Explicit Speciation with few *a priori* Parameters for Dynamic Optimization Problems

Christopher Ronnewinkel
Institute for Neuro- and Bioinformatics
Medical University Lübeck
Seelandstraße 1a, 23569 Lübeck
ronne@inb.mu-luebeck.de

Thomas Martinetz
Institute for Neuro- and Bioinformatics
Medical University Lübeck
Seelandstraße 1a, 23569 Lübeck
martinetz@inb.mu-luebeck.de

Abstract

We present a multipopulation concept that detaches global from local search. A search population constantly searches for new regions of high fitness, whereas colonies track already found fitness peaks. The colonies interact via a *parameterless* speciation criterion. In a performance test on multiple stochastically moving fitness peaks on the hypercube, we show that our approach outperforms tag sharing, deterministic crowding, restricted tournament selection, and multiniche crowding. It is argued, that for dynamic environments explicit speciation is generally advantageous because it enables the use of arbitrary (here generational) evolutionary operators for local search.

1 Introduction

Evolutionary algorithms (EAs) have been successfully applied to a large number of real-world problems. In many cases, changes of the environment lead to effectively dynamic optimization problems. In order to be prepared for certain changes in the environment, it is favorable to track several suboptimal solutions in parallel, as each might become globally optimal in the future. In an EA, such behavior can be achieved by *speciation*. Recently, approaches to dynamic optimization that implement *explicit* speciation by *self-organized* formation of autonomous subpopulations have been proposed by Branke and Schmeck (2000) and Ursem (2000).

Contrasting to these concepts, there exist *implicit* speciation methods that accomplish speciation within a single population. In the following, we will consider tag sharing without tag mutation (TS) (Liles and DeJong 1999), deterministic crowding (DC) (Mahfoud 1995), multi-niche crowding (MNC) (Cedeno and Vemuri 1997) and restricted tournament selection (RTS) (Harik 1995). DC without crossover (HC) essentially performs parallel hillclimbing (Mahfoud 1995) using a mutation rate instead of single bitflips. From the principles on which they are build, these methods are capable of tracking suboptimal solutions. Hence, the question arises whether the additional organizational structure necessary in multipopulation approaches to achieve explicit speciation *de facto* leads to an advantageous optimization performance compared to implicit methods. The experimental results we present at the end will shed some light on this question. In the following, we first introduce the speciation criterion and present the used multipopulation framework. In conclusion, we report on the experiments with stochastically moving peaks on the boolean hypercube and give an interpretation to our results.

2 Speciation Criterion

During an evolutionary run, selection drives individuals towards regions of high fitness and the population becomes implicitly clustered in the presence of multiple peaks. However, for standard selection methods genetic drift renders these clusters highly unstable. DC, RTS and MNC fight genetic drift by restricting replacements of individuals only to similar individuals. TS stabilizes the existence of different tag groups by fitness sharing. In order to split up the population without such replacement restrictions or fitness sharing, one needs to detect the presence of fitness induced clusters as early as possible. A standard cluster analysis using only the distance structure of the population might be misleading for small population sizes and broad peaks. Therefore, Ursem (2000) uses additional fitness evaluations to detect fitness valleys between individuals which imply the presence of more than one fitness peak. We favor a more approximate, but with regard to fitness evaluations more efficient way, which is to test the *fitness envelope* (a definition is given in Pètrowski and Genet (1999)) for monotony. For real-valued encoding, Pètrowski and Genet (1999) have shown that this criterion leads in combination with k-means clustering to an efficient explicit speciation method that outperforms fitness sharing and DC on static multimodal function optimization problems.

In general search spaces, the envelope criterion can still be used, but reasonable *centroids* need to be chosen for clustering. For this, we added an index list which keeps for each distance interval the individual that set the envelope for that interval. Figure 1 shows an example. If the envelope is not monotonous, we choose the fittest individual c_0 and the next fit individual c_1 that is separated by an envelope valley as centroids for splitting. All other individuals are assigned to the closest centroid c_0 or c_1. Due to its low computational cost (linear in population size), the envelope test can be done after each selection, which allows to detect fitness induced clusters before they are destroyed by genetic drift. This procedure proved to work very reliable in our experiments on the boolean hypercube. As it does not use any representation specific mathematical operations (e.g. real-valued encoding: arithmetic averaging) it is applicable to arbitrary search spaces, as long as a metric can be defined. In addition, it is parameterless. Let us now shortly review our multi-population framework.

