cdots + \text{NRMSE}_k}{k}. \tag{6.3}$$

The index l in Equation (6.2) varies from 1 to ℓ/k for each NRMSE in Equation (6.3). The best NRMSE model (*i.e.* with optimal hyperparameters' values and most important variables) is related to the smallest value for the mean validation NRMSE.

After finding such a model, a retraining step takes place, in which the SVR optimization problem with all ℓ observations is solved. Finally, the test set is used to estimate the generalization ability of the obtained model by means of the test NRMSE, when $l = 1, \ldots, m$. The adoption of a cross-validation technique in the fitness evaluation step is also an extesion of the SVR + PSO presented in Lins *et al.* (2010a).

6.2 Application Example

The resulting PSO + SVR methodology is used for the prediction of TBFs of onshore oil wells located in the Northeast of Brazil. Onshore activities in this area date back to the beginning of the oil exploration in Brazil (ZAMITH & SANTOS, 2007). In spite of being related to mature wells of low productivity, these activities are responsible for a non-negligible part of the overall production. For example, for the period 2000-2010, onshore wells provided about 12% of the national oil production (ANP, 2011). Also, the prediction of TBFs of onshore wells permit the implementation of preventive actions to reduce or avoid failures and production downtime.

The data considered in this example originate from a database containing observations from 1983 to 2006 of TBFs and various aspects of different onshore wells located in the Northeast of Brazil. The database was analyzed by Barros Jr. (2006) and the rods were identified as one of the most critical components related to well failures. These equipments are responsible for transmitting the rotational energy of an engine to a pump, which artificially lifts the oil to the surface.

In this work, given the importance of the rods to the proper well operation, wells' failures are deemed to occur upon the failures of their installed rods. Hence, the present application aims at predicting wells' TBFs by means of SVR. The regressors shown in Table 6.1 are believed to influence the rods' (and wells') performance. Some of them are related to operational and environmental characteristics (x_1 - x_{11}) and others are associated with the previous failure and maintenance of the rods (x_{12} - x_{18}). For example, variables x_4 - x_7 regard the combination of rods of different sizes installed in the well, whereas x_{14} is related to the level of rods substitution in the previous maintenance action. The idea is to select a group of the the most relevant variables among x_1 - x_{18} concurrently with the choice of SVR hyperparameters by means of the PSO methodology presented in Section 5.3. The classification of the variables (C – categorical; N – numerical) are also given in Table 6.1. The categorical variables x_4 - x_{18} were handled by means of 0-1 dummy variables (MONTGOMERY *et al.*, 2006).

Table 6.1: Variables that can influence wells' TBFs

Var.	**Description**	**Type**	**Observed range or categories**
x_1	Well depth (m)	N	$[640, 830]$
x_2	Well production (m^3)	N	$[0.4, 22.4]$
x_3	% of water and solids	N	$[43, 98.3]$
x_4	Presence of 1" rods	C	1 - No, 2 - Yes
x_5	Presence of 7/8" rods	C	1 - No, 2 - Yes
x_6	Presence of 3/4" rods	C	1 - No, 2 - Yes
x_7	Presence of 5/8" rods	C	1 - No, 2 - Yes
x_8	Level of H_2S	C	1 - No, 2 - Low, 3 - High
x_9	Level of paraffin	C	1 - No, 2 - Low, 3 - High
x_{10}	Type of artificial oil lifting	C	1 - Mechanical, 2 - Progressive cavities
x_{11}	Filter type	C	1, 2, 3
x_{12}	Location of previous failure (on rods)	C	1, 2, 3
x_{13}	Rods' mode of previous failure	C	1, 2, 3, 4
x_{14}	Substitution of rods in prev. maint.	C	1 - None, 2 - Partial, 3 - All
x_{15}	State of 1" rods installed in prev. maint.	C	1 - New, ..., 3 - Old, 4 - Other
x_{16}	State of 7/8" rods installed in prev. maint.	C	1 - New, ..., 4 - Old, 5 - Other
x_{17}	State of 3/4" rods installed in prev. maint.	C	1 - New, ..., 4 - Old, 5 - Other
x_{18}	State of 5/8" rods installed in prev. maint.	C	1 - New, ..., 4 - Old, 5 - Other
Y	Time between failures (TBF)	N	$[10, 2039]$

After preprocessing the original database, a subset of 242 observations related to 26 wells located in the same geographical area was selected. From these, 192 (ℓ) were allocated to the training+validation phase and the remaining 50 (m) formed the test data, which yielded a proportion of approximately 4:1 between training+validation and test data. A 5-fold cross-validation techique is also considered. The PSO + SVR methodology was replicated 100 times to evaluate its stochastic performance and all of them involved the same PSO parameters ($n_{\text{part}} = 20$, $n_{\text{neigh}} = 8$, $n_{\text{iter}} = 5000$, $\chi = 0.7298$, $c_1 = c_2 = 2.05$) and decision variables' characteristics (Table 6.2). In order to avoid numerical problems because of the different scales of the output and input variables, each of them was scaled in $[0.1, 0.9]$. The scale parameters (minimum and maximum values of each variable) were obtained in the training+validation set and were also used to scale the test set. Thus, the valid range of ε is related to the scaled output and the obtained parameter values for c, γ and ε itself are all related to scaled data.

Table 6.2: Characteristics of PSO decision variables

Decision variable	**Range**	**Initial $v_j^{\max}$**
c	$[100, 1500]$	140
ε	$[0.0021, 0.0322]$	0.0301
γ	$[0.1, 150]$	14.9900
r_h, $h = 1, \ldots, 18$	$[0, 1]$	0.1000

The summary of the 100 PSO + SVR replications are shown in Table 6.3, in which the mean validation and test NRMSE values concern the original scale of the output variable. By means of the standard deviation of the mean validation NRMSE in the 100 runs, one can infer that the PSO was able to find essentially the same best value for the fitness function. The variation of the PSO decision variables indicates that the mean validation NRMSE is quite difficult to be tuned, since it may present several local minima with slightly different values. The model associated with the smallest test NRMSE, that is, the one with best generalization ability is described in Table 6.4 and it is referred henceforth as "optimal reduced model". It is important to emphasize that the test predictions to calculate the test NRMSE were obtained by means of the retrained SVR model considering $\ell = 192$ observations.

Table 6.3: Summary of 100 PSO+SVR replications

	Minimum	**Median**	**Maximum**	**Mean**	**Std. Dev.**
c	101.5147	826.0446	1498.9536	826.1644	359.6575
ε	0.0199	0.0255	0.0322	0.0276	0.0039
γ	0.1002	144.8759	149.9988	122.3001	40.6190
Number of selected variables	4	12	13	10.4300	2.3323
Mean validation NRMSE	0.7127	0.7148	0.7711	0.7303	0.0198
Test NRMSE (after retraining)	0.7108	0.9637	0.9874	0.9082	0.0883

From Table 6.4, notice that only 9 of the original 18 regressors were identified as relevant to describe the wells' TBFs. PSO + SVR returned a regression model in which the TBFs are function of the presence of 1" and 3/4" rods (x_4 and x_6), levels of H_2S and paraffin (x_8 and x_9), type of filter (x_{11}), mode of previous failure (x_{13}), level of rods' substitution in previous maintenance (x_{14}) and the state of 1" and 5/6" rods in previous maintenance (x_{15} and x_{18}). The level H_2S, for example, may influence the metal corrosion. Also, given that imperfect maintenance is usually performed, *i.e.* the system

Table 6.4: Characterization of the optimal reduced and full SVR models

	Optimal reduced model	**Optimal full model**
Hyperparameter	**Value**	**Value**
c	240.8256	178.5738
ε	0.0320	0.0231
γ	103.6400	149.9975
Selected variables	$x_4, x_6, x_8, x_9, x_{11}, x_{13}, x_{14}, x_{15}, x_{18}$	–
NRMSE	Value	Value
Mean validation	0.7509	0.7767
Training (after retraining)	0.6322	0.1251
Test (after retraining)	0.7108	0.8545
Number of support vectors	141	177

is not restored to its new condition, the variables related to the preceding failure and maintenance may indeed affect the time to next failure and are reasonable to be included in the model.

For comparison purposes, an SVR regression model considering all 18 variables presented in Table 6.1 was estimated by means of the PSO + SVR methodology. In this case, PSO was used only for tuning the SVR hyperparameters c, ε and γ. Once more 100 replications were performed and the model with least test NRMSE was chosen as the "optimal full model". Its characteristics are also shown in Table 6.4, in which the NRMSE values concern the original data scale. In spite of the greater training NRMSE related to the reduced model, it presented a general better generalization, given the evidence of the smaller mean validation and test NRMSE values when compared to the full model counterparts. Also, the reduced model is able to summarize the essential information of the training set with less support vectors (141 vs. 177). This reflects the principle of parsimony, which states that if two techniques adequately model a given data set, the one with less parameters may have superior predictive ability when handling new data (SEASHOLTZ & KOWALSKI, 1993). The prediction results provided by the optimal reduced and full models are presented in Figures 6.2 and 6.3, for the training and test sets, respectively. It can be noticed that the predictions of the full model over the test set assume the same value for a number of different cases. Otherwise, the reduced model has a better performance over the same test set, as its predictions tend to approach the real observations.

Additionally, with the plausible consideration of independence between PSO + SVR runs with and without variable seletion, a Wilcoxon-Mann-Whitney statistical test (HIGGINGS, 2004) was applied to compare the hyperparameters values provided by both approaches. The PSO + SVR with variable selection tends to return higher values for c

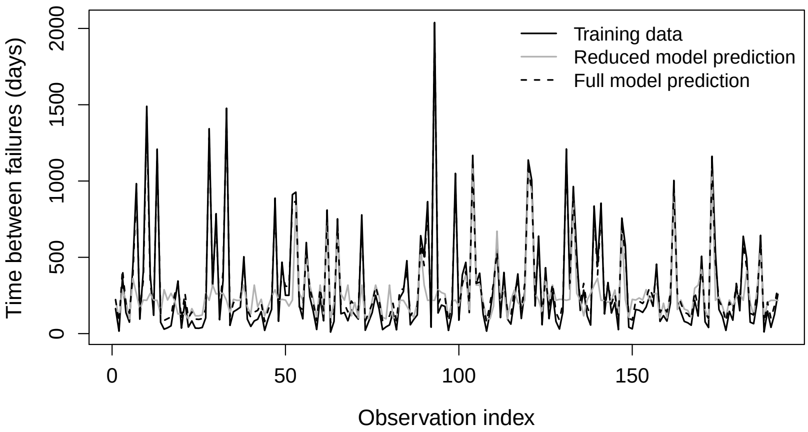


Figure 6.2: SVR training results

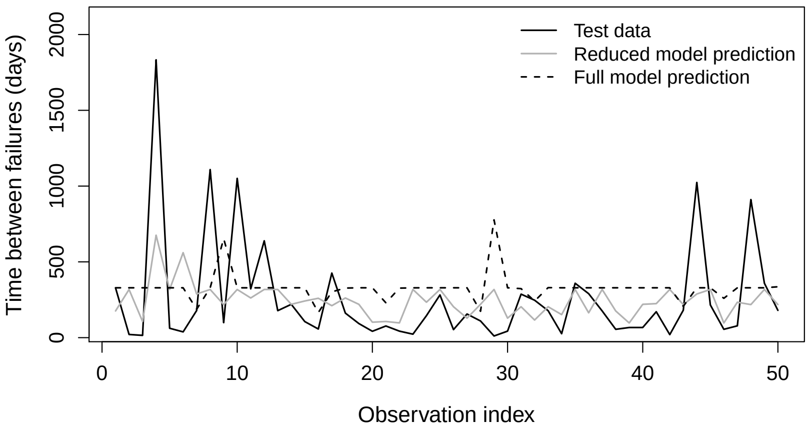


Figure 6.3: SVR test results

and ϵ and lower values for γ when compared to PSO + SVR without variable selection. The obtained p-value was 0.0905 for the test concerning c and, for the tests regarding the other two hyperparameters, the p-values were both smaller than $2.2 \cdot 10^{-16}$. Such an outcome suggests the importance of SVR hyperparameters' tuning whenever the set of considered variables changes.

