Variable Selection in Cox’s Proportional Hazards Model Using a Parallel Genetic Algorithm

Mu Zhu and Guangzhe Fan
Department of Statistics and Actuarial Science
University of Waterloo
Waterloo, Ontario N2L 3G1, Canada
Email: {m3zhu,gfan}@uwaterloo.ca

September 7, 2006

Abstract

One of the most common objectives for analyzing censored survival data with covariates is to identify a small number of key risk factors. Cox’s proportional hazards model is perhaps the most widely used statistical model in such a context. We propose a parallel genetic algorithm (PGA) to select important covariates in the Cox model. Our simulation study shows that PGA is capable of finding the correct subset of covariates with significantly higher probability than stepwise selection. We also illustrate our method with a real data set concerning the treatment of primary biliary cirrhosis of the liver.

Key Words Akaike Information Criterion (AIC); Bayesian Information Criterion (BIC); Bayesian model averaging; LASSO; Majority vote; Stepwise selection.

1 Introduction

Suppose we have survival data \( \{(y_i, \mathbf{x}_i, c_i) : i = 1, 2, ..., n\} \), where \( y_i \) is the survival time of individual \( i \); \( \mathbf{x}_i \in \mathbb{R}^p \) is a vector of covariates; and \( c_i \) is an indicator variable such that \( c_i = 0 \) means no event up to time \( y_i \) and hence the survival time \( y_i \) is right-censored. Cox’s proportional hazards model (Cox 1972) is perhaps the most popular statistical model for analyzing such data; it assumes that the hazard rate at time \( t \) given the covariates \( \mathbf{x} \) is equal to

\[
\lambda(t|\mathbf{x}) = \lambda_0(t) \exp (\mathbf{\beta}^T \mathbf{x}).
\]
The baseline hazard function $\lambda_0(t)$ is regarded as a nuisance parameter whereas the key parameter of interest, $\beta$, is estimated by maximizing the partial likelihood function,

$$l(\beta) = \prod_{i=1}^{n} \left( \frac{\exp(\beta^T x_i)}{\sum_{r \in R_i} \exp(\beta^T x_r)} \right)^{c_i},$$

where $R_i$ denotes the risk set at time $y_i$, i.e., the set of individuals who have not yet experienced an event. Hereafter, we will refer to this model simply as the “Cox model.”

Let $C = \{x_1, x_2, ..., x_p\}$ be the set containing a total of $p$ possible covariates, which may include functions of variables such as main effects and interaction terms. When $p$ is large, it is often the case that only a small number of them actually affect the hazard rate. Under such circumstances, it becomes desirable to work with a small subset of $C$. Let $\Omega$ be the collection of all subsets of $C$, e.g., $\omega = \{x_1, x_4, x_5\} \in \Omega$. The goal of variable selection is to find the best $\omega \in \Omega$ that contains the key risk factors.

Finding the best $\omega \in \Omega$ is a typical combinatorial optimization problem. There are altogether $2^p - 1$ non-trivial subsets of $C$. Even for moderately large $p$, an exhaustive search is not possible. By far, the most widely used methods for variable selection in the Cox model are stepwise procedures such as forward selection, backward elimination, or a combination of both. It is well-known that these are greedy methods that can easily be trapped at various local sub-optima.

Recently, some other methods have been proposed. Tibshirani (1997) proposed an $L_1$-norm penalized regression procedure called LASSO, which performs regression shrinkage and variable selection simultaneously. Faraggi and Simon (1998) proposed a Bayesian approach; and Hoeting et al. (1999) showed that tools for Bayesian model averaging can be used to calculate the marginal posterior effect probabilities of the variables and hence help us make variable selection decisions.

In this article, we propose a new approach using a parallel genetic algorithm (PGA). PGA was first developed by Zhu and Chipman (2005) in the context of multiple linear regression. Here we propose a parallel genetic algorithm for the Cox model with censored survival data. The article is organized as follows: In Section 2, we describe the main method. In Section 3, we conduct a number of simulation studies. In Section 4, we present a real data example. Finally, Section 5 contains some discussions and a short summary.