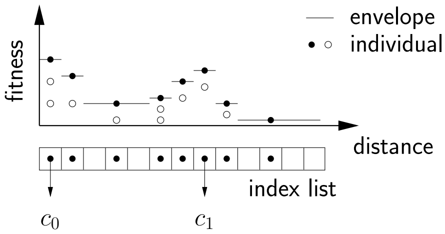

Figure 1: The population is plotted into a fitness-distance diagram. The corresponding envelope is shown as line. In the depicted case, the envelope is not monotonous and c_0 and c_1 will be chosen as centroids to split up into two subpopulations.

3 Multipopulation Concept

We distinguish between colonies, which are autonomous populations tracking single fitness peaks, and a search population, which explores the common complement of all colonies. The fittest individual of a colony is called its *centroid*. The search population has fixed size and its individuals are not allowed to

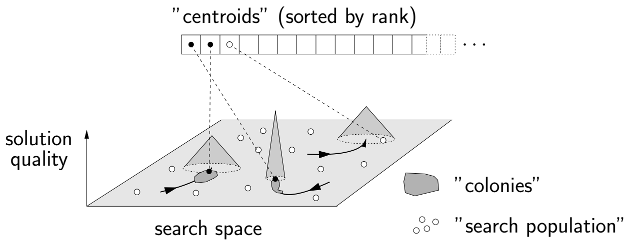

Figure 2: General concept for a division of global and local search by means of explicit speciation. Evolutionary operators can be chosen individually on both levels. Colonies interact via a speciation criterion and via a deterministic regulation of population sizes based on the centroids' properties.

be closer to a centroid than the maximal distance between that centroid and the corresponding colony individuals. As a result, it is constantly searched for new, promising centroids, which are injected in regular intervals into the centroid list (see below). In addition, new colonies can emerge from a division of an existing colony into parts depending on the above speciation criterion. To check, whether two colonies should be merged, the speciation criterion is applied to the union of both colonies. If the envelope of the union is monotonous, the colonies are merged and the worse centroid is removed from the centroid list. Since the colonies are explicit populations, their sizes can be regulated and stabilized in deterministic fashion. We distribute individuals only over the k_c best centroids and within these proportional to the centroids' fitness. If the centroid list is made longer than k_c, centroids with rank $\geq k_c$ remain in the list and are reevaluated. A new centroid from the search population always replaces the worst centroid in the list (see above). If a centroid's fitness increases, it moves forward in the ranked list and will obtain a finite colony size, i.e. found a new colony, if its rank becomes $< k_c$. Thereby, the centroid list works as a *memory*. In Figure 2, the concept is schematically depicted. Due to the strict division into search population and colonies, different evolutionary operators can be chosen on local and global level. For example, it has been very successful, to use a diversity preserving selection (e.g. DC) for the search population, because its aim is to spread out and quickly find new local fitness optima. On the local level, each colony can adapt its own strategy for tracking the corresponding fitness peak.

There are similarities between the previously proposed multipopulation approaches and the above concept. However, we want to stress neither having introduced additional algorithm parameters like diameters, nor using additional fitness evaluations for speciation in our approach.

4 Experiments

In order to test the capabilities of the above described multipopulation concept (MPGA), we performed experiments on the boolean hypercube. As benchmark problem, we have chosen the analogue to the real-valued moving peaks benchmark (MPB) (Branke and Schmeck 2000). In particular, we defined the fitness of an individual x as

$$f(x) = \max_{i=1,\ldots,n_p} \left\{ 1, 1 + (h_i - 1)^{\frac{w_i - d(x,p_i)}{w_i}} \right\}, \qquad (1)$$

where p_i denotes the position, h_i the height, w_i the width of peak i, and $d(x,y)$ the Hamming distance. For each peak, the fitness decreases linearly with Hamming distance. Motions of peaks are generated by applying a jumprule every τ evaluations to all peak positions p_i. Furthermore, the heights h_i and widths w_i are changed like in the original MPB. Note, that changes are synchronized with evaluations, *not* with population generations.