6.3 Summary and Discussion

Previous works have shown the importance of SVR hyperparameters tuning to improve the SVR predicting performance. In this paper, besides the quest for the most suitable SVR hyperparameters, a variable selection procedure was implemented. The adoption of a PSO algorithm allowed the simultaneous application of both procedures.

The proposed PSO + SVR methodology was used on an application example from the Brazilian oil industry. The results show that the variable selection procedure enhanced the predictive ability of SVR. Also, a statistical test confirmed the necessity of the SVR hyperparameters' adjustment for the specific data set comprising the selected regressors. In general, the outcomes indicate that PSO + SVR is a promising tool for reliability prediction and it could be part of maintenance framework so as to support decisions concerning preventive actions.

7 UNCERTAINTY ASSESSMENT BY COUPLING BOOTSTRAP AND SUPPORT VECTOR REGRESSION

In order to add uncertainty-related information to the SVR point estimates of the response variable, a methodology involving bootstrap methods and SVR for the construction of confidence and prediction intervals is presented. Indeed, two different approaches based on two bootstrap methods frequently used in regression problems are provided. The first one involves pairs sampling and the second relies on residuals sampling. The boostrapped SVRs are applied to an artificial example and to a case study involving the prediction of scale growth rate on an equipment of the offshore oil industry. The performance of both bootstrapped SVRs are discussed. Some of the findings in this chapter can be found in Lins *et al.* (2012b).

7.1 Bootstrapped Support Vector Regression

There are two different ways of bootstrap sampling a regression model: one is based on pairs and the other on residuals (EFRON & TIBSHIRANI, 1993). The pairs scheme involves bootstrap samples $D^b = \{(x_{i_1^b}, y_{i_1^b}), \ldots, (x_{i_\ell^b}, y_{i_\ell^b})\}$, for $b = 1, \ldots, B$. The indices $i_1^b, \ldots, i_\ell^b$ are uniformly generated with replacement from the $1, \ldots, \ell$. Thus, the observed pairs $(x_1, y_1), \ldots, (x_\ell, y_\ell) \in D$, introduced in Section 5.1, are directly used to form the bootstrap sets. This type of resampling requires no assumptions on the errors $\epsilon(x)$ in Equation (5.1) other than independence (DAVISON & HINKLEY, 1997).

The second approach relies on the model adjusted over the original D, henceforth called f_0^0. This model provides the estimates $\hat{y}_l^0$, which enables the computation of the residuals $\hat{\epsilon}_l = y_l - \hat{y}_l^0, \forall\, l$. Given that the error term is supposed to have zero mean, one may adopt the centralized version of the raw residuals, $\tilde{\epsilon}_l = \hat{\epsilon}_l - \sum_l \hat{\epsilon}_l/\ell, \forall\, l$. Additionally, in order to consider the more general case involving heteroskedastic errors, instead of directly sampling the re-centered residuals, these are combined to Rademacher variables – defined as $\eta_l = 1$ with probability 0.5 and $\eta_l = -1$ with probability 0.5, $\forall\, l$ – given their efficient performance in practice (LIU, 1988; DAVIDSON *et al.*, 2007). Therefore, $\breve{\epsilon}_l = \tilde{\epsilon}_l \eta_l$ are sampled with replacement in order to obtain the sets $D^b = \{(x_1, \hat{y}_1^0 + \breve{\epsilon}_{i_1^b}), \ldots, (x_\ell, \hat{y}_\ell^0 + \breve{\epsilon}_{i_\ell^b})\}$, for $i_1^b, \ldots, i_\ell^b$ as previously described. In this setting the input vectors x remain fixed and the predictions $\hat{y}^0$ have their values perturbed by the residuals $\breve{\epsilon}$.

In both schemes, after the generation of each bootstrap set D^b, an SVR training is performed and the corresponding adjusted model f_0^b is stored. The bootstrapped SVR based on pairs and on residuals are summarized in Algorithms 9 and 10, respectively.

For a given observation of the input vector x_+, instead of taking $\hat{y}_+^0$ as estimate for

```
1. Train an SVR over $D$; obtain $f_0^0$.
2. For $b = 1, \dots, B$:
   a. Generate $i_1^b, \dots, i_\ell^b$ from $1, \dots, \ell$.
   b. Set $D^b = \{(x_{i_1^b}, y_{i_1^b}), \dots, (x_{i_\ell^b}, y_{i_\ell^b})\}$.
   c. Train an SVR over $D^b$; obtain $f_0^b$.
```

Algorithm 9: Sampling pairs

```
1. Train an SVR over $D$; obtain $f_0^0$.
2. Calculate residuals $\hat{\epsilon}_l = y_l - \hat{y}_l^0, \forall l$.
3. Re-center residuals, $\tilde{\epsilon}_l = \hat{\epsilon}_l - \sum_l \hat{\epsilon}_l/\ell, \forall\, l$.
4. For $b = 1, \dots, B$:
   a. Generate Rademacher variables $\eta_1, \dots, \eta_\ell$.
   b. Calculate $\breve{\epsilon}_l = \tilde{\epsilon}_l \eta_l, \forall\, l$.
   c. Generate $i_1^b, \dots, i_\ell^b$ from $1, \dots, \ell$.
   d. Set $D^b = \{(x_1, \hat{y}_1^0 + \breve{\epsilon}_{i_1^b}), \dots, (x_\ell, \hat{y}_\ell^0 + \breve{\epsilon}_{i_\ell^b})\}$.
   e. Train an SVR over $D^b$; obtain $f_0^b$.
```

Algorithm 10: Sampling residuals

$\mu_Y(x_+)$, the bagging (*b*ootstrap *agg*regat*ing*) predictor is taken (BREIMAN, 1996):

$$\hat{y}_+^{bag} = \frac{\hat{y}_+^0 + \sum_{b=1}^{B} \hat{y}_+^b}{B+1}, \tag{7.1}$$

where $\hat{y}_+^b$ is obtained from the respective adjusted model f_0^b (see Algorithm 11).

In this work, percentile intervals are adopted since they present advantages over intervals based on normality assumptions constructed with bootstrapped standard errors: they satisfy the transformation-respecting and range-preserving properties with no requirements of previous knowledge about appropriate transformations or specific ranges that the random variables must respect (EFRON & TIBSHIRANI, 1993).

The construction of percentile confidence intervals for $\mu_Y(x_+)$ is straightforward. For a given significance level α, one may take the $\alpha/2$ and $1-\alpha/2$ quantiles of $\hat{y}_+^0, \dots, \hat{y}_+^B$ as lower and upper bounds of the interval, in this order. Algorithm 11 summarizes these procedures.

In order to determine prediction intervals for Y_+ by bootstrapping residuals, an additional sampling is required to simulate the variation of Y_+ about its mean $\mu_Y(x_+)$. This is accomplished by estimating the distribution of the prediction error $\delta_+ = Y_+ - f_0(x_+)$ via $\hat{\delta}_+^{bm} = \left[\hat{y}_+^0 + \breve{\epsilon}_+^m\right] - \hat{y}_+^b$, where $m = 1, \dots, M$ and M is the number of resamplings. Afterwards, the $\alpha/2$ and $1-\alpha/2$ quantiles of $\hat{\delta}_+^{11}, \dots, \hat{\delta}_+^{1M}, \dots, \hat{\delta}_+^{B1}, \dots, \hat{\delta}_+^{BM}$ are taken to form the $(1-\alpha)100\%$ prediction interval for Y_+, whose limits are set as $\hat{y}_+^{bag} + \hat{\delta}_+^{\alpha/2}$ and $\hat{y}_+^{bag} + \hat{\delta}_+^{1-\alpha/2}$ (see Algorithm 12). Note that no other SVR trainings are demanded, only the modified version of the residuals $\breve{\epsilon}$ are constructed with no significant increase in computational effort.

Also, one might adopt the bootstrap based on pairs and use Algorithm 4 to simulate

```
1. For a given $x_+$:
   a. Obtain $\hat{y}_+^0, \ldots, \hat{y}_+^B$ from $f_0^0, \ldots, f_0^B$.
   b. Obtain $\hat{y}_+^{bag}$, Equation (7.1).
   c. $\mathrm{CI}[\mu_Y(x_+); \alpha] = \left[\hat{y}_+^{\alpha/2}, \hat{y}_+^{1-\alpha/2}\right]$.
```

Algorithm 11: Percentile confidence interval for $\mu_Y(x_+)$ by pairs or residuals bootstrapping

```
1. For a new $x_+$:
   a. For $b = 1, \ldots, B$:
         I. For $m = 1, \ldots, M$:
               i. Generate Rademacher variable $\eta_+^m$.
              ii. Generate $i_+^m$ from $1, \ldots, \ell$.
             iii. Calculate $\breve{\epsilon}_+^m = \tilde{\epsilon}_{i_+^m} \eta_+^m$.
              iv. Compute $\hat{\delta}_+^{bm} = (\hat{y}_+^0 + \breve{\epsilon}_+^m) - \hat{y}_+^b$.
   b. $\mathrm{PI}(Y_+; \alpha) = \left[\hat{y}_+^{bag} + \hat{\delta}_+^{\alpha/2}, \hat{y}_+^{bag} + \hat{\delta}_+^{1-\alpha/2}\right]$.
```

Algorithm 12: Percentile prediction interval for Y_+ by residuals bootstrapping

prediction intervals. But then f_0^0 would be used to obtain the residuals, which is against the essence of the pairs scheme of being totally free of model assumptions. It might be expected that a second level of bootstrap sampling would generate the desired variability of Y_+ about its mean. Thus, a double bootstrap (DAVISON & HINKLEY, 1997) was performed over the application examples presented in Section 7.2. However, the obtained prediction intervals were close to the confidence intervals at the expense of a more computer-intensive procedure as additional SVR trainings were required. Alternatively, one could estimate a prediction interval by means of a wider confidence interval, *e.g.* a 95% prediction interval could be approximated by a 99% confidence interval. However, preliminary experiments involving such an approach provided prediction intervals with low coverages. For further details on bootstrap, the interested reader can consult Efron & Tibshirani (1993) and Davison & Hinkley (1997).

