Reconstructing Cross Cut Shredded Documents with a Genetic Algorithm with Solution Archive*

Benjamin Biesinger, Christian Schauer, Bin Hu, and Günther R. Raidl

Institute of Computer Graphics and Algorithms
Vienna University of Technology
Favoritenstrae 911/1861, 1040 Vienna, Austria
{biesinger|schauer|hu|raidl}@ads.tuwien.ac.at

1 Introduction

The reconstruction of shredded documents is of high interest not only in forensic science but also when documents are destroyed unintentionally.

Reconstructing cross-cut shredded documents (RCCSTD) is particularly difficult since the documents are cut into rectangular pieces of equal size. Since shape information along the edges—in contrast to hand torn pieces—cannot be exploited, the reconstruction solely depends on the information written on the shreds. Therefore, we use a metric for calculating the number of gray value mismatches along the edges of two shreds put next to each other either horizontally or vertically. Consequentially, we model the document reconstruction as a combinatorial optimization problem minimizing the overall mismatch of the reconstructed document. Since we focus in this work on the combinatorial aspect of the problem we use this simple metric which can be replaced in future work by more advanced pattern recognition techniques, see [2] for a sample method.

In previous work, Prandtstetter and Raidl [3] developed an Ant Colony Optimization and a Variable Neighborhood Search (VNS) for the RCCSTD, while Schauer *et al.* [5] proposed a Memetic Algorithm (MA). Sleit *et al.* [6] proposed a different approach by iteratively merging two clusters that fit together well and repairing possibly occurring conflicts.

In this work the MA from [5] is adapted and extended by a complete solution archive in order to avoid duplicate solutions by efficiently storing all visited solutions in a special data structure. If a duplicate solution is detected it is converted into a similar yet unconsidered one. This is done to preserve the diversity of the population and to avoid unnecessary re-evaluations of already visited solutions. This approach is a rather new method for duplicate detection and conversion which was successfully applied on several binary problems (e.g., MAX-SAT) in [4] as well as on the generalized minimum spanning tree problem [1].

2 A Genetic Algorithm with Solution Archive

For a detailed description of the GA and its operators, which is extended by our solution archive, see [5]. To encode solutions the authors used an $n \times n$ array that

* This work is supported by the Austrian Science Fund (FWF) under grant P24660.

represents the absolute position of each shred. In the current work we propose a more compact solution representation using an 1-dimensional array that does not store blank shreds at the end of a row. Therefore, a special character for a line break is introduced. Theoretically this solution representation is not necessarily more compact than the original one, but in practice we are able to reduce the average length of a solution significantly.

The underlying data structure of the solution archive is a trie, a tree data structure commonly used for dictionaries. Each node of the trie has n children representing the possible alleles. The height of the trie is the length of the genome. In the commonly used *indexed trie* the children of each trie node are stored in an array. Preliminary tests showed that using an *indexed trie* needs a huge amount of memory. Therefore, several modifications are made to save memory. The most memory saving change was to use a *linked trie*, in which the children of the nodes are stored as linear lists.

The solution conversion that is performed after detecting a duplicate is entirely carried out in the trie. A conversion level l, which corresponds to the l-th gene in the solution array, is chosen randomly. On this level another valid child that has not been visited yet, which is represented as a *null* pointer, is chosen and the solution array is altered accordingly. If there is no *null* pointer in the current node, we follow the old solution one level down and the procedure is repeated. Since subtries where all children are *complete* are pruned, we know that there must be a node with at least one *null* child. Hence, this method guarantees that the generated solution has not been visited yet.

3 Results and Conclusions

We tested the proposed algorithm on several benchmark instances using different cutting patterns and compared the GA with and without the solution archive. By using the indexed trie as data structure, the memory consumption increased clearly too strongly after a relatively small number of iterations. Therefore the GA had to stop when it ran out of memory instead of when it converged, which produced worse results than the GA without archive. The linked trie variant, without causing additional run-time, consumed only one fourth of the memory. Unfortunately, even with this improvement the preliminary results were only on par with those generated by the GA without archive when using the same amount of time as stopping criterion. The reason is that the solution archive in this case is not able to fully compensate its overhead by saving the effort for re-evaluating duplicate solutions. However, when using a fixed number of iterations, the results of the GA with archive were better by far.

We conclude that on combinatorial optimization problems where the solution representation is not compact, using a solution archive with linked trie results in a substantial memory advantage over the indexed trie variant.

References

1. Hu, B., Raidl, G.R.: An evolutionary algorithm with solution archives and bounding extension for the generalized minimum spanning tree problem. In: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation (GECCO). pp. 393–400. ACM Press, Philadelphia, PA, USA (2012)
2. Perl, J., Diem, M., Kleber, F., Sablatnig, R.: Strip shredded document reconstruction using optical character recognition. In: 4th International Conference on Imaging for Crime Detection and Prevention 2011 (ICDP 2011). pp. 1–6 (2011)
3. Prandtstetter, M., Raidl, G.R.: Meta-Heuristics for Reconstructing Cross Cut Shredded Text Documents. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO). pp. 349–356. ACM Press (2009)
4. Raidl, G.R., Hu, B.: Enhancing genetic algorithms by a trie-based complete solution archive. In: Cowling, P., Merz, P. (eds.) Evolutionary Computation in Combinatorial Optimization. LNCS, vol. 6022, pp. 239–251. Springer Berlin Heidelberg (2010)
5. Schauer, C., Prandtstetter, M., Raidl, G.R.: A memetic algorithm for reconstructing cross-cut shredded text documents. In: Proceedings of the 7th international conference on Hybrid metaheuristics. pp. 103–117. HM2010, Springer-Verlag (2010)
6. Sleit, A., Massad, Y., Musaddaq, M.: An alternative clustering approach for reconstructing cross cut shredded text documents. Telecommunication Systems pp. 1–11 (2011)