Available online at www.sciencedirect.com

SCIENCE DIRECT®

BioSystems xxx (2006) xxx–xxx

Bio Systems

www.elsevier.com/locate/biosystems

On the use of multi-objective evolutionary algorithms for survival analysis

Christian Setzkorn*, Azzam F.G. Taktak, Bertil E. Damato

Royal Liverpool University Hospital, Liverpool, United Kingdom

Received 7 December 2005; received in revised form 3 March 2006; accepted 3 March 2006

Abstract

This paper proposes and evaluates a multi-objective evolutionary algorithm for survival analysis. One aim of survival analysis is the extraction of models from data that approximate lifetime/failure time distributions. These models can be used to estimate the time that an event takes to happen to an object. To use of multi-objective evolutionary algorithms for survival analysis has several advantages. They can cope with feature interactions, noisy data, and are capable of optimising several objectives. This is important, as model extraction is a multi-objective problem. It has at least two objectives, which are the extraction of accurate and simple models. Accurate models are required to achieve good predictions. Simple models are important to prevent overfitting, improve the transparency of the models, and to save computational resources. Although there is a plethora of evolutionary approaches to extract models for classification and regression, the presented approach is one of the first applied to survival analysis. The approach is evaluated on several artificial datasets and one medical dataset. It is shown that the approach is capable of producing accurate models, even for problems that violate some of the assumptions made by classical approaches.

Keywords: Survival analysis; Evolutionary algorithms; Radial basis function networks

1. Introduction

Survival analysis involves the estimation of the distribution of the time it takes for an event to occur to an object depending on its features (Kleinbaum, 1996). In a medical domain, objects often correspond to patients and their features, which are also known as explanatory variables, predictors and covariates, could be demographic information and/or physiological information. Events may correspond to the recurrence of a disease or the death of a patient. Hence, the distribution of the time to a specific event for an object is also referred to as lifetime/failure time distribution. To estimate the lifetime distribution, or conversely the probability of survival, has many benefits. It allows clinicians to devise a suitable treatment regime and counsel patients about their prognosis. Hence, it helps patients to plan their lives and provide future care for their dependents.

Survival analysis is also widely used in the social and economic sciences, as well as in engineering. Here the objects could correspond to customers, machines/systems and the event of interest may be that the customer 'churns' or the failure of the machine. Survival analysis is therefore also referred to as reliability and failure time analysis (Afifi et al., 2003).

The distribution of the time to a specific event dependent upon the features of an object can be represented by four closely related functions, which are listed as follows:

* Corresponding author. Tel.: +44 151 706 4214; fax: +44 151 706 5803.

E-mail address: chris@csc.liv.ac.uk (C. Setzkorn)

0303-2647/$ – see front matter © 2006 Published by Elsevier Ireland Ltd.
doi:10.1016/j.biosystems.2006.03.002

- density function $f(t, x)$ (p.d.f.);
- cumulative distribution function $F(t, x)$ (c.d.f.);
- survival function $S(t, x)$;
- hazard function $h(t, x)$.

These functions are related to each other as shown in Eqs. (1)–(4) (Allison, 1997; Collet, 1994):

$$f(t,x) = h(t,x)\mathrm{e}^{-\int_0^t h(u)\,\mathrm{d}u} = \frac{\mathrm{d}F(t)}{\mathrm{d}t} = -\frac{\mathrm{d}S(t)}{\mathrm{d}t} = \lim_{\mathrm{d}t\to 0}\frac{P(t \le T < t+\mathrm{d}t, x)}{\mathrm{d}t} \quad (1)$$

$$F(t,x) = \int_0^t f(u)\,\mathrm{d}u = P(T < t, x) = 1 - S(t,x) \quad (2)$$

$$S(t,x) = \mathrm{e}^{-\int_0^t h(u)\,\mathrm{d}u} = 1 - F(t,x) = P(T \ge t, x) \quad (3)$$

$$h(t,x) = -\frac{\mathrm{d}}{\mathrm{d}t}\log(S(t)) = \frac{f(t,x)}{S(t,x)} = \lim_{\Delta t\to 0}\frac{P(t \le T < t+\mathrm{d}t | T \ge t, x)}{\Delta t} \quad (4)$$

The survival and hazard function are the most popular ways of describing the lifetime distribution dependent upon of the features x of an object. The hazard function is also often referred to as force of mortality, instantaneous death rate, or failure rate in a medical domain (Afifi et al., 2003). It is important to note that, although it may be helpful to think of the hazard as an instantaneous probability, it is not a probability as it can take on values greater than one (Allison, 1997).

A popular way of estimating the survival function is illustrated using a dataset taken from Freireich et al. (1963). The dataset contains the survival times of 42 leukaemia patients with one dichotomous feature that indicates whether or not the patient received a particular treatment. Table 1 contains the data separated according to the treatment feature.

The time is measured in weeks, and the maximum time horizon of the study (the follow up time) is 35 weeks. Patients shown without plus signs died during the follow up time, whereas patients shown with plus sign were censored. In essence, censoring occurs if one knows the status of an object for a particular period of time, but not for the complete follow up time.

Table 1
Leukaemia data taken from Freireich et al. (1963)

Group 1 (treatment)	Group 2 (placebo)
6, 6, 6, 7, 10, 13, 16, 22, 23, 6+, 9+, 10+, 11+, 17+, 19+, 20+, 25+, 32+, 32+, 34+, 35+	1, 1, 2, 2, 3, 4, 4, 5, 5, 8, 8, 8, 8, 11, 11, 12, 12, 15, 17, 22, 23

The numbers correspond to the survival times in weeks after the patient entered the study. The plus sign indicates censoring.

Generally there are three different forms of censoring, which are listed as follows (Kleinbaum, 1996):

(1) the event does not happen to the object before the maximum time horizon of the study;
(2) the object is lost to follow-up during the study;
(3) the object is withdrawn from the study because a different event made it impossible to follow it up any further.

The first type of censoring indicates that the event did not happen to the object during the complete follow up time (e.g. the patient did not die during the study). Note that the patient can usually enter the study at any time during the follow up time.

The second type of censoring occurs when one does not know the status of the object after a particular point in time. For example, the patient may have failed to attend an appointment in the clinic (Griffin, 1998).

The third type of censoring occurs because an event that is not relevant to the study happened to the object and made it impossible to follow the object up any further (e.g. the event of interest may be ‘cancer related death’ but the patient died from a car accident).

All three types of censoring are often referred to as right censoring, which is the most common form of censoring (Lawless, 1982). Another form of censoring is known as left censoring. It occurs if the status of an object is unknown at the left side of the follow-up period (e.g. the diagnosis of a disease does not necessarily mean that one knows when the disease started). If the population is both right and left censored one speaks of interval censoring.

One non-parametric approach for estimating the survival function is the Kaplan–Meier method. Its computation is summarised in Eq. (5):

$$\hat{S}(t) = \prod_{j=1}^{k}\left(\frac{n_j - d_j}{n_j}\right) = \prod_{j=1}^{k}\left(1 - \frac{d_j}{n_j}\right) = \hat{S}(t-1)\left(1 - \frac{d_t}{n_t}\right) \quad (5)$$

Here, d_j corresponds to the number of events (e.g. deaths) at time j where one or more events occurred and n_j corresponds to the number of objects (e.g. patients) that are still observed at time j. In the medical domain, these patients are also called the risk set. Given the structure of Eq. (5), it is not surprising that the Kaplan–Meier method is often referred to as the Product-limit estima-

Table 2
Kaplan–Meier estimates for the leukaemia data of patients with treatment

t_j	n_j	d_j	q_j	$S(t_j)$
0	21	0	0	21/21 = 1.00
6	21	3	1	$1 \times 18/21 = 0.8571$
7	17	1	1	$0.8571 \times 16/17 = 0.8067$
10	15	1	2	$0.8067 \times 14/15 = 0.7529$
13	12	1	0	$0.7529 \times 11/12 = 0.6902$
16	11	1	3	$0.6902 \times 10/11 = 0.6275$
22	7	1	0	$0.6275 \times 6/7 = 0.5378$
23	6	1	5	$0.5378 \times 5/6 = 0.4482$

Here, t_j is the point in time where at least one event occurs, n_j the size of the risk set, d_j the number of events at time t_j, and q_j is the number of patients who were censored at time t_j.

tor. The survival function values for the leukaemia data for patients with treatment are summarised in Table 2 and for patients without treatment in Table 3.

It should be noted that Kaplan–Meier estimates could not be used directly to approximate the lifetime distribution depending upon the explanatory variables. Instead, groups containing patients with particular feature values have to be determined beforehand. Hence, the estimated survival curves are only reliable if the number of samples within each group is large and the amount of censoring is low.

The corresponding survival curves are shown in Fig. 1. Here, the solid line corresponds to the survival curve for patients who were treated, and the dashed line corresponds to the survival curve for patients without treatment.

It can clearly be seen that the survival function estimates of patients without treatment is worse than of treated patients.

Table 3
Kaplan–Meier estimates for the leukaemia data of patients without treatment

t_j	n_j	d_j	q_j	$S(t_j)$
0	21	0	0	21/21 = 1.00
1	21	2	0	19/21 = 0.90
2	19	2	0	17/21 = 0.81
3	17	1	0	16/21 = 0.76
4	16	2	0	14/21 = 0.67
5	14	2	0	12/21 = 0.57
8	12	4	0	8/21 = 0.38
11	8	2	0	6/21 = 0.29
12	6	2	0	4/21 = 0.19
15	4	1	0	3/21 = 0.14
17	3	1	0	2/21 = 0.10
22	2	1	0	1/21 = 0.05
23	1	1	0	0/21 = 0.00

Here, t_j is the point in time where at least one event occurs, n_j the size of the risk set, d_j the number of events at time t_j, and q_j is the number of patients who were censored at time t_j.

