# Adaptive Markov Recombination

## Genetic Engineering with Active Partial Solution Preservation

Arvid Halma
University of Amsterdam, Informatics Institute
Kruislaan 403, 1098 SJ Amsterdam
ahalma@science.uva.nl

Remi Turk
University of Amsterdam, Informatics Institute
Kruislaan 403, 1098 SJ Amsterdam
rturk@science.uva.nl

## Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods and Search*

## General Terms

Algorithms

## Keywords

Adaptation/self-adaptation, Evolution strategies, Genetic algorithms, Recombination operators

## 1. PARTIAL SOLUTION PRESERVATION

A new class of 'crossover' algorithms is proposed that is used in what we coin the *Genetic Engineering* strategy. These algorithms explicitly consider the fitness of all subsets of candidate solutions when creating the next iteration of candidate solutions. If the fitness of individual solutions is positively correlated to the fitness of their subsets, one could optimize faster by decreasing the probability that good subsets are destroyed while performing recombination. Finding promising partial solutions turns out to be simply a matter of counting. Our implementation, Markov recombination, creates a histogram of all symbol transitions in all candidate solutions in the population at a certain time step. Randomized Markov chains is then be used to generate offspring.

## 2. THE NEXT EVOLUTIONARY STRATEGY

A hierarchy of increasingly advanced optimizers may begin with *generate and test*, that can be surpassed by *hill climbing* and *simulated annealing* and from these "asexual evolution" optimizers to their "sexual" successor recognized in *genetic algorithms*.

We will extend the list with the metaphorical Genetic Engineering laboratory. In such a laboratory one possesses a large amount of genetic material of various individuals at a certain time. One may analyze the building blocks of this material and is then able to carefully select promising building blocks to breed promising individuals. This preservation of good partial solutions is a virtue, since a blind crossover operator is likely to cut them in meaningless pieces and therefore decreases the speed of convergence.


*GECCO'07,* July 7–11, 2007. London, England, United Kingdom.
ACM 978-1-59593-697-4/07/0007.

The question now is what good partial solutions are and how to find them. We can elegantly resolve this issue as follows. Holland showed in his proof of convergence of classical genetic algorithms that individuals with good partial solutions (schemata with an above average fitness) will increase in number over time [1]. It turns out, however, that this relation is symmetrical. Therefore we assume that frequently occurring subsets in a population are good partial solutions that should be incorporated in successors.

Intuitively, frequently occurring subsets somehow managed to survive the selection process over a large number of epochs, and therefore must be responsible for a high evaluation.

## 3. A MARKOV CHAIN IMPLEMENTATION

At every time step, a histogram of all symbol transitions, occurring in all individuals of the population can be used to model the fitness of subsequent subsets (substrings). In the recombination step this histogram is used to generate offspring solutions by randomly adding new symbols to a solution according to their occurrence frequencies. Frequently occurring, successful substrings in a population are more likely to be generated again in the offspring and are therefore preserved. These substrings tend to grow over time as the optimal solution becomes clearer.

## 4. RESULTS AND CONCLUSION

The algorithm was tested against single-point and uniform crossover methods on the Traveling Salesman Problem and a path planning problem. From our empirical results we can conclude that our method has two major advantages: fitness improves faster in the first few iterations, and it converges to a better solution. This is particularly nice for an algorithm that is independent of the fitness function and has the same time complexity as the classical crossover techniques.

As the No Free Lunch theorem states, it is impossible to create a generally better performing optimizer [2]. Our approach makes a somewhat bigger assumption on composibility of partial solutions. This leads to faster convergence for the tested domains, but does not necessarily hold for other domains.

## 5. REFERENCES

[1] J. H. Holland. Adaptation in natural and artificial systems. 1975.
[2] D. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Trans. Evolutionary Computation*, 1(1):67–82, 1997.