Read's linear codes and genetic programming

Martin Pelikán, Vladimír Kvasnička, and Jiří Pospíchal

Department of Mathematics, Slovak Technical University, 812 37 Bratislava, Slovakia
Martin.Pelikan@st.fmph.uniba.sk, {kvasnic, pospich}@cvt.stuba.sk,

ABSTRACT

A theoretical base for efficient handling of mutation and crossover in Read's coding of rooted trees is given, which keeps correctness. Theorem for checking, whether for a given code there exists a corresponding rooted tree, is used by an algorithm for efficient generation of a correct code.

Discussion

Standard evolutionary computation applications are usually performed over populations of solutions that are represented by linear strings of digits. To work with trees more efficiently in compiled languages without handling tree structures by pointers of nodes, we have used Read's coding (Read, 1972), see Figure 1, which implementation shows very good performance. The code also allows its easy parsing and implementation of Koza's ADF. It is not necessary to code graphs uniquely, isomorphic rooted trees may have different code.

Bibliography

Read, R.C., 1972. Coding of Unlabeled Trees. In Read R.C. (editor). *Graph Theory and Computing*. New York: Academic Press. Pages 153-182.

Kvasnička V., Pospíchal J. 1992. An Existence Theorem for Molecular Graphs Determined by a Sequence of Valence States. *J. Math. Chem.* 11, 353-364.

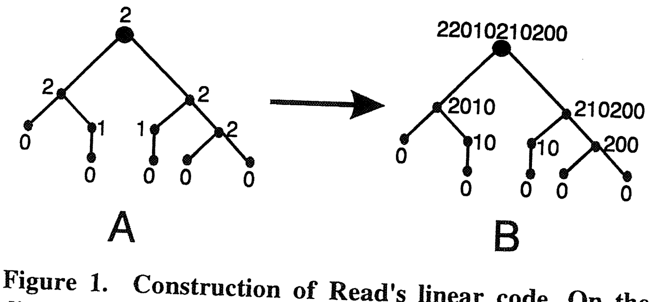


Figure 1. Construction of Read's linear code. On the diagram A each vertex of the tree is evaluated by an integer (either a valence for the root or a valence decreased by one for other nonroot vertices). The code in diagram B is constructed so that going successively in a bottom-up manner we concatenate the evaluation of the given vertex and codes of bottom vertices.

```
procedure Generation_Code(output:p,α);
begin p:=p_min+rand(p_max-p_min+1);
      α_1:=1+rand(p-1); d_1:=p-1;
      for j:=2 to p-1 do
      begin d_j:=d_j-1-α_j-1;
            if d_j=p-j then
            α_j:=1+rand(d_j)else
            α_j:=rand(d_j+1);
      end;
      α_p:=0;
end;
```

Algorithm 1. Random generation of Read's linear code α of the length p bounded by $p_{min} \le p \le p_{max}$. The generated code is automatically graphable (Kvasnička, 1992). Function rand(I) generates integers with uniform distribution from the closed interval [0,I-1].

```
procedure
Mutation(input:p,α;output:p',α');
begin τ:=2+rand(p-2);
      r:=length of a subcode α_2 of α
         starting at τ;
      r':=p_min-p+r+rand(p_max-p_min+1);
      Generation_Code(r',α'_2);
      α'=(α_1,α'_2,α_3); p':=p-r+r';
end;
```