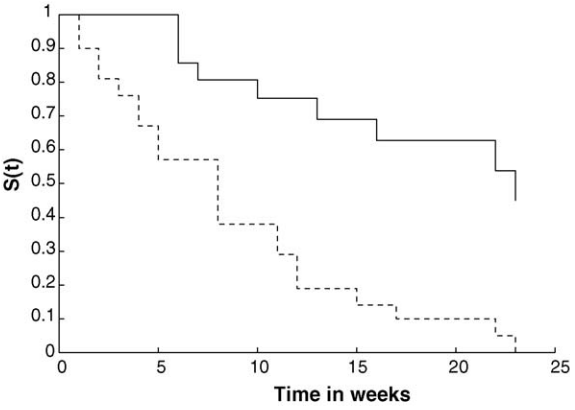

Fig. 1. Kaplan–Meier curves for the leukaemia data summarised in Table 1. The solid line corresponds to the Kaplan–Meier curve for patients who were treated and the dashed line to those without treatment.

Another method for estimating lifetime distributions is the proportional hazards model. It is also known as the Cox model (Cox, 1972). The hazard function of an individual is estimated using Eq. (6):

$$h_i(t; x) = h_0(t)\psi(x_i) \tag{6}$$

Here, $h_0(t)$ is the baseline hazard and $\psi(x_i)$ is the relative hazard. The baseline hazard is the hazard for individuals with $x = 0$. The relative hazard is a factor that makes the hazard of individual i proportional to the baseline hazard. Proportionality means that none of the survival curves of the population cross (Collet, 1994). It also means that the logarithm of the estimated hazard functions has a constant distance (Marubini and Valsecchi, 1995). In other words, they should be strictly parallel (Allison, 1997). Of course, this assumption is not very realistic as one group of patients might have higher hazard values at the beginning of the study but lower hazard values later. If the converse would occur for another group of patient, the survival curves of both groups would cross.

To model the dependency of the hazard on the feature values of an object one usually replaces the relative hazard with a function that depends on, for example, a linear combination of the features. The exponential function is used commonly to ensure that the relative hazard remains positive. This leads to Eq. (7):

$$h_i(t) = h_0(t)\mathrm{e}^{(\beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi})} = h_0(t)\mathrm{e}^{(\hat{\beta} x_i)} \tag{7}$$

The linear combination of the features is also referred to as the *risk score* or *prognostic index*. Unfortunately, the coefficients $\hat{\beta}$ cannot be estimated using the ordinary maximum likelihood method, because the baseline hazard h_0 is not defined parametrically. To estimate the coefficients $\hat{\beta}$, Cox put forward the partial likelihood method. This method allows the estimation of the coefficients $\hat{\beta}$ without specifying the baseline hazard h_0. The computation of the partial likelihood is summarised in Eq. (8):

$$\mathrm{PL} = \prod_{i=1}^{n} \left[\frac{\mathrm{e}^{(\beta' x_i)}}{\sum_{j=1}^{n} Y_{ij}\, \mathrm{e}^{(\beta' x_j)}} \right]^{\delta_i} \tag{8}$$

Here, $Y_{ij} = 1$ if $t_j \geq t_i$ and $Y_{ij} = 0$ if $t_j < t_i$, where t corresponds to the time of the event or the time of censoring. This is a convenient way to define the risk set in the denominator at the time of the ith event. The exponent δ_i indicates whether the object/patient was censored ($\delta_i = 0$) or whether the event occurred ($\delta_i = 1$). The partial likelihood can be maximised using, for example, a version of the Newton–Raphson algorithm. It has to be noted that Eq. (8) is only valid for data without ties in which no two events occur at the same time. The computation of the exact partial likelihood for tied data can be a daunting task (see for example Allison, 1997). Methods for the approximation of the partial likelihood for tied data are discussed in Allison (1997), Collet (1994) and Therneau and Grambsch (2000). Unfortunately, the standard Cox model (Eq. (7)) has several shortcomings, which are summarised as follows:

- It assumes proportional hazards;
- The baseline hazard has to be determined in order to obtain values of the hazard/survival function for an individual;
- The linear combination of the features cannot be used to model interaction effects;
- The estimation of the partial likelihood is computationally very expensive when the data contain ties. This is very likely as time is often measured in a discrete domain;
- It cannot be used (in its standard form) to model data containing time-dependent features.

These shortcomings can be alleviated using methods that are, for example, discussed in Marubini and Valsecchi (1995) and Therneau and Grambsch (2000). However, it has to be emphasised that the correct use of these methods is quite difficult, as they require extensive statistical knowledge. It is therefore not surprising that researchers are striving to develop more practical approaches, which make fewer assumptions and do not presume extensive statistical knowledge.

This paper investigates the use of a multi-objective evolutionary algorithm (MOEA) for survival analysis. A similar MOEA has already successfully been applied to several classification problems (Setzkorn and Paton, 2005). Although, there are many successful applications of evolutionary algorithms to regression and classification problems, we believe that this is the first study that uses a MOEA for survival analysis. This serves to demonstrate the versatility of evolutionary approaches for model extraction from data.

2. Existing work

Many proposed methods to overcome the shortcomings of the Cox model use the fact that the hazard corresponds to a conditional probability in the discrete time domain as shown in Eq. (9) (Lawless, 1982; Willett, 1993):

$$\hat{h}(t_j, x) = Pr(T = t_j | T \geq t_j, x) \tag{9}$$

This is in contrast to the continuous time domain in which the hazard corresponds to a rate that can have values that are greater than one. In the discrete case, the survival probabilities can be computed according to Eq. (10) for each time interval t_j:

$$\hat{S}(t_j, x) = \prod_{k=1}^{j} (1 - \hat{h}(t_j, x)_k) \tag{10}$$

It follows that the original survival analysis problem can be cast as a classification problem that requires the estimation of a conditional probability. However, the original data need to be pre-processed due to the problem of censoring. The pre-processing is described with the help of the artificial data in Table 4. Each object is uniquely identified by its ID, the feature Gender and the features $S1$–$S6$ which represent a time-varying feature S for six time intervals. The column Time shows how long the individual was observed. The column Censor indicates whether or not the object was censored.

Table 5 contains the pre-processed data. Each object was repeated according to the number of time intervals it was observed. The features $S1$–$S6$ are represented by one time-varying feature S. The feature Event indicates whether the event occurred when the object was last observed. The event has to be predicted. This can be achieved by estimating the conditional probability $P(\text{Event}|x)$, where the feature vector x consists of the three features Gender, S, and Time.

The conditional probability $P(\text{Event}|x)$ can be estimated by models that approximate the likeli-

Table 4
Original survival data

ID	Gender	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	Time	Censor
01	1	0	1	–	–	–	–	2	0
02	0	1	0	1	–	–	–	3	1

Table 5
Pre-processed survival data

ID	Gender	S	Time	Event/indicator
01	1	0	1	0
01	1	1	2	0
02	0	1	1	0
02	0	0	2	0
02	0	1	3	1

hood ratio $P(x|\text{Event})/P(x|\overline{\text{Event}})$ and the prior ratio $P(\text{Event})/P(\overline{\text{Event}})$ within the logistic link function. The derivation of the logistic function is shown in Eqs. (11)–(14):

$$P(\text{Event}|x) = \frac{P(x|\text{Event})P(\text{Event})}{P(x)} \tag{11}$$

$$P(\text{Event}|x) = \frac{P(x|\text{Event})P(\text{Event})}{P(x|\text{Event})P(\text{Event}) + P(x|\overline{\text{Event}})P(\overline{\text{Event}})} \tag{12}$$

$$P(\text{Event}|x) = \frac{1}{1 + \exp\left(-\log\left[\frac{P(x|\text{Event})}{P(x|\overline{\text{Event}})}\right] - \log\left[\frac{P(\text{Event})}{P(\overline{\text{Event}})}\right]\right)} \tag{13}$$

$$P(\text{Event}|x) = \frac{1}{1 + \mathrm{e}^{-\xi}} \tag{14}$$

If ξ represents a linear combination of the features, the model corresponds to the logistic regression equation. In this case the parameters can be optimised using standard statistic packages (Willett, 1993). In contrast to the standard Cox model, this simple model can already estimate the dependency of the hazard on time without estimating the baseline hazard. This is because the pre-processed data contain the feature Time. Furthermore, the model can be used for time-varying features such as S in Table 5.

Of course, it is also possible to model non-linear relationships between the features and the hazard. This is achieved by using a more complicated relationship for ξ as shown in Eq. (15):

$$\xi = \alpha + \sum_{h=1}^{H} w_h \phi_h \left(\alpha_h + \sum_{j=1}^{J} w_{ij} x_{ij} \right) \tag{15}$$

Here, ϕ corresponds to the logistic function and α, w_h, α_h, and w_{ij} are parameters that have to be estimated. If one substitutes Eq. (15) into Eq. (14) one obtains a particular type of artificial neural network (ANN) with a single hidden layer. An ANN like this was used in Biganzoli et al. (1998) and is referred to as Partial Logistic Artificial Neural Network (PLANN). It can be optimised using, for example, the back-propagation algorithm and the cross-entropy error function which is shown in Eq. (16) (Bishop, 1995):

$$E = -\sum_{i=1}^{N} \sum_{j=1}^{L} \{d_{ij} \log[h(t_j, x)] + (1 - d_{ij}) \log[1 - h(t_j, x)]\} \tag{16}$$

Here, d_{ij} is the indicator value (see Event column in Table 5) for the ith vector for the jth time interval. The hazard $h(t_j, x)$ is the output of the model.

An advantage of PLANNs is that they can directly produce smooth estimates for the discrete hazard without estimating the baseline hazard. In addition, PLANNs can cope with non-proportional hazards as they account for interactions between the features and time implicitly. This makes this model very powerful. However, this apparent advantage can also be a problem as it enables the model to fit chance fluctuations within the data that are due to there being only a finite number of samples (Burges, 1998; Hand, 1997). Finite numbers of samples sparsely cover the feature space, especially within high dimensional feature spaces. Hence, the classifier has to extrapolate the samples in a non-trivial way. This problem is also known as the 'curse of dimensionality' (e.g. Geman et al., 1992).

