# Reconstructing Cross-Cut Shredded Text Documents: A Genetic Algorithm with Splicing-Driven Reproduction

Yong-Feng Ge[a,b], Yue-Jiao Gong[a,b,*], Wei-Jie Yu[b,c], Xiao-Min Hu[b,d] and Jun Zhang[a,b]
[a] School of Advanced Computing, Sun Yat-sen University
[b] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education
[c] School of Information Management, Sun Yat-sen University
[d] School of Public Health, Sun Yat-sen University, P.R. China
*(Corresponding author) gongyuejiao@gmail.com

## ABSTRACT
In this work we focus on reconstruction of cross-cut shredded text documents (RCCSTD), which is of high interest in the fields of forensics and archeology. A novel genetic algorithm, with splicing-driven crossover, four mutation operators, and a row-oriented elitism strategy, is proposed to improve the capability of solving RCCSTD in complex space. We also design a novel and comprehensive objective function based on both edge and empty vector-based splicing error to guarantee that the correct reconstruction always has the lowest cost value. Experiments are conducted on six RCCSTD scenarios, with experimental results showing that the proposed algorithm significantly outperforms the previous best-known algorithms for this problem.

## Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.7.m [**Document and Text Processing**]: Miscellaneous.

## General Terms
Algorithms, Performance, Experimentation.

## Keywords
Document reconstruction; genetic algorithm; row-oriented elitism strategy.

## 1. INTRODUCTION
Although electronic documents are now commonly used in most fields, the printed documents are still necessary due to the long-time storage requirements and some legal reasons. In this case, document shredding is a frequently used method to obfuscate sensitive documents, which happens either by purpose or by accident. The former is related to forensic science whereas the latter attracts the interest of archeology. Specifically, paper can be destroyed by hands or, more professionally, by shredders. Within this work, we focus on the reconstruction of cross-cut shredded text documents (RCCSTD) problem [1], in which documents are cut into rectangular shreds of equal size and shape by shredders.


*GECCO'15, July 11-16, 2015, Madrid, Spain.*
 ISBN 978-1-4503-3472-3/15/07…$15.00
http://dx.doi.org/10.1145/2739480.2754677

In previous literature, some approaches have been developed to solve the RCCSTD problem. Clustering method is used by Sleit *et al*. [2], as part of the reconstruction process instead of a preprocessing step. Variable neighborhood search [3] is embedded in evolutionary algorithms in [4]-[5] and it works as a global improvement procedure.

Two problems related to RCCSTD problem in the field of document reconstruction are restoration of hand torn paper documents and reconstruction of strip shredded text documents (RSSTD). On one hand, different from the shredder cut problem, each piece of the hand torn problem has a unique shape. In this case, geometric information could be utilized efficiently during the reconstruction process. Justino *et al.* [6] applied a polygonal approximation to reduce the complexity of construction. On the other hand, Prandtstetter and Raidl [7] proposed some effective approaches to solve the RSSTD problem, which contain a variable neighborhood search approach with a semi-automatic system in the optimization process. Moreover, they proved that the reconstruction of strip shredded documents is NP-complete. Since RSSTD problem could be regarded as a special case of RCCSTD problem, we conclude that RCCSTD problem is also NP-complete. Possibility of using MPEG-7 descriptors for the strip content description is showed in [8].

Genetic algorithm [9] has gained successes in various fields [10]-[11]. In the area of reconstruction, an enhanced genetic algorithm with solution archive is applied in [12]. Solution archive helps avoid the duplicate solutions. Accordingly, diversity of population is maintained. Moreover, the genetic algorithm is extended to a memetic algorithm by embedding the variable neighborhood search in [4].

Different from the previous genetic algorithms for document reconstruction, the process of our proposed algorithm has a tighter connection with reconstruction. In this algorithm, a novel splicing-driven crossover is defined to emphasize the adjacency information in the sequences. Most correctly positioned links are extracted and appear in the offspring. Several mutation operators are designed according to common misleading situations in the reconstruction. In addition to introducing diverse genetic materials, these mutation operators are also efficient in removing incorrect links. Additionally, a row-oriented elitism strategy is proposed based on the fact that the performance of traditional genetic algorithm often deteriorates rapidly as the dimensionality of the problem increases [13]. Similar to the effective decomposition strategy applied in large scale optimization problem, this elitism strategy focuses on the rows which are more likely to be correctly positioned.

