TREE STRUCTURES FOR EVOLUTIONARY COMPUTING

by

Esin SAKA

Middle East Technical University

Department of Computer Engineering

Ankara, Turkey

December, 2003

ABSTRACT

One of the most important factors on the success of genetic programming about trees is the representation of them. The representation should exhibit efficiency, locality and heritability to enable effective evolutionary computing.

Neville proposed three different methods for encoding labeled trees. In 2001, it is reported that, Prüfer-numbers is a poor representation of spanning trees for evolutionary search. But there is no information about the performance of Neville's second type of encoding.

In this report we examine methods for coding labeled trees, explain locality and heritability and give four possible testing problems to compare different codings in evolutionary algorithms: the degree-constrained minimum spanning tree problem, the optimal communication spanning tree problem, the rectilinear Steiner problem and the fixed-charge transportation problem.

TABLE OF CONTENTS

INTRODUCTION

An evolutionary algorithm (EA) studies on a population of data structures that represent candidate solutions to a problem. In a decoder based EA, each data structure can be thought as a instruction list to a decoder. The decoder builds the solution which is represented by the data structure. The data structure is the genotype of the solution and the decoded solution is the phenotype of solution.

In EAs, how the data structure is implemented is important. Because, the decoder should be fast and the coding implemented in genotype-decoder pair should exhibit locality and heritability. In other words, small changes in genotypes should correspond to small changes in the solutions they represent and solutions generated by crossover should combine features of their parents.

EAs have been applied to problems that search spaces of spanning trees. It is easy to find an unconstrained minimum spanning tree but when constraints are around, it becomes computationally hard to solve.

Prüfer and Neville studied on the problem of representing the trees as efficient and suitable as possible. In 1918 Prüfer suggested a deceptively elegant coding of spanning trees. It is a one-to-one mapping between spanning trees on n nodes and strings of n-2 node labels. Conventionally the integers from 1 to n label the nodes and the strings of labels are called Prüfer numbers. Later Neville offered three codings [3]. Bryant, Julstrom, Raidl and Rothlauf claimed that Prüfer numbers is a poor representation of spanning trees for evolutionary search [2]. They compared Prüfer numbers with other

codings of spanning trees in EAs for four problems: the degree-constrained spanning tree problem, the communication spanning tree problem, the rectilinear Steiner problem and fixed-charge transportation problem. In every case, evolutionary search is more effective with other codings than Prüfer numbers [2].

Although Prüfer numbers are said to be poor representation, there is no study about Neville's codings. This report gives the necessary information to start such a study.

EXISTING METHODS FOR ENCODING LABELED TREES

Prüfer's method encodes a tree by iteratively deleting the leaf node with minimum label and recording the neighbor until only one edge remains. One drawback of the Prüfer code is its lack of structure for determining the diameter or center(s) of the tree directly from the code without first constructing the tree. For the tree in Figure 1, Prüfer code is (9, 7, 7, 3, 10, 4, 4, 7, 1).

Neville proposed three methods to encode a labeled tree. The first one is similar to Prüfer's method. A root node is chosen and it is never deleted. When deleting the leaves, the one with smallest label is chosen. If the node with biggest label is chosen as the root node, the method becomes same with Prüfer's method. For the tree in Figure 1, Neville's first code is (9, 7, 7, 3, 10, 4, 4, 7, 1).

Neville's second method proceeds in k stages where k is the radius of the tree. In each stage, the leaves are deleting in the ascending order of their labels and their neighbors are recorded. The process is repeated until one edge remains. If we apply this method for the tree in Figure 1, at first stage, we will delete 2, 5, 6, 8 and 11. Their

neighbors to record are: 9, 7, 7, 3, 1. At second stage, 1, 3, 9 will be deleted and 7, 10 and 4 will be recorded. At last stage, 7 will be deleted and 4 will be recorded. Then the code is (9, 7, 7, 3, 1, 7, 10, 4, 4). This algorithm lead to a simple algorithm for computing the diameter of a tree directly from the code. However, there is no known O(n)-time algorithm to encode or decode the trees. Besides, at each stage there is a necessity of ordering the leaf nodes.

The third approach of Neville studies on sub-trees. Firstly it chooses the leaf with smallest label. It removes the leaves until no leaves remain in that sub-tree. When there is no leaf to remove in the sub-tree, it removes the leaf with smallest label at the tree. Similar to other encodings, whenever one node is deleted, it neighbor is recorded. This procedure continues until one edge remains. For the tree given in Figure 1, the leaf with smallest label is 2. Then we remove 2 and record 9. Now, 9 is a leaf and we remove 9, record 4. Then we do not have a leaf to delete at the sub-tree. The next leaf to delete is 5, since it is the one with smallest label at hand. 5 is deleted and 7 is recorded. 6 is deleted, 7 recorded. 8, 3, 10, 4 and 7 are deleted in order from the same sub-tree and edge connecting 1 and 11 remained. So, the code is (9, 4, 7, 7, 3, 10, 4, 7, 1).