The PLANN can become very specific to the data by fitting chance fluctuations. Hence, the model does not estimate the general mechanism that produced the data (Dietterich, 1995). Such models are called over-fitted. The extreme scenario is that the model exactly represents/memorises the data. On the other hand, if the model is too simple it might not achieve a good fit to the data (it is under-fitted). This problem can be summarised with help of the following metaphor taken from Burges (1998):

> An over-fitted model is like a botanist with a photographic memory who, when presented with a new tree, concludes that it is not a tree because it has a different number of leaves from anything she has seen before; an under-fitted model is like the botanist's lazy brother, who declares that if it's green, it's a tree.

Over-fitted models exhibit a high variance, because their fit to the data (bias), although good on particular data, is worse when they are tested on new data sampled from the same population. This phenomenon is also known as the bias/variance dilemma (Geman et al., 1992). There are several methods to prevent over-fitting, which are reviewed in Setzkorn (2005). Numerous approaches to avoid over-fitting of ANNs are discussed in Lisboa et al. (2003) and Biganzoli et al. (1998).

This paper uses a multi-objective evolutionary algorithm (MOEA) to extract radial basis function networks (RBFNs) from survival data, instead of using PLANNs and optimisation algorithms such as back-propagation. It builds upon the work presented in Setzkorn and Paton (2005). RBFNs are a specific type of ANN: advantageously, RBFNs have simpler structures relative to PLANNs. This results in less complex search spaces and thus shorter extraction times (Bishop, 1995). In addition, the simpler structure of RBFNs allows an easier interpretation of the parameters of the model. Hence, one major criticism of ANNs, namely their difficult interpretability (Clark et al., 2003), might be overcome.

MOEAs are powerful optimisation algorithms, which can optimise several incommensurable objectives without making any assumptions about their importance. This is important in the context of model extraction, which is a multi-objective problem. It has at least two objectives, which are the extraction of accurate and simple models from data. Accurate models are required to achieve exact predictions whereas simple models are required to understand the data generation process. In fact, it is often argued that only simple models are adopted in practice due to their transparency (Elder and Pregibon, 1996; Humphrey et al., 1998; Pazzani et al., 1997). The extraction of simple models is also important because complex models tend to be over-fitted, require longer execution times, and more storage space.

There are additional reasons for the preference towards MOEAs for model extraction. For example, MOEAs are less prone to feature interactions and can cope better with noisy data. This is in contrast to other greedy search algorithms (Dhar et al., 2000; Freitas, 2002; Shi et al., 1999). In addition, MOEAs can extract several models (trade-off solutions for the given objectives) from a dataset in a single run. This is because MOEAs deploy a number of candidate solutions/models to search a given search space (Michalewicz and Fogel, 2005). This has the advantage that, if the preferences of the decision maker change, (s)he could choose another trade-off solution/model from the solution set (e.g. (s)he might prefer more accurate models over simpler models). This saves valuable (computational) resources, because the search does not have to be repeated. Generated trade-off solutions can also be combined. Combined model may offer better generalization in some cases, and worse in others (Kuncheva, 2004).

3. Using a multi-objective evolutionary algorithm for survival analysis

This section describes the implemented MOEA. It begins with a brief summary of the algorithm and then describes its particular components. Fig. 2 depicts the structure of the algorithm. It is similar to other evolutionary algorithms (e.g. Michalewicz and Fogel, 2005).

In broad terms, the algorithm proceeds as follows. Candidate solutions/individuals (i.e. a population of RBFNs) are initialized randomly. After this, the variation operators are applied to some of the RBFNs to recombine and/or change them. The fitness evaluation then determines the performance (fitness/objective values) of each RBFN.

The selection process generates a new population of individuals by sampling from the current population and the archive. The archive stores the best (elite) individuals found by the MOEA. This prevents the loss of good candidate solutions due to the randomness of the selection process (Zitzler et al., 2002). The use of an archive is a form of elitism, which can help to create better individuals (Deb, 2001). In fact, the implemented MOEA uses an archiving strategy that ensures diversity within the population and prevents premature convergence of the algorithm (Laumanns et al., 2002a). The selection process is followed by the termination test, which either terminates the MOEA or transmits the current population (generation) to the process that applies the variation operators.

Better individuals (RBFNs) will be produced over time, as the above sequence is repeated. If the MOEA terminates (e.g. after a maximum number of generations) the RBFNs within the current population and the archive are evaluated on a validation dataset, which was not used

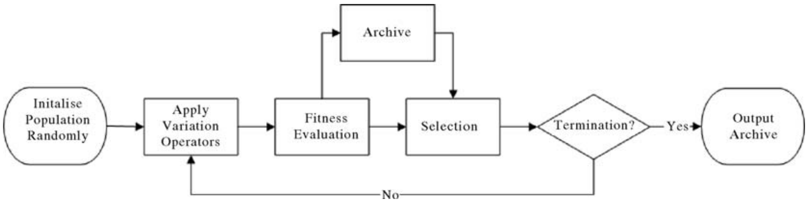

Fig. 2. Structure of the implemented MOEA.

during the induction process. The final output of the system is the updated set of individuals within the archive. Here follows a more detailed description of the components of the MOEA.

3.1. The representation scheme

Radial basis function networks can be used to estimate conditional probabilities (Biganzoli et al., 2001; Bishop, 1995), such as the discrete hazard defined in Eq. (9), by replacing ξ in Eq. (14) with

$$\xi_k = \sum_{j=1}^{M} w_{kj} z_j(x) + b_k \tag{17}$$

Here, w_{kj} are coefficients (weights), z_{kj} the output of the jth basis functions, and b is a bias. The index k denotes the class to be predicted. This index is only necessary if one intends to predict more than two classes. The present study used a Gaussian basis function, which is defined in Eq. (18):

$$f(x;\sigma,\mu) = \mathrm{e}^{-(x-\mu)^2/2\sigma^2} \tag{18}$$

Here, σ is the variance and μ the mean. Note that other basis functions (kernels) could also be used.

The implemented representation scheme consists of basis functions (see Eq. (17)), which are represented as trees (see Fig. 3). This part of the representation scheme was inspired by the representation scheme of genetic programming (GP) (Cramer, 1985; Koza, 1998). However, in contrast to GP, the basis functions are not combined into one tree: rather they are kept apart in order to simplify the application of problem-specific variation operators and to maintain transparency.

Non-terminal nodes (circles in Fig. 3) correspond to the product operator. The terminal nodes (squares in Fig. 3) correspond to Gaussian basis functions (see Eq. (18)). A tree can contain at most one basis function for each feature. Note that neither the number of basis functions nor the number of features used within a tree is preset. They can be between a minimum and maximum value enabling the MOEA to adapt the complexity of the RBFN. This also means that the MOEA can perform an implicit feature selection, because it is not forced to use all features within the trees.

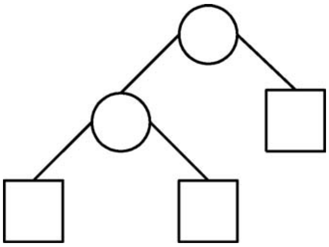

Fig. 3. Basis function tree.

Each tree also has k associated weight vectors, which contain values between zero and one (see Eq. (17)). Hence, this representation scheme could also be used to model competing risks (e.g. Biganzoli et al., 2001). The kth bias is a real number. The structure of the tree as well as the parameters (basis function parameters, weights, number of basis functions, etc.) can be changed by several variation operators that are described in Section 3.4.

3.2. The fitness evaluation

The implemented MOEA minimises the following objectives using the fitness assignment of the second Strength Pareto Evolutionary Algorithm (SPEA2) (Zitzler et al., 2002):

- Objective 1: measures the fit of the RBFN to the data using Eq. (16);
- Objective 2: measures the number of basis functions within the RBFN;
- Objective 3: measures the number of different features used within the RBFN.

To assign a scalar fitness to an individual with several objective values, the fitness evaluation makes use of the Pareto dominance relation, which is explained in Definition 1.

Definition 1 (Pareto dominance relation). A solution x_1 is said to dominate a solution x_2, also expressed as $x_1 \succ x_2$, if x_1 is at least as good as x_2 in all objectives and better with respect to at least one objective. This can be expressed more formally as: $\forall i \in \{1,\ldots,n\} : f_i(x_1) \leq f_i(x_2) \wedge \exists j \in \{1,\ldots,n\} : f_j(x_1) < f_j(x_2)$.

As mentioned earlier, the use of the Pareto dominance relation during the fitness selection enables the algorithm to optimise several incommensurable objectives without making any assumptions about their importance. The fitness $F(i)$ of an individual i is computed according to Eq. (19) (Zitzler et al., 2002):

$$F(i) = R(i) + D(i) \tag{19}$$

The value of $R(i)$ captures dominance information (see Eqs. (20) and (21)) and $D(i)$ captures density information (see Eq. (22)) of the ith individual:

$$R(i) = \sum_{j \in P_t + \bar{P}_t,\, j \succ i} S(j) \tag{20}$$

$$S(i) = |\{j | j \in P_t + \bar{P}_t \wedge i \succ j\}| \tag{21}$$

Here, P_t and $\bar{P}_t$ refer to individuals from the population and the archive, respectively. The expression $i \succ j$ denotes the dominance relation between individuals i and j. Eq. (20) determines the strength of the dominators of the ith individual. A high value means that the ith individual is dominated by many individuals, which in turn dominate other individuals. If the value of R_i is zero the individual i is non-dominated. The density information is computed according to Eq. (21) and is an adaptation of the kth nearest neighbour method (Silverman, 1999):

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{22}$$

Here, σ_i^k is the Euclidean distance between the objective values of the kth and the ith individual. The value for k is equal to the square root of the sample size: $k = \sqrt{N + \bar{N}}$ (Silverman, 1999). The values N and $\bar{N}$ denote the number of individuals in the population and archive, respectively.