The remainder of this paper is organized as follows. Section 2 is a definition of RCCSTD problem. In Section 3, the cost function is defined in detail. A genetic algorithm with splicing-driven reproduction is proposed in Section 4. Experimental results are showed in Section 5, followed by conclusions drawn in Section 6.

## 2. PROBLEM DEFINITION

Based on [1], the formal definition of RCCSTD problem is expressed as follows. A set of rectangular shreds $S = \{1,\ldots,n\}$ with identified numbers containing text of original document is given firstly. These shreds must appear exactly once in the solution. To simplify this process, the correct orientation of each shred is assumed to be known, whose related pattern recognition technique is proposed in [14]. Standard solution of RCCSTD problem could be expressed as a matrix, whose elements represent the shreds in set *S*. The aim of RCCSTD problem is to find an assignment of shreds with the lowest cost. The cost functions in two directions are defined in the next section. The horizontal function $H_{i,j}$ shows the cost made by positioning shred *i* left to shred *j*. Analogously, the vertical function $B_{i,j}$ returns the cost value when shred *i* is on the top of *j*.

## 3. COST FUNCTION DEFINITION

One crucial issue in solving RCCSTD problem is the definition of an appropriate cost function, which will be utilized to evaluate the individuals of GA afterwards. It is remained to be a challenging task to define a perfect cost function for RCCSTD. If the function is not sophisticatedly defined, in some scenarios, there always exists a situation that an incorrect solution has a lower cost value than a correct one does. Such situations would result in selection error and hence bad performance of GA. To avoid such situations, a new cost function is defined in this section.

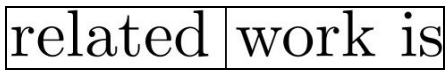
Since the majority of text documents are black text on the white background, we take the gray-scale (which ranges from 0 to 255) information as input. The edges of shreds are represented as vectors. Suppose $\mathbf{A_i}$ represents the edge of shred *i* and $\mathbf{B_j}$ represents the edge of shred *j*. Then we have

$$\begin{aligned}\mathbf{A_i} &= \left[a_1^i, a_2^i, a_3^i \ldots a_h^i\right], a_n^i \in [0,255] \\ \mathbf{B_j} &= \left[b_1^j, b_2^j, b_3^j \ldots b_h^j\right], b_n^j \in [0,255]\end{aligned} \tag{1}$$

where *h* is the length of the edge.

### 3.1 Edge Splicing Cost

The edge splicing cost function makes use of the pixel information on each edge of two shreds. The function can either contain the "non-blank edge cost" when neither of these two edges is blank, or the "blank edge cost" otherwise. The two situations are illustrated in Figure 1 and the corresponding functions are defined as follows.




(a) Two shreds are linked with two blank edges.




(b) Two shreds are linked without blank edge.

**Figure 1. Two situations to consider.**

"Non-blank edge cost function" adopts the idea in [7]. When considering the difference of current two pixels on each edge, information of two pixels above and two pixels below to each current pixel is included. It is proved that the application of continuous pixels information helps reflect the consistency of text indeed. The cost $d_{i,j}$ when shred *j* is positioned next to shred *i* is defined as:

$$d_{i,j} = \sum_{k=3}^{h-2} p_{i,j}^k \tag{2}$$

$$p_{i,j}^k = \begin{cases} 1, if\ p'^k_{i,j} \geq \tau \\ 0, otherwise \end{cases} \tag{3}$$

$$p'^k_{i,j} = \left| \begin{aligned} &0.7(a_k^i - b_k^j) + 0.1(a_{k+1}^i - b_{k+1}^j) + 0.1(a_{k-1}^i - b_{k-1}^j) + \\ &0.05(a_{k+2}^i - b_{k+2}^j) + 0.05(a_{k-2}^i - b_{k-2}^j) \end{aligned} \right| \tag{4}$$

where $p'^k_{i,j}$, $p^k_{i,j}$ refer to the actual and boolean differences of shred *i* and shred *j* at the point *k* respectively. The threshold value $\tau$ should be carefully defined, which has a great influence on computing results.

On the other hand, the "blank edge cost function" utilizes the length of blank space generated by connecting two shreds. In general, the blank spaces in a routine text document have a standard range, which could help detect the possibility of two shreds with blank edges positioned correctly. Either a larger one or a smaller one indicates a wrong connection. The cost $b_{i,j}$ when shred *j* is positioned next to shred *i* is defined as:

$$b_{i,j} = \begin{cases} 0, if\ (c^i + d^j) \in standard\_blank\_range \\ length, otherwise \end{cases} \tag{5}$$

where $c^i$ is the length of shred *i*'s margin. Accordingly, $d^j$ represents the length of margin on shred *j*. To unify the costs here with results of "non-blank edge cost function", the minimum and maximum values are set to 0 and length of edge.