7.2 Numerical Experiments

In both examples presented in this Section, the output variable as well as each input variable are scaled on $[0.1, 0.9]$ at each bootstrap iteration; the test set is scaled on the same interval but considering the lower and upper observed values of y and each component of x on the respective training set. The hyperparameters of SVR are $c = 100$ and $\varepsilon = 0.0025$ for all experiments, whereas $\gamma = 10$ for the simulated example and $\gamma = 150$ for the case study. These hyperparameters values have been found by trial. The PSO + SVR presented in Lins *et al.* (2010a), Lins *et al.* (2012a) and in Chapter 6 was not adopted as it would introduce an additional source of uncertainty, involving c, ε, γ. In this chapter, the goal is to create samples that could have been observed via bootstrap schemes and to assess their effects on the variability of the response variable. Additionally, since the bootstrap samples are different from each other, a PSO run would be demanded for each of them, which would require a prohibitive computational effort. Yet, as the bootstrap samples are all of equal sizes and originated from a unique source (*i.e.* the available data set), no drastic variations are expected on the values of the SVR hyperparameters. Thus, a single set of values for c, ε, γ may not harm the performance of the SVR models.

7.2.1 Simulated case

In order to evaluate the ability of the bootstrapped SVR in providing accurate point and interval estimates, the following deterministic process is analyzed (HESKES, 1997; ZIO, 2006):

$$\mu_Y(x) = \sin(\pi x)\,\cos(1.25\pi x) \tag{7.2}$$

with normally distributed heteroskedastic errors of zero mean and variance given by

$$\sigma_\epsilon^2(x) = 0.0025 + 0.0025\,[1 + \sin(\pi x)]^2. \tag{7.3}$$

Training sets $D = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ of different sizes ($\ell = 100, 200, 400$) were constructed by means of the Latin Hypercube Sampling (LHS – Helton & Davis (2003)) scheme with $f_X(x) = |x|$ as probability density function of the inputs $x_i \in (-1, 1)$. The main idea of LHS is to divide the range of the input variables into ℓ disjoint intervals of equal probability and then to select one value at random from each interval.

The test set comprised $n = 100$ samples, with x values equally spaced over the $(-1, 1)$. It is important to emphasize that the test points were generated independently from the training points and were, by no means, used to construct the bootstrap samples: they were treated as new observations.

The Mean Squared Error (MSE) (Equation (7.4)) and the quadratic bias (Equation (7.5)) were used as performance metrics for the bootstrapped SVR over the test set. The MSE measures the quadratic deviations occurred due to the adoption of $\hat{y}^{bag}$ as estimate for y and $\widehat{\text{bias}^2}$ is estimated by the average of the quadratic difference between $\hat{y}^{bag}$ and the true $\mu_Y(x)$. Note that $\widehat{\text{bias}^2}$ can be calculated because the true function of the mean is known, which is not usually possible in practice.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{bag})^2 \tag{7.4}$$

$$\widehat{\text{bias}^2} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{y}_i^{bag} - \mu_Y(x_i)\right]^2 \tag{7.5}$$

Bootstrap simulations with $B = 99, 499, 999$ were performed and the number of iterations to construct the prediction intervals via residuals sampling was defined as $M = (B + 1)/2$. The MSE and $\widehat{\text{bias}^2}$ values for the pairs and residuals approaches are presented, respectively, in Tables 7.1 and 7.2 for each combination of B and ℓ. Regarding the performance of the $\hat{y}^{bag}$ point estimates, the pairs sampling had overall advantage over the residuals scheme. For $\ell = 200$ and $\ell = 400$, the MSE values were smaller than for $\ell = 100$ and were quite similar within each bootstrap type with slightly better results for $\ell = 200$, mainly in the pairs scheme. For both bootstraps, the $\widehat{\text{bias}^2}$ values decreased as the training set sizes increased. For a fixed ℓ, the increase of B did not greatly impact the performance of $\hat{y}^{bag}$, specially when considering MSE.

Table 7.1: Simulated case – MSE (left) and $\widehat{\text{bias}^2}$ (right) over test set by bootstrapping pairs

B	$\ell = 100$	$\ell = 200$	$\ell = 400$
99	$6.583 \cdot 10^{-3}$; $6.764 \cdot 10^{-4}$	$5.564 \cdot 10^{-3}$; $3.113 \cdot 10^{-4}$	$5.877 \cdot 10^{-3}$; $9.158 \cdot 10^{-5}$
499	$6.603 \cdot 10^{-3}$; $6.912 \cdot 10^{-4}$	$5.530 \cdot 10^{-3}$; $2.844 \cdot 10^{-4}$	$5.900 \cdot 10^{-3}$; $9.820 \cdot 10^{-5}$
999	$6.522 \cdot 10^{-3}$; $6.555 \cdot 10^{-4}$	$5.520 \cdot 10^{-3}$; $2.693 \cdot 10^{-4}$	$5.876 \cdot 10^{-3}$; $9.348 \cdot 10^{-5}$

Table 7.2: Simulated case – MSE (left) and $\widehat{\text{bias}^2}$ (right) over test set by bootstrapping residuals

B	$\ell = 100$	$\ell = 200$	$\ell = 400$
99	$8.320 \cdot 10^{-3}$; $2.167 \cdot 10^{-3}$	$5.932 \cdot 10^{-3}$; $5.975 \cdot 10^{-4}$	$6.106 \cdot 10^{-3}$; $2.742 \cdot 10^{-4}$
499	$8.278 \cdot 10^{-3}$; $2.162 \cdot 10^{-3}$	$6.091 \cdot 10^{-3}$; $6.868 \cdot 10^{-4}$	$6.081 \cdot 10^{-3}$; $2.567 \cdot 10^{-4}$
999	$8.295 \cdot 10^{-3}$; $2.181 \cdot 10^{-3}$	$6.034 \cdot 10^{-3}$; $6.451 \cdot 10^{-4}$	$6.094 \cdot 10^{-3}$; $2.646 \cdot 10^{-4}$

As an example, Figure 7.1 depicts $\mu_Y(x)$ and $\hat{y}^{bag}$ for the test set considering the pairs and residuals schemes, $\ell = 200$ and $B = 499$. Notice that both approaches convey bagging estimates very near the true mean, which indicates the ability of the bootstrapped SVR in providing accurate point estimates. For the same setting, Figures 7.2 and 7.3 show the simulated confidence intervals ($\alpha = 0.05$) for bootstrapping pairs and residuals, respectively. The graph related to the residuals scheme (Figure 7.3) also depicts the prediction intervals for the response variable, which may be compared to the true intervals $[\mu_Y(x_i) \pm 1.96 \cdot \sigma(x_i)], i = 1, \ldots, n$. Note that the simulated intervals closely approximate the true ones. Confidence and prediction intervals are represented by lines so as to facilitate visualization.

In order to assess the coverage properties of the confidence and prediction intervals provided by the bootstrapped SVR, a Monte Carlo experiment with 5000 replicates was performed for each combination of B and ℓ. The 5000 training sets were generated by the

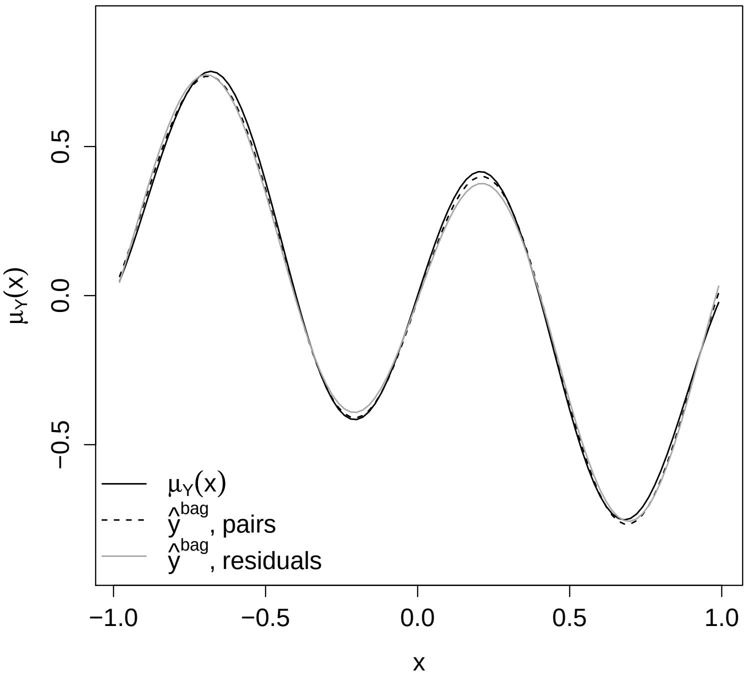


Figure 7.1: Simulated case – pairs and residuals bagging estimates vs. true mean

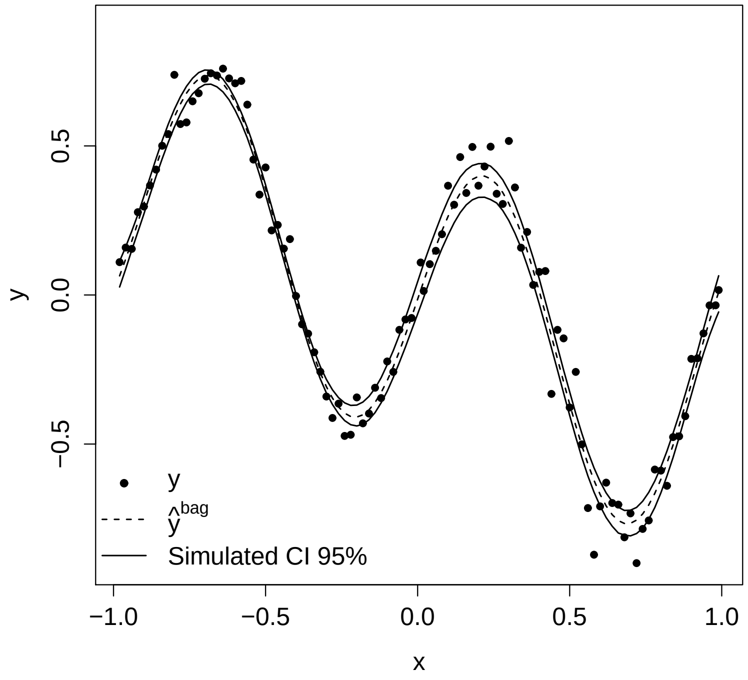


Figure 7.2: Simulated case – results over test set by bootstrapping pairs

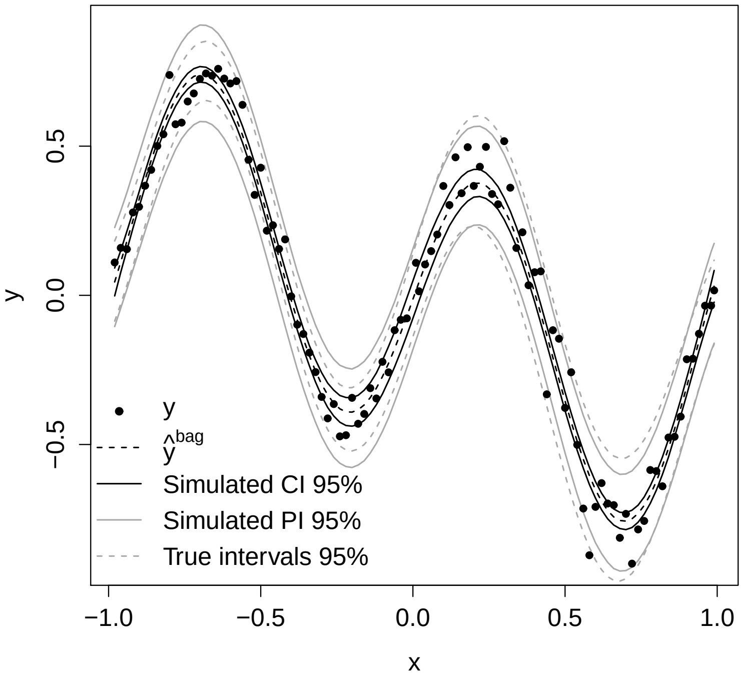


Figure 7.3: Simulated case – results over test set by bootstrapping residuals

LHS approach. The test sets were all formed by selected x_+^j, $j = 1, \ldots, 10$ (Table 7.3) and the corresponding y_+^j were randomly created from Equations (7.2) and (7.3).