3.3. The selection

The selection process produces a new population of individuals from the current population and the archive using binary tournament selection (Zitzler et al., 2002). Two individuals are sampled randomly without replacement from either the population or the archive. Whether an individual is selected from the archive or the population is determined by the 'elitism degree' (ED). The value of ED is computed according to Eq. (23) (Laumanns et al., 2000):

$$\mathrm{ED} = \begin{cases} 1 - \frac{|P_t|}{|\bar{P}_t \cup P_t|} & \text{if } |\bar{P}_t| \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

Here, $|P_t|$ is the size of the current population and $|\bar{P}_t \cup P_t|$ is the size of the archive and the current population. Hence, the larger the archive, the more likely it is that an individual is sampled from the archive. The individual with the lowest fitness value (see Eq. (19)) is declared as the winner of the 'binary tournament' and inserted into the new population. If a tie occurs, an individual is chosen with a uniform probability. This procedure is repeated until the new population has reached the size of the old population.

3.4. The variation operators

It is well known that the deployment of several problem-specific variation operators (VOs) can improve the evolutionary search (Spears, 1995; Janikow, 1993; Grefenstette, 1991; Wolpert and Macready, 1997). It was therefore decided to implement several problem-specific VOs. These VOs work on different levels of the individual (e.g. basis function level and weight level) to achieve an appropriate exploitation and exploration of the search space.

Two types of VOs were implemented. The first type (VO1) can change one individual and is also known as the mutation operator. The second type (VO2) can change two individuals and is also known as the crossover operator. There are several operators of each type. Each VO is applied to an individual with a low probability, which is determined by the parameters 'crossover probability' and 'mutation probability'. A particular VO is chosen with a uniform probability.

To illustrate the working principle of the VOs, the MOEA was applied to an artificial dataset that consisted of 1000 samples. Each sample had two features ($X1$ and $X2$), which were sampled with a uniform probability from the interval $[-10.0, \ldots, 10.0]$.

3.4.1. VO1$_1$ operator

The VO1$_1$ operator reinitialises one terminal node of one tree of the individual. The tree and the node are chosen with a uniform probability. The working principle of the operator is illustrated in Fig. 4.

Fig. 4 shows two individuals with two basis functions as contour diagrams within the feature space defined by $X0$ and $X1$. The left part depicts the individual before the VO1$_1$ operator is applied. The right part of Fig. 4 depicts the individual after the application of the operator. It can clearly be seen that the Gaussian kernel for feature $X0$ of the dotted basis function was changed.

3.4.2. VO1$_2$ operator

The VO1$_2$ operator reinitialises an individual. This operator is expected to be very disruptive but may help to prevent the premature convergence of the algorithm by introducing new 'genetic material' into the population.

3.4.3. VO1$_3$ operator

Similarly to the VO1$_1$ operator, the VO1$_3$ operator reinitialises one node of one tree in the individual. The tree and the node are chosen with a uniform probability. A new node or a sub-tree (depending on the aforementioned restrictions) replaces the node.

3.4.4. VO1$_4$ operator

The VO1$_4$ operator removes one basis function tree from the individual if the resulting individual

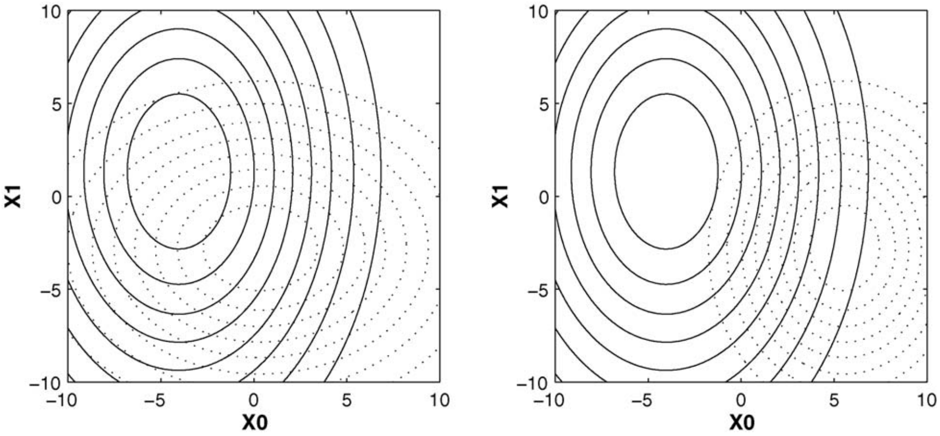

Fig. 4. The left part shows an individual with two basis functions before the VO1$_1$ operator is applied. The right part depicts the individual after the VO1$_1$ operator was applied.

would not contain fewer basis functions than allowed. The basis function is chosen with a uniform probability.

3.4.5. *VO1$_5$ operator*

The VO1$_5$ operator adds one new basis function tree to the individual if the resulting RBFN does not exceed the maximum number of trees.

3.4.6. *VO1$_6$ operator*

The VO1$_6$ operator changes one Gaussian kernel of one basis function slightly. The basis function and the kernel are chosen with a uniform probability. A Gaussian kernel can be changed in three different ways. It can be shrunk, extended, or moved. How the kernel is changed is decided upon with a uniform probability.

The 'slight' change of a parameter is achieved by applying Eq. (24). The parameter α could, for example, correspond to the centre of the Gaussian kernel μ in Eq. (18):

$$\acute{\alpha} = \begin{cases} \alpha + \delta(t, \alpha_{\mathrm{UB}} - \alpha) \\ \alpha - \delta(t, \alpha - \alpha_{\mathrm{LB}}) \end{cases} \tag{24}$$

Here, $\acute{\alpha}$ denotes the changed parameter and t the current generation. The values α_{LB} and α_{UB} denote the lower and upper bound of the parameter α. The value of δ is computed according to Eq. (25) (Michalewicz, 1996):

$$\delta(t, y) = y(1 - r^{(1-(t/T))^b}) \tag{25}$$

Here, r is a random number that is sampled with uniform probability from the interval $[0, \ldots, 1]$ and T denotes the maximum number of generations. Hence the change ratio of parameter α is decreased as the evolutionary search progresses, depending on the value of b. The value of b was set to a value of two for all runs of the MOEA.

3.4.7. *VO1$_7$ operator*

The VO1$_7$ operator makes a copy of one basis function tree of the individual and adds it to the individual. This is only done if the resulting individual would not contain more trees than allowed. All terminal nodes of both trees (the cloned and the original tree) are slightly changed, as described for the VO1$_6$ operator.

3.4.8. *VO1$_8$ operator*

The VO1$_8$ operator extends the Gaussian kernel of one terminal node by changing σ slightly (see VO1$_6$ and Eq. (18)). The tree and the terminal node are chosen with a uniform probability. This operator could increase the sensitivity of the RBFN.

3.4.9. *VO1$_9$ operator*

The VO1$_9$ operator extends the Gaussian kernel of one terminal node by changing σ slightly (see VO1$_6$ and Eq. (18)). The tree and the terminal node are chosen with a uniform probability. This operator could increase the specificity of the RBFN.

3.4.10. *VO1$_{10}$ operator*

The VO1$_{10}$ operator moves the Gaussian kernel of one terminal node by changing μ slightly (see VO1$_6$ and Eq. (18)). The tree and the terminal node are chosen with a uniform probability.

3.4.11. *VO1$_{11}$ operator*

The VO1$_{11}$ operator negates one bit of the bit string that corresponds to a weight w in Eq. (17). The bit is chosen with a uniform probability.

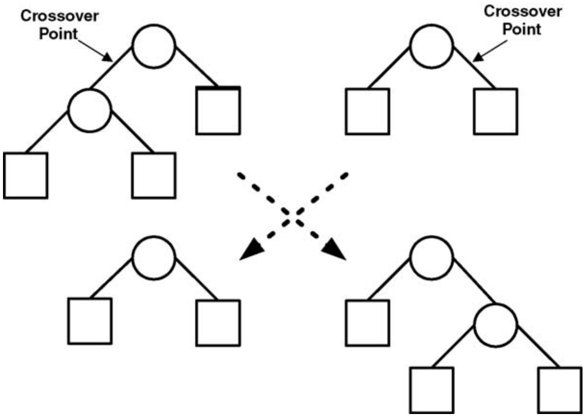

Fig. 5. Crossover between two basis function trees. The upper part shows the trees before the $VO2_1$ operator was applied. The lower part shows the trees after the application of the $VO2_1$.

3.4.12. $VO1_{12}$ operator

The $VO1_{12}$ swaps the weights of two randomly chosen basis functions.

3.4.13. $VO1_{13}$ operator

The $VO1_{13}$ slightly changes the bias value by adding/subtracting a small real number. The real number is sampled from the interval $[-1, \ldots, 1]$ with a uniform probability.

3.4.14. $VO2_1$ operator

The $VO2_1$ operator performs an exchange (crossover) of randomly chosen parts between two basis function trees from two individuals. The two individuals and trees are chosen with a uniform probability. Fig. 5 illustrates the working principle of this operator.

The upper part of Fig. 5 depicts the trees before the application of this operator. The lower part of Fig. 5 depicts the resulting trees. Fig. 5 also shows the crossover points which mark the parts of each tree that are exchanged. The crossover points are chosen such that the resulting trees do not contain more nodes than allowed, and such that no feature is used more than once within the resulting trees. Fig. 6 illustrates the working principle of the $VO2_1$ operator in the feature space.

The left part of Fig. 6 shows two individuals before the $VO2_1$ operator was applied. The right part of Fig. 6 shows the two individuals after the application of the $VO2_1$ operator. It can clearly be seen that the $VO2_1$ operator changed the basis function trees that are depicted as dotted lines (the Gaussian kernels of feature $X0$ were exchanged).

3.4.15. $VO2_2$ operator

The $VO2_2$ operator removes a complete basis function tree from each individual, which is chosen with a uniform probability. The tree is then added to the other individual.

3.4.16. $VO2_3$ operator

The $VO2_3$ operator merges two individuals. If the resulting RBFN contains more basis functions than allowed, trees are removed randomly until this constraint is not violated anymore. The resulting RBFN replaces each of the original two individuals.