# 2 Method

## 2.1 Genetic Algorithm (GA)

The genetic algorithm (e.g., Goldberg 1989) is a stochastic-search, global optimization algorithm. Start with a number of randomly generated candidates (the initial population). Using
the Darwinian principle of "the survival of the fittest," GA gradually eliminates the weaker (less optimal) candidates and allows the stronger ones to survive and generate offsprings. This goes on for a number of generations until, in the end, good solutions are produced.

In a typical setting, each individual \( \omega \) is represented by a binary string, say, of length \( p \), which is treated as the genetic code (DNA) of \( \omega \); each position can be regarded as a single gene. Starting with a randomly generated population of size \( m \), \( \{ \omega_1, \omega_2, \ldots, \omega_m \} \), a new generation is produced with three genetic operations: selection, reproduction, and mutation.

### Selection
Each individual is evaluated by a fitness function \( F \), often the objective function for the underlying optimization problem, and assigned a score. When the goal is to maximize (minimize) \( F \), those with high (low) scores are given higher chances to survive to the next generation. See Table 1.

<table>
<thead>
<tr>
<th>( \omega_i )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
<th>( p )</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega_1 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>( s_1 )</td>
</tr>
<tr>
<td>( \omega_m )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>( s_m )</td>
</tr>
</tbody>
</table>

### Reproduction
Two individuals are selected at random to produce a child. Typically, a cross-over position is chosen at random between 1 and \( p \), say \( j \); the child then inherits the first \( j \) genes from the father and the rest \( p - j \) genes from the mother. See Table 2; the cross-over position in the illustration is between 3 and 4.
Mutation. At birth, an individual is allowed, with a certain small probability (called the mutation rate), to alter its genetic code at a randomly chosen position. In the typical setting where binary codes are used, this amounts to flipping a 0 to a 1 and vice versa. See Table 3.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>After</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

To implement the GA in practice, there are a number of control parameters that must be preselected, e.g., the population size, the mutation rate, and the decision rule that controls which individuals can survive to the next generation. We delay these discussions to Section 2.3.

### 2.2 Parallel Evolution

The fact that candidate solutions are often represented as binary strings makes the GA appear particularly well suited for the variable selection problem, because subsets of variables can be coded as binary strings in a rather trivial manner, e.g., \( \omega = \{x_1, x_4, x_5\} \) is the same as writing \( \omega = (1, 0, 0, 1, 1, 0, ..., 0) \); in this case \( \Omega \) can be thought of as the space of all possible such binary strings. A number of studies (e.g., Draper and Fouskakis 2000; Zhu and Chipman 2005) have shown, however, that the GA is not an effective variable selection tool.

Zhu and Chipman (2005) proposed the idea of parallel evolution and showed it to be very effective in the context of multiple linear regression.

The main idea behind parallel evolution is as follows: Instead of running one long evolutionary path, paying careful attention to the choices of the various control parameters, simply create a number of “parallel universes” and run a relatively mindless and short evolutionary path in each one of them. Here, “mindless” means using a standard set of control parameters for every problem and “short” means stopping each evolutionary path well before it has converged; more details are given below in Section 2.3.

Let there be a total of \( B \) parallel universes. We start a different evolutionary path in each universe \( b = 1, 2, ..., B \), each for a total of \( N \) generations, where \( N \) is relatively small. As Zhu and Chipman (2005), we assume the population size in each universe is fixed at \( m \) in every generation. Let

\[
\mathcal{P}(b, t) = \text{generation-}t \text{ population in universe } b.
\]

The main idea is to define \( 0 \leq r(j, b) \leq 1 \) as a measure of the importance for variable \( j \)
based on $N$ generations of evolution in universe $b$:

$$r(j, b) = \frac{1}{m} \sum_{i=1}^{m} \omega_i(j) \quad \text{for} \quad \omega_1, \omega_2, \ldots, \omega_m \in \mathcal{P}(b, N).$$

In other words, for each universe $b$, $r(j, b)$ is the percentage of last-generation candidates that contain variable $j$. The important variables are then selected by taking a majority vote across all universes. That is, we define

$$\bar{r}_j = \frac{1}{B} \sum_{b=1}^{B} r(j, b)$$

and use $\bar{r}_j$ as an importance measure to rank variable $j$; the top $q < p$ variables are selected.