Table 1: Benchmark parameters: τ: change frequency, h_0, w_0: initial height and width, $h_<, h_>, w_<, w_>$: height and width boundaries, $\delta h, \delta w$: height and width severity.

τ	h_0	$h_<$	$h_>$	δh	w_0	$w_<$	$w_>$	δw
5000	50	30	70	7	8	6	10	1

Table 2: Algorithm parameters: n_p: number of peaks, m: total population size, μ, χ: mutation, crossover (2pt) rate.

	n_p	m	μ	χ
MPGA search	5/50	15	0.050	0.7
MPGA colonies	5/50	80	0.008	0.7
HC/DC	5	15	0.030	0.0/0.2
HC/DC	50	25	0.020	0.0/0.2
RTS	5/50	35/60	0.020	0.2
MNC	5/50	100/200	0.020	0.7
TS	5/50	200	0.008	0.7

4.1 Tracking, Crossing, Memory

Experiments have shown that the MPGA solves problems like crossing of two peaks, tracking of independently moving peaks, or memorization of periodic motions very efficiently (due to space limitations we will not present the details here).

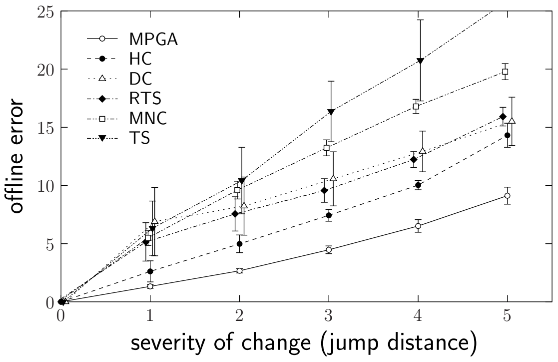

Figure 3: Offline error for a 5 peaks problem (averaged over 400 changes and 20 runs).

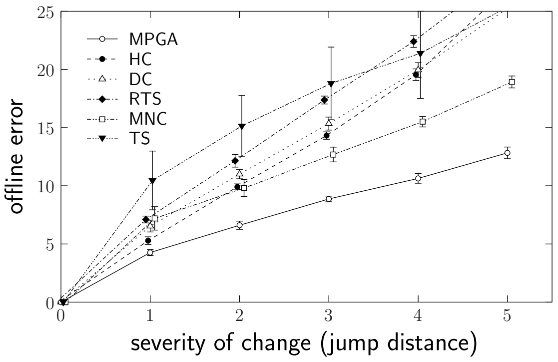

Figure 4: Offline error for a 50 peaks problem (averaged over 400 changes and 20 runs).

4.2 Performance on Moving Peaks

In order to study the performance of the above methods on more difficult problems, we applied the algorithms to problems with multiple, stochastically moving peaks. We used bitstrings of length $l = 40$. The selection method in MPGA colonies was Whitley's linear ranking with $\eta = 2$ and $k_c = 10/20$, for 5/50 peaks, respectively. For RTS, the full population was chosen as window. For TS, we chose the number of tag bits $n_t = 5$. For MNC, the involved crowding factors were chosen as $(C_s, C_f, s) = (10, 5, 20)$. At the beginning of each run, n_p peak positions were generated at random and initialized with the parameters of Tab. 1. The jumprule was: flip d randomly chosen bits. In Figs. 3 and 4, the offline error (Branke and Schmeck 2000) for the cases of 5 and 50 peaks are shown. The algorithm parameters (Tab. 2) have been fine tuned to a jump distance $d = 1$. For larger distances the setting was kept fixed to show the sensitivity of the algorithms performance to changes in the jump severity.

In the case of 5 peaks, MPGA was able to cover all peaks with colonies, and hence performed very well.