The coverage results for both bootstrap schemes are reported in Table 7.3; the nominal level of significance was set to $\alpha = 0.05$, thus the nominal coverage was 95%. By the analysis of Table 7.3, it can be noticed the superior performance of the confidence intervals given by the pairs sampling over the ones provided by the residuals sampling. Indeed, the coverages of the confidence intervals of the latter approach are (much) lower than the nominal value. In this way, the reliance on f_0^0 as the true model that characterizes the residuals scheme negatively impacts the performance of the bootstrapped SVR. Nevertheless, the related prediction intervals presented excellent coverage values. The greater and the lower coverages – when compared to the nominal 95% – for the negative and positive x_+^j, respectively, are justified by the behavior of $\sigma_\epsilon^2(x)$ over the interval $(-1, 1)$ (Equation (7.3), Figure 7.4). Note that smaller variances are associated with the negative part of the interval, whereas greater variances are related to positive x. Hence, if a positive x is observed, the prediction of the response variable tends to be more difficult than if a negative x had been observed. Additionally, the increase of ℓ enhanced the performance of the bootstrapped SVR, since more information about the process was given in the SVR training step. On the other hand, the effects of the number of bootstrap iterations were less evident, as $B = 499$ and $B = 999$ provided similar results with a slight advantage over $B = 99$.

Table 7.3: Simulated case – coverage results (%) of the Monte Carlo simulation with 5000 replicates

		$\ell = 100$			$\ell = 200$			$\ell = 400$		
		Pairs	**Residuals**		**Pairs**	**Residuals**		**Pairs**	**Residuals**	
	x_+^j	CI	CI	PI	CI	CI	PI	CI	CI	PI
$B = 99$	-0.9	86.96	64.96	98.78	89.48	75.48	98.80	93.30	87.34	98.98
	-0.7	94.82	95.72	99.68	95.42	96.84	99.62	95.14	98.04	99.86
	-0.5	75.16	55.76	99.48	78.40	62.48	99.62	81.58	73.70	99.86
	-0.3	62.74	35.96	99.34	72.26	49.48	99.68	81.34	68.48	99.62
	-0.1	90.50	72.74	98.04	89.92	77.16	98.46	91.16	84.36	99.14
	0.1	84.42	55.92	91.48	87.40	61.96	93.06	89.72	72.14	93.94
	0.3	81.94	38.48	85.70	86.24	46.98	87.24	88.80	58.50	87.90
	0.5	85.92	44.64	83.70	87.50	50.10	84.42	89.34	59.26	84.32
	0.7	94.62	79.64	88.30	94.22	80.38	88.04	95.16	83.76	87.88
	0.9	88.70	56.58	93.90	89.90	64.64	94.04	92.76	77.80	94.18
$B = 499$	-0.9	88.00	65.62	98.68	90.48	76.36	98.82	93.90	88.32	99.02
	-0.7	95.76	96.46	99.70	95.88	97.14	99.60	96.58	98.58	99.88
	-0.5	77.14	57.28	99.42	79.06	63.84	99.62	83.12	76.06	99.84
	-0.3	63.76	36.92	99.40	73.54	49.00	99.66	82.04	70.04	99.64
	-0.1	91.66	73.76	98.02	90.74	79.02	98.56	92.36	85.48	99.06
	0.1	85.52	56.84	91.32	88.00	63.18	93.20	90.84	72.98	94.10
	0.3	82.18	38.58	85.58	87.32	47.32	87.56	89.78	58.86	88.00
	0.5	87.28	45.16	83.86	88.26	50.68	84.44	90.32	59.18	84.64
	0.7	95.44	80.34	88.40	95.14	80.60	88.04	95.72	84.54	87.70
	0.9	89.84	56.56	93.80	91.38	65.48	94.16	94.04	78.66	94.16
$B = 999$	-0.9	88.24	65.72	98.70	90.70	76.74	98.80	94.02	88.08	99.00
	-0.7	95.78	96.70	99.70	95.88	97.30	99.60	96.44	98.82	99.88
	-0.5	76.92	56.90	99.46	79.26	64.20	99.62	83.54	76.08	99.84
	-0.3	63.96	37.18	99.38	73.48	49.06	99.66	82.36	70.24	99.64
	-0.1	92.14	73.70	97.96	91.02	79.02	98.52	92.42	85.46	99.06
	0.1	85.62	56.46	91.32	88.28	63.14	93.20	90.88	73.12	93.96
	0.3	82.84	38.88	85.62	87.56	47.88	87.40	90.08	59.20	87.98
	0.5	87.80	45.30	83.80	88.26	50.56	84.34	90.54	59.44	84.54
	0.7	95.42	80.18	88.42	95.18	81.44	88.06	95.94	84.74	87.64
	0.9	90.14	56.64	93.92	91.64	65.74	94.14	94.12	78.62	94.12

As a general outcome, the bootstrapped SVR gave accurate interval estimates for the mean response, mainly by the pairs approach, and for the response variable itself via the residuals scheme. Also, great values for B are not necessarily required in order to obtain satisfactory results; this is an interesting feature as far as computational effort is concerned. However, in practical problems, the number of training examples is often a non-adjustable parameter as it is determined by budget constraints, *e.g.* an additional observation can represent more experiments involving the phenomenon under analysis.

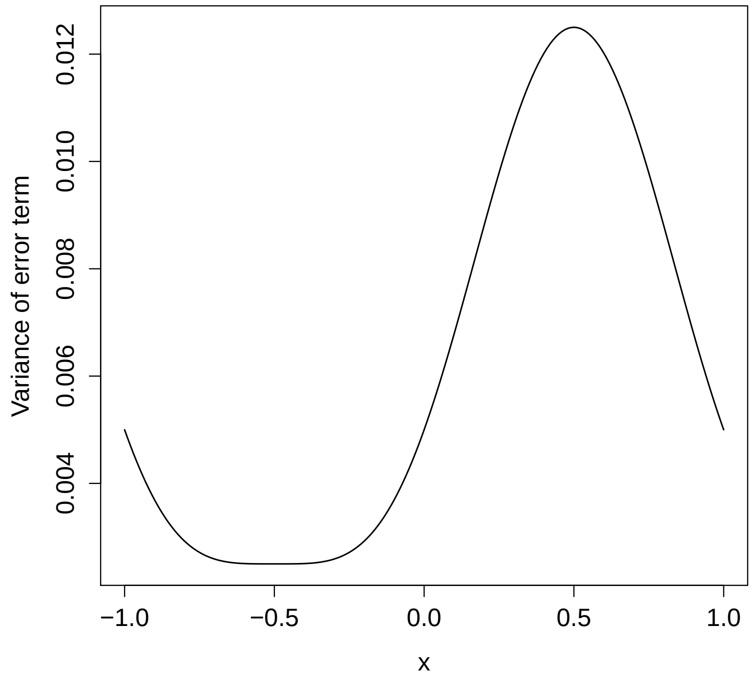


Figure 7.4: Simulated case – variance behavior over the interval of x

7.2.2 Case Study: Prediction of Scale Rate on Metal Surfaces

Scaling build-up in subsea oil well systems is a result of the combination of a set of interacting variables, such as reservoir temperature, pressure and water composition, that characterize the subsea environment. These factors can be tracked to predict the amount of scale that will be deposited in the future and determine the time to next maintenance action for removing the scale layer before it leads to equipment failure (MOURA *et al.*, 2011; AK *et al.*, 2012).

Some kinetic approaches have been used to predict scale formation, accounting for the effects of environmental variables (ZHANG *et al.*, 2001; DYER & GRAHAM, 2002; CHEN *et al.*, 2005). These prediction models take the form of a deterministic mathematical formula relating the scale output variable to the multi-dimensional input variable. Establishing such formula is often not easy in practice, and uncertainties need also to be taken into account.

Thus, in the present case study, the proposed bootstrapped SVRs are applied so as to give point and interval predictions of the scale growth rate on metal surfaces of an equipment used in offshore oil wells. If the scale layer achieves a predefined width, the equipment fails to properly perform its function. Some experiments were performed as an attempt to reproduce the subsea environment so as to observe the deposited scale layer. The following influential variables were observed: (i) temperature (T) and (ii) pressure (P) maintained over the tests, (iii) water composition (W), which characterizes the concentration of carbonates that might be expected in the real scenario and (iv) fluid velocity (V) near the metal surfaces. The response variable Y – scale growth rate – can be described by the general formula of Equation 5.1 with $x = (t, p, w, v)$.

The experiments performed gave a set of 131 observations. From these, about 90% ($\ell = 118$) are allocated for SVR training and the remaining 10% ($n = 13$) comprise the test set. Confidence and prediction intervals have been obtained for the test samples via the bootstrapped SVR with $B = 99$. For the construction of prediction intervals by the residuals scheme, M is set equal to 50.

The MSE values are $6.1234 \cdot 10^{-3}$ and $5.8613 \cdot 10^{-3}$ for the pairs and residuals bootstrapping, respectively. Differently from the simulated example, for this case study the residuals approach has a slightly superior performance. Figures 7.5 and 7.6 depict the real observed values y, the estimates $\hat{y}^{bag}$ and the confidence intervals for the mean response. Figure 7.6 also presents the prediction intervals for the various test points. Once again, line representation is adopted only for visualization purposes.

