PAPER

An Accident Severity Classification Model Based on Multi-Objective Particle Swarm Optimization

Chunlu WANG†, Chenye QIU†‡, Xingquan ZUO†, Nonmembers, and Chuanyi LIU†, Member

SUMMARY Reducing accident severity is an effective way to improve road safety. In the literature of accident severity analysis, two main disadvantages exist: most studies use classification accuracy to measure the quality of a classifier which is not appropriate in the condition of unbalanced dataset; the other is the results are not easy to be interpreted by users. Aiming at these drawbacks, a novel multi-objective particle swarm optimization (MOPSO) method is proposed to identify the contributing factors that impact accident severity. By employing Pareto dominance concept, a set of Pareto optimal rules can be obtained by MOPSO automatically, without any pre-defined threshold or variables. Then the rules are used to form a non-ordered classifier. A MOPSO is applied to discover a set of Pareto optimal rules. The accident data of Beijing between 2008 and 2010 are used to build the model. The proposed approach is compared with several rule learning algorithms. The results show the proposed approach can generate a set of accurate and comprehensible rules which can indicate the relationship between risk factors and accident severity.

key words: accident severity, particle swarm optimization, multi-objective optimization, ROC curve

1. Introduction

Traffic accidents have been one of the most leading causes of death and injury worldwide, resulting in estimated 1.2 million deaths and 50 million injuries worldwide each year (World Health Organization, 2009). Risky factors that influence accident severity are of special concern to researchers in traffic safety, since reducing accident severity is an effective way to improve road safety. Understanding risky factors which would lead to increased accident severity is the main objective in accident severity analysis. This study employs a novel multi-objective particle swarm optimization approach to identify the risk factors that can significantly influence accident severity.

The most commonly used accident severity prediction models are the statistical models, particularly regression models. Many researchers have employed regression models to identify risky factors affecting accident severity [1]–[6]. However, most regression models have their own assumptions and pre-defined underlying relationships between dependent and independent variables (i.e. linear relations between the variables) [7]. If these assumptions are violated, the model could lead to erroneous estimations of the likelihood of severe injury.

In order to deal with this problem, some non-parametric methods without any pre-defined underlying relationship between the target (dependent) variable and the predictors (independent variables) are proposed to identify the risk factors affecting injury severity levels in traffic accidents, including neural network, Bayesian network, CART, and PART [8]–[10]. These models show promising results. However, there are still two main disadvantages. First, most prediction methods are difficult to be interpreted by the end users. For example, the neural network model is a black box model. The results of a trained neural network are many weights through the network which are not understandable. Second, most works focus on building accurate classifier, as measured by classification error rate. These classification models focus on building accurate classifier, as measured by classification error rate. One drawback of these models is that they may still achieve high overall accuracy even when a large number of instances in the minority class (i.e. the class with few representative cases in the database) are misclassified. This may cause a serious problem when the minority class is of particular interest. Accident severity analysis is such a problem in which the patterns of fatal accidents are important to traffic management departments.

In order to decrease the abovementioned disadvantages, a novel approach based on multi-objective particle swarm optimization (MOPSO) is employed to solve the accident severity analysis problem. Multi-objective optimization algorithm has been widely used in data mining tasks [11]–[13]. MO algorithm can extract a set of rules from the training dataset. Rules can provide knowledge in a simple and intuitive way. Hence it is very suitable for accident analysis problem. The rules generated by MO algorithm can reveal the relationships between risky factors and accident severity and improve the understanding of accident severity. The rules can be easily understood by end users and help them with their decision making. MO based classification rule mining has shown to be a powerful tool in many types of rule mining tasks. Results of applying the MO meta-heuristic algorithms across a range of datasets from the UCI repository showed that the MO algorithm can generate accurate and comprehensible rules.

However, employing MO algorithm to accident severity problem still faces two main challenges. (1) Previous studies used some benchmark problems, such as the UCI dataset. There are few MO algorithms for solving real world problems, especially in the traffic accident data analysis. To the best of our knowledge, we are the first to apply multi-objective particle swarm optimization algorithm

Manuscript received February 28, 2014.
Manuscript revised June 10, 2014.
†The authors are with Beijing University of Posts and Telecommunications, Beijing, China.
a) E-mail: qiuchenye@gmail.com
DOI: 10.1587/transinf.2014EDP7069

Copyright © 2014 The Institute of Electronics, Information and Communication Engineers
to accident severity analysis.

(2) The other challenge is to build a classifier with a set of rules. Some researchers only focus on finding a set of classification rules from the data. They do not build a classifier with the rule set [10]–[13]. Among the researches about building classifiers with rule sets, many existing methods aim at maximizing the predictive accuracy in the training set [14]. Recent studies show that using accuracy as a metric is not appropriate in the condition of unbalanced dataset [15]. High overall accuracy can be achieved even when a large number of instances in the minority class (i.e., the class with few representative cases in the database) are misclassified.