In the fine tuned case $d = 1$, HC's offline error is approximately twice as large as MPGA's offline error. Unrestricted crossover is disruptive for the hillclimbing process such that DC and RTS perform worse than HC. MNC and TS perform in the fine tuned case comparable to RTS and DC, but they are more sensitive against changes of the jump distance. In the case of 50 peaks, MPGA covers ≈ 10 peaks for a total number of colony individuals $m = 80$. Too small colonies fail to successfully track their peak and loose their individuals via size regulation due to inferior fitness. Experiments with larger population sizes (and slower dynamics) confirm, that the peak coverage has a strong impact on the offline error. Thus, MPGA does not perform much better than HC in this case. However, for larger jump distances MPGA's performance remains quite robust even with unchanged parameters. MNC shows similar behavior. All other methods drift off more quickly to higher offline errors. For TS, sometimes the same peak is covered by more than one tag which might explain the worse performance already for 5 peaks.

Coming back to the question raised in the beginning, we can state that MPGA *does* show advantageous performance compared to the considered implicit methods, in particular HC. How can that be on a landscape of fitness hills? The implicit methods HC, DC, and RTS need to perform tournaments between parents and kids. In static environments only the kids' fitnesses need to be evaluated, whereas in dynamic environments, the parents need to be reevaluated as well. This leads from the viewpoint of generations to twice as fast changes. Contrasting to that, no tournaments are needed when the population is explicitly split up into autonomous colonies. MNC circumvents reevaluations by replacement of the "worst (old fitness) among similar", but is still outperformed by MPGA. In conclusion, a major advantage of explicit speciation seems to be the possibility to apply arbitrary (here generational) evolutionary operators for local search, which is problematic in implicit speciation approaches due to the complex coupling of local and global search.

5 Conclusions

We have presented a multipopulation concept (MPGA) for explicit speciation using a parameterless fitness envelope criterion. The MPGA is able to efficiently solve problems with crossing fitness peaks and is able to memorize periodic movements of fitness peaks.

The overall performance of MPGA is coupled to the fraction of fitness peaks covered by colonies. However, in a performance comparison between implicit speciation methods and MPGA on MPB problems, MPGA outperforms the considered implicit methods. The main reasons are an increased number of evaluations (HC, DC, RTS) in dynamic environments and the lack of a capable interaction between species (TS). The outperformance of MPGA supplies experimental evidence that *explicit* speciation is useful for solving *dynamic* optimization problems.

The development of methods that use the statistics of centroid properties for an adaptation of each colony's local exploration to some given dynamics is a very promising and important topic for future research. In this context, a better understanding of real-world fitness landscapes within the moving peaks picture would be very helpful.

Acknowledgments

This work is supported by BMBF grant No. 01-IB-802-A9 (LEONET).

References

Branke, J. and H. Schmeck (2000). Designing evolutionary algorithms for dynamic optimization problems. In S. Tsutsui and A. Ghosh (Eds.), *Theory and Application of Evolutionary Computation: Recent Trends.* in press. Springer, UK.

Cedeno, W. and V. R. Vemuri (1997). On the use of niching for dynamic landscapes. In *International Conference on Evolutionary Computation*, pp. 361–366. IEEE Piscataway, NJ.

Harik, G. (1995). Finding multimodal solutions using restricted tournament selection. In *Proceedings 6th International Conference on Genetic Algorithms*, pp. 24–31. Morgan Kaufmann, San Mateo, CA.

Liles, W. and K. DeJong (1999). The usefulness of tag bits in changing environments. In *Proceedings of CEC 1999*, pp. 2054–2060. IEEE Piscataway, NJ.

Mahfoud, S. W. (1995). Niching methods for genetic algorithms. IlliGAL Report 95001, Illinois Genetic Algorithm Laboratory.

Pètrowski, A. and M. G. Genet (1999). A classification tree for speciation. In *Proceedings of CEC 1999*, pp. 204–211. IEEE Piscataway, NJ.

Ursem, R. K. (2000). Multinational GAs: Multimodal optimization techniques in dynamic environments. In *Proceedings of GECCO 2000*, pp. 19–26. Morgan Kauffman, San Francisco, CA.