The number $q$ can be determined by plotting $\bar{r}_j$ from the largest to the smallest and looking for a “big gap,” much like using the scree plot in principal component analysis (e.g., Mardia et al. 1979; Jolliffe 2002; Zhu and Ghodsi 2005). A simple rule-of-thumb is given by Zhu and Chipman (2005) for assessing the statistical significance of the gap as well as choosing the total number of parallel paths used ($B$): A gap of size $G$ can be regarded as statistically significant and the number of paths $B$ can be regarded as adequate if

$$G \geq 0.8225B^{-1/2}.$$  

Typically, one would start with a fairly small $B$, say $B = 20$ or $B = 25$, find the largest gap and check whether this gap is statistically significant. If not, one could easily run a few more paths.

### 2.3 Some Details

The standard set of control parameters used by Zhu and Chipman (2005) for every parallel path is as follows: For each parallel path, the population size $m$ is always set to be either $p$ or $p+1$ — whichever is even — where $p$ is the total number of candidate covariates; the mutation rate is always set to be equal to $1/m$; and the survival pool always consists of the top half of the current population (as measured by their fitness values). To ensure termination of each path before convergence, Zhu and Chipman (2005) proposed the following strategy: Run the GA a couple of times with different initial population, record the average number of generations needed to achieve convergence and use half that number of generations for each individual path. In this article, we use these default settings throughout and refer the readers to Zhu and Chipman (2005) for more details.

### 2.4 Fitness Function

PGA was originally presented in the context of multiple linear regression. To apply PGA to selecting important variables in the Cox model, we only need to specify an appropriate
fitness function. As mentioned earlier, the Cox model is often fitted by maximizing the
partial likelihood function (1). Given a subset \( \omega \), we use an AIC-typed criterion as the
fitness function to evaluate the corresponding Cox model,

\[
F(\omega) = 2 \log(l(\hat{\beta}_\omega)) - 2q_\omega.
\] (3)

In equation (3), \( l(\hat{\beta}_\omega) \) is the partial likelihood as given in (1); \( \hat{\beta}_\omega \) is the (MLE) estimate of \( \beta \) when the covariates are restricted to the subset \( \omega \); and \( q_\omega = |\omega| \) is the size of the subset \( \omega \).

For any given subset \( \omega \), we obtain \( \hat{\beta}_\omega \) by maximizing the partial likelihood (1) using the
function `coxph` in the package `R` (R Development Core Team 2004); we use Efron’s method
for dealing with ties.

While it is well-known that AIC tends to favor subsets that contain more variables than
is necessary (e.g., Kou and Efron 2002), a very important message from Zhu and Chipman
(2005) is that PGA has the built-in capability to filter out spurious variables even though AIC
is used as the fitness criterion to drive the entire procedure. Early termination of individual
paths and majority vote across many parallel paths are the two critical ingredients of PGA
that give it such a nice property; see Section 2.5 below. We will see an example of this effect
later in Section 3 (Table 4).