For dealing with the above issues, a novel approach based on a MO particle swarm optimization with K-means guide selection strategy (KMOPSO) is proposed to identify risk factors affecting accident severity [16]. KMOPSO employs the Pareto dominance concept to extract a set of Pareto optimal rules automatically, without any predefined threshold. A K-means algorithm based guide selection strategy is proposed to select global best in KMOPSO in order to spread the rules along the Pareto front. Then a non-ordered classifier based on the rule set is built and the proposed approach is evaluated with the ROC curve (Receiver Operating Characteristic) [17]. ROC curve has the advantage of insensitive to the distribution of class. So it is very suitable for the accident analysis problem in which the number of fatal accident is much smaller. Recent researches have shown MOPSO with ROC curve analysis can generate accurate and comprehensible classification models in unbalanced dataset [15], [18].

The proposed algorithm has several specific points:

(1) MOPSO is applied to the accident analysis problem for the first time. The novel approach can extract a set of rules from the accident dataset. Knowledge represented in the form of rules can be easily understood by end users.

(2) ROC curve is used to evaluate the performance of classifier which can solve the problem of unbalanced data distribution.

(3) The Pareto approach is employed to choose only the non-dominated rules for each class. Hence this study does not need any predefined threshold like support, confidence, or number of rules.

(4) A novel global best selection strategy based on K-means algorithm is used to find the global best from the non-dominated rules during each run of the algorithm. This global best selection method can ensure the rules spread over the entire non-dominated front.

The remainder of this paper is organized as follows: Sect. 2 introduces accident data used in this study and the method. Section 3 gives a brief concept about multi-objective optimization. Section 4 describes the multi-objective particle swarm optimization used for accident severity analysis in detail. Section 5 reports the experimental results. Section 6 concludes the paper.

2. Accident Data and the Method

2.1 Accident Data

Accident data were obtained from the Beijing Traffic Management Bureau, which included the reported accidents of Beijing between 2008 and 2010. Only accidents happened on the non-intersection roads, i.e., the accidents not related to intersections, were chosen to build the model as the characteristics of non-intersection roads and intersections are quite different, so they cannot be mixed together. The total number of accidents obtained on non-intersection roads during this period was 3,651. 523 accidents with missing values were screened out. These data missed at least one variable used for accident analysis. 23 accidents with questionable information were deleted. For example, records with conflicting variables were screened out. Finally, the database contained 3105 accident instances, with 1996 non-fatal accidents (64.3%) and 1109 fatal accidents (35.7%).

Twelve variables and one class variable were used to identify the risk factors contributing to the accident and injury severity. Accident data include information on accident severity, time of accident, involved driver (e.g., age, gender, if the driver wears seat belt), accident type, accident cause, and involved vehicle (e.g., vehicle type). Roadway data include characteristics of the roadway on which the accidents occurred (e.g., sight distance, divided or undivided). Weather data include the weather conditions when the accident occurred. The description and levels of these variables are given in Table 1.

2.2 Accident Severity Analysis by KMOPSO

Many accident severity analysis models are built to identify factors that are associated with accident severity, using roadway variables, driver characteristics, weather data, and accident data as predictors. Classification rule mining is one of the most used forms to find knowledge in data mining context since the rules can be easily understood by users.

In accident severity analysis, the data were divided into two classes: fatal accidents and non-fatal accidents. In this paper, KMOPSO is used to mine classification rules from the traffic accident dataset. By fixing the class attribute, KMOPSO can generate a set of classification rules for each class automatically, without any predefined threshold. These rules can represent some relationships between risky factors and accident severity. They can indicate which combinations of risky factors increases the severity of accident severity.

Then a non-ordered classifier is built based on these rules with a voted classification method [18], ROC curve is used to evaluate the performance of the proposed algorithm. Compared with accuracy metric, ROC curve is insensitive to the unbalanced distribution of dataset.
Table 1 Description of the variables.

| Variables     | Values          | Codes |
|---------------|-----------------|-------|
| accident time | day             | 1     |
|               | night           | 2     |
| protection    | no              | 1     |
| facilities    | yes             | 2     |
|               | none            | 1     |
| road divider  | median strip    | 2     |
|               | motor/non-motor divided | 3 |
|               | median strip & motor/non-motor divided | 4 |
| sight distance| [50,100]        | 2     |
|               | [100,200]       | 3     |
|               | >200            | 4     |
| seat belt     | not wear        | 1     |
|               | wear            | 2     |
| driver        | non-professional| 2     |
|               | without license | 3     |
| vehicle       | small passenger car | 1 |
|               | large passenger car | 2 |
|               | small truck     | 3     |
|               | large truck     | 4     |
|               | other           | 5     |
|               | sideswipe       | 1     |
|               | pedestrian-vehicle | 2 |
| collision type| head-on        | 3     |
|               | fixed-object    | 4     |
|               | rollover        | 5     |
|               | follow too close| 6     |
|               | fail to brake   | 1     |
| accident cause| drunk driving  | 2     |
|               | not yield as it has to | 4 |
|               | drive in the wrong lane | 5 |
|               | speeding        | 6     |
| weather       | sunny           | 1     |
| gender        | male            | 2     |
|               | female          | 3     |
| age           | <26             | 1     |
|               | [26,55]         | 2     |
|               | >55             | 3     |
| accident severity | non-fatal | 0     |
|               | fatal           | 1     |

Fig. 1 Concept of Pareto optimality.

