Learning Offspring Optimizing Mate Selection

Ekaterina A. Holdener
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri, USA
holdener@acm.org

Daniel R. Tauritz
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri, USA
dtauritz@acm.org

ABSTRACT

This poster paper presents a methodology for removing user-defined parameters at the parent selection stage, by allowing all individuals in the population to self-organize into pairs of mates.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; G.1.6 [**Numerical Analysis**]: Optimization

General Terms

Algorithms

Keywords

Mate selection, parent selection, offspring optimization

1. MATE SELECTION

Traditionally, the parent selection stage of Genetic Algorithms (GAs) requires user-defined parameters, while mate pairing is done at random. We propose a new mate selection method for GAs, Learning Offspring Optimizing Mate Selection (LOOMS), which removes the need for user-defined parameters at the parent selection stage. LOOMS matches each individual in the population with a mate that the individual believes to be its best match in terms of offspring quality. In LOOMS, individuals learn which qualities (alleles) of their mates result in good offspring by observing the results of their own reproduction. The obtained knowledge is passed on to the offspring, so as the evolution continues the mate selection process becomes more refined.

At the parent selection stage of LOOMS, each individual in the population ranks all other individuals from most likable to least likable. Based on these rankings, an instance of the stable marriage problem [2] is created.

LOOMS does not assume that the individuals in the population are of the opposite sex. To create an instance of the stable marriage problem, each individual is considered twice: once as a male and once as a female. So if there are N individuals in the population, N couples are created using the Gale and Shapley algorithm [2]. Each individual is exactly in two couples (once as a male and once as a female). These couples are the selected mate pairings that go on to the reproduction stage. Note that the number of children produced by each couple is user defined, but at least N children are produced at each generation.

In addition to the bit string genotype, s_j, encoding individual j's trial solution to a problem, each individual contains a real-valued vector d_j encoding the desirable qualities of his potential mates. We refer to d_j as the *desiredFeatures* vector of individual j. The length of d_j equals the length of s_j, and each element i of d_j, $d_j[i]$, represents how much individual j wants the i^{th} bit of its potential mate to be 1. We define the Mate Ranking Function (MRF) of individual j on individual k to be:

$$MRF(j,k) = \sum_{i=1}^{L} d_j[i] \cdot (-1)^{(1-s_k[i])}, \tag{1}$$

where L is the length of the bit string solution and $s_k[i]$ is the i^{th} bit of k's trial solution. In other words, if the i^{th} bit of individual k is 1, the value of $d_j[i]$ is added to the total score, otherwise the value of $d_j[i]$ is subtracted from the total score. Individuals with higher MRF values are more likable than those with lower MRF values, so preference lists are composed by sorting the population by the MRF value. Note that LOOMS assumes fixed length solution representation. All elements of the *desiredFeaturs* vectors are initialized to uniformly distributed random values in the range between -1 and 1, but are not bounded to any range during the learning process.

When offspring are produced via crossover, the *desiredFeatures* vectors of the parents are crossed as well. If an offspring obtained the i^{th} bit of its trial solution from parent 1, then the i^{th} element of the offspring's *desiredFeatures* vector is copied from the i^{th} element of the *desiredFeatures* vector of parent 1 as well. This process ensures that the parents' knowledge about desirable mate qualities relevant to the offspring's genotype is passed on to the offspring.

The knowledge about desirable mate qualities is obtained by observing the outcome of reproduction. Each time an individual participates in reproduction, he updates his *desiredFeatures* vector depending on how the offspring's fitness compares to his own. The offspring also updates his *desiredFeatures* vector after comparing his fitness to his parents' fitness values. The detailed procedure of how these updates are performed can be found in Section 5.1.2 of [3].

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

Table 1: Parameters used in the experiments

parameter	value
replacement strategy	$(\mu + \lambda)$
μ for DTRAP with L=100	100
μ for DTRAP with L=500	500
t_c	2, 3, 4, 5, 6, 7
t_p for TGA	1, 2, 3, 4
λ	μ
recombination	one point crossover
mutation (when applied)	bit flip
bit flip mutation chance	$1/L$
number of runs	60

2. EXPERIMENTS AND RESULTS

We compare the performance of a GA with LOOMS to the performance of a traditional GA (TGA) with panmictic parent selection on instances of a fully deceptive trap problem (DTRAP) with traps of size four [1]. In a DTRAP instance, the bit string is split into n-bit traps, with $n = 4$, and the fitness of a trial solution is the sum of fitnesses of each trap. The fitness of each trap is a function of the number of ones in that trap. Let t be the number of ones in a 4-bit trap. Then the trap function used in the experiments was defined as:

$$f(t) = \begin{cases} 3 - t & \text{if } t < 4 \\ 4 & \text{if } t = 4 \end{cases}$$

These experiments used DTRAP problem instances of lengths $L = 100$ and $L = 500$.