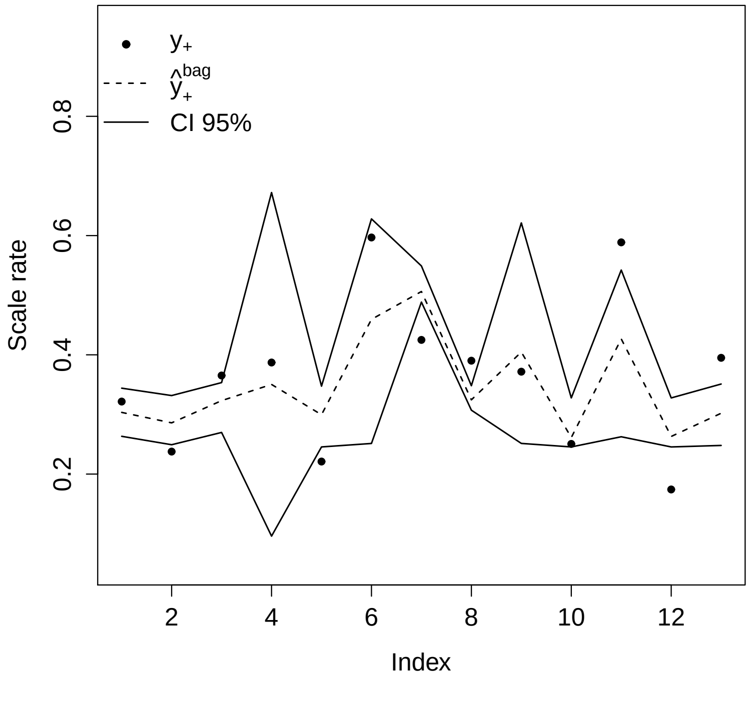


Figure 7.5: Study case – results over test set by bootstrapping pairs

Apart from obtaining accurate point estimates for Y, the bootstrapped SVR provides interval estimations related to the output variable Y. As seen in Figures 7.5 and 7.6, the confidence intervals provided by the residuals approach have smaller widths than the ones given by the pairs scheme. This is probably due to the use of f_0^0 in the sampling of the bootstrap based on residuals, as discussed in Section 7.2.1.

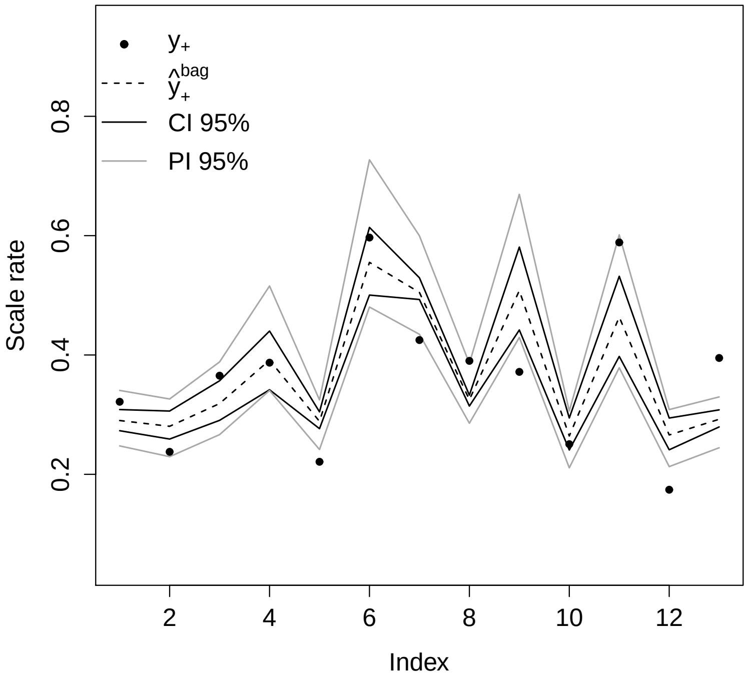


Figure 7.6: Study case – results over test set by bootstrapping residuals

With the interval estimates for the scale rate, a threshold for the thickness of the scale layer and the fact that scale rate = thickness/time, point and interval estimates related to the time to attain the threshold can be obtained so as to support maintenance-related decisions. As an illustrative example, consider the first test point x_1. Suppose that the scale rate is given in cm/day and that 5cm is the threshold value. Table 7.4 presents the point and interval estimates associated to the scale rate, which are directly provided by the bootstrapped SVR along with the corresponding estimates for the time to reach the threshold. The results indicate that a prudent decision would be to perform a preventive maintenance action for the removal of the scale layer on day 14.

Table 7.4: Illustrative example – point (PE) and interval (CI, PI) estimates for the scale rate and for the time to attain threshold

		Scale rate (cm/day)	**Time to attain threshold (days)**
Pairs	PE	0.3059	16
	CI	$0.2637 < \mu_Y(x_1) < 0.3512$	$14 < \mu_{\text{time}}(x_1) < 19$
Residuals	PE	0.2886	17
	CI	$0.2749 < \mu_Y(x_1) < 0.3051$	$16 < \mu_{\text{time}}(x_1) < 18$
	PI	$0.2488 < Y(x_1) < 0.3447$	$14 < \text{time}(x_1) < 20$

7.3 Summary and Discussion

In this chapter, different bootstrap methods (pairs and residuals samplings) were combined with SVR for the evaluation of uncertainty about the response variable when analytical and/or stochastic models describing it are not available. Moreover, instead of using the outcome of a singe model, *e.g.* the one trained over the original D, as a point estimate, the bootstrapped SVR provided the more accurate bagging estimates (BREIMAN, 1996).

The proposed methodology was applied to a simulated example in order to assess its ability in providing accurate point and interval estimates. The small values of MSE and $\widehat{\text{bias}^2}$ indicate that the outcomes of the bootstrapped SVR are accurate. For the evaluation of the coverage properties of the obtained intervals, a Monte Carlo experiment was performed and analyses about the effects of the bootstrap schemes, of the number of bootstrap iterations and of the training set size were also carried out. As general results: (i) confidence intervals are better estimated by the pairs scheme and the prediction intervals given by the residuals sampling have good coverages; (ii) increasing the bootstrap replications does not necessarily improve the performance of the bootstrapped SVR; (iii) larger training sets are expected to give better coverages, but for practical applications the data set is often subject to budget constraints.

A real case study involving the prediction of scale growth rates on metal surfaces as a function of four influential variables representing the environment was successfully handled by the bootstrapped SVR. Also, an example to illustrate how the obtained results can support maintenance-related decisions was provided. In this way, the bootstrapped SVR can be part of a more general framework for the establishment of maintenance policies.

The proposed methodology is based on non-parametric techniques (SVR and bootstrap), which enables its application in a variety of practical situations as long as empirical data of inputs and related response is available. The SVR allows for the treatment of complex processes for which the mapping of inputs into output is unknown. However, once the regression function is estimated in the training step, the prediction becomes a trivial mathematical exercise and the application of a given observation of the inputs on the estimated formula will always return the same predicted value for the response, independently of how many times it is calculated. This drawback of SVR (and of all regression methods, *e.g.* linear regression) is overcome by its coupling with bootstrap techniques that have the advantage of not requiring assumptions of the probabilistic models related to the inputs and to the response. In this way, the bootstrapped SVR provides not only a point estimate of the response, but also an associated interval of probable values, which is a valuable information in practical decision-making.

8 CONCLUSION

This work proposed models for risk and reliability quantification of systems and equipments in different phases of their life cycle (design and operation). Concerning the design phase and in order to solve a generalization of RAPs, a MOGA was developed. Not only the trade-off between cost and availability was handled, but also the behavior of the system was taken into account during the optimization procedure. The results from the validation examples indicate a superior performance of the proposed MOGA when compared to a MOACO, as the simulated fronts provided by the former were closer to the real Pareto fronts than the simulated fronts given by the latter.

Regarding the operational phase, a similar MOGA was adapted to the context of multi-objective inspection plans in which cost and risk were the objectives to be both minimized. The proposed MOGA + RBI was used in an example involving a oil and gas separator vessel, three different inspection techniques and a planning horizon of 20 years. As an outcome, nondominated inspection schedules were obtained. In this way, the shortcomings of the RBI methodology when applied alone were overcome, since a risk target level was not required and the costs related to inspection activity were considered as an objective itself.

In both proposed MOGAs, the genetic operators sampling, crossover and mutation were devised to provide only feasible individuals. In the case of multi-objective RAPs, the configurations were subject to physical constraints concerning the minimum and maximum number of components per subsystem, whereas the inspection plans were subject to regulations' requirements. Such an adaptation of the genetic operators enabled a reduction of the search space explored by MOGA that was prevented from being lost in unfeasible regions. Additionally, unnecessary fitnesses evaluations – sometimes involving expensive computational procedures – of unfeasible individuals were not performed.

After obtaining the Pareto front from MOGA, a ROI analysis was suggested to aid the decision maker in choosing a specific solution (system design and number of maintenance teams or an inspection plan). It was observed that considerable investments in system design are not necessarily translated into a great gain on system mean availability. Analogously, considerable investments in inspection do not certainly imply a great reduction in risk.

Also related to the operational phase of systems and equipments, the SVR learning method was used for the prediction of TBFs of onshore oil wells and of failures related to the scale growth rate on an equipmet used in offshore systems. In the first case, the SVR was combined to a PSO so as to select the most important influential variables of the TBFs concurrently to the SVR hyperparameters' adjustment. The obtained results suggest that whenever a variable is introduced into or removed from the data set, a tuning of the SVR hyperparameters must be performed so as to avoid reduction of the SVR performance.

In the second case, pairs and residuals bootstrap schemes were combined to SVR for the construction of confidence and prediction intervals related to the response variable. In this way, along with the accurate point predictions resulted from the adopted bagging estimator, interval inferences could also be performed. Additionally, since both SVR and bootstrap are non-parametric techniques, a reduced number of probabilistic assumptions were required. This is an interesting property of the proposed method, as it can be applied when no probabilistic models fit the phenomenon under analyais and / or when a large amount of data can not be obtained, *e.g.* due to budget constraints. The proposed methods were validated on an simulated example and a Monte Carlo experiment was performed in order to assess their coverage properties. Such a simulation indicated that the coverages are better when the training data set increases and that a greater number of bootstrap iterations does not necessarily enhance the coverages. Also, an illustration of how the presented methodology could support decision making concerning preventive maintenance actions was also provided; it was based on the interval estimates resulted from the boostrapped SVR and on a given threshold related to failure occurrence.

8.1 Limitations and Suggestions for Future Works

Although the presented techniques can be, in principle, used in various contexts, the obtained results as well as the related inferences may not be generalized to other applications without a preliminary investigation.