### 3.2 Empty Vector-Based Splicing Cost

Although the edge splicing cost function returns logical results in most cases, we need another cost function named "empty vector-based splicing cost function", especially when considering whether two shreds are from different rows in the document (see Figure 2). "Empty vector" is a vector has the same height as shreds, whose value in *k*th row represents whether the *k*th row in the shred is blank. In most cases, shreds on different lines in the correct solution have clear differences in empty vectors. The cost between shreds *i* and *j* is defined as:

$$f_{i,j} = \sum_{k=1}^{h} (e_k^i \oplus e_k^j) \tag{6}$$

$$e_k^i = \begin{cases} 1, if\ row_k\ is\ blank \\ 0, otherwise \end{cases} \tag{7}$$

where $e_k^i$ represents the truth value of whether the *k*th row of shred *i* is blank.

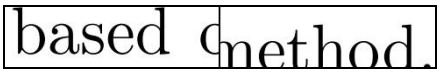



**Figure 2. Two shreds from different rows are linked by mistake.**

## 3.3 Overall Cost Function

In the horizontal direction, to utilize the results of two functions efficiently, they are combined as follows. "non-blank edge cost function" and "blank edge cost function" are used based on the defined situations. The horizontal cost when positioning shred *i* left to shred *j* is defined as:

$$H_{i,j} = \begin{cases} d_{i,j} + f_{i,j}, if\ c^i + d^j = 0 \\ b_{i,j} + f_{i,j}, otherwise \end{cases} \quad (8)$$

where $c^i$ and $d^j$ represent the same meanings in Eqs. (5). Since the text on the columns is not aligned on the same lines, "empty vector-based splicing cost function" is not appropriate on the vertical direction. In this occasion, vertical cost is simpler and defined as:

$$B_{i,j} = \begin{cases} d_{i,j}, if\ c^i + d^j = 0 \\ b_{i,j}, otherwise \end{cases} \quad (9)$$

An example of overall cost is defined as:

$$Cost = H_{a,b} + H_{b,c} + B_{a,d} + H_{d,e} + B_{b,e} + H_{e,f} + B_{c,f} \quad (10)$$

where shreds *a-f* form a 2×3 solution for reconstruction in turn.

# 4. The PROPOSED ALGORITHM

## 4.1 Initial Population

A solution with *n* shreds is represented by an $x \times y$ matrix, whose elements stand for the shreds in RCCSTD problem and labeled with letters. This form is used in the following process. Besides the traditional random building method (RBM), greedy building heuristic (GBH) is proposed to create initial solutions. A good initial heuristic could help supply more effective genetic materials. Accordingly, it helps increase the efficiency of the whole algorithm. In this algorithm, 40% of the individuals in initial population are created by GBH while the other 60% are generated by RBM.

### *4.1.1 RBM*

Lesson from the original genetic algorithm, random initial method could help increase the diversity of initial individuals. In this case, the shreds in initial solutions built by this method are completely randomly chosen.

### *4.1.2 GBH*

In general, text documents are laid out such that there is a blank margin on the left of every page. In this way, it is very likely that those shreds with blank left margins are on the left edge in the final solution. The GBH firstly chooses a shred with the left blank margin. Subsequently, the initial solution is built row by row selecting shreds by greedy strategy. This greedy strategy is with respect to the evaluation of cost function, which is the sum of cost values generated by already processed left and top shreds, if exist. Additionally, the shreds on first row of solution just focus on their left neighbors, while shreds on leftmost edge only concern their top neighbors.

## 4.2 Recombination

Although it is very unlikely that the optimal solution is one of the initial individuals, it is common that some of those initial solutions have perfectly arranged parts. Those members turn out to be more competitive in the evolving population and be more likely to be chosen as parents in the following crossover operators, respecting the nature selection. As an ideal result of recombining individuals with the perfect parts, an offspring with more shreds on correct positions should be generated.

Different genetic sequencing operators emphasize different factors during the recombination. Some focus on the order, such as order crossover [15] and position-based crossover, while others highlight the position information just like partially mapped crossover (PMX) [16] and cycle crossover [17].

Respect to the fact that a solution with more correct links is likely to provide more available information from the text document, we here adopt the idea of enhanced edge recombination (EER) proposed in [19], which emphasizes the adjacency information in the sequence. However, although EER is an efficient recombination operator in solving TSP problem, it is useless to be directly applied in RCCSTD problem due to the differences in dimension and direction. A new crossover operator based on EER is proposed here named "EER-2D".