The algorithms employed $(\mu + \lambda)$ replacement strategy with tournament selection of various tournament sizes, t_c, and two sets of reproductive operators: 1) one-point crossover and 2) one-point crossover followed by mutation. In TGA, parents were also selected with tournament selection of various tournament sizes, t_p. All parameters used to configure the GA with LOOMS and TGA are shown in Table 1. The performance of the algorithms was compared based on the Mean Best Fitness (MBF) statistic - best fitness found by each run of the algorithm, averaged over all runs. MBF was computed as the percentage of the best possible fitness for the given problem instance.

We statistically compared the performance of the two algorithms by applying the two-tailed t-test assuming unequal variances with $\alpha = 0.05$ to the best MBF values (out of all tested parameter sets) found by each algorithm with and without mutation on each problem instance. The best performance of each algorithm on each problem instance along with the parameters resulting in such performance and the results of the t-test are shown in Table 2. In three out of four cases, the GA with LOOMS outperformed TGA. Although in the case of DTRAP with $L = 100$ the average MBF of the GA with LOOMS was higher than that of TGA, there was no statistically significant difference in the performance of the two algorithms.

In all four cases ($L = 100$ with and without mutation and $L = 500$ with and without mutation), TGA achieved its best performance when parent selection tournament size was 1, which is equivalent to random parent selection. This lack of parent selection pressure was compensated by the competition selection pressure. The GA with LOOMS also lacks parent selection pressure, since the mate selection algorithm

Table 2: Best results of each algorithm on each problem instance

reproduction	problem instance	measure	TGA	GA with LOOMS
no mutation	$L = 100$	params.	$t_p = 1$, $t_c = 6$	$t_c = 6$
		MBF	84.48	87.08
		st. dev	3.29	2.62
	$L = 500$	params.	$t_p = 1$, $t_c = 6$	$t_c = 3$
		MBF	79.68	82.63
		st. dev	2.03	2.02
with mutation	$L = 100$	params.	t_p=1, $t_c = 7$	$t_c = 6$
		MBF	91.60	91.67
		st. dev	2.16	2.12
	$L = 500$	params.	$t_p = 1$, $t_c = 7$	$t_c = 6$
		MBF	88.33	89.69
		st. dev	0.99	1.38

does not consider individuals' fitnesses, and relies on the selection pressure at the competition stage. However, mate selection in the GA with LOOMS is not completely random, as all individuals learn which mate qualities result in high fitness offspring. The results of the experiments show that LOOMS helps evolve better or equal quality final solutions while removing all parent selection parameters, as compared to the final solutions evolved by TGA.

3. CONCLUSIONS

LOOMS eliminates the need for manual tuning of parameters for the parent selection stage of GAs, but relies on the selection pressure at the competition stage. Preliminary experimental comparison of GA with LOOMS to TGA on DTRAP instances showed that there is no performance sacrifice associated with the removal of parent selection related parameters.

One drawback of the GA with LOOMS is the computational overhead of the parent selection stage. Since all individuals have to rank all other individuals in the population, the parent selection stage can significantly slow down the algorithm. Further evaluation of ELOOMS on DTRAP as well as on other problem classes will determine the impact of this method more accurately.

4. REFERENCES

[1] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. In *Proceedings of FOGA II: the Second Workshop on Foundations of Genetic Algorithms*, pages 93–108, Vail, Colorado, USA, July 1992. Morgan Kaufmann Publishers Inc.

[2] D. Gale and L. Shapley. College admissions and the stability of marriage. *The Amer. Math. Monthly*, 69:9–15, 1962.

[3] E. Holdener. *The art of parameterless evolutionary algorithms*. PhD thesis, Missouri University of Science and Technology, Rolla, MO, USA, 2008.