3. Multi-Objective Particle Swarm Optimization

Many real-world optimization problems involve multiple objectives that should be optimized simultaneously. Sometimes, these objectives are even conflicting. Improve one objective would worsen at least one other objective. Contrary to single objective (SO) optimization problem, there is no single optimal solution in multi-objective (MO) optimization problem, but a set of trade-off solutions, known as Pareto Optimal solutions.

Without loss of generality, we consider a multi-objective maximization problem. It can be stated as:

\[ \text{Max } y = f(x) = (f_1(x), \ldots, f_n(x)) \]

\[ x = (x_1, \ldots, x_m) \in X, \ y = (y_1, \ldots, y_n) \in Y \] (1)

A solution \( x_k \) is said to dominate another vector \( x_l \), denoted as:

\[ \forall i \in 1, 2, \ldots, n: f_i(x_k) \geq f_i(x_l) \]

\[ \exists i \in 1, 2, \ldots, n: f_i(x_k) > f_i(x_l) \] (2)

A solution \( x_k \) is called Pareto optimal if there does not exist another \( x_l \in X \) that dominates it.

The concept of Pareto dominance is illustrated in Fig. 1. In these three solutions, rule 1 is not dominated by any other rule because it has the biggest value on comprehensibility. Rule 2 is also not dominated by any other rule due to its highest accuracy. Rule 3 is not dominated by Rule 1 for its higher accuracy but it’s dominated by Rule 2. Therefore, solution 1 and solution 2 form the Pareto optimal rule set. In MO problems, the goal is to find the set of Pareto optimal solutions.

4. KMOPSO for Classification Rule Learning

Particle swarm optimization (PSO) is a relatively new algorithm, which is inspired by the social interaction of bird flocking [19]. PSO has been proved to be very effective in a wide variety of optimization problems due to its fast convergence and ease of implementation. It has been successfully used in many areas.

The original PSO was proposed to solve single objective problems. Extending original PSO to multi-objective PSO requires a redefinition of the global best (gbest). In MOPSO, there is no single gbest, but a set of non-dominated solutions. The common way is to use an external archive to store all the non-dominated solutions and choose gbest from the archive. A K-means based gbest selection strategy is proposed in this paper.

Employing KMOPSO to solve classification rule mining also needs some adaptation, such as the rule representation, the rule evaluation, and the evolutionary process of particles. These aspects of KMOPSO will be described in detail in this section.

4.1 Particle Swarm Optimization

PSO is a heuristic technique inspired by the social behavior of bird flocking. A standard particle swarm optimization includes a swarm of particles which represents solutions of the problem. Particles fly in a multi-dimensional search space looking for the optimal position according to its own flying experience and the experience of the best particle in the swarm. The best particle is found according to the fitness value, which is the objective function of the problem. In each generation, all the particles’ positions and velocity are
updated.

Let \( x_i = (x_{i1}, x_{i2}, \cdots, x_{iD}) \) be the \( i \)th particle in the swarm. \( D \) is the dimension of the search space. Its current velocity is \( v_i = (v_{i1}, v_{i2}, \cdots, v_{iD}) \). In the basic PSO algorithm, the positions of particles are updated as follows:

\[
\begin{align*}
\dot{v}_{id}^{t+1} &= \omega \times v_{id}^t + c_1 \times r_1^t \times (pbest_{id}^t - x_{id}^t) \\
&\quad + c_2 \times r_2^t \times (gbest_{id}^t - x_{id}^t) \\
x_{id}^{t+1} &= x_{id}^t + v_{id}^{t+1}
\end{align*}
\]

where \( v_{id}^t \) is the \( d \)th dimension of the velocity of particle \( i \) in cycle \( t \); \( x_{id}^t \) the \( d \)th dimension of the position of particle \( i \) in cycle \( t \); \( pbest_{id}^t \) is the \( d \)th dimension of the position of personal best of particle \( i \) in cycle \( t \); \( gbest_{id}^t \) is the \( d \)th dimension of the position of global best in cycle \( t \); \( \omega \) is the inertial weight, which plays an important role in balance global and local search. A large inertial weight promotes global search and a small inertial weight is more appropriate for local search. The value is typically set between 0 and 1. \( c_1 \) is the cognitive weight and \( c_2 \) is the social weight. \( r_1^t \) and \( r_2^t \) are two random numbers.

### 4.2 Rule Representation

In this paper, the Michigan approach is employed. Each particle in the swarm represents a single classification rule. All the \( V_i \) are the rule antecedent which comprises a conjunction of attributes, and \( C \) is the rule consequent, specifying the value of goal attribute, i.e. the class attribute.

Samples with the same goal attribute are associated. In accident severity analysis, goal attribute is accident severity and it can take “fatal accident” or “non-fatal accident”. By fixing on each class attribute, rules can be obtained for each class. These rules indicating an increased chance of accident severity can be used to reduce accident severity.

Each particle in the population represents a classification rule in the form of “if \( A \) then \( C \)”. Only the antecedent is shown in the particle. The consequent part of the rule is pre-defined in each run of the algorithm. Hence it is not represented in the encoding. \( A \) represents the position of a particle. If there are \( m \) decision attributes, the size of a particle is \( m \). The encoding is shown in Fig. 2. \( A_i \) is the \( i \)th attribute. All the \( A_i \) are in the range \([0, 1]\).