### *4.2.1 EER*

In this operator, "edge table" is a list structure which storages the adjacency information extracted from parents. Take a sequence [*b d e a c*] as an example, it contains the connections [*bd*, *de*, *ea*, *ac*, *cb*]. Based on this structure, the generation of offspring proceeds as following steps: (1) Select the initial member from one of the parents. (2) Choose a subsequent member according to the edge table and already processed members. (3) Repeat step 2 until a valid solution is complete. The word "enhanced" in EER operator refers to the strategy that the links appear in both parents are labeled with minus and have higher priority to get emersion in the offspring.

Figure 3 shows an example of parents and their corresponding edge table with offspring. Suppose element *a* is selected randomly to start the offspring. Element *a* has links to *b*, *c* and *e*. Since link "*a-b*" appears in both parents and is labeled with minus in the edge table, it is chosen to get emersion in offspring. Element *d* is then chosen because it has lower quantity of remaining links. This process continues until offspring is complete.

**Parent 1**: a b c d e **Parent 2**: d b a c e

**Offspring**: a b d e c

**Edge table**

**a**: -b, c, e **b**: -a, c, d **c**: a, b, d, e

**d**: b, c, -e **e**: a, c, -d

**Figure 3. Elements of EER operator.**

### *4.2.2 EER-2D*

According to the differences between RCCSTD problem and traditional TSP problem, some ideas are proposed in this new recombination method named EER-2D.

On one hand, recombination of RCCSTD problem is associated with both horizontal direction and vertical direction. In this way, there should be two edge tables in EER-2D operator, namely, "horizontal edge table" and "vertical edge table". Links on these two directions are added to respective edge tables. As a result of using two edge tables, two results would be generated during the process of recombination. If two results are not the same, a selection priority is important and defined as: the result from horizontal table has higher priority on the first row while vertical result has higher priority on the first column; they two have the

same priority in other positions with being chosen in random fashion.

On the other hand, the graph in TSP problem is directed while it is undirected in RCCSTD problem. Specifically, link "*a*-*b*" and link "*b*-*a*" both means "the road between these two cities" in TSP problem. However, these two links indicate opposite meanings in RCCSTD problem: the former link indicates that shred *a* is left to shred *b*; the latter link means shred *a* is right to shred *b*. In this case, successive information in the two edge tables of EER-2D is not sufficient as the adjacency information in original EER's single edge table. For instance, link "*a*-*b*" is in the list of shred *a*, but it is not in the list of shred *b*.

To ensure there is enough successive information to generate the offspring, the third genetic individual besides traditional two parents is necessary. In the genetic process in natural world, environment plays the role of the third factor impacting on the generation of offspring. Based on this theory, the third individual is named "environment". The rule in EER operator is still adopted, successive links with two or more appearances are labeled with minus and have higher priority to be chosen.

To sum up, edge tables in two directions and the third genetic parent named "environment" are incorporated into EER-2D to improve the performance of recombination in two-dimensional structure. Figure 4 shows an example of parents with their corresponding edge tables and offspring. Firstly, element *a* is selected randomly to start the offspring. Secondly, element *d* is inserted based on the links in horizontal edge table. The left positions on the first row are filled with the help of horizontal edge table. Element *e* is chosen due to the link "*a*-*e*" in vertical edge table. Next, element *f* is chosen by two directions. This process continues until offspring is complete.

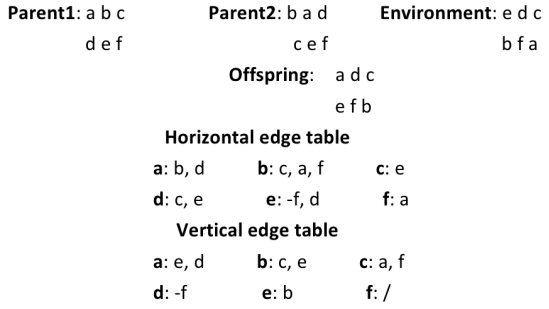



**Figure 4. Elements of EER-2D operator.**

## 4.3 Mutation

Mutation is an operator of innovation, which helps introduce new genetic materials to prevent the algorithm from converging too fast. The appropriate mutation operators could additionally help remove the incorrect and misleading links. The following mutation operators are proposed based on this thought.