3.4.17. $VO2_4$ operator

The $VO2_4$ operator performs crossover between two bit strings (e.g. Michalewicz, 1996). A bit string is chosen with a uniform probability in each individual.

3.5. Archive

As mentioned earlier, an archive contains the best (elite) individuals that the MOEA has found so far. The archive ensures that the best individuals are preserved, as they could otherwise get lost due to the randomness of the selection process (Zitzler et al., 2002). For practical reasons, an archive can only store a limited number of individuals (large numbers of individuals increase the memory demands and the execution time of the algorithm). However, this can result in the loss of non-dominated solutions, which is a problem known as partial deterioration (Laumanns et al., 2002b) and is illustrated in Fig. 7.

The leftmost subplot depicts the archive of a MOEA after the first generation ($g = 1$). During the next generation, the candidate solution denoted as A (depicted in the objective vector space as diamond) is replaced by another incomparable candidate solution.[1] The replacement of individuals in the archive can happen because the size of the archive is limited. The new member of the archive is depicted as a diamond and denoted as B in the second subplot from the left ($g = 2$). The new member of the archive is then replaced by another incomparable candidate solution during the next generation, which is again depicted as a diamond and denoted as C in the third subplot from the left ($g = 3$). All candidate solutions that were produced in the last three generations are shown in the right subplot. It can clearly be seen that

[1] An individual is incomparable if it is not dominated by any other individual in the population, or conversely if it belongs to the set of non-dominated candidate solutions (see also Definition 1).

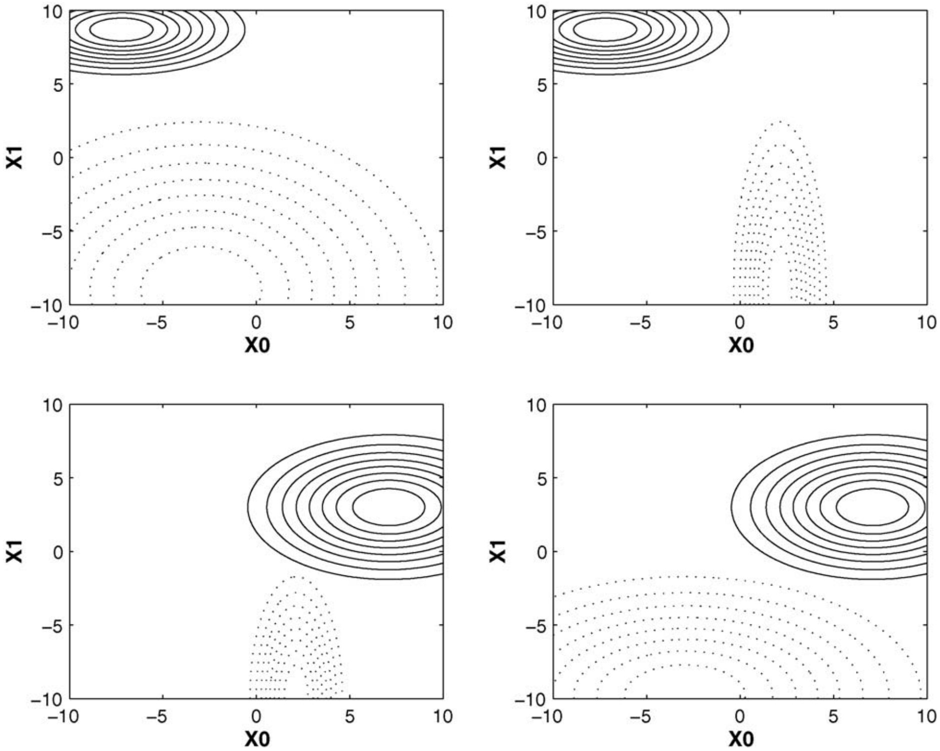

Fig. 6. The left part shows two RBFNs before the application of the VO2$_1$ operator. The right part shows the two RBFNs after the VO2$_1$ operator was applied. The VO2$_1$ operator changed the basis function trees depicted as dotted lines (the Gaussian kernels of feature $X0$ were exchanged).

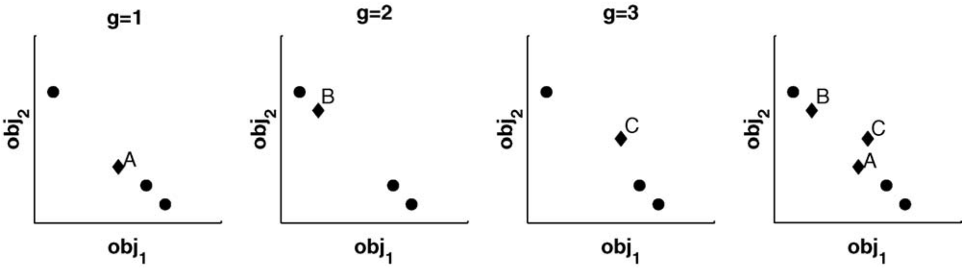

Fig. 7. Development of a size limited archive (both objectives have to be minimised).

the approximation of the Pareto set has deteriorated, as candidate solution C is clearly dominated by all other candidate solutions. It could only become a member of the archive due to the removal of candidate solution A after the second generation. Laumanns et al. (2002b) have proposed an archiving strategy that uses an archive of bounded size, but does not exhibit the problem of partial deterioration. For the present purposes, the implemented MOEA deploys this archiving strategy.

4. Results and discussions

The implemented MOEA is evaluated on several benchmark datasets. In addition, it is evaluated on a 'real-world' medical dataset. The benchmark datasets comprise the leukaemia dataset (Freireich et al., 1963) (see also Section 1) and four artificial datasets.

The datasets were randomly split into a training dataset and test dataset for each experiment. The training datasets contained two-thirds and the test dataset one-third of the original dataset. Before the MOEA was run a holdout dataset was randomly selected from the training dataset. It contained one-third of the training dataset. The MOEA was then run and the generated individuals in the archive and the population were re-evaluated on the holdout dataset. As each run of the MOEA produces several trade-off solutions, the RBFN with the best fit to the holdout dataset (see Eq. (16)) was chosen as the final

Table 6
Parameter values used for each MOEA run

Parameter	Parameter value
Population size	100
Number of generations	500
Crossover probability	0.7
Mutation probability	0.3

output of the algorithm. It should be noted, that if there were several RBFNs with the same fit to the validation data, the RBFN with the smallest number of basis functions and features was chosen. To choose the model with the best fit to the holdout dataset rather than the training dataset makes it more likely that the model generalises on unseen data. This method is also referred to as holdout method (Bishop, 1995). During the experiments the parameters summarised in Table 6 were used.

4.1. Evaluation on the leukaemia data

This section applies the implemented MOEA to the leukaemia data (see Section 1). Fig. 8 shows the Kaplan–Meier estimates and the model estimates.

It can clearly be seen that the estimates agree with the Kaplan–Meier estimates.

4.2. Evaluation on the artificial dataset 1

This artificial dataset was inspired by the well-known XOR classification problem. It has two binary features: $X0$ and $X1$ and a third binary feature that has to be predicted. Two new features replaced the third feature in order to simulate a survival analysis problem. The first feature (I) indicated whether the event occurred to the sample and the second feature (Time) determined the observation time of the sample. The maximum observation time was set to a value of ten. The actual data consisted of 50 samples for each feature value combination (see Table 7).

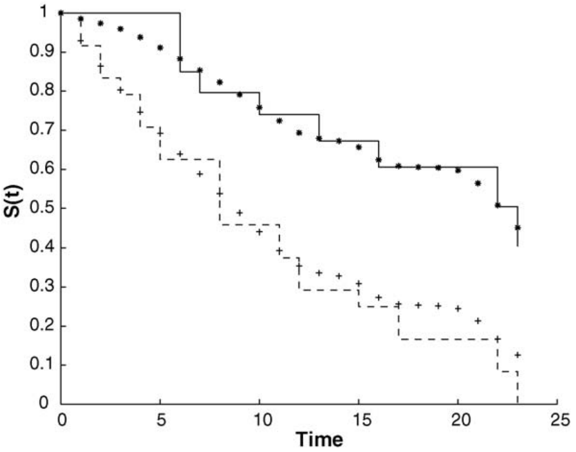

Fig. 8. Kaplan–Meier estimates for the leukaemia dataset together with the yearly estimations of the generated model. The solid line corresponds to the Kaplan–Meier estimates for patients who were treated and the dashed line to those without treatment. The stars correspond to the estimations of the model for treated patients whereas the crosses correspond to the estimations for untreated patients.

Table 7
Possible combinations of the feature values (two left columns)

x_0	x_1	μ	σ	Samples
0	0	1.1	0.15	50
0	1	1.8	1.1	50
1	0	1.8	1.1	50
1	1	1.1	0.15	50

Parameters of the inverse lognormal distribution (two right columns).

Two values were generated to obtain the observation time for a sample. The first value (α) was sampled from the interval $[0, \ldots, 10]$ with a uniform probability. The second value (β) was sampled from a inverse lognormal distribution (Evans et al., 1993). This distribution is characterised by the parameters μ and σ. The parameter values for a particular sample (feature value combination) are summarised in Table 7. To simulate censoring the actual observation time and the indicator value were determined according to Eq. (26):

$$(I, \text{Time}) = \left| \begin{array}{ll} (1, \beta) & \text{for}\, \alpha \geq \beta \\ (0, \alpha) & \text{otherwise} \end{array} \right. \qquad (26)$$

Fig. 9 depicts the probability density functions (left) and the survival function (right) for the parameters in Table 7.

Here, the dashed lines correspond to the parameters $\mu = 1.1$ and $\sigma = 0.15$ and the solid lines to the parameters $\mu = 1.8$ and $\sigma = 1.1$. It can clearly be seen that this problem does not exhibit proportional hazards as the survival curves cross. Hence, the standard Cox model would be unsuitable (see also Section 1). Fig. 10 shows the responses of the evolved RBFN for each possible feature value combination together with the 'true' survival functions.