In order to form a rule, the bit string should be translated into the original information. The translation is as follows:

\[
V_i = \text{int}(A_i \ast (\text{Count}_i + 1))
\]

where \( V_i \) means the value translated from the particle for the \( i \)th attribute, \( \text{Count}_i \) is the total number of different values of the \( i \)th attribute, \( \text{int}(x) \) represents the biggest integer number smaller than \( x \).

| \( A_1 \) | \( A_2 \) | \( A_3 \) | \( \cdots \) | \( \cdots \) | \( \cdots \) | \( A_m \) |
|---|---|---|---|---|---|---|

Fig. 2 Rule representation of a particle.

All twelve variables are considered to be important, so they are used to build the model. But a rule with twelve attributes is too long for users to understand. Users prefer rules that can easily be interpreted. Hence the rule in this study is variable-length. If \( V_i \) is 0, this means the \( i \)th attribute is absent in the rule. During the iteration of the algorithm, rules with different combinations of attributes will be generated automatically and they will be evaluated with the sensitivity and specificity. The antecedent part consists of at least one attribute.

For example, if a particle’s position is:

\[
<0.12, 0.63, 0.34, 0.74, 0.43, 0.27, 0.86, 0.61, 0.11, 0.09, 0.33, 0.57>
\]

Suppose the pre-defined consequent is 0, i.e., non-fatal accident. According to Table 1 and Eq. (5), the particle can be transformed to the rule:

\[
\begin{align*}
&\text{if (protection facilities = yes) \& (road divider = none) \& (sight distance = [100, 200]) \& (seat belt = not wear) \& (vehicle = other) \& (collision type = fixed-object) \& (gender=male) \& (age = [26, 55]) } \\
&\Rightarrow \text{(accident severity = non-fatal)}
\end{align*}
\]

Three variables (accident time, accident cause, weather) are absent from the rule.

### 4.3 Rule Evaluation

In this study, sensitivity and specificity are chosen to evaluate the rules. The sensitivity and specificity criteria are directly related to the ROC curve. Sensitivity measures how much the classifier predicts correctly the positive examples. Specificity measures how much the classifier predicts correctly the negative examples. They are defined as follows:

\[
\begin{align*}
\text{Sensitivity} &= \frac{TP}{TP + FN} \\
\text{Specificity} &= \frac{TN}{TN + FP}
\end{align*}
\]

where TP (True Positive) is the number of instances covered by the rule that are correctly classified, i.e., its class matches the training target class. FP (False Positive) is the number of instances covered by the rule that are wrongly classified, i.e., its class differs from the training target class. TN (True Negative) is the number of instances not covered by the rule, and its class differs from the training target class. FN (False Negative) is the number of instances not covered by the rule, and its class matches the training target class.

### 4.4 Rule Learning with KMOPSO

In the beginning of the algorithm, the positions of all the particles are randomly initialized. Their speed is set as 0. Their \( pbest \) (personal best) are their present positions. All the non-dominated solutions are stored in an external archive. A global best (\( gbest \)) selection strategy based on \( K \)-means algorithm is used to choose the \( gbest \) from the archive in order to spread the particles along the entire Pareto front. The
archive is updated after each cycle.

The algorithm of KMOPSO is as follows.

- $M$ is the population size
- $x$ is the position of particle
- $v$ is the speed of each particle
- $P$ is the population
- $t$ is the iteration counter
- $K$ is the number of clusters

(1) Initialize the population:
   a. $x_i$ randomly
   b. $v_i = 0$

(2) Evaluate all the particles.
(3) Store the non-dominated vectors in $P$ into the external archive $EA$.
(4) Initialize the personal best of each particle: $pbest_i = x_i$
(5) While $t < \text{maximum number of iterations}$

   DO
   (1) For each particle, $gbest$ is chosen from the archive with the $K$-means guide selection strategy. The method will be discussed later.
   (2) Compute the speed of each particle with Eq. (5).
   (3) Compute the new position of particle with Eq. (6).
   (4) If $x_i$ goes beyond its search boundaries, we take two measures: 1) set the decision variable the value of its corresponding lower of upper boundary; 2) its velocity is multiplied by $-1$ in order to make it searches the opposite direction.
   (5) Evaluate all the particles in population.
   (6) Update the external archive $EA$. Insert all the currently non-dominated solutions into $EA$. Delete any dominated solution from $EA$.
   (7) Update $pbest$ for each particle. If the current solution is dominated by the current $pbest$, the $pbest$ is kept; otherwise, the particle position is updated. If neither of them is dominated by the other, one of them is selected as the new $pbest$ randomly.
   (8) Increment the loop counter: $t = t + 1$.

End while.