### *4.3.1 Slide on Line (SOL)*

Generally, a small difference in two costs is caused by the sliding of small sets of genetic sequences. In the SOL operator, mutation makes the selected gene segment slide to a random position on its current line. The length and position of the segment are chosen by random. An example of SOL operator is showed in Figure 5, where shred *i*, *j* slide to the head of the second row.

$$\begin{bmatrix} a & b & c & d & e \\ f & g & h & \boldsymbol{i} & \boldsymbol{j} \\ k & l & m & n & o \end{bmatrix} \Rightarrow \begin{bmatrix} a & b & c & d & e \\ \boldsymbol{i} & \boldsymbol{j} & f & g & h \\ k & l & m & n & o \end{bmatrix}$$

**Figure 5. SOL operator.**

### *4.3.2 Swap within Line (SWL)*

The basic idea of SWL operator is to amend the reverse pairs within the same row. During a SWL operator, two shreds on the same are chosen by random and exchanged.

Moreover, since some rows of the individual are more likely to be misled than others, the possibility $p_n$ that $n$th row is chosen for mutation is defined to be variable. Since adaptive strategy is proved to be efficient in most evolutionary algorithms [19]-[21] in recent years, a simple self-adaptive strategy is designed to change its value:

$$p_n = \frac{k_n}{k_{sum}} \tag{11}$$

$$k_{sum} = \sum_{n=1}^{row} k_n \tag{12}$$

$$k_n = \begin{cases} k_n + 1, if\ generate\ a\ better\ result\ and\ p_n < 2/row \\ k_n - 1, if\ generate\ a\ worse\ result\ and\ p_n > 1/row \\ k_n, otherwise \end{cases} \tag{13}$$

where parameter $k_n$ is adjusted to keep $p_n$ in the range defined above. $k_{sum}$ is the total value of $k_n$ in the solution. The same strategy is also used in SOL operator.

### *4.3.3 Slide through Line (STL)*

Distinct from horizontal SOL operator, STL operator is carried out vertically. It is designed to focus on the rows with blank edges, which is more likely to appear in the wrong position. In the STL operator, a randomly chosen row slides to a random position and the other rows in the solution move accordingly.

### *4.3.4 Swap between Line (SBL)*

SBL is the simplest but also most flexible in these four operators. The swap of random two shreds from different rows could help improve the diversity of genetic population.

## 4.4 Row-Oriented Elitism Strategy (ROES)

In the genetic process, finding a correct solution needs cooperation of a large amount of genetic individuals during hundreds of generations. To shorten this time-consuming genetic process as well as improve the possibility of finding correct solution, a decomposition strategy is proposed, which is named "row-oriented elitism strategy".

It is obvious that the correctly positioned rows are not difficult to find as the correct solution. Since these correct rows are necessary elements of final solution, they are named "final rows". Furthermore, considering that the judgment of whether the current rows are final rows is impossible, the rows with lowest costs are put into use, which are more likely to be the final rows. These rows are named "elitism-rows" and identified by the first elements of the rows.

The ROES operator contains the following two steps. Firstly, a construction rule is defined to select appropriate elitism-rows and assemble these rows. The approach used here is to follow the

leading shreds on the rows of best individual in current generation. Then elitism-rows beginning with these shreds are chosen and stitched together according to the order in the best individual. An original "elitism-individual" is generated through the first step.

Secondly, there might be several shreds appearing more than once in the original elitism individual. To make the solution available and logical, these redundant shreds are randomly replaced by the unused shreds. A logical elitism-individual is generated through the second step.

The ROES operator proceeds as the following steps: (1) Elitism-rows are selected and stored during every genetic generation. (2) Elitism-individual is generated through the two steps mentioned above. (3) Elitism-individual is inserted into population of next generation.

Figure 6 shows an example of step (2) in the ROES operator. Shreds *a* and *d* are leading two rows in the best individual. The first and fourth elitism-rows which begin with the leading shreds are chosen and stitched together. However, there are two shreds *c* in original elitism-individual. To make this individual available, one of shreds *c* is replaced by shred *f* in the logical elitism-individual.