The 'true' survival function for the specific feature value combination is shown as dotted line. Fig. 10 shows that the evolved model predicts the survival functions correctly. The r^2 measure was used to measure the correlation between the expected values and the predicted values. The evolved model achieved an r^2 value of 0.9154. This shows that the proposed approach can be applied to survival data with non-proportional hazard distributions.

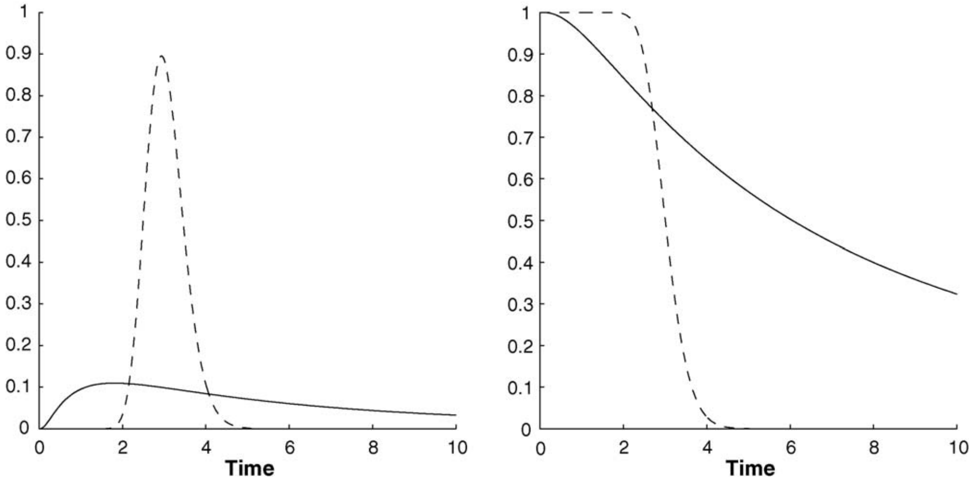

Fig. 9. Probability density functions (left) and the survival functions (right) for the parameters $\mu = 1.1, \sigma = 0.15$ (dashed line) and $\mu = 1.8, \sigma = 1.1$ (solid line).

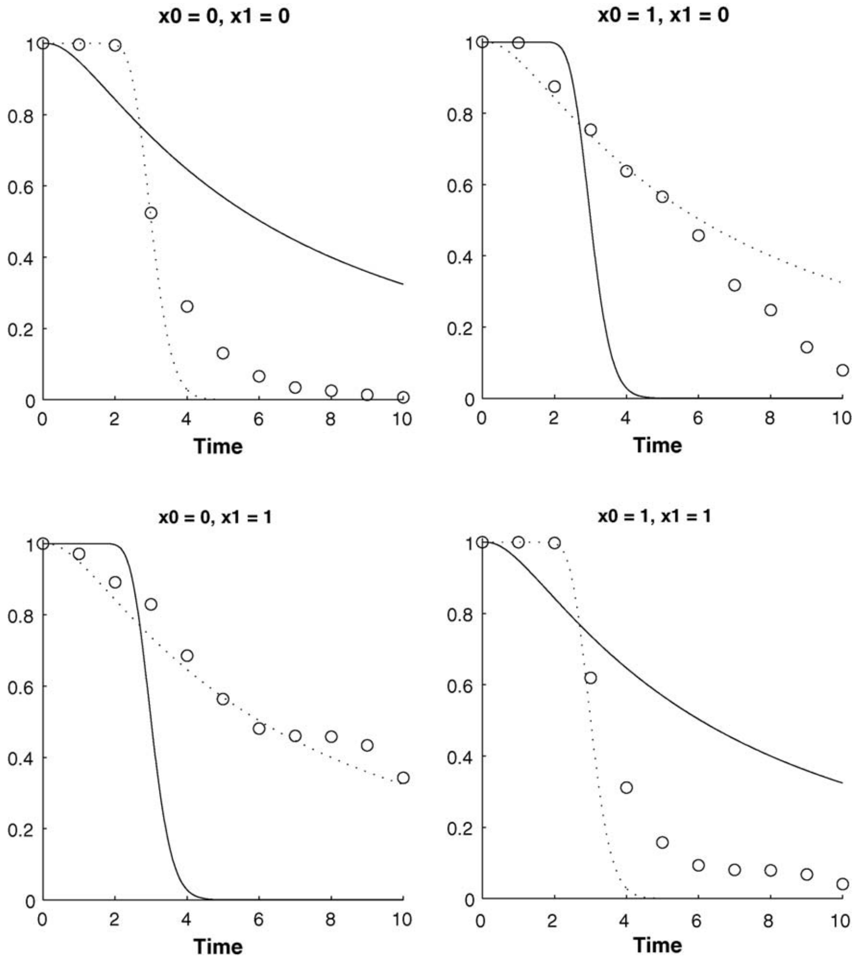

Fig. 10. Response values of the evolved model for each feature value combination. The correct survival function is shown as dotted line.

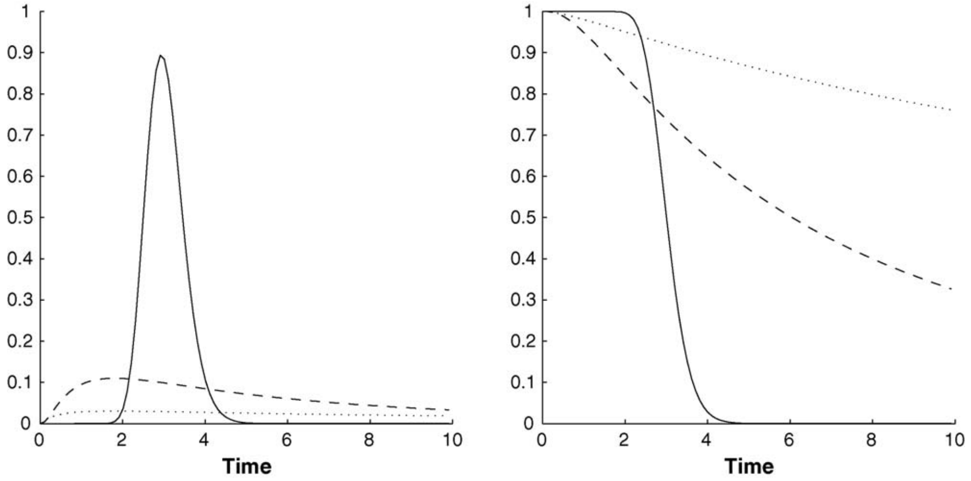

Fig. 11. Probability density functions (left) and the survival functions (right) for the parameters $\mu = 1.1$, $\sigma = 0.15$ (solid line); $\mu = 1.8$, $\sigma = 1.1$ (dashed line); $\mu = 3.5$, $\sigma = 1.7$ (dotted line).

4.3. Evaluation on the artificial dataset 2

This artificial dataset was generated in the same manner as the first artificial dataset. However, it contains an additional ‘noise’ variable, which was generated by sampling from the interval $[0, \ldots, 1]$ with a uniform probability. The evolved model achieved an r^2 measure of 0.9219 and excluded the ‘noise’ variable. This shows that the implemented MOEA can be applied to non-proportional and noisy hazard distributions.

4.4. Evaluation on the artificial dataset 3

This artificial dataset was generated in a similar manner as the first artificial dataset. However, it contains three features that can have the value 1 or 0. Hence, there are eight possible combinations as shown in Table 8, which also contains the corresponding parameter values for the inverse lognormal distributions.

Table 8
Possible combinations of the feature values (two left columns)

x_0	x_1	x_2	μ	σ
0	0	0	1.10	0.15
0	0	1	3.50	1.70
0	1	0	1.80	1.10
0	1	1	3.50	1.70
1	0	0	1.80	1.70
1	0	1	3.50	1.70
1	1	0	1.10	0.15
1	1	1	3.50	1.70

Parameters for the inverse lognormal distribution (two right columns).

The probability density functions and survival functions for the three lognormal distributions are shown in Fig. 11.

Fifty samples were produced for each feature value combination. Fig. 12 depicts the estimations of the evolved model for each feature value combination as circles together with the ‘true’ survival functions, which are depicted as dotted line.

It can clearly be seen that the evolved model estimates these artificial survival functions. The evolved model achieves an r^2 value of 0.9491.

4.5. Evaluation on the artificial dataset 4

This artificial dataset was originally proposed in Eleuteri et al. (2003). The artificial lifetime distribution is a mixture of a Weibull and gamma distribution. The parameters of these distributions depend on the values of the two features x_1 and x_2 as shown in Eq. (27):

$$f(t|x) = x_2 W(x_1^2, \sin(x_1)^3 + 2) + (1 - x_2) G((x_1+1)^2, \times \exp(\cos(x_1) + 2)) \tag{27}$$

The values of x_1 and x_2 were first sampled from a bivariate Gaussian with the mean vector $(0, 0)$ and the covariance $[1.0, -0.5; 1.0, 0.5]$. The variable x_2 was then transformed to a binary indicator (values greater than or equal to zero were set to one, values less than zero were set to zero). Five hundred training and test data samples were produced. Each sample was censored with a probability of 0.27 (see Eleuteri et al., 2003).