4.4.1 Gbest Selection Strategy Based on K-Means Algorithm

In MOPSO, $gbest$ is very important in guiding the entire population moving towards the true Pareto front. Different from single objective optimization problem, there exists a set of non-dominated solutions. In our algorithm, we implement elitism by incorporating an external archive. A fixed-size external archive is used to store all the non-dominated solutions found along the search process in order to prevent the loss of good solutions. In each cycle, we need to choose $gbest$ from the archive for each particle in the population to guide their fly. The archive is updated in each cycle. If the candidate solution is not dominated by any solution in the archive, it will be added to the archive. Any archive members dominated by this solution will be removed from the archive.

The $gbest$ in the PSO has a great impact on convergence and diversity of solutions. Researchers have proposed many methods to choose $gbest$, such as the crowding distance [20], the sigma method [21], etc. These methods all have some drawbacks. For example, the crowding distance method only chooses $gbest$ from the sparse regions which may decrease the convergence speed. The sigma method may lead to premature when the initial particles are badly-distributed [22]. This paper proposed a novel $gbest$ selection method based on $K$-means algorithm and proportional distribution in order to lead to a diverse and uniformly distributed set of solutions.

This paper introduced a $gbest$ selection strategy based on $K$-means algorithm in order to lead to a diverse and uniformly distributed set of solutions. As shown in Fig. 3, the first step of the $gbest$ selection method is to divide the particles in the archive into $K$ clusters according to their corresponding objective function values. It operates as follows:

(1) Randomly choose $K$ solutions, each of which represents a cluster mean or center.
(2) For each of the remaining solutions, a solution is assigned to the cluster to which it is the most similar. The similarity is evaluated by the Euclidean distance between the solution and the cluster center.
(3) Compute the new center of each cluster.
(4) Iterate step 2 and 3 until convergence of objective function:

$$E = \sum_{i=1}^{K} \sum_{f \in C_i} |f - m_i|$$

where $E$ is the sum of the square error for all solutions in the archive; $f$ is the point in space representing a given solution; $m_i$ is the mean of the cluster $C_i$.

(5) In each cluster, find the solution nearest to the centroid, and consider it as the representative solution of the cluster.

After the clustering process, the clusters are separated from each other and the solutions in the same cluster are similar. The representative solutions lie in diverse regions of the non-dominated front. The next step is to choose global best from the $K$ representative particles for each particle in the population. In the $K$ clusters, they contain different
numbers of particles. The more particles a cluster contains, the more crowded it is. If a cluster has a large number of particles, it means this region is very crowded. We should encourage fewer particles move towards these crowded regions. A proportional distribution method is used to encourage particles in the population move to the sparse regions. The probability of a representative particle $i$ being chosen as the global guide is calculated as follows:

$$p_i = \frac{1/\text{num}_i}{\sum_{i=1}^{\text{num}} 1/\text{num}_i}$$

(9)

where $p_i$ is the probability of the $i$th representative particle being chosen as the global best. $\text{num}_i$ is the number of particles the $i$th cluster contain. As is shown in Eq. (5), the representative particle $i$ has more opportunity to be chosen as the global best if the corresponding cluster has less particles.

This $g_{best}$ selection algorithm considers both the global and local information of the non-dominated front. The $g_{best}$ chosen by this method can represent the distribution of all the non-dominated solutions and encourage the solutions in the sparse regions. They can guide the particles in the population move towards different regions in order to get a uniformly distributed Pareto front. The particles in the crowded regions also have the opportunity to be chosen as the global guide. Hence this algorithm would not decrease the speed of convergence.

4.4.2 Time Decreasing Inertia Weight

Inertia weight plays an important role in balancing global and local search. A large inertia weight promotes global search and a small inertia weight is more appropriate for local search. Reference [23] proposed a time-decreasing inertia weight. The inertia weight is shown as follows:

$$w' = w_{\text{max}} - \frac{t}{\text{maxgen}}(w_{\text{max}} - w_{\text{min}})$$

(10)

where $w_{\text{max}}$ and $w_{\text{min}}$ are the maximum and minimum values of the inertia weight; $t$ is the current iteration and $\text{maxgen}$ is the maximum iteration of the algorithm. In this paper, we adopted this time decreasing inertia weight. Therefore, the algorithm explores the search space initially and later focuses on the most promising regions.

4.5 Building a Classifier with Rule Sets

Our task is to build a classifier with the rules mined by KMOPSO. Many previous researchers build their classification models aiming at maximizing prediction accuracy [14]. The accuracy metric measures how many instances are correctly classified. But this metric is not appropriate in the circumstances of uneven data distribution. In accident severity analysis, there are much more non-fatal accidents in the original dataset.

In order to deal with this problem, the ROC curve is used to measure the performance of the classifier. ROC curve has been widely used in medical diagnosis since the 1970’s and it has gained more and more attention in machine learning area in recent years. ROC curve relates the false positive rate (FP) (axis-x) and the true positive rate (TP) (axis-y) of a classifier. It can give a clear visualization of the performance of a classifier. A ROC curve can be obtained by a single rule or a set of classifiers. In order to compare different algorithms, AUC (Area under the ROC Curve) value was proposed. When using ROC analysis, the goal is to maximize the AUC value. AUC value can describe the trade-off between true classification rate and false classification rate. ROC curve has the advantage of being insensitive to the distribution of dataset. Hence, it is suitable for the accident analysis problem.