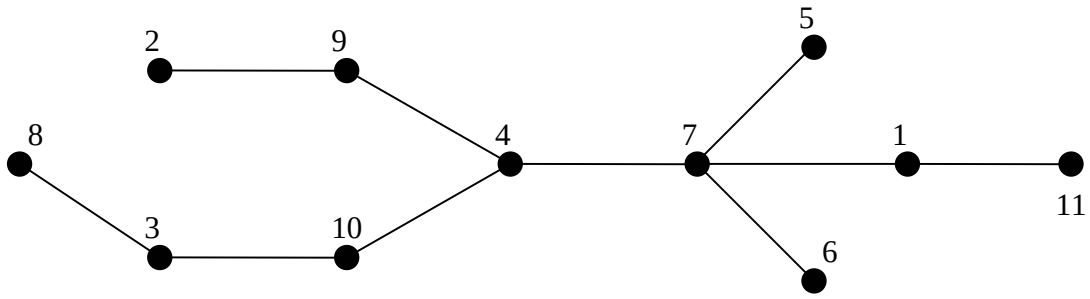

Figure 1: A tree on 11 nodes

PROPERTIES

Efficiency and O(n)-time complexity are important for encoding and decoding methods. But for an coding to be considered as a good representation for labeled trees, it should also have high locality and heritability.

A coding has high locality if mutating a genotype changes the corresponding phenotype only slightly. Prüfer numbers are said to have poor locality [2].

A coding has high heritability with respect to a crossover operator, if offspring phenotypes consist mostly of substructure of their parents' phenotypes.

SAMPLE PROBLEMS

Four problems are suggested to use for testing a coding method.

The Degree-Constrained Minimum Spanning Tree Problem

Given a weighted undirected graph G, the degree-constrained minimum spanning tree problem (d-MSTP) seeks a spanning tree on G of minimum weight whose degree does not exceed d>1.d-MSTP is NP-hard [2].

The Optimal Communication Spanning Tree Problem

Consider a collection of nodes. Each node demands communicating with each other. In Optimal Communication Spanning Tree Problem (OCSTP), a tree-structured network of minimum total cost of connecting all nodes is looked for. A links flow is the sum of the communication demands between all pairs of nodes communicating directly or indirectly

over the link. The cost for each link is not fixed a priori but depends on the length and capacity of the link. The links capacity must satisfy the link's flow and this flow depends on the entire tree structure. OCSTP is NP-hard [2].

The Rectilinear Steiner Problem

Given a collection of points in a plane, a rectilinear Steiner tree (RStT) is a tee of horizontal and vertical line segments that connects them all. The length of a RStT is the sum of its segments' lengths. The search of minimum length on a set of points is the rectilinear Steiner problem (RStP) and it is NP-hard.

The Fixed-charge Transportation Problem

Consider m sources and n destinations. The problem is to distribute a commodity from sources to destinations with minimum cost of transportation. The amounts of commodity available at each source and required at each destination are known.

CONCLUSION AND FUTURE WORK

One of the most important factors on the success of genetic programming about trees is the representation of them. The representation should exhibit efficiency, locality and heritability to enable effective evolutionary computing.

There are studies on Prüfer numbers and it is said to be a poor representation of spanning trees for evolutionary search. Although there exist some algorithms for Neville's methods, no consideration of them on evolutionary computing is available. Is suggested to examine algorithms for encoding and decoding of Neville's encodings,

examine the properties of Neville's encodings and compare it with other codings in evolutionary algorithms for four problems: the degree-constrained minimum spanning tree problem, the optimal communication spanning tree problem, the rectilinear Steiner problem and the fixed-charge transportation problem

APPENDICES

[1] N. Deo, P. Micikevicius. Prüfer-like codes for labeled trees. Congressus Numerantium, vol. 151, pp. 65-73, 2001.

[2] J. Gottlieb, B. A. Julstrom, G.R.Raidl, F. Rotlauf. Prüfer Numbers: A poor representation of spanning trees for evolutionary search. Technical Report, 2001.

[3] E. H. Neville. The codifying of tree-structure. Proceedings of Cambridge Philosophical Society, vol 49, pp. 381-385, 1953.

[4] N. Deo, P. Micikevicius. A new encoding for labeled trees employing a stack and a queue. Bulletin of the ICA, vol. 34, pp. 77-85, 2002.

[5] N. Deo, P. Micikevicius. Parallel algorithms for computing Prüfer-like codes for labeled trees. Technical Report CS-TR-01, Department of Computer Science, University of Central Florida, Orlando, 2001.