MOGA is a probabilistic model and does not guarantee that the true optimal solutions have been obtained. However, such a drawback can be partially overcome if various MOGA runs are performed with different GA parameter values (*e.g.* probabilities of crossover and mutation, numbers of individuals and generations). Each of these sets of parameters is expected to enable the exploration of a specific location of the search space and, in the end, the comparison of the results from each run may provide an overall better front. Besides, MOGA is a flexible optimization technique that permits the coupling with other methods for the generation of more realistic frameworks to support decision making. Also, it is specially useful in the absence of analytical objective functions, as in the case of superimposing GRPs, when the problem is related to a prohibitive number of combinations for an exhaustive evaluation of solutions, and / or multiple objectives need to be separately handled. In fact, there are many real cases in which the optimal solution or the Pareto front is not known or the objective functions and / or constraints are intractable. If the methods used to compute the objectives is time consuming (*e.g.* DES), the previously commented approach of GA parameter variation to provide an overall better front may be too expensive. Like MOGA, PSO is a probabilistic optimization model and there are no guarantees that the optimal solution have been attained. However, it has been successfuly applied, mainly to real-valued problems and analogously to the case of MOGA, different PSO parameters can be used to explore various locations of the search space so as to provide overall better solutions, mainly if the fitness evaluation step is not time consuming.

In the multi-objective redundancy allocation problems, only the constraints related to the minimum and maximum number of components per subsystem were considered. Other constraints related, for example, to volume and weight, can be taken into account. Also, instead of GRPs, other methods describing the failure-operation process of components that incorporate the effects of repairs can be used, *e.g.* modulated power law processes (RIGDON & BASU, 2000; SALDANHA & FRUTUOSO E MELO, 2012).

In this work, ROI analysis was suggested to guide the decision concerning the selection of a solution (*e.g.* system design and number of maintenance teams or inspection plan) from the obtained Pareto front. Such an approach can be combined, for example, to elicitation methods (KAHNEMAN *et al.*, 1982; MOSLEH *et al.*, 1988; COOKE, 1991; CAMPELLO DE SOUZA, 2007) that capture the preferences of the decision maker so as to construct a circumstance-adapted post-optimization method.

Inspections reduce the uncertainty about the deterioration state of the equipment but no intervention is performed in order to reduce the associated risk level. In this way, they are able to suggest the performance of preventive maintenance so as to reduce and / or delay the action of the damage mechanism. In this work, however, the effects of preventive maintenance actions were not taken into account in the elaboration of multi-objective inspection plans. The combination of multi-objective inspection and preventive maintenance plans would consider the impacts of the maintenance actions on the deterioration state of the equipment, which would be incorporated in the risk computation by the RBI. Thus, the inspection and the maintenance activities could provide information to each other so as to efficiently determine their scheduling.

Indeed, RBI is used for calculating risk of equipment from oil and petrochemical industries. In this way, the proposed elaboration of multi-objective inspection plans can only be used in such a context. However, other methods for risk evaluation applicable to other industries can be combined to MOGA, so as to provide non-dominated inspection plans with respect to risk and cost.

The predictive ability of SVR was basically assessed by NRMSE and MSE on a test set. However, even though a good performance on unseen data is expected, there are no guarantees that the estimated SVR model will properly function. But, if an SVR gives poor predictions on unseen cases, it can be interpreted as an indicator for a retraining procedure incorporating new data that have become available. In this way, as more data are collected, the SVR could be retrained and updated.

Not only SVR but also bootstrap methods are suitable for situations where small to moderate observation sets are accessible. Nevertheless, the quality of the estimated models and values directly depends on the quality of the data used in the estimation step. Hence, in order to construct accurate data bases it is essential to give the appropriate importance for the data gathering procedure. If this is accomplished, more observations mean that more information about the phenomenon or process under analysis become available, which may positively impact the accuracy and precision of the SVR and bootstrap models and estimates.

Although the variable selection procedure by PSO + SVR suggests the subset of the most relevant input variables, the variability of the response that can be apportioned to them are not quantified. In this way, a global sensitivity analysis (SALTELLI *et al.*, 2004) approach could be combined to the PSO + SVR so as to allow for the ranking of the input variables according to their contribution for the variability of the response variable.

The methods and techniques considered and proposed in this work have potential to be combined in several manners so as to compose more general and comprehensive frameworks, as discussed in Section 1.2. For example, the components in the MOGA + DES were supposed to have times between failures governed by Weibull probability densities. However, it might not be the case and historic data concerning the components' failures could feed an SVR. For a given component, the SVR could provide the MOGA + DES with the time of its next failure. Thus, data related to the operational phase of the equipment could furnish valuable information for the design of systems. The same reasoning could be applied to the logistic time and to the TTRs. Also, the simulation block could incorporate not only failure times and corrective maintenance, but also inspection and preventive actions, given the importance of these activities for critical components. Finally, each proposed method or their combination is able to support decision making involving risk and reliability aspects of systems.

REFERENCES

AK, L.; LI, Y.; VITELLI, V.; ZIO, E.; DROGUETT, E. L. NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment. *Expert Systems with Applications*, v. 40, p. 1205–1212, 2013.

ANGUITA, D.; BONI, A.; RIDELLA, S. Evaluating the generalization ability of support vector machines through the bootstrap. *Neural Processing Letters*, v. 11, p. 51–58, 2000.

ANP. Agência Nacional do Petróleo, Gás Natural e Biocombustíveis. 2011. Available at: <http://www.anp.gov.br>. Date of access: 30-Mar-2011.

API. American Petroleum Institute. *API publication 581: risk-based inspection base resource document*. Washington, DC, United States, 2008.

BABAOGLU, I.; FINDIK, O.; ÜLKER, E. A comparison of feature selection models utilizing binary particle swarm optimization and genetic algorithm in determining coronary artery disease using support vector machine. *Expert Systems with Applications*, v. 37, p. 3177–3183, 2010.

BARROS JR., P. F. R. *Uma metodologia para análise de disponibilidade de sistemas complexos via hibridismo de redes Bayesianas e processos Markovianos*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Recife, 2006.

BOWLES, J. B. Comementary - caution: constant failure-rate models may be hazardous to your design. *IEEE Transactions on Reliability*, vol. 51, n. 3, p. 375–377, 2002.

BOYD, S.; VANDENBERGHE, L. *Convex optimization*. Cambridge: Cambridge University Press, 2004. Available at: <http://www.stanford.edu/∼boyd/cvxbook>.

BRABANTER, K. D.; BRABANTER, J. D.; SUYKENS, J. A. K.; MOOR, B. D. Approximate confidence and prediction intervals for least squares support vector regression. *IEEE Transactions on Neural Networks*, v. 22, n. 1, p. 110–120, January 2011.

BRATTON, D.; KENNEDY, J. Defining a standard for particle swarm optimization. In: *Proceedings of the IEEE Swarm Intelligence Symposium*. Honolulu, United States: 2007.

BREIMAN, L. Bagging predictors. *Machine Learning*, v. 24, p. 123–140, 1996.

BRYANT, R. Symbolic Boolean manipulation with ordered binary decision diagram. *ACM Computing Surveys*, v. 24, p. 375–377, 1992.

BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, v. 2, p. 121–167, 1998.

BUSACCA, P. G.; MARSEGUERRA, M.; ZIO, E. Multiobjective optimization by genetic algorithms: application to safety systems. *Reliability Engineering & System Safety*, vol. 72, p. 59–74, 2001.

CADINI, F.; ZIO, E.; KOPUSTINKAS, V.; URBONAS, R. A model based on boostrapped neural networks for computing the maximum fuel cladding temperature in an Rmbk-1500 nuclear reactor accident. *Nuclear Engineering and Design*, v. 238, p. 2165–2172, 2008.

CAMPELLO DE SOUZA, F. M. *Decisões racionais em situações de incerteza.* 2ed. Recife: 2007.

CANTONI, M.; MARSEGUERRA, M.; ZIO, E. Genetic algorithms and Monte Carlo simulation for optimal plant design. *Reliability Engineering & System Safety*, vol. 68, p. 29–38, 2000.

CHANG, C.-C.; LIN, C.-J. *LIBSVM: a library for support vector machines.* 2001. Available at: <http://www.csie.ntu.edu.tw/∼cjlin/libsvm>.

CHANG, M.-W.; LIN, C.-J. Leave-one-out bounds for support vector regression model selection. *Neural Computation*, v. 17, n. 5, p. 1188–1222, 2005.

CHAPELLE, O.; VAPNIK, V.; BOUSQUET, O.; MUKHERJEE, S. Choosing multiple parameters for support vector machines. *Machine Learning*, v. 46, p. 131–159, 2002.

CHEN, K.-Y. Forecasting systems reliability based on support vector regression with genetic algorithms. *Reliability Engineering and System Safety*, v. 92, p. 423–432, 2007.

CHEN, T.; NEVILLE, A.; YUAN, M. Calcium carbonate scale formation: assessing the initial stages of precipitation and deposition. *Journal of Petroleum Science and Engineering*, v. 46, p. 185–194, 2005.

CHIANG, C.-H.; CHEN, L.-H. Availability allocation and multiobjective optimization for parallel-series systems. *European Journal of Operational Research*, vol. 180, p. 1231–1244, 2007.

COELLO, C. A. C.; VELDHUIZEN, D. A. V.; LAMONT, G. B. *Evolutionary algorithms for solving multiobjective problems.* New York: Kluwer Academic, 2002.

COOKE, R. *Experts in uncertainty: opinion and subjective probability in science.* New York: Oxford University Press, 1991.

CRIBARI-NETO, F. Asymptotic Inference Under Heteroskedasticity of Unknown Form. *Computational Statistics & Data Analysis*, v. 45, p. 215–233, 2004.

CRIBARI-NETO, F.; LIMA, M. G. A. Heteroskedasticity-consistent interval estimators. *Journal of Statistical Computation and Simulation*, v. 79, n. 6, p. 787–803, 2009.

CRISTINIANI, N.; SHAWE-TAYLOR, J. *An introduction to support vetor machines and other kernel-based learning methods.* Cambridge: Cambridge Universty Press, 2000.

DAVIDSON, J.; MONTICINI, A.; PEEL, D. Implementing the wild bootstrap using a two-point distribution. *Economics Letters*, v. 96, p. 309–315, 2007.

DAVISON, A. C.; HINKLEY, D. V. *Bootstrap methods and their applications.* New York: Cambridge University Press, 1997. (Cambridge Series in Statistical and Probabilistic Mathematics).

DEB, K. Evolutionary algorithms for multi-criterion optimization in engineering design. In: *Evolutionary Algorithms in Engineering and Computer Science (EUROGEN'99).* 1999.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, vol. 6, n. 2, p. 182–197, 2002.

DOYEN, L.; GAUDOIN, O. Classes of imperfect repair models based on reduction of failure intensity or virtual age. *Reliability Engineering & System Safety*, vol. 84, p. 45–56, 2004.

DYER, S. J.; GRAHAM, G. M. The effect of temperature and pressure on oilfield scale formation. . *Journal of Petroleum Science and Engineering*, v. 35, p. 95–107, 2002.

EFRON, B. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, v. 7, n. 1, p. 1–26, 1979.

EFRON, B.; TIBSHIRANI, R. J. *An introduction to the bootstrap.* New York: Chapman & Hall/CRC, 1993. (Monographs on Statistics and Applied Probability 57).

ELEGBEDE, C.; ADJALLAH, K. Availability allocation to repairable systems with genetic algorithms: a multiobjective formulation. *Reliability Engineering & System Safety*, vol. 82, p. 319–330, 2003.