The AUC value for non-ordered rule set was calculated with a voted classification method [18]. For each instance in the testing dataset, a score is assigned. The initial score is zero. For each instance, if one rule of the positive class votes, one point is added to its score. If one rule of the negative class covers the instance, one point is subtracted from the score. After this process, each instance has a corresponding score. We can obtain a numeric rank for all the instances in the test dataset. This rank can be used to build a binary classifier with a threshold. If the instance’s score goes beyond the threshold, the result is yes, otherwise no. Each threshold can produce a classifier, i.e., a single point in the ROC curve. Hence, by varying the threshold from $-\infty$ to $+\infty$, we can get a curve. AUC value can be obtained by calculating the area under the curve.

5. Experiments

5.1 Parameter Settings

KMOPSO uses 100 particles in the population and runs 500 generations. The maximum number of the archive is 100. The learning rates $c_1$ and $c_2$ are 1.0, 1.0 respectively. $w$ is time decreasing in order to balance global and local search. $w_{\text{max}}$ is 0.9 and $w_{\text{min}}$ is set as 0.4. The number of cycles in the $K$-means algorithm is 20. The number of clutter ($K$) is time varying. It is shown as follows:

$$K = \begin{cases} |\text{EA}| & \text{when } 0 < \text{EA} \leq 3 \\ 3 & \text{when } 3 < \text{EA} \leq 10 \\ 5 & \text{when } 10 < \text{EA} \leq 30 \\ 10 & \text{when } 30 < \text{EA} \leq 100 \end{cases}$$

(11)

where $\text{EA}$ represents the archive. According to our experiment, when the number of particles in the archive is more than 30, the number of $K$ should be between 10 and 15 in order to achieve good performance.

The dataset is divided into two parts: training set (3/4 of the dataset) and test set (1/4 of the dataset). Algorithm runs on the training set to get a set of rules and the rules are evaluated on the test set. The algorithm produces rules of the $i$th class in the $i$th run of the algorithm [12]. Hence, we need to run the algorithm separately to get rules for both
classes and find how different combinations of risk factors affect the accident severity.

5.2 Comparing with other MOPSO

KMOPSO is compared with another MOPSO based rule learning algorithm (SMOPSO) to testify the quality of rules [15]. The main difference between KMOSPO and the algorithm employed by Carvalho is the way of choosing \( g_{best} \). In their approach, a sigma method is used to select \( g_{best} \) for each particle in the population to encourage convergence and diversity [21].

Figure 4 gives the non-dominated rules mined by KMOPSO for fatal accident and non-fatal accident, related to sensitivity and specificity. Figure 5 gives the non-dominated rules mined by SMOPSO for fatal accident and non-fatal accident, related to sensitivity and specificity. Each point represents a non-dominated rule. The more northeast a point is, the better is the quality of the rule. All the rules form a non-dominated front.

In this run, KMOPSO generated 120 rules from the training dataset, 61 rules for the non-fatal accident class and 59 rules for the fatal accident class. SMOPSO mined 65 rules, 34 rules for the non-fatal accident class and 31 rules for the fatal accident class.

From the above figures, it can be noticed that both algorithms tend to generate rules with high specificity and low sensitivity. Most of the rules locate in the northwest part of the figure. This is caused by the unbalanced data distribution. The negative instance is the majority class. Even only a small part of negative instances are classified as positive class, the sensitivity would be very low. However, when some positive instances are wrongly classified, the specificity would still be quite high. However, KMOPSO can still generate some rules with high sensitivity and low specificity, although the number is relatively small. These rules can predict most of the positive instance correctly, i.e., the minority class. SMOPSO cannot cover the entire non-dominated front.

The main difference between KMOPSO and SMOPSO is the guide selection strategy. The sigma method can guide the particles move to the Pareto front directly and show good convergence ability. However, this behavior may lead to premature and bad diversity when the initial particles of the archive are bad-distributed. The \( g_{best} \) selection strategy based on \( K \)-means algorithm shows better diversity than the sigma method [16].

MOPSO is a heuristic with some random factors that would affect the final result, such as the initial positions of particles and some random variables in Eq. (3). Hence, each algorithm was run 20 times to restrict the influence of random effects. Table 2 to 4 shows the numerical results produced by two MOPSO algorithms considering three performance metrics.

In order to provide a quantitative assessment for the quality of the rules generated by both algorithms, three measures are used: the coverage metric, the spacing metric and the maximum spread metric [20]. There three metrics can measure the convergence, diversity and spread of the generated rules, respectively.

From Table 2, we can see KMOPSO shows better performance in terms of convergence. \( C(A, B) \) equals 1 in some runs in the non-fatal class while \( C(A, B) \) equals in all the 20 runs in the fatal class. This means the rules mined by KMOPSO shows better sensitivity and specificity. It can be shown from Table 3 that KMOPSO has a better distribution of the non-dominated rules in the non-fatal class while SMOPSO performs better on the fatal class. By looking at

![Fig. 4](image-url) Rules mined by KMOPSO.