### 2.5 Brief Explanation

The reason why spurious variables are often included for most model evaluation criteria
(and especially the AIC) is because the objective functions used to evaluate models are often
highly asymmetric, in the following sense: As far as model performance is concerned, it
is often a lot worse to exclude an important variable than to include a spurious variable.
Therefore, in the evolutionary process, a model that contains all the important variables
will generally have a high fitness score. As a result, all the other spurious variables that
are contained in that model (by random chance) will also be regarded as important by the
evolutionary force. Since the spurious variables are there by chance, when the GA has not
fully converged, these variables will most likely be different across the parallel universes.
Therefore, by running a number of short paths (paths that have not fully converged), these
spurious variables can be effectively eliminated. More specifically, if a variable \( j \) is truly
important, \( r(j, b) \) will be high for all or most \( b \), whereas if a variable \( j \) is spurious, \( r(j, b) \) will
be high (by chance) only for some \( b \). Therefore when averaged over \( b = 1, 2, ..., B \), \( \bar{r}_j \) will
only be high for variables that are truly important. This is why parallel evolution is such an
effective tool for making variable selection choices.
3 Simulation Studies

In this section, we conduct simulation studies to evaluate the performance of PGA systematically. Two efforts are made to create variable selection problems that are reasonably difficult. First, the candidate variables are generated not independently but to correlate with each other. Second, the effects of the variables in the true model are chosen to be rather small, which gives rise to a fairly low signal-to-noise ratio.

Altogether, three simulation studies are conducted, each consisting of 100 runs. In each run, we generate $n = 80$ observations with $p = 20$ predictor variables. The predictors are generated according to

$$x_j = z + \epsilon_j; \quad j = 1, 2, ..., 20, \quad \epsilon_j, z \sim \text{iid} N_{80}(0, I).$$

This induces a pairwise correlation of 0.5 among all the variables. The response $y_i$ is generated from an exponential distribution whose hazard function is given by

$$\lambda_i(t) = \lambda_0(t) \exp(0.5x_{i,5} + x_{i,10} + 1.5x_{i,15}) \quad \text{where} \quad \lambda_0(t) = 1.$$

In other words, only three predictors are useful: $x_5$, $x_{10}$, and $x_{15}$. Moreover, variable 15 is the strongest predictor, having the largest effect, whereas variable 5 is the weakest variable. The effects of all three predictors are deliberately chosen to be fairly small in order to make this a fairly hard variable selection problem.

In the first simulation, we generate data without any censoring. In the second and third simulation, the following censoring mechanism is used: For each observation, we generate a censoring time $t_i^*$ independently from a uniform distribution on the interval $(0, \gamma)$. If $y_i > t_i^*$, then observation $i$ is censored, i.e., we replace $y_i$ with $t_i^*$ and let $c_i = 0$; otherwise $c_i = 1$. The parameter $\gamma$ is chosen to obtain an overall censoring rate of approximately 20% (second simulation; $\gamma = 45$) and 40% (third simulation; $\gamma = 4$).

We compare three variable selection algorithms: stepwise AIC, stepwise BIC, and PGA. For the stepwise procedures, we use the function stepAIC in R (R Development Core Team 2004) — by replacing the default penalty size of 2 with $\log(n)$, the function stepAIC can be used to perform stepwise search using the BIC rather than the AIC as the criterion. For PGA, we use $B = 25$ parallel paths, each evolving for $N = 6$ generations. Using the default population size of $m = p = 20$, this amounts to evaluating $m \times N \times B = 20 \times 6 \times 25 = 3,000$ models, which is a tiny fraction of the entire model space of $2^{20}$.

Table 4 reports the success rate (based on 100 trials) that each method is able to find the correct subset of variables: 5, 10, and 15. Notice that, for any binary random variable $z$ with $p = P(z = 1)$, its variance is $p(1 - p) \leq 0.25$. Therefore, the maximal standard deviation for every entry in Table 4 is equal to $\sqrt{0.25/100} = 0.05$. We can see that PGA is significantly better at selecting the right variables.
Table 4: Simulation study. Success rate based on 100 repetitions.