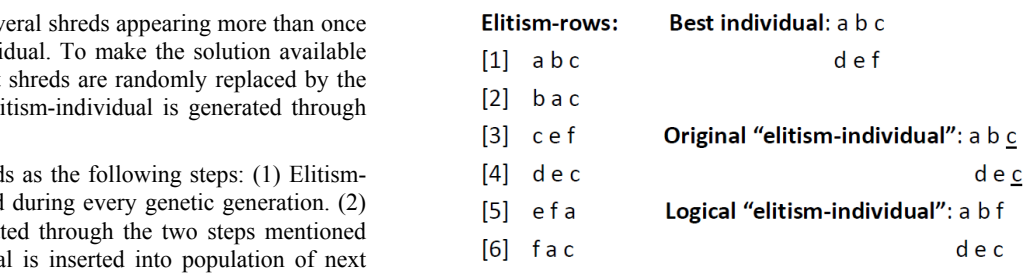
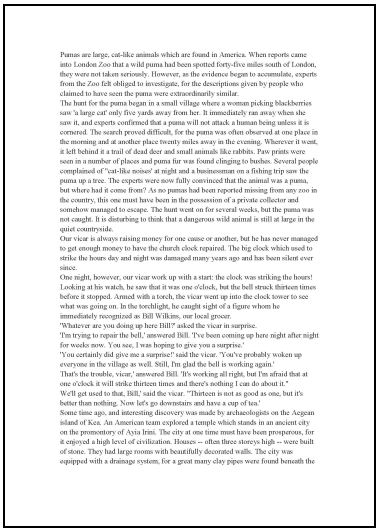


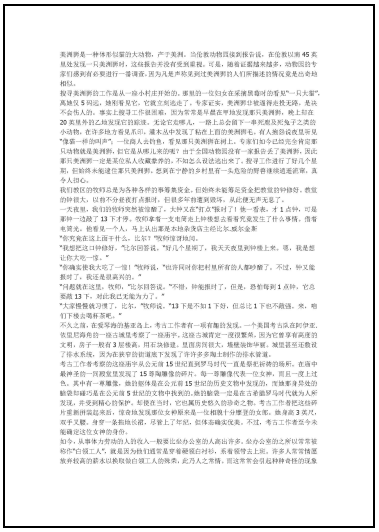

**Figure 6. Elements of ROES operator.**

**Procedure** Proposed Algorithm
**Begin**
*t*←0;
initialize (*P*(*t*));
evaluate (*P*(*t*));
**Do**
*P*(*t*) ← selection (*P*(*t*));
*P*(*t*) ← crossover (*P*(*t*));
*P*(*t*) ← mutation(*P*(*t*));
*P*(*t*) ← ROES operator(*P*(*t*));
*t* ← *t*+1;
*P*(*t*) ← *P*(*t*-1);
**While** (Reach process not converged and allowed generation not reached)
**Return** best individual
**End** Procedure

**Figure 7. Pseudo code of the combing algorithm.**



(a) e1



(b) c1

**Figure 8. Pages e1 and c1.**

## 4.5 Overall Process

In this section, a genetic algorithm with splicing-driven reproduction is proposed, whose whole procedure is showed in Figure 7. ROES operator is inserted into the process of genetic algorithm and cooperates with other operators.

# 5. EXPERIMENTAL RESULTS

## 5.1 Experimental Setup

Within this section, three methods are presented to compare. Besides the proposed algorithm, a method named "EER-2D" is involved to show the effect of genetic operations in the paper, which represents the proposed method without ROES operator. The genetic part of $HV^2$ method proposed in [4] is also used. To unify the cost values for comparison, all three methods use the same cost functions and initial methods in this paper. In the proposed algorithm, the rates of mutation operators are set to 0.02-0.08, and the rates of crossover operator and ROES operator are both set to 0.8.

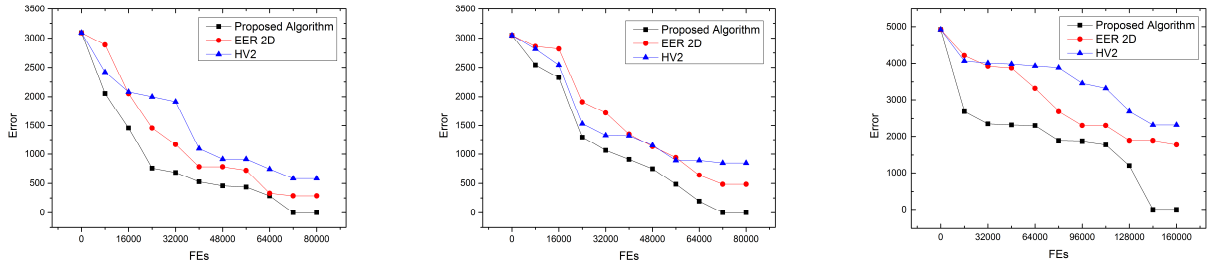
All the methods were implemented in C and performed on a Core i3-3240 CPU with 4GB RAM. The test instances were generated by shredding text documents using three different cutting patterns, which involve 36 to 81 shreds. Pages e1 and c1 are composed of continuous text only and showed in Figure 8. Furthermore, e2 and c2 contain tables with horizontal and vertical lines while e3 and c3 show listings. Respectively, pages begin with "e" are English text documents while the others are in Chinese.