![Fig. 5](image-url) Rules mined by SMOPSO.

| Table 2 | Results for the coverage (C) metric for non-fatal and fatal classes, where A is KMOPSO and B denotes SMOPSO. |
|---------|----------------------------------------------------------------------------------------------------------|
| \( C(A, B) \) | \( C(B, A) \) |
| --- | --- | --- |
| Non-fatal | Max 1 | 0.3548 |
| Avg 0.9709 | 0.12134 |
| Min 0.8158 | 0.2934 |
| F a t a l | Max 1 | 0.3729 |
| Avg 0.9574 | 0.11232 |
| Min 0.871 | 0.0364 |

| Table 3 | Results for the spacing (S) metric for non-fatal and fatal classes. |
|---------|------------------------------------------------------------------|
| KMOPSO | SMOPSO |
| --- | --- | --- |
| Non-fatal | Max 0.0358 | 0.1064 |
| Avg 0.03548 | 0.05314 |
| Min 0.0172 | 0.0143 |
| F a t a l | Max 0.0343 | 0.0577 |
| Avg 0.03275 | 0.0284 |
| Min 0.0311 | 0.0098 |

| Table 4 | Results for the maximum spread (MS) metric for non-fatal and fatal classes. |
|---------|-------------------------------------|
| KMOPSO | SMOPSO |
| --- | --- | --- |
| Non-fatal | Max 1 | 0.5514 |
| Avg 0.9434 | 0.5154 |
| Min 0.4359 | 0.1296 |
| F a t a l | Max 1 | 1 |
| Avg 1 | 0.32202 |
| Min 1 | 0.0797 |
In this section, KMOPSO was compared with some classical rule learning algorithms. Three rule learning algorithms include: PART (a rule induction algorithm based on a partial decision tree) [24], RIPPER (repeated incremental pruning to produce error reduction, a propositional rule learner) [25], and C4.5 (a decision tree algorithm) [26].

Table 6 presents the results obtained by the four algorithms. KMOPSO is a random search algorithm. Hence the result of KMOPSO is a mean value of 20 runs of the algorithm. It can be seen that KMOPSO achieves the best AUC value. The other three algorithms fall behind KMOPSO, and there is little difference within these three algorithms.

6. Conclusions

This paper proposed a novel KMOPSO approach for the accident severity analysis problem. The approach contains two steps: a set of non-dominated rules are mined by the KMOPSO, and then the rules are used to build a classifier. The rules can indicate some relationships between risk factors and the accident severity. And the rules are easy to be interpreted by users, unlike some black box models. The traffic accident data of Beijing were collected to build the model. KMOPSO was compared with four well-known rule learning algorithms. The experimental results show the following advantages of the proposed algorithm:

(1) A MOPSO is introduced into the accident severity analysis problem. The adoption of the Pareto concept allows a set of simple rules in a single run of the algorithm. The IF-THEN rules can be easily understood.

(2) The novel $gbest$ selection method can improve the searching ability of MOPSO. It can spread the particles along the entire Pareto front. When compared with SMOPSO, KMOPSO shows better convergence, diversity and spread of solutions.

(3) AUC value is used to evaluate the proposed approach. AUC can deal with the unbalanced dataset. The experimental results show KMOPSO gets the best AUC value among all the algorithms.

References

[1] A.S. Al-Ghamdi, “Using logistic regression to estimate the influence of accident factors on accident severity,” Accident Analysis and Prevention, vol.34, no.6, pp.729–741, Nov. 2002.

[2] J.C. Milton, V.N. Shankar, and F.L. Mannering, “Highway accident severities and the mixed logit model: An exploratory empirical analysis,” Accident Analysis and Prevention, vol.40, no.1, pp.260–266, Jan. 2008.

[3] M. Bédard, G.H. Guyatt, M.J. Stones, and J.P. Hirdes, “The independent contribution of driver, crash, and vehicle characteristics to driver fatalities,” Accident Analysis and Prevention, vol.34, no.6, pp.717–727, Nov. 2002.

[4] K.K.W. Yau, H.P. Lo, and S.H.H. Fung, “Multiple-vehicle traffic accidents in Hong Kong,” Accident Analysis and Prevention, vol.38, no.6, pp.1157–1161, Nov. 2006.

[5] T. Yamamoto and V.N. Shankar, “Bivariate ordered-response probit model of driver’s and passenger’s injury severities in collisions with fixed objects,” Accident Analysis and Prevention, vol.36, no.5, pp.869–876, Sept. 2004.

[6] K.M. Kockelman and Y.J. Kweon, “Driver injury severity: An application of ordered probit models,” Accident Analysis and Prevention, vol.34, no.3, pp.313–321, May 2002.

[7] L.Y. Chang and H.W. Wang, “Analysis of traffic injury severity: an application of non-parametric classification tree techniques,” Accident Analysis and Prevention, vol.38, no.5, pp.1019–1027, Sept. 2006.

[8] D. Dursun, S. Ramesh, and B. Max, “Identifying significant predictors of injury severity in traffic accidents using a series of artificial neural networks,” Accident Analysis and Prevention, vol.38, no.3, pp.434–444, May 2006.

