Autonomous Selection in Evolutionary Algorithms

A. E. Eiben[1]
[1]Dept. of Computer Science,
Vrije Universiteit Amsterdam
The Netherlands

Marc Schoenauer[2]
[2]Projet TAO
INRIA Futurs
Orsay, France

D. W. F. van Krevelen[1], M. C. Hobbelman[1], M. A. ten Hagen[1], R. C. van het Schip[1]

Abstract: This work introduces *Autonomous selection* in EAs to escape the need for some central control during the selection phases of an EA. The results demonstrate that this is a viable idea that needs further investigation.

The main idea is to make the decisions about (de)selection on local level (by the individuals) in a decentralized manner (without global coordination), in such a way that individuals with above/below average fitness have a high/low probability of surviving and producing offspring. The proposed mechanism is based on 1) information about the population's average fitness available at each individual, 2) a function that determines (de)selection probabilities, based on the individual's own fitness and the population's average fitness. This study concentrates on the selection mechanism, and assumes that the average fitness is locally available to all individuals. Indeed, in P2P networks, it is possible to gather an approximation of such statistics without any central control. However, in order to study the selection mechanism in its pure form, it is assumed that an "oracle" provides individuals with the actual average.

The parental and survival probabilities of a given individual $\bar{x}$ are (sigmoid) functions of its fitness deviation from average $\Delta f(\bar{x})$, and depend on two parameters each, s_a and m_a (with a=s or f, for **s**urvival or **r**eproduction). The *shift s* determines where the transition from low to high probability takes place, and increasing s will increase the selection pressure. The *multiplier m* determines how sharp the transition from very low to very high probability is. More formally, the probability P_a that $\bar{x}$ survives or reproduces is

$$P_a(\bar{x}) = \frac{1}{1+e^{-m_a \cdot (\Delta f(x) - s_a)}}$$

Each individual in turn dies, or survives and reproduces, stochastically with the corresponding probabilities. The population size can hence greatly vary during evolution, as offspring are created locally without any information about global population size.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Autonomous Selection, Distributed EAs

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

Experiments and Results

The tested algorithms are SGA, a simple GA, and ASGA, the Autonomous Selection GA. Both use random initialization, $f(\bar{x}) = 1$ or 10,000 evaluations as termination condition, 2-points crossover, with probability 0.8 for SGA and 0.2 for ASGA, and bit-flip mutation with probability 0.01 for SGA and 0.05 for ASGA (set by preliminary experimentations). SGA uses 2-tournament, and ASGA the autonomous selection described above. The first test suite is made of three 100-bits artificial landscapes with 1, 10, and 100 binary peaks, whose heights are linearly distributed, and where the lowest peak is 0.5. The performance of different algorithms are classically compared using the Mean Best Fitness (MBF) and the Success Rate (SR) computed over 25 independent runs.

Shifting: A first series of experiments explores the combined effect of the shift parameters for survival and fertility, keeping the multipliers steady ($m_s = 1000$, $m_f = 60$), while the values for shift are varied: $s_s = \{$-0.08, -0.1, -0.12, -0.14, -0.16, -0.18, -0.2$\}$ for survival and $s_f = \{$0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16$\}$ for fertility. Conclusions of those experiments are

- More than half of the population must survive ($s_s < 0$) and less than half of the population must reproduce ($s_f > 0$).
- The success rate follows the MBF-plane only in the lower right corner, where less individuals survive and reproduce.
- Changing s_f has a larger effect on the MBF and the SR than changing s_s.

Surviving: A second series looks closer at survival selection and tests different m_s and s_s values: $s_s = \{$-0.08, -0.1, -0.12, -0.14, -0.16, -0.18, -0.2$\}$ and $m_s = 1, 10, 100, 1000, 10000$. Conclusions are

- The success rate drops when the shift for survival leaves the $-.14$ setting, i.e. whether the surviving part of the population is increased or decreased, while the population fitness drops only when less of the population survives.
- Decreasing the multiplier for survival selection below 100 severely affects population fitness and the algorithm is no longer successful.

SGA vs. ASGA: Finally, using the "optimized" settings $m_s = 1000$, $s_s = -.14$, $m_f = 60$, and $s_f = .12$, ASGA is validated against SGA on the same test suite plus the famous Ugly, Royal Road, and LongPath problems. On the multi-peaks landscapes (where ASGA parameters were tuned), ASGA equals or outperforms SGA. On the Long Path and the Royal Road problems, the differences are insignificant. On the ugly 30-bits problem, SGA wins by a large margin w.r.t. SR and by a small margin w.r.t. MBF.