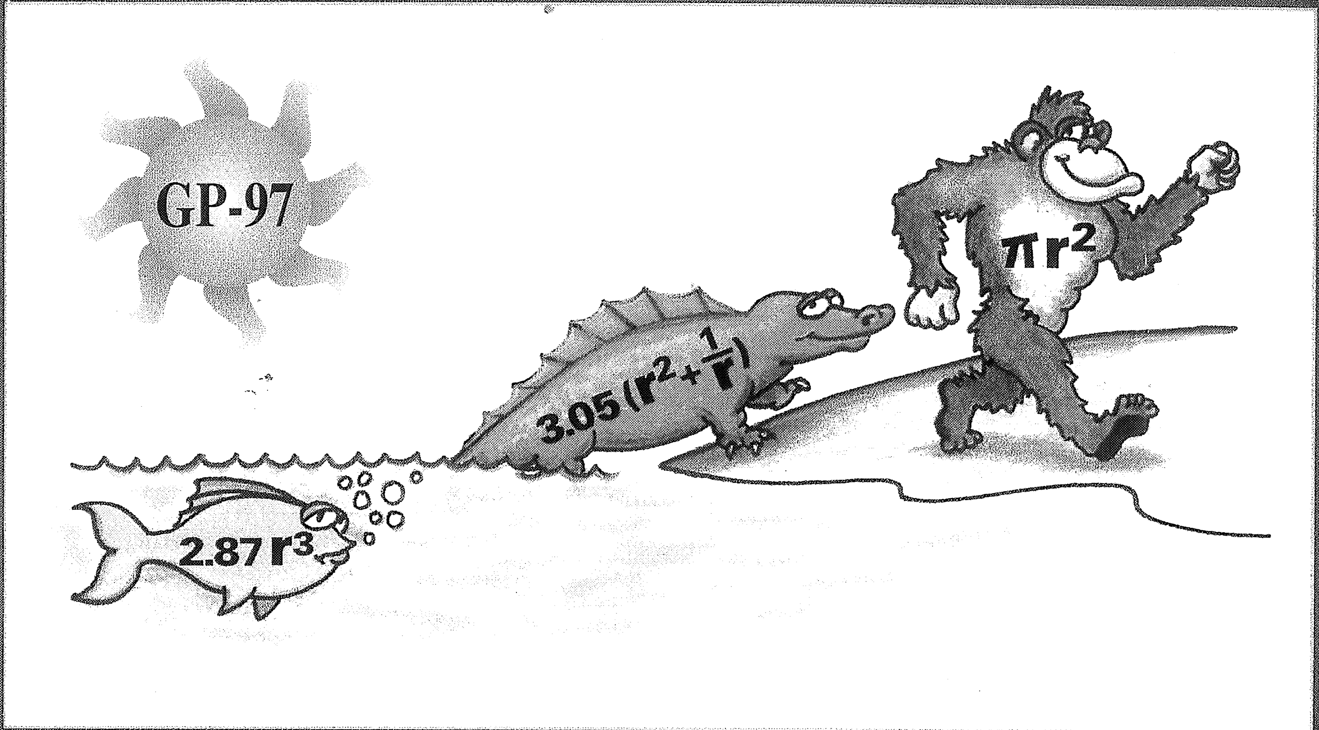
Algorithm 2. Implementation of the mutation process applied to a code α, the resulting code is denoted by α'. The mutation point τ is random, the length of the subcode that starts at this point is denoted by r. The length of new randomly generated code is denoted by r'.

```
procedure Crossover(input:p,p̃,α,β;
          output: p', p̃',α',β');
begin τ:=2+rand(p-2);τ̃:=2+rand(p̃-2);
      α'=(α_1,β_2,α_3); β'=(β_1,α_2,β_3);
      p':=p-|α_2|+|β_2|; p̃':=p̃-|β_2|+|α_2|;
end;
```

Algorithm 3. Implementation of the crossover operation for two parental linear codes α and β that are modified to offspring codes α' and β'. Crossover points τ and $\tilde{\tau}$ are properly randomly selected so that the resulting lengths satisfy $p_{min} \le |\alpha'| \le p_{max}$ and $p_{min} \le |\beta'| \le p_{max}$.

CONFERENCE PROCEEDINGS

Genetic Programming 1997



Proceedings of the Second Annual Conference
July 13-16, 1997, Stanford University

Edited by
John R. Koza
Kalyanmoy Deb
Marco Dorigo
David B. Fogel
Max Garzon
Hitoshi Iba
Rick L. Riolo