[9] J.D. Oha, R.O. Mujalli, and F.J. Calvo, “Analysis of traffic accident injury severity on Spanish rural highways using Bayesian networks.”
WANG et al.: AN ACCIDENT SEVERITY CLASSIFICATION MODEL BASED ON MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

[10] T. Beshah and S. Hill, “Mining road traffic accident data to improve safety: Role of road-related factors on accident severity in Ethiopia,” Proc. AAAI Artificial Intelligence for Development, 2010.

[11] B. Alatas and E. Akin, “Multi-objective rule mining using a chaotic particle swarm optimization algorithm,” Knowledge-Based Systems, vol.22, no.6, pp.455–460, Aug. 2009.

[12] S. Dehuri and R. Mall, “Predictive and comprehensible rule discovery using a multi-objective genetic algorithm,” Knowledge-Based Systems, vol.19, no.6, pp.413–421, Oct. 2006.

[13] B. de la Iglesia, G. Richards, M.S. Philpott, and V.J. Rayward-Smith, “The application and effectiveness of a multi-objective metaheuristic algorithm for partial classification,” European Journal of Operational Research, vol.169, no.3, pp.898–917, March 2006.

[14] B. Liu, W. Hsu, and Y. Ma, “Integrating classification and association rule mining,” Proc. 1998 International Conference on Knowledge Discovery and Data Mining, pp.80–86, New York, 1998.

[15] A.B. De Carvalho, A. Pozo, and S.R. Vergilio, “A symbolic fault-prediction model based on multi-objective particle swarm optimization,” J. Systems and Software, vol.83, no.5, pp.868–882, May 2010.

[16] C.Y. Qiu, C.L. Wang, and X.Q. Zuo, “A novel multi-objective particle swarm optimization with K-means based global best selection strategy,” Int. J. Computational Intelligence Systems, vol.6, no.5, pp.822–835, Sept. 2013.

[17] T. Fawcett, “Using rule sets to maximize ROC performance,” Proc. IEEE International Conference on Data Mining, pp.131–138, San Jose, California, USA, 2001.

[18] A.P.G. De Toracio and A.T.R. Pozo, “Multiple objective particle swarm for classification-rule discovery,” Proc. IEEE Congress on Evolutionary Computation 2007, pp.684–691, Singapore, Sept. 2007.

[19] J. Kennedy and R.C. Eberhart, “Particle swarm optimization,” Proc. IEEE International Conference on Neural Networks, pp.1942–1948, Perth, Australia, 1995.

[20] C.R. Raquel and P.C. Naval, “An effective use of crowding distance in multiobjective particle swarm optimization,” Proc. Genetic and Evolutionary Computation Conference (GECCO’2005), pp.257–264, Washington DC, USA, 2005.

[21] S. Mostaghim and J. Teich, “Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO),” Proc. 2003 IEEE Swarm Intelligence Symposium, pp.26–33, Indianapolis, 2003.

[22] K. Ali, S. Manganaris, and R. Srikant, “Partial classification using association rules,” Proc. Third International Conference on Knowledge Discovery and Data Mining, pp.115–118, Menlo Park, California, Aug. 1997.

[23] R.C. Eberhart and Y.H. Shi, “Comparing inertia weights and constriction factors in particle swarm optimization,” Proc. IEEE Congress on Evolutionary Computation, pp.84–88, La Jolla, USA, 2000.

[24] E. Frank and I.H. Witten, “Generating accurate rule sets without global optimization,” Proc. 15th International Conference Machine Learning (ICML’98), pp.144–151, San Francisco, 1998.

[25] W.W. Cohen, “Fast effective rule induction,” Proc. Twelfth International Conference on Machine Learning, pp.115–123, Tahoe City, California, USA, 1995.

[26] A.A. Freitas, “A genetic algorithm for generalized rule induction,” in Advances in Soft Computing-Engineering Design and Manufacturing, eds. R. Roy et al., pp.340–353, New York, 1999.

Chunlu Wang is an associate professor in School of Computer, Beijing University of Posts and Telecommunications. She received her M.S. degree in computer science and technology from Harbin Institute of Technology, China, in 1994. Her research interests include Intelligent Transportation, Cloud Computing, File and Storage Systems. She has published over 60 research papers in journals and conferences.

Chengye Qiu received his B.S. degree (2007) in information engineering from Beijing University of Posts and Telecommunications. He is now a PhD student in Beijing University of Posts and Telecommunications.

Chuanyi Liu is an Associate Professor in School of Computer, Beijing University of Posts and Telecommunications. He received the Ph.D. degree in control theory and control engineering from Harbin Institute of Technology, Harbin, China, in 2004. From 2004 to 2006, he was a Postdoctoral Research Fellow in Automation Department of Tsinghua University. From 2012 to 2013, he was a Visiting Scholar in Industrial and System Engineering Department, Auburn University, USA. He has published over 50 research papers in journals and conferences, two books and several book chapters.

Chunliu Wang is now an associate professor of computer science & engineering at Beijing University of Posts and Telecommunications. He received his PhD (2009) in computer science and technology from Tsinghua University. He has broad research interests in computer systems, including Computer Architecture, Cloud Computing & Cloud Security, File and Storage Systems, Information Security and Data Protection. He spent one year as a visiting scholar at the Digital Technology Center of the University of Minnesota.