FEI, S.-W.; WANG, M.-J.; MIAO, Y.-B.; TU, J.; LIU, C.-L. Particle swarm optimization-based support vector machine for forecasting dissolved gases content in power transformer oil. *Energy Conversion and Management*, v. 50, p. 1604–1609, 2009.

FONSECA, C. M.; FLEMING, P. J. Genetic algorithms for multi-objective optimization: formulation, discussion and generalization. In: *Proceedings of the Fifth International Conference on Genetic Algorithms.* 1993.

FRIEDMAN, J. H.; TUCKEY, J. W. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, v. 23, n. 9, p. 881–890, 1974.

FRÖHLICH, H.; CHAPELLE, O.; SCHÖLKOPF, B. Feature selection for support vector machines by means of genetic algorithms. In: *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence.* 2003.

FURTADO, M. J. H.; SOARES, R. F; LINS, I. D.; MOURA, M. C; DROGUETT, E. L; FIRMINO, P. R. A. Multi-objective optimization of risk and cost for risk-based inspection plans. In: *Proceedings of the 11th International Probabilistic Safety Assessment and Management Conference (PSAM 11) and of the European Safety & Reliability Association Conference (ESREL).* Helsinki, Finland: 2012.

GOLDBERG, D. E. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, 1989.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, v. 3, p. 1157–1182, 2003.

GUYON, I.; WESTON, J.; BARNHILL, S.; VAPNIK, V. Gene selection for cancer classification using support vector machines. *Machine Learning*, v. 46, p. 389–422, 2002.

HELTON, J. C.; DAVIS, F. J. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, v. 81, p. 23–69, 2003.

HESKES, T. Practical confidence and prediction intervals. In: MOZER, M.; JORDAN, M.; HESKES, T. (Ed.). *Advances in neural information processing systems 9*. Cambridge: MIT Press, 1997. p. 466–472.

HIGGINGS, J. J. *Introduction to modern nonparametric statistics*. Pacific Groove: Brooks/Cole – Thomson Learning, 2004.

HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. E. A niched Pareto genetic algorithm for multiobjective optimization. In: *Proceedings of the First IEEE Conference on Evolutionary Computation*. 1994.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. *A practical guide to support vector classification*. 2003. Available at: <http://www.csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf>.

ITO, K.; NAKANO, R. Optimizing support vector regression hyperparameters based on cross-validation. In: *Proceedings of the International Joint Conference on Neural Networks*. Portland, United States: 2003.

JOACHIMS, T. Making large-scale SVM learning practical. In: SCHÖLKOPF, B.; BURGES, C.; SMOLA, A. J. (Ed.). *Advances in kernel methods: support vector learning*. Cambridge: The MIT Press, 1999. p. 169–184.

JONES, J.; HAYES, J. Estimation of system reliability using a “non-constant failure rate” model. *IEEE Transactions on Reliability*, vol. 50, p. 286–288, 2001.

JUANG, Y.-S.; LIN, S.-S.; KAO, H.-P. A knowledge management system for series-parallel availability optimization and design. *Expert Systems with Applications*, vol. 34, p. 181–193, 2008.

KAHNEMAN, D.; SLOVIC, P.; TVERSKY, A. *Judgement under uncertainty: heuristics and biases*. New York: Oxford University Press, 1982.

KECMAN, V. *Learning and soft computing: support vector machines, neural networks and fuzzy logic models*. Cambridge: The MIT Press, 2001.

KECMAN, V. Support vector machines: an introduction. In: WANG, L. (Ed.). *Support vector machines: theory and applications*. Berlin Heidelberg: Springer-Verlag, 2005, (Studies in Fuzziness and Soft Computing, v. 177). p. 1–47.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*. Perth, Australia: 1995.

KENNEDY, J.; EBERHART, R.; SHI, Y. *Swarm intelligence*. San Francisco: Morgan Kaufmann, 2001.

KIJIMA, M.; SUMITA, N. A useful generalization of renewal theory: counting process governed by non-negative Markovian increments. *Journal of Applied Probability*, vol. 23, p. 71–88, 1986.

KNOWLES, J.; CORNE, D. On metrics for comparing nondominated sets. In: *Proceedings of the 2002 Congress on Evolutionary Computation Conference (CEC02)*. IEEE Press, 2002. p. 711–716.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. *Artificial Intelligence*, v. 97, p. 273–324, 1997.

KUO, W.; PRASAD, V. R.; TILLMAN, F. A.; HWANG, C.-L. *Optimal reliability design: fundamentals and applications*. United Kingdom: Cambridge University Press, 2001.

KUO, W.; WAN, R. Recent advances in optimal reliability allocation. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 37, n. 4, p. 143–156, 2007.

LIN, C. J.; WENG, R. C. *Simple probabilistic predictions for support vector regression*. 2004.

LIN, H.-T.; LIN, C.-J. *A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods*. 2003.

LIN, S.-W.; YING, K.-C.; CHEN, S.-C.; LEE, Z.-J. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications*, v. 35, p. 1817–1824, 2008.

LINS, I. D.; DROGUETT, E. L. Multiobjective optimization of redundancy allocaion problems in systemas with imperfect repairs via ant colony and discrete event simulation. In: *Proceedings of the European Safety & Reliability Conference (ESREL)*. Valencia, Spain: 2008.

LINS, I. D.; DROGUETT, E. L. Multiobjective optimization of availability and cost in repairable systems via genetic algorithms and discrete event simulation. *Pesquisa Operacional*, vol. 29, p. 43–66, 2009.

LINS, I. D.; DROGUETT, E. L. Redundancy allocation problems considering systems with imperfect repairs using multi-objective genetic algorithms and discrete event simulation. *Simulation Modelling Practice and Theory*, v. 19, n. 1, p. 362–381, 2011.

LINS, I. D.; MOURA, M. C.; DROGUETT, E. L. *Support vector machines and particle swarm optimization: applications to reliability prediction*. Saarbrücken, Germany: Lambert Academic Publishing, 2010a. ISBN 9783838319407.

LINS, I. D.; MOURA, M. C.; DROGUETT, E. L.; ZIO, E. Reliability prediction of oil production wells by particle swarm optimized support vector machines. In: *Proceedings of the European Safety & Reliability Conference (ESREL)*. Rhodes, Greece: 2010b.

LINS, I. D.; MOURA, M. C.; DROGUETT, E. L.; ZIO, E.; JACINTO, C. M. Reliability prediction of oil wells by support vector machine with particle swarm optimization for variable selection and hyperparameter tuning. In: *Proceedings of the European Safety & Reliability Conference (ESREL)*. Troyes, France: 2011b.

LINS, I. D.; MOURA, M. C.; ZIO, E.; DROGUETT, E. L. A particle swarm-optimized support vector machine for reliability prediction. *Quality and Reliability Engineering International*, v. 28, n. 2, p. 141–158, 2012a.

LINS, I. D.; MOURA, M. C.; DROGUETT, E. L.; ZIO, E.; JACINTO, C. M. Uncertainty assessment through bootstrapped support vector regression. In: *Proceedings of the 11th International Probabilistic Safety Assessment and Management Conference (PSAM 11) and of the European Safety & Reliability Association Conference (ESREL)*. Helsinki, Finland: 2012b.

LINS, I. D.; RÊGO, L. C.; MOURA, M. C.; DROGUETT, E. L. Security system designs via games of imperfect information and multi-objective genetic algorithms. In: *Proceedings of the International Topical Meeting on Probabilistic Safety Assessment and Analysis (PSA)*. Wilmington, NC, USA: 2011a.

LINS, I. D.; RÊGO, L. C.; MOURA, M. C.; DROGUETT, E. L. Selection of security system design via games of imperfect information and multi-objective genetic algorithm. *Reliability Engineering & System Safety*, v. 112, p. 59–66, 2013.

LINS, S. *Princípios de enumeração*. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 1981.

LITTLEWOOD, B. Letter to the Editor. *Reliability Engineering & System Safety*, vol. 93, p. 1287, 2008.

LIU, R. Bootstrap procedures under some non-i.i.d. models. *The Annals of Statistics*, v. 16, n. 4, p. 1696–1708, 1988.

MARANGONE, F. C.; FREIRE, J. L. F. Gerenciamento da integridade de um vaso de pressão utilizando a inspeção baseada em risco. In: *Proceedings of the* 8ª *Conferência sobre Tecnologia de Equipamentos*. Salvador, BA, Brazil: 2005.

MARSEGUERRA, M.; ZIO, E.; PODOFILLINI, L. Multiobjective spare part allocation by means of genetic algorithms and Monte Carlo simulation. *Reliability Engineering & System Safety*, vol. 87, p. 325–335, 2005.

MESSAC, A.; SUNDARARAJ, G. J.; TAPPETA, R. V.; RENAUD, J. E. Ability of objective functions to generate points on nonconvex Pareto frontiers. *AIAA Journal*, vol. 38, n. 6, p. 1084–1091, 2000.

MICHALEWICZ, Z. *Genetic algorithms + data structures*. 3ed. Berlin: Springer-Verlag, 1996.

MODARRES, M. *Risk analysis in engineering: techniques, tools and trends*. Boca Raton: Taylor & Francis, 2006.

MODARRES, M.; KAMINSKY, M.; KRIVTSOV, V. *Reliability engineering and risk analysis*. New York: Marcel Dekker, 1999.

MOMMA, M.; BENNETT, K. P. A pattern search method for model selection of support vector regression. In: *Proceedings of the 2002 SIAM International Conference on Data Mining*. 2002. p. 261–274.

MONTGOMERY, D. C.; PECK, E. A.; VINING, G. G. *Introduction to linear regression analysis*. 4ed. Hoboken: John Wiley & Sons, 2006.

MOSLEH, A.; BIER, V. M.; APOSTOLAKIS, G. A critique of current practice for the use of expert opinions in probabilistic risk assessment. *Reliability Engineering & System Safety*, v. 20, p. 63–85, 1988.

MOURA, M. C.; ROCHA, S. P. V.; DROGUETT, E. A. L.; JACINTO, C. M. C. Bayesian assessment of maintenance efficiency via generalized renewal process. In Portuguese. *Pesquisa Operacional*, vol. 27, p. 569–589, 2007.

MOURA, M. C.; ZIO, E.; LINS, I. D.; DROGUETT, E. L. Failure and reliability prediction by support vector machines regression of time series data. *Reliability Engineering & System Safety*, v. 96, p. 1527–1534, 2011.

NOCEDAL, J.; WRIGHT, S. J. *Numerical optimization.* 2ed. New York: 2006.

PAI, P.-F. System reliability forecasting by support vector machines with genetic algorithms. *Mathematical and Computer Modelling*, v. 43, p. 262–274, 2006.

PLATT, J. C. Fast training of support vector machines using sequential minimal optimization. In: SCHÖLKOPF, B.; BURGES, C. J. C.; SMOLA, A. (Ed.). *Advances in kernel methods: support vector machines.* Cambridge: The MIT Press, 1998.

RAKOTOMAMONJY, A. Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, v. 3, p. 1357–1370, 2003.