<table>
<thead>
<tr>
<th>Method</th>
<th>None</th>
<th>20%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stepwise AIC</td>
<td>0.02</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>Stepwise BIC</td>
<td>0.39</td>
<td>0.38</td>
<td>0.34</td>
</tr>
<tr>
<td>PGA</td>
<td>0.63</td>
<td>0.67</td>
<td>0.57</td>
</tr>
</tbody>
</table>

4 Example: Primary Biliary Cirrhosis

We now analyze a real data set. A double-blind randomized trial was conducted by Mayo Clinic from 1974 to 1984 to study the drug D-penicillamine (DPCA) for treating primary biliary cirrhosis (PBC) of the liver. This data set has been widely used in the literature for variable selection, including Tibshirani (1997), Faraggi and Simon (1998) and Hoeting et al. (1999). It is available in a number of slightly different versions; the version we use here is the same as Hoeting et al. (1999, Section 7.1). Table 5 lists a total of 15 covariates. The censoring rate for this data set is approximately 60%.

Table 5: PBC Data. List of covariates.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$ (Treat):</td>
<td>treatment (1 = DPCA, 2 = placebo)</td>
</tr>
<tr>
<td>$x_2$ (Age):</td>
<td>age of patient in years</td>
</tr>
<tr>
<td>$x_3$ (Sex):</td>
<td>sex of patient (0 = male, 1 = female)</td>
</tr>
<tr>
<td>$x_4$ (Asc):</td>
<td>presence of ascites (1 = present, 0 = not)</td>
</tr>
<tr>
<td>$x_5$ (Hep):</td>
<td>presence of hepatomegaly (1 = present, 0 = not)</td>
</tr>
<tr>
<td>$x_6$ (Spiders):</td>
<td>presence of spiders (1 = present, 0 = not)</td>
</tr>
<tr>
<td>$x_7$ (Edema):</td>
<td>presence of edema</td>
</tr>
<tr>
<td>$x_8$ (Bil):</td>
<td>serum bilirubin level, log scale</td>
</tr>
<tr>
<td>$x_9$ (Albu):</td>
<td>albumin level, log scale</td>
</tr>
<tr>
<td>$x_{10}$ (UC):</td>
<td>urine copper level, log scale</td>
</tr>
<tr>
<td>$x_{11}$ (Alk):</td>
<td>alkaline phosphates</td>
</tr>
<tr>
<td>$x_{12}$ (SGOT):</td>
<td>SGOT level</td>
</tr>
<tr>
<td>$x_{13}$ (Platelets):</td>
<td>platelet count</td>
</tr>
<tr>
<td>$x_{14}$ (Proth):</td>
<td>prothrombin time, in seconds</td>
</tr>
<tr>
<td>$x_{15}$ (Hist):</td>
<td>histologic state of disease, coded 1-4</td>
</tr>
</tbody>
</table>
Figure 1 shows the result from PGA, using $B = 40$ parallel paths each evolving for $N = 8$ generations. The left panel plots the importance measure $\bar{r}_j$ (2) from the largest to the smallest — we call such a plot the “order plot.” There is a (statistically significant) large gap between the 7th and 8th variable, indicating that seven covariates are more important than others; the right panel of Figure 1 tells us which seven (see also Table 6) — we call the plot on the right the “bubble plot” because important variables float on the top like bubbles.

![Order Plot: B= 40 Parallel Paths](image1)

![Bubble Plot: B = 40 Parallel Paths](image2)

**Figure 1:** *PBC Data Set. Left: The order plot, plotting the importance measure $\bar{r}_j$ from the largest to the smallest. Right: The bubble plot.*

Table 6 lists and compares solutions found by different techniques, including PGA, stepwise AIC and stepwise BIC. Hoeting et al. (1999) analyzed this data set using Bayesian model averaging (BMA); the line “BMA” in Table 6 lists the estimated posterior effect probabilities, $P(\beta_j \neq 0|\text{data})$, taken from Hoeting et al. (1999, Tables 1 and 2). As we can see, stepwise BIC gives the most conservative solution, choosing six variables, whereas stepwise AIC chooses two more variables, “Hist” and “SGOT.” PGA, on the other hand, chooses the variable “Hist” but not the variable “SGOT,” suggesting that “Hist” is somewhat more important than “SGOT.” This is consistent with the BMA result; from Table 6, we can see that $P(\beta_{Hist} \neq 0|\text{data}) = 34\% > 22\% = P(\beta_{SGOT} \neq 0|\text{data})$.