## 5.2 Solution Measurement

To find out the effect of introduced methods, two standards are defined to help reflect the quality of solutions. Firstly, the quality of a single solution is described by the proportion of correct links and shreds appearing in total links and shreds. A link is defined to be correct if it also appears in the correct solution. A shred is correct if it has the same position as the same shred in correct solution. In general, a solution with higher proportion of correct links and shreds could help extract more useful information from documents. It is worth noting that the percentage gap with respect to the objective value of original document page is more likely to be useless in this quality measure. This is motivated by the fact that the percentage gap is directly affected by the definition of cost functions. Secondly, the quality of a set of solutions is described by the standard deviation of the proportion, which could help reflect the stability of the method. Since the experimental results are based on 20 tests, the mean and stand deviation of the defined proportion are applied.

**Table 1. Results and comparisons of the three algorithms**

| Methods | | Proposed Algorithm | | EER 2D | | $HV^2$ | |
|---|---|---|---|---|---|---|---|
| Page | Patterns | Mean % | Std % | Mean % | Std % | Mean % | Std % |
| e1 | 6×6 | **100** | **0** | **100** | **0** | **100** | **0** |
| | 6×9 | **100** | **0** | **100** | **0** | **100** | **0** |
| | 9×9 | **81.67** | 14.01 | 49.56 | 2.15 | 47.39 | **1.69** |
| e2 | 6×6 | **100** | **0** | **100** | **0** | **100** | **0** |
| | 6×9 | **98.16** | **3.85** | 81.73 | 11.44 | 75.61 | 11.57 |
| | 9×9 | **82.67** | 11.50 | 45.17 | 1.37 | 46.11 | **1.11** |
| e3 | 6×6 | **100** | **0** | **100** | **0** | **100** | **0** |
| | 6×9 | **100** | **0** | **100** | **0** | **100** | **0** |
| | 9×9 | **89.78** | 8.72 | 48.22 | **2.32** | 47.94 | 2.52 |
| c1 | 6×6 | **100** | **0** | **100** | **0** | 97.81 | 9.54 |
| | 6×9 | **100** | **0** | **100** | **0** | **100** | **0** |
| | 9×9 | **83.28** | 10.87 | 64.44 | **4.41** | 67.61 | 7.67 |
| c2 | 6×6 | **100** | **0** | **100** | **0** | **100** | **0** |
| | 6×9 | **100** | **0** | 92.82 | 5.23 | 70.54 | 4.75 |
| | 9×9 | **87.5** | 11.2 | 65.28 | 8.62 | 62.06 | **7.69** |
| c3 | 6×6 | **94.38** | **6.06** | 84.90 | 14.83 | 80.52 | 13.12 |
| | 6×9 | **100** | **0** | **100** | **0** | **100** | **0** |
| | 9×9 | **95.61** | 8.35 | 61.78 | 6.86 | 53.28 | **4.39** |




(a) 6×6 shreds (b) 6×9 shreds (c) 9×9 shreds

**Figure 9. Convergence curves of the three algorithms.**

**Table 2. Average computation time of the three algorithms (ms).**

| Patterns | Proposed Algorithm | EER 2D | $HV^2$ |
|---|---|---|---|
| 6×6 | 6104.71 | 6008.25 | 5811.79 |
| 6×9 | 8866 | 8478.39 | 7991.25 |
| 9×9 | 28381.09 | 27486.83 | 26867.61 |

## 5.3 Comparison of Results

Table 1 shows the results obtained by applying proposed algorithm; EER-2D and $HV^2$ in the instances mentioned above and the best results are marked in bolt. It can be observed that: the proposed algorithm clearly outperforms other two algorithms. On one hand, for instances of 36 and 54 shreds, the proposed algorithm exhibits higher degree of stability. On the other hand, for instances of 81 shreds, the proposed algorithm could achieve higher accuracy rates during the same generation. Comparing the proposed algorithm with EER-2D algorithm, we can see that the ROES operator helps improve the search efficiency of proposed algorithm. Furthermore, the results of EER-2D algorithm show that the proposed mutation and crossover operators are efficient for this problem.