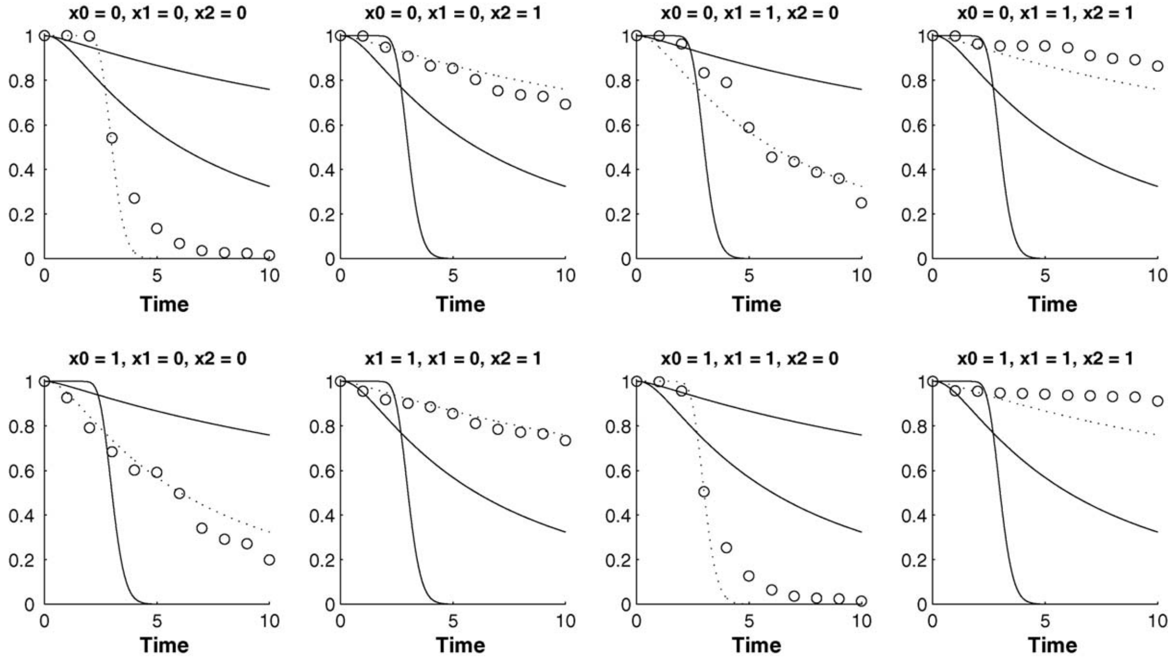

Fig. 12. Estimations of the evolved model for each feature value combination (circles). The dotted lines depict the ‘true’ survival functions for the particular feature value combination.

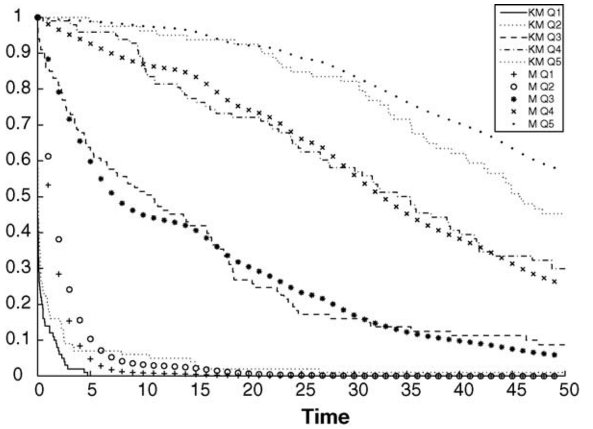

Fig. 13. Kaplan–Meier estimates (KM Q1–KM Q5) and average survival function estimates of the evolved model (M Q1–M Q5) for all five groups.

The estimates of the evolved RBFN at $t = 8$ were then ordered and divided into five groups of equal size.[2] The Kaplan–Meier estimates and the average survival function estimates of the evolved RBFN for each group are shown in Fig. 13.

The model captures the distribution of the data if both estimates agree (Eleuteri et al., 2003). This can be observed in Fig. 13.

4.6. Evaluation on a medical problem

This section evaluates the implemented MOEA on a ‘real-world’ medical dataset of uveal melanoma patients (Damato, 2000). Uveal melanomas, which have an occurrence rate of six per million per year (Damato, 2005), arise from melanocytes in the uvea. The uvea consists of the choroid, ciliary body and iris. Patients with uveal melanoma usually have symptoms, such as blurred vision, flashing lights and visual field loss. Without treatment, many eyes become blind, painful and cosmetically unsightly. Approximately 50% of all patients with uveal melanoma ultimately die of this disease, nearly always as a result of haematogenous spread of tumour to the liver (i.e. through the blood circulation). Estimating the probability of survival for uveal melanoma patients has many benefits. It allows clinicians to review their practice and advice their patients on the best course of treatment. Furthermore, it allows patients to plan their lives and provide future care for their dependents.

The samples consisted of six features, which are summarised in Table 9. The training data consisted of 1820 samples and the test data of 978 samples. The former contained 549 events whereas the latter contained 286 events.

The MOEA was applied to the training data and the evolved model was evaluated on the test data us-

[2] The median survival time is 8.4 (Eleuteri et al., 2003).

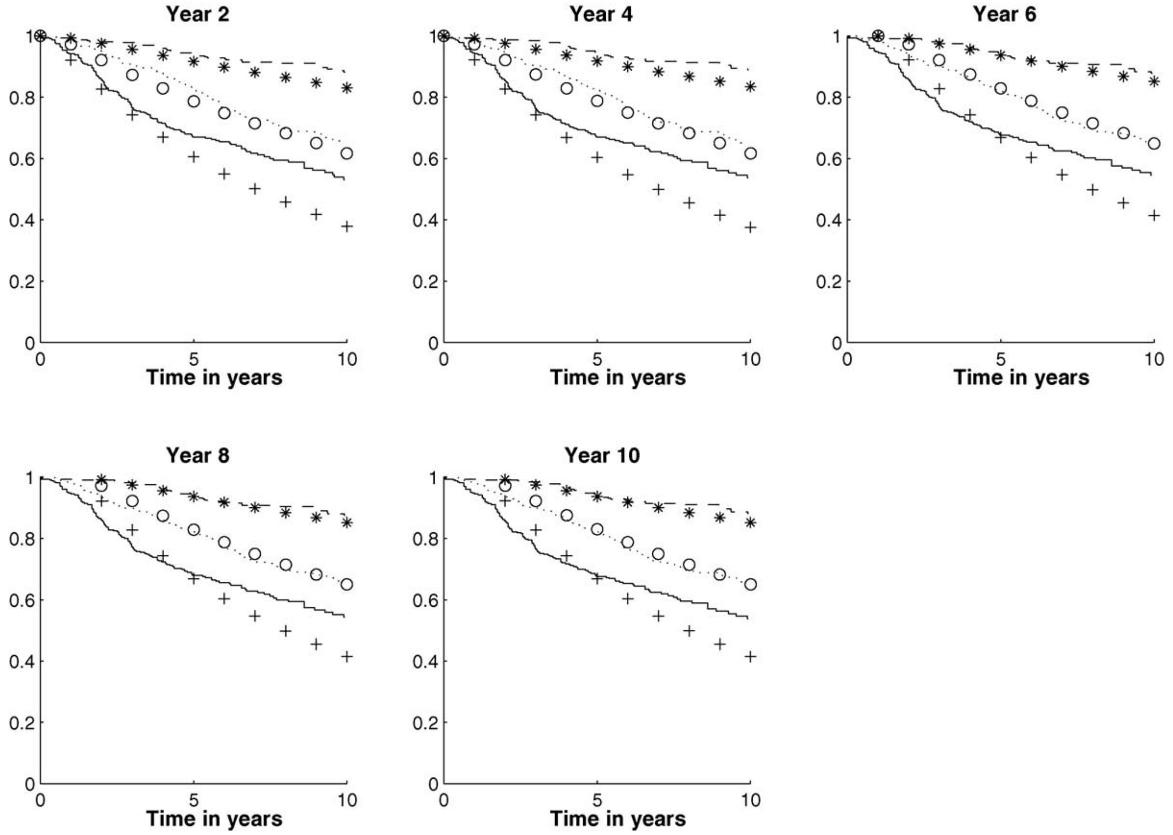

Fig. 14. Kaplan–Meier estimates (dashed, dotted and solid lines) together with the average group estimates of the model (stars, circles, and ses) for each group created at time 2, 4, 6, 8 and 10.

ing the rank-based discrimination index (C^{td}) proposed by Antolini et al. (2005). This measure was inspired by the *C-index* put forward by Harrell et al. (1996). The *C-index* is equivalent of the AUC measure (Hanley and McNeil, 1982) for survival data. The closer the C^{td} index is to one the better the model discriminates the data. The evolved model achieved a value of 0.696 with a standard deviation of 0.0142.

Table 9
Features of the uveal melanoma dataset

Name	Type	Description
Antora	Dichotomous	Indicates whether the tumour is at the front or the back of the eye (anterior choroid or posterior choroid)
Age	Continuous	Age of the patient when (s)he entered the study
Ludb	Continuous	Tumour dimension as measured by ultrasonography
Gender	Dichotomous	n/a
Time	Continuous	Observation time of the patient
Indicator	Dichotomous	Indicates whether or not the patient was censored

The survival probability values for the test data were arranged in ascending order for time 2, 4, 6, 8 and 10. Three groups were created for these time values with an equal number of samples. The Kaplan–Meier estimates were computed for each group. Fig. 14 shows the Kaplan–Meier estimates (dashed, dotted and solid lines) together with the average group estimates of the model (stars, circles, and ses) for each group and time.

It can be observed that the model produces estimates that approximate the Kaplan–Meier estimates, apart from the group with the worst survival curve. Here,the model produces more pessimistic estimates as time progresses. It has to be noted, however, that censoring also increases as time progresses making the Kaplan–Meier estimates less accurate.

The distribution of the grouped samples was also analysed using the approach described in Setzkorn and Paton (2005) (the samples were classified according to their group membership). Interestingly, it was observed that the produced model classifies samples according to the existing TNM staging (Sobin and Wittekind, 2002) of uveal melanoma patients. TNM is an abbreviation for tumour nodes metastasis.

5. Conclusions and further work

A multi-objective evolutionary algorithm for the extraction of models for survival analysis has been proposed and evaluated. The evaluation of the MOEA on several benchmark datasets and one medical problem has shown that the approach is capable of producing accurate and valid models. The evaluation on the artificial dataset also emphasised that the approach can cope with interaction effects and noisy non-proportional hazard distributions. This is in contrast to, for example, the Cox model. In its standard form, it does not consider interaction effects and non-proportional hazard distributions. Only extensive statistical knowledge allows one to apply it to such dataset successfully.