A number of other studies have been conducted using slightly different versions of this data set. Tibshirani (1997) considered two more potential covariates, serum cholesterol level and triglycerides level. It turns out that a fairly large number of observations contain missing values for these two variables. Tibshirani (1997) eliminated observations with missing values. The line “LASSO” in Table 6 lists the absolute $Z$-scores from LASSO, taken from Tibshirani (1997, Table I). Despite the differences in the data sets being analyzed, the LASSO solution is essentially the same as our stepwise AIC solution, with a few extra variables with small but near-zero effects. Faraggi and Simon (1998) used a Bayesian variable selection (BVS)
Table 6: PBC Data. Variables selected, estimated posterior effect probabilities (for BMA) or absolute Z-scores (for LASSO).

<table>
<thead>
<tr>
<th>Method</th>
<th>Age</th>
<th>Edema</th>
<th>Bil</th>
<th>Albu</th>
<th>UC</th>
<th>Proth</th>
<th>Hist</th>
<th>SGOT</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stepwise BIC</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PGA</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Stepwise AIC</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>-</td>
</tr>
<tr>
<td>BMA</td>
<td>100%</td>
<td>84%</td>
<td>100%</td>
<td>100%</td>
<td>72%</td>
<td>78%</td>
<td>34%</td>
<td>22%</td>
<td>small</td>
</tr>
</tbody>
</table>

Results from Other Studies Using Slightly Different Versions of the Same Data Set:

<table>
<thead>
<tr>
<th>Method</th>
<th>Age</th>
<th>Edema</th>
<th>Bil</th>
<th>Albu</th>
<th>UC</th>
<th>Proth</th>
<th>Hist</th>
<th>SGOT</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>1.89</td>
<td>1.71</td>
<td>2.97</td>
<td>2.27</td>
<td>1.98</td>
<td>0.97</td>
<td>2.28</td>
<td>1.04</td>
<td>≪ 1 or 0</td>
</tr>
<tr>
<td>BVS</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
<td>-</td>
</tr>
</tbody>
</table>


method but used a version of the data set that contains only 8 covariates — in fact, their 8 covariates coincide with the ones selected by stepwise AIC. They evaluated “all” $2^8$ models and listed (in their Table 2) the best models for each size: 1, 2, ..., 8. The line “BVS” in Table 6 gives their best model of size 7 (the same size as the PGA solution). It contains the variable “SGOT” but not “UC,” while the other 6 variables are the same as the PGA solution. From Table 6, we can see that stepwise BIC, BMA and LASSO would all agree with PGA and select “UC” over “SGOT.”

**Remark** In practice, it is often helpful to make a sequence of bubble plots as $B$ increases and coalesce the entire sequence of plots into a movie, treating each individual plot as a frame. Usually, one can very quickly spot the important covariates moving up to the top while the rest fluctuate at the bottom. For obvious reasons, we could not display such a movie here; an example is available at the first author’s web site.

## 5 Conclusion

PGA seems to be a worthy competitor for selecting important risk factors from the Cox model when compared with stepwise selection, Bayesian methods, as well as LASSO. Some critical features of PGA include: global optimization (the genetic algorithm), early termination (short evolutionary paths) and majority vote using an ensemble (parallel evolutionary paths). In simulation studies, PGA selects the correct model with a significantly higher probability
than stepwise selection using either AIC or BIC as the criterion. Unlike Bayesian methods, PGA does not require the specification of any prior distributions.

Finally and most importantly, there is a considerable amount of flexibility in PGA for specifying the fitness function. This makes PGA readily applicable to other (more complex) regression models for dealing with censored survival data.

**Acknowledgment**

Both authors are partially supported by the Natural Science and Engineering Research Council (NSERC) of Canada.

**References**