In addition, the convergence curves of the three algorithms are plotted in Figure 9. The figure clearly shows that the proposed algorithm converges fastest to achieve the correct solution. Moreover, the figure also shows the strong search ability of proposed algorithm at the beginning of reconstruction. The data of average computation time in table 2 shows that the efficiency of proposed algorithm is enhanced without costing too much computation time. To summarize, the proposed algorithm is a very competitive algorithm in solving RCCSTD problem.

# 6. CONCLUSION

In this work we propose a genetic algorithm with splicing-driven reproduction for the reconstruction of cross-cut shredded text documents. To improve the capability of solving this problem, a splicing-driven crossover, four mutation operators, and a row-oriented elitism strategy is designed. Additionally, a new cost function based on both edge and empty vector-based splicing error is defined. To test the performance of the algorithm, shreds from six documents in two languages are used. Based on the experimental results, this proposed algorithm clearly outperforms the previous best-known algorithms.

# 7. ACKNOWLEDGMENTS

This work was supported in part by the High-Technology Research and Development Program (863 Program) of China No. 2013AA01A212, in part by the NSFC for Distinguished Young Scholars 61125205, in part by the NSFC Joint Fund with Guangdong under Key Projects U1201258 and U1135005.

# 8. REFERENCES

[1] Prandtstetter M. 2009. Hybrid optimization methods for warehouse logistics and the reconstruction of destroyed paper documents. Ph.D. thesis, Vienna University of Technology.

[2] Sleit A, Massad Y, Musaddaq M. 2013. An alternative clustering approach for reconstructing cross cut shredded text documents. *Telecommunication Systems*, *52*(3), 1491-1501.

[3] Mladenović N, Hansen P. 1997. Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097-1100.

[4] Schauer C, Prandtstetter M, Raidl G R. 2010. A memetic algorithm for reconstructing cross-cut shredded text documents. In *Hybrid Metaheuristics*, 103-117.

[5] Prandtstetter M, Raidl G R. 2009. Meta-heuristics for reconstructing cross cut shredded text documents. In *Genetic and Evolutionary Computation Conference*, 349-356.

[6] Justino E, Oliveira L S, Freitas C. 2006. Reconstructing shredded documents through feature matching. *Forensic science international*, *160*(2), 140-147.

[7] Prandtstetter M, Raidl G R. 2008. Combining forces to reconstruct strip shredded text documents. In *Hybrid Metaheuristics*, 175-189.

[8] Ukovich A, Ramponi G, Doulaverakis H, et al. 2004. Shredded document reconstruction using MPEG-7 standard descriptors. In *International Symposium on Signal Processing and Its Applications*, 334-337

[9] Holland J H. 1975. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press.

[10] Leardi R. 2000. Application of genetic algorithm–PLS for feature selection in spectral data sets. *Journal of Chemometrics*, *14*(5-6), 643-655.

[11] Rahmat-Samii Y, Michielssen E. 1999. Electromagnetic optimization by genetic algorithms. John Wiley & Sons Inc.

[12] Raidl G R, Hu B. 2010. Enhancing genetic algorithms by a trie-based complete solution archive. In *Evolutionary Computation in Combinatorial Optimization*, 239-251.

[13] Li X, Yao X. 2012. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Transactions on Evolutionary Computation*, *16*(2), 210-224.

[14] Lu S, Tan C L. 2007. Automatic detection of document script and orientation. In *International Conference on Document Analysis and Recognition,* 237-241.

[15] Davis L. 1985. Applying adaptive algorithms to epistatic domains. In *International Joint Conference on Artificial Intelligence*, 162-164.

[16] Goldberg D E, Lingle R. 1985. Alleles, loci, and the traveling salesman problem. In *International conference on Genetic Algorithms and Their Applications*, 154-159.

[17] I. Oliver, D. Smith, and J. Holland. 1987. A Study of permutation crossover operators on the traveling salesman problem. In *International Conference on Genetic Algorithms*, 224-230.

[18] Starkweather T, McDaniel S, Mathias K E, et al. 1991. A comparison of genetic sequencing operators. In *International Conference on Genetic Algorithms*, 69-76.

[19] Zhan Z H, Zhang J, Li Y, et al. 2009. Adaptive Particle Swarm Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *39*(6), 1362-1381.

[20] Yu W J, Shen M, Chen W N, et al. 2014. Differential evolution with two-level parameter adaptation. *IEEE Transactions on Cybernetics*, *44*(7), 1080-1099.

[21] Gong Y J, Zhang J. 2013. Small-world particle swarm optimization with topology adaptation. In *Genetic and Evolutionary Computation Conference*, 25-32.