The experiments have also shown that the generated models can produce survival function values for given feature values without estimating the baseline hazard first. Another advantage of the proposed approach is that it could be used to model cause-specific hazards as suggested in Biganzoli et al. (2001) because it can cope with several indicator values.

One drawback of the current approach is that the original data have to be time-coded. This results in a large datasets and long execution times. Hence, if the original number of samples is large, the application of the approach can be impractical (Baesens et al., 2005). However, this disadvantage could be alleviated by executing the proposed approach in parallel by harvesting the computational power of idle computers as suggested in Setzkorn and Paton (2004).

The approach is currently being evaluated in an international double-blind study, which applies several existing and new approaches for survival analysis to a large uveal melanoma dataset. The approaches include the Cox model, the Lognormal model, the PLANN model (Biganzoli et al., 1998), the PLANNARD model (Lisboa et al., 2003), and the presented MOEA. The training data and test data were distributed via the General Ocular Oncology Database, Geoconda (Setzkorn et al., 2005; Taktak and Fisher, in press), which is an Internet environment for ocular cancer research. Participants of this study had only access to the test data that did not contain indicator values. Participants submitted their results (response of the models for the test data) to an independent evaluation committee who determined, for example, the C^{td} index of the particular model. Note that, the evaluation committee did not know the origin of the response values. This removed possible biases.

Acknowledgements

We would like to express our gratitude to Dr. Jessica Grabham, Dr. Tony Fisher and Dr. Peter McBurney for the suggestions they made to improve the paper.

References

Afifi, A., Clark, V., May, S., 2003. Computer-Aided Multivariate Analysis. Chapman and Hall.

Allison, P., 1997. Survival Analysis using SAS: A Practical Guide. SAS Publishing.

Antolini, L., Boracchi, P., Biganzoli, E., 2005. A time-dependent discrimination index for survival data. Stat. Med. 24 (24), 3927–3944.

Baesens, B., Gestel, T.V., Stepanova, M., Vanthienen, J., 2005. Neural network survival analysis for personal loan data. J. Oper. Res. Soc. 56 (9), 1089–1098.

Biganzoli, E., Boracchi, P., Mariani, L., Marubini, E., 1998. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat. Med. 17 (10), 1169–1186.

Biganzoli, E., Boracchi, P., Marubini, E., 2001. Modelling cause specific hazards with radial basis functions artificial neural networks: an application to 2233 breast cancer patients. Stat. Med. 20, 3677–3694.

Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Oxford.

Burges, C., 1998. A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Disc. 2 (2), 121–167.

Clark, T., Bradburn, M., Love, S., Altman, D., 2003. Survival analysis. Part IV. Further concepts and methods in survival analysis. Brit. J. Cancer 89, 781–786.

Collet, D., 1994. Modelling Survival Data in Medical Research. Chapman and Hall, London.

Cox, D., 1972. Regression models and life tables (with discussion). J. Roy. Stat. Soc. Ser. B 74, 187–220.

Cramer, N., 1985. A representation for the adaptive generation of simple sequential programs. Proceedings of the First International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Inc., 183–187.

Damato, B., 2000. Ocular Tumours: Diagnosis and Treatment. Butterworth Heinemann.

Damato, B., 2005. Current management of uveal melanoma. Eur. J. Cancer Suppl. 3 (3), 433–435.

Deb, K., 2001. Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Europe.

Dhar, V., Chou, D., Provost, F.J., 2000. Discovering interesting patterns for investment decision making with glower—a genetic learner overlaid with entropy reduction. Data Min. Knowl. Disc. 4 (4), 251–280.

Dietterich, T., 1995. Overfitting and undercomputing in machine learning. ACM Comput. Surv. 27 (3), 326–327.

Elder, J., Pregibon, D., 1996. A statistical perspective on knowledge discovery in databases. In: Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (Eds.), Advances in Knowledge Discovery and Data Mining. AAAI Press/The MIT Press, pp. 83–113.

Eleuteri, A., Tagliaferri, R., Milano, L., De Placido, S., De Laurentiis, M., 2003. A novel neural network-based survival analysis model. Neural Networks 16 (5–6), 855–864.

Evans, M., Hastings, N., Peacock, B., 1993. Statistical Distributions. John Wiley and Sons Inc.

Freireich, E., Gehan, E., Frei, E., Schroeder, L., Wolman, I., Anbari, R., Burgert, E., Mills, S., Pinkel, D., Selawry, O., Moon, J., Gendel, B., Spurr, C., Storrs, R., Haurani, F., Hoogstraten, B., Lee, S., 1963. The effect of 6-mercaptopurine on the duration of steroid-induced remissions in acute leukaemia. Blood 21, 699–716.

Freitas, A., 2002. Data Mining and Knowledge Discovery With Evolutionary Algorithms. Springer-Verlag.

Geman, S., Bienenstock, E., Doursat, R., 1992. Neural networks and the bias/variance dilemma. Neural Comput. 4, 1–58.

Grefenstette, J., 1991. Lamarckian learning in multi-agent environments. In: Belew, R., Booker, L. (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kaufman, San Mateo, CA, pp. 303–310.

Griffin, S., 1998. Lost to follow-up: the problem of defaulters from diabetes clinics. Diab. Med. 15 (3), 14–24.

Hand, D., 1997. Construction and Assesment of Classification Rules. Wiley, New York.

Hanley, J., McNeil, B., 1982. The meaning and use of the area under a receiver operating characteristic. Radiology 143, 29–36.

Harrell, F., Lee, K., Mark, D., 1996. Tutorial in biostatistics, multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Stat. Med. 15, 433–435.

Humphrey, M., Cunningham, S., Witten, I., 1998. Knowledge visualization techniques for machine learning. Intell. Data Anal. 2, 333–347.

Janikow, C., 1993. A knowledge-intensive genetic algorithm for supervised learning. Mach. Learn. 13, 189–228.

Kleinbaum, D., 1996. Survival Analysis: A Self-Learning Text. Springer.

Koza, J., 1998. Genetic programming. In: Williams, J., Kent, A. (Eds.), Encyclopedia of Computer Science and Technology, vol. 39. Marcel-Dekker, pp. 29–43.

Kuncheva, L., 2004. Combining Pattern Classifiers: Methods and Algorithms. John Wiley and Sons Inc.

Laumanns, M., Thiele, L., Deb, K., Zitzler, E., 2002a. Archiving with guaranteed convergence and diversity in multi-objective optimization. Proceedings of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann Publishers, pp. 439–447.

Laumanns, M., Thiele, L., Deb, K., Zitzler, E., 2002b. Combining convergence and diversity in evolutionary multi-objective optimisation. Evolut. Comput. 10, 263–282.

Laumanns, M., Zitzler, E., Thiele, L., 2000. A unified model for multi-objective evolutionary algorithms with elitism. Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000). IEEE Press, Piscataway, New Jersey, pp. 46–53.

Lawless, J., 1982. Statistical Models and Methods for Lifetime Data. Wiley, New York.

Lisboa, P., Wong, H., Harris, P., Swindell, R., 2003. A bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer. Artif. Intell. Med. 28 (1), 1–25.

Marubini, E., Valsecchi, M., 1995. Analysing Survival Data from Clinical Trials and Observational Studies. Wiley.

Michalewicz, Z., 1996. Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin.

Michalewicz, Z., Fogel, D., 2005. How to Solve it: Modern Heuristics, 2nd ed. Springer, Berlin.

Pazzani, M., Mani, S., Shankle, W., 1997. Comprehensible knowledge discovery in databases. In: Shafto, M., Langley, P. (Eds.), Proceedings of the 19th Annual Conference of the Cognitive Science Society. Lawrence Erlbaum, pp. 596–601.

Setzkorn, C., 2005. On the use of multi-objective evolutionary algorithms for classification rule induction. Ph.D. Thesis. University of Liverpool, Department of Computer Science, Liverpool, United Kingdom.

Setzkorn, C., Paton, R., 2004. Javaspaces—an affordable technology for the simple implementation of reusable parallel evolutionary algorithms. In: López, J., Benfenati, E., Dubitzky, W. (Eds.), Knowledge Exploration in Life Science Informatics—KELSI 2004 (LNAI 3303). Springer-Verlag New York, Inc., pp. 151–161.

Setzkorn, C., Paton, R., 2005. On the use of multi-objective evolutionary algorithms for the induction of fuzzy classification rule systems. BioSystems 81 (2), 101–112.

Setzkorn, C., Taktak, A., Damato, B., 2005. Geoconda: a web environment for multi-centre research. Tech. Re ULCS-05-011. Department of Computer Science, University of Liverpool, Liverpool, United Kingdom.

Shi, Y., Eberhart, R., Chen, Y., 1999. Implementation of evolutionary fuzzy systems. IEEE Trans. Fuzzy Syst. 7 (2), 109–119.

Silverman, B., 1999. Density Estimation for Statistics and Data Analysis. Chapman and Hall.

Sobin, L., Wittekind, C., 2002. Classification of Malignant Tumours. Wiley/Liss.

Spears, W., 1995. Adapting crossover in evolutionary algorithms. In: McDonnell, J.R., Reynolds, R., Fogel, D. (Eds.), Proceedings of the Fourth Annual Conference on Evolutionary Programming. MIT Press, Cambridge, MA, pp. 367–384.

Taktak, A., Fisher, A., in press. Outcome Prediction in Cancer. Elsevier.

Therneau, T., Grambsch, P., 2000. Modeling Survival Data: Extending the Cox Model. Springer.

Willett, J.S.J., 1993. It's about time: using discrete-time survival analysis to study duration and the timing of events. J. Educ. Stat. 18 (2), 155–195.

Wolpert, D., Macready, W., 1997. No free lunch theorems for optimisation. IEEE Trans. Evolut. Comput. 1 (1), 67–82.

Zitzler, E., Laumanns, M., Thiele, L., 2002. Spea2: improving the strength pareto evolutionary algorithm. In: Giannakoglou, K. (Ed.), Proceedings of the EUROGEN2001 Conference. Barcelona, Spain, pp. 95–100.