RAUSAND, M.; HOYLAND, A. *System reliability theory: models and statistical methods.* 2ed. New York: John Wiley & Sons, 2004.

RAUZY, A. Mathematical foundations of minimal cutsets. *IEEE Transactions on Reliability*, v. 50, p. 389–396, 2001.

RIGDON, S. E.; BASU, A. P. *Statistical methods for the reliability of repairable systems.* New York: John Wiley & Sons, 2000.

ROSS, S. M. *Simulation.* 3ed. San Diego: Academic Press, 2002.

ROSS, S. M. *Introduction to probability models.* 10ed. San Diego: Academic Press, 2010.

SALDANHA, P. L. C.; FRUTUOSO E MELO, P. F. Analysis of the qualified life extension of nuclear safety systems through modulated Poisson point processes. *Progress in Nuclear Energy*, v. 60, p. 117–128, 2012.

SALTELLI, A.; TARANTOLA, S.; CAMPOLONGO, F.; RATTO, M. *Sensitivity analysis in practice: a guide to assessing scientific models.* Chichester: Wiley, 2004.

SANTOS, C. R.; LINS, I. D.; FIRMINO, P. R. A.; MOURA, M. C.; DROGUETT, E. L. A method for optimal allocation of defensive alternatives: analysis of a strategic interaction with a multi-objective approach. In: *Proceedings of the 10th International Probabilistic Safety Assessment and Management Conference (PSAM 10).* Seattle, WA, USA: 2010.

SCHAFFER, J. D. Multiple objective optimization with vector evaluated genetic algorithms. In: *Proceedings of the First International Conference on Genetic Algorithms.* 1985.

SCHÖLKOPF, B.; SMOLA, A. J. *Learning with kernels: support vector machines, regularization, optimization and beyond*. Cambridge: The MIT Press, 2002.

SEASHOLTZ, M. B.; KOWALSKI, B. The parsimony principle applied to multivariate calibration. *Analytica Chimica Acta*, v. 277, n. 2, p. 165–177, 1993.

SECCHI, P.; ZIO, E.; MAIO, F. D. Quantifying uncertainties in the estimation of safety parameters by using bootstrapped artificial neural networks. *Annals of Nuclear Energy*, v. 35, p. 2338–2350, 2008.

SRINIVAS, N. K.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Journal of Evolutionary Computation*, vol. 2, n. 3, p. 221–248, 1994.

TABOADA, H. A.; BAHERANWALA, F.; COIT, D. W.; WATTANAPONGSAKORN, N. Practical solutions for multi-objective optimization: an application to system reliability design problems. *Reliability Engineering & System Safety*, vol. 92, p. 314–322, 2007.

TABOADA, H. A.; ESPIRITU, J.; COIT, D. W. MOMS-GA: a multiobjective multi-state genetic algorithm for system reliability optimization design problems. *IEEE Transactions on Reliability*, vol. 57, p. 182–191, 2008.

TSUJITANI, M.; TANAKA, Y. Cross-validation, bootstrap, and support vector machines. *Advances in Artificial Neural Systems*, v. 3, p. 45–55, 2011.

VAPNIK, V. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, v. 10, n. 5, p. 988–999, 1999.

VAPNIK, V. N. *The nature of statistical learning theory*. 2ed. New York: Springer-Verlag, 2000.

WOODSEND, K. *Using interior point methods for large-scale support vector machine training*. Tese (Doutorado) – University of Edinburgh, 2009.

WOODSEND, K.; GONDZIO, J. Exploiting separability in large-scale linear support vector machine training. *Computational Optimization and Applications*, v. 49, p. 241–269, 2011.

WRIGHT, S. J. *Primal-Dual Interior Point Methods*. SIAM, 1997.

WU, K.-P.; WANG, S.-D. Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. *Pattern Recognition*, v. 42, p. 710–717, 2009.

YAN, J.; LI, C.; WANG, Z.; DENG, L.; SUN, D. Diversity metrics in multiobjective optimization: review and perspective. In: *IEEE International Conference on Integration Technology*. Shenzhen, China: 2007.

YAN, W.; SHAO, H.; WANG, X. Soft sensing modeling based on support vector machines and Bayesian model selection. *Computers and Chemical Engineering*, vol. 28, p. 1489–1498, 2004.

YAÑES, M.; JOGLAR, F.; MODARRES, M. Generalized renewal process for analysis of repairable systems with limited failure experience. *Reliability Engineering & System Safety*, vol. 77, p. 167–180, 2002.

YANG, J.-B.; ONG, C.-J. Feature selection for support vector regression using probabilistic prediction. In: *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Washington, DC, United States: 2010. p. 343–351.

ZAMITH, R.; SANTOS, E. M. dos. *Atividades onshore no Brasil: regulação, políticas públicas e desenvolvimento local*. São Paulo: Annablume; Fapesp, 2007.

ZHANG, Y.; SHAW, H.; FARQUHAR, R.; DAWE, R. The kinetics of carbonate scaling: application for the prediction of downhole carbonate scaling. *Journal of Petroleum Science and Engineering*, v. 29, p. 85–95, 2001.

ZIO, E. A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes. *IEEE Transactions on Nuclear Science*, v. 53, n. 3, p. 1460–1478, June 2006.

ZIO, E.; APOSTOLAKIS, G. E.; PEDRONI, N. Quantitative functional failure analysis of a thermal–hydraulic passive system by means of bootstrapped artificial neural networks. *Annals of Nuclear Energy*, v. 37, p. 639–649, 2010.

ZIO, E.; BROGGI, M.; GOLEA, L.; PEDRONI, N. Failure and reliability predictions by Infinite Response Locally Recurrent Neural Networks. In: *Proceedings of* 5th *European Congress on Computational Methods in Applied Science and Engineering (ECCOMAS)*. Venice, Italy: 2008.

ZITZLER, E. *Evolutionary algorithms for multiobjective optimization: methods and applications*. Tese (Doutorado) – Swiss Federal Institute of Technology Zurich, 1999.

Appendix

Lagrangian and KKT First Order Conditions for SVR Training Problem

The Lagrangian related to the training problem (5.3)-(5.7) is:

$$\begin{aligned}\mathcal{L}(w,b,\xi,\xi^*,\alpha,\alpha^*,\beta,\beta^*) = \frac{1}{2}w^T w + \, c\cdot\sum_{l=1}^{\ell}(\xi_l+\xi_l^*) - \sum_{l=1}^{\ell}\alpha_l\cdot[w^T\phi(x_l)+b-y_l+\varepsilon+\xi_l] \\ -\sum_{l=1}^{\ell}\alpha_l^*\cdot[y_l-w^T\phi(x_l)-b+\varepsilon+\xi_l^*]-\sum_{l=1}^{\ell}(\beta_l\xi_l+\beta_l^*\xi_l^*)\end{aligned} \tag{A.1}$$

in which $\alpha,\alpha^*,\beta,\beta^*$ are the ℓ-dimensional vectors of Lagrange multipliers associated to constraints (5.4) and (5.5) respectively. Note that α_l and α_l^* can not be strictly positive simultaneously, given that there is no point satisfying both (5.4) and (5.5) at the same time. Hence, $\alpha_l\alpha_l^*=0$. The Lagrangian in (A.1) must be minimized with respect to primal variables w,b,ξ,ξ^* and maximized with respect to dual variables $\alpha,\alpha^*,\beta,\beta^*$. Then the saddle point $(w_0,b_0,\xi_0,\xi_0^*,\alpha_0,\alpha_0^*,\beta_0,\beta_0^*)$ of $\mathcal{L}$ has to be found. The related KKT first order optimality conditions are:

$$\frac{\partial\,\mathcal{L}(w,b,\xi,\xi^*,\alpha,\alpha^*,\beta,\beta^*)}{\partial\,w}=0,\quad w_0=\sum_{l=1}^{\ell}(\alpha_{l0}-\alpha_{l0}^*)\phi(x_l) \tag{A.2}$$

$$\frac{\partial\,\mathcal{L}(w,b,\xi,\xi^*,\alpha,\alpha^*,\beta,\beta^*)}{\partial\,b}=0,\quad \sum_{l=1}^{\ell}(\alpha_{l0}-\alpha_{l0}^*)=0 \tag{A.3}$$

$$\frac{\partial\,\mathcal{L}(w,b,\xi,\xi^*,\alpha,\alpha^*,\beta,\beta^*)}{\partial\,\xi_l}=0,\quad C-\alpha_{l0}=\beta_{l0},\quad \forall\,l \tag{A.4}$$

$$\frac{\partial\,\mathcal{L}(w,b,\xi,\xi^*,\alpha,\alpha^*,\beta,\beta^*)}{\partial\,\xi_l^*}=0,\quad C-\alpha_{l0}^*=\beta_{l0}^*,\quad \forall\,l \tag{A.5}$$

$$w_0^T\phi(x_l)+b_0-y_l+\varepsilon+\xi_{l0}\geq 0,\quad \forall\,l \tag{A.6}$$

$$y_l-w_0^T\phi(x_l)-b_0+\varepsilon+\xi_{l0}^*\geq 0,\quad \forall\,l \tag{A.7}$$

$$\xi_{l0}\geq 0,\quad \forall\,l \tag{A.8}$$

$$\xi_{l0}^*\geq 0,\quad \forall\,l \tag{A.9}$$

$$\alpha_{l0}\geq 0,\quad \forall\,l \tag{A.10}$$

$$\alpha_{l0}^*\geq 0,\quad \forall\,l \tag{A.11}$$

$$\beta_{l0}\geq 0,\quad \forall\,l \tag{A.12}$$

$$\beta_{l0}^*\geq 0,\quad \forall\,l \tag{A.13}$$

$$\alpha_{l0}\cdot[w_0^T\phi(x_l)+b_0-y_l+\varepsilon+\xi_{l0}]=0,\quad \forall\,l \tag{A.14}$$

$$\alpha_{l0}^{*} \cdot [y_l - w_0^T \phi(x_l) - b_0 + \varepsilon + \xi_{l0}^{*}] = 0, \quad \forall\, l \tag{A.15}$$

$$\beta_{l0}\xi_{l0} = 0, \quad (c - \alpha_{l0}) \cdot \xi_{l0} = 0, \quad \forall\, l \tag{A.16}$$

$$\beta_{l0}^{*}\xi_{l0}^{*} = 0, \quad (c - \alpha_{l0}^{*}) \cdot \xi_{l0}^{*} = 0, \quad \forall\, l \tag{A.17}$$

By replacing equalities (A.2)-(A.5) in (A.1), the objective function (5.10) is defined. The constraint (A.3) remains in the problem. Constraints (5.12) and (5.13), in turn, are obtained by the combination of Equations (A.4), (A.8), (A.10), (A.12) and Equations (A.5), (A.9), (A.11), (A.13), rescpectively. Note that problem (5.10)-(5.13) presents only the dual variables α_l and α_l^* as unknowns, which have c as upper bound, so as to respect the non-negativity of β_l and β_l^*, see (A.12) and (A.13).