On Repair by Binary Interpolation: A Genetic Operator Having Offspring at a Constraint Transition*

Steven Orla Kimbrough
University of Pennsylvania
kimbrough@wharton.upenn.edu

David Harlan Wood
University of Delaware
wood@cis.udel.edu

Categories and Subject Descriptors: I.2.8 [Problem Solving, Control Methods, and Search]: Heuristic Methods

General Terms: Algorithms

Keywords: binary interpolation, boundary searching, genetic operator

1. CONSTRAINTS AND INFEASIBILITY

One ongoing theme of our research [*e.g.* PPSN VIII, Yao,ed. LNCS 3242, pp. 292–301, Springer-Verlag, 2004.] is the value of infeasible solutions of constrained optimization problems. Infeasible solutions violate the constraints but they can nevertheless contain valuable genetic information. Particularly valuable ones might be only a single mutation away from feasibility, or even optimality.

Consider also that an optimal solution is likewise only a single mutation away from infeasibility provided that the constraints are relevant (which we assume). The guiding principle is that for *constrained* optimization problems, evolutionary computation methods should pay particular attention to the region near the feasible/infeasible boundary. A desire for boundary exploration led us to create the Feasible-Infeasible Two Population Genetic Algorithm (FI-2PopGA). This model alternates between (1) evolving feasible solutions solely to *improve* them by seeking better objective function values and (2) evolving infeasible solutions solely to *repair* them by lessening their constraint violations.

Let us give a general description of the observed behavior of the FI-2PopGA (*op. cit.*). A first order effect is that the infeasible population seeks out the boundary indiscriminately (exploration), and the feasible population seeks out the optima which will be located on or near the boundary (exploitation). Upon reaching the region of the boundary, a population can produce mutant children that end up in the other population where they are subjected to a different competitive criterion. A very interesting second order effect is obtained when mutant grandchildren end up reentering the population of their grandparents. This means that their genetic material has survived competitions involving two criteria: both approaching optima and also diminishing constraint violations. This makes both populations gradually become genetically similar and cluster on opposite sides of the boundary near optima. Of course, both boundary searching and dual population evolution have been studied before (see references in *op. cit.*). Our approach is one of the few that avoid attempting to heuristically evaluate candidate solutions by two unrelated criteria at the same time.

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

2. INTERPOLATION TO THE BOUNDARY

Conspicuously missing from the FI-2PopGA is any interbreeding *between* the two populations. After all, with n variables, why not just interpolate between a feasible parent and an infeasible in n-dimensional space to reach the boundary? This could produce two children with nearly identical genetic material, but with one feasible and the other one infeasible. This could replace some or all of the more leisurely and explorative evolution to the boundary and the waiting for chance mutants to cross the boundary. The highly exploitive nature of interpolation could be eased by interpolating along selected continuous nonlinear curves rather than straight lines.

However, we are most concerned with binary (yes-no) variables, where it is less clear how interpolation might be carried out. This brings us, at last, to the subject of this note. We find ourselves in a Hamming space where the points are vectors of bits (zeros and ones), normally referred to as bitstrings. If we have a master list of all possible bitstrings of length n, than we can interpolate to the boundary in the following way. Suppose we have two parent bitstrings, one feasible and the other one infeasible. Find where these two bitstrings are located on the master list. Probabilistically pick out a bitstring roughly halfway between the two parents. Test whether this bitstring is feasible or not. In either case, we now have determined another feasible/infeasible pair of parent bitstrings that are closer together on the list than the original parents. Repeat this shrinking until we obtain a feasible/infeasible pair that are adjacent on the list. Since each feasibility test cuts the sublist approximately in half, about n tests, at worst, are required to obtain adjacency.

But two adjacent children may not be genetically similar. For example, if the master list is just the binary form of the numbers from 0 to 63, interpolation might produce the adjacent but genetically very different bitstrings $[001111]$ and $[010000]$. So, we require the master list to have the following property: adjacent bitstrings must be on the boundary, i.e. differ by a single bit. That may seem like asking for a lot, but there there are many such lists; they are known as a Gray codes. Actually, nobody knows how many Gray codes there are. So we can ask for more. Balanced Gray codes, for example, may be useful to us; in these any one bit position has an equal (as possible) probability of changing as one goes through the list.

There are various circumstances where binary interpolation can be used. Initial populations might be generated by binary interpolation between a known feasible/infeasible pair such as the all-zero bit string and the all-ones bitstring. Interpolating parents that are on the boundary but genetically dissimilar would generically explore *along* the boundary. In problems where it is quite difficult to find solutions that are feasible, interpolation may be a valuable source of new feasible solutions. And, if we interpolate between parents that are close on the Gray code list, and the feasible parent is near an optimum, the children will also be near the optimum.