# COMPRESSED LINEAR GENETIC PROGRAMMING: EMPIRICAL PARAMETER STUDY ON THE EVEN-N-PARITY PROBLEM

Johan Parent [a] Ann Nowé [b] Anne Defaweux [b] Kris Steenhaut [a]

[a] *Vrije Universiteit Brussel, Faculty of Engineering, ETRO*
[b] *Vrije Universiteit Brussel, Faculty of Science, COMO*

**Abstract**

Modularisation is of intrest to the EA community as a mechanism to allow evolution to address larger problems. Starting from basic elements modularisation could make it possible to reuse higher level functionalities discovered during the evolutionary search process. This text presents a compression based extension of the standard genetic algorithm. By adding a compression operation in the loop of a genetic algorithm (GA) a simple modularisation mechanism is obtained. This compression is applied to the genotypes present in the population. Promising allele combinations are replaced by a shorter representation, i.e. a single symbol, thus results in a compressed representation. This *placeholder* symbol is then added to the genetic alphabet. The GA extended with compression is called the compressed genetic algorithm (cGA). The compression scheme fullfils two goals. First, compression aims at protecting building blocks from crossover by shortening them. A second goal is to obtain a form of code reuse, this is achieved by complementing the basic genetic alphabet with the placeholder symbols. The combination of the protection against crossover and code reuse makes it possible for the cGA to improve upon the GA. Critical to the performance of the cGA is the length $l$ of the allele combinations being compressed as well the identification of *good* combinations. The length of the combinations is a parameter set by the user. The selection of the combinations is based on their repeated occurence within a pool of above average individuals. This criterion has the advantage of relying on the information already present in the population. When combined with an evaluator the cGA forms a compressed linear genetic programming (cl-GP) system. This system uses a linear encoding reminescent from the underlying GA. The program execution is handled by a problem dependent evaluator. This paper presents a study of the impact of parameters added by the cGA compared to a canonical GA. Different instances of the Even-n-Parity problem were used as a benchmark. The cGA introduces the population compression parameter ($\kappa$), the length of the compressed allele combinations ($l$) and the size of the pool used to identify good allele combinations ($N$). Experiments [1] illustrate the strong influence of the combination length $l$. This can readily be explained by the implicit assumption of tight linkage between genes when applying a substitution based compression scheme.

# 1 Results

All the results presented here are the average of 100 independent runs, using randomly seeded initial populations of 500 individuals. The genotype length is 32. Other settings were: crossover rate 80%, mutation rate 5% and the top 5% of the population was kept at every generation. Every experiment lasted 50 generations. If not mentioned otherwise the following values were used for the runs using the cl-GP: $\kappa = 0.30$ (tournament size 4) and the length of the dictionary entries is 2. A pool of 10 individuals (fitness proportional selected) is used to build the compression dictionary.

## 1.1 Population compression, $\kappa$

Table 1 contains the raw fitness for the best individuals of the population for different problem instances. The second column, 0% of population selected for compression, serves as the reference since in this case the cl-GP is equivalent to a *normal* l-GP. As was expected high values do not lead

| n | $\kappa = 0.0$ | 0.3 | 0.6 | 0.9 |
|---|---|---|---|---|
| 5 | 0.891 | 0.964 | 0.950 | 0.949 |
| 7 | 0.680 | 0.783 | 0.796 | 0.769 |
| 9 | 0.5 | 0.528 | 0.530 | 0.536 |

Table 1: The number of individuals that are selected for compression influence the performance of the system. High values (60% and 90%) reduce the diversity and reduce the overall performance.

| n | $N = 10$ | 20 | 50 | 100 |
|---|---|---|---|---|
| 5 | 0.964 | 0.964 | 0.964 | 0.965 |
| 7 | 0.783 | 0.795 | 0.785 | 0.774 |
| 9 | 0.528 | 0.537 | 0.531 | 0.524 |

Table 2: The raw fitness of the best of population as a function of the pool size used to build the dictionary of the substitution coder. The difference between various pool size is less pronounced compared to influence of $\kappa$ on the performance.

to improved performance. This is most noticeable for small problem instances ($n = 5$ for example). Surprisingly, this tendency seems to reverse as the problem size increases.

### 1.2 Pool size, $N$

Table 2 presents the raw fitness of the best of population when using different pool sizes. The use of fitness proportional selection (with overselection) makes the cl-GP relatively insensitive to the pool size. This can be explained by the fact that overselection makes it possible for the some individuals to be present several time in the pool. As a result pools of different sizes can in fact show little differences in diversity.

### 1.3 Substring length, $l$

Table 3 contains the fitness of the best individual when using different substring lengths. Using longer dictionary entries decreases the performance. This decreasing trend can be observed for all the problem instances. As the substrings get longer it becomes harder for the cl-GP to protect good schemata. This clearly illustrates that by protecting sub-optimal building blocks the performance of the cl-GP is penalized.

## References

[1] DEFAWEUX Anne STEENHAUT Kris PARENT Johan, NOWE Ann. Compressed linear genetic programming: empirical parameter study on the even-n-parity problem. *WSEAS TRANSACTIONS ON INFORMATION SCIENCE AND APPLICATIONS*, 2(5):540–545, 2005.

| n | $l = 2$ | 3 | 4 | 5 |
|---|---|---|---|---|
| 5 | 0.964 | 0.948 | 0.931 | 0.909 |
| 7 | 0.783 | 0.753 | 0.750 | 0.717 |
| 9 | 0.528 | 0.524 | 0.527 | 0.522 |

Table 3: Increasing the length of the dictionary entries makes it less likely for the assumption of tight linkage between gene values to hold. For the problem sizes the raw fitness decrease as the substring length increases.