Introducing Distance Tracing of Evolutionary Dynamics in a Feasible-Infeasible Two-Population (FI-2Pop) Genetic Algorithm for Constrained Optimization

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

Ming Lu
University of Pennsylvania
milu@wharton.upenn.edu

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

July 10, 2004

Abstract

We explore data-driven methods for gaining insight into the dynamics of the FI-2Pop GA (explained below), which has been effective for constrained optimization problems. We track and compare one population of feasible solutions and another population of infeasible solutions. Feasible solutions are selected and bred to improve their objective function values. Infeasible solutions are selected and bred to reduce their constraint violations. Interbreeding between populations is completely indirect, that is, only through their offspring that happen to migrate to the other population. We introduce an empirical measure of distances between individuals and population centroids to monitor the progress of evolution. We find that the centroids of the two populations approach each other and stabilize. This is a valuable characterization of convergence. We find the infeasible population influences, and sometimes dominates, the genetic material of the optimum solution. Since the infeasible population is not evaluated by the objective function, it is free to explore boundary regions, where the optimum may be found.

Contents

This paper introduces and uses a particular empirical technique for gaining insight into the operation and dynamics of genetic algorithms (GAs) for constrained optimization. The technique applies to GAs and more generally to evolution programs (EPs).[1] Given the generally limited analytic results for understanding GAs (and EPs), and the No Free Lunch theorems [25, 28], empirical investigations of these heuristics are desirable, even unavoidable, if we are to understand them better. To that end, we have sought data-driven methods for communicating insight into the evolutionary dynamics of the FI-2Pop GA, a variety of GA (explained below) that has proven effective for constrained optimization problems. We introduce and report here on results for one such empirical method, involving distances between individuals and population centroids arising during a run of the GA. Our discussion hardly exhausts the scope of our distance-based method, and that method is only one of very many methods for empirically studying EPs. Even so, insights or at least intriguing hypotheses are produced. The contribution here is by its nature something of a case study, yet it adds to what must be a long-term piling up of scientific evidence on how GAs work.

1 Context and Background

Genetic algorithms (GAs) meliorize. Like all heuristics, including the more general category of evolution programs (EPs), they seek better and better decisions in the face of a given objective. In doing so, they offer neither a practical guarantee that an optimal decision will be found, nor an ability to recognize one if it were found. These properties are disadvantages, compared to many optimization procedures established in the Operations Research (OR) literature. On the other hand, when problems are nonlinear and have mixtures of integers and reals in the decision variables, these disadvantages also attend the traditional OR solvers. Even if they did not, there remains the question of what value GAs (and EPs) bring to the table as a practical matter in solving difficult optimization problems. If a GA can get a better decision, even if it takes longer, or if a GA can get an equally good decision faster, even if this exploits parallel computing, or if a GA can return a slightly poorer decision but provide valuable decision making information not otherwise readily available, then there is a case to be made for their deployment and use.

In all these ways, GAs are promising, are contenders. There is, however, a fundamental impediment to their deployment and use for problems of *constrained* optimization. The difficulty is that the genetic operations—including mutation and crossover—are not guaranteed to preserve feasibility. A single mutation in a highly-fit individual can, and often in practice will, result in an infeasible decision. Similarly, offspring produced by crossover from two highly-fit parents need not be, and often are not, feasible. We give, in §2, a brief introduction to constraint handling in GAs. Further elaboration would divert us from the main purposes of this paper (but see references cited). In a nutshell, although the problem of constraint handling is a severe one for GAs applied to constrained optimization problems and although much attention has been given to this problem, no generally accepted solution has emerged.

Recent work, however, has shown considerable promise and prowess for a feasible-infeasible two-population (FI-2Pop) genetic algorithm for constrained optimization [14, 17, 15, 16]. The FI-2Pop GA has beaten standard methods for handling constraints in GAs [15]; has regularly produced

[1] Roughly, any metaheuristic that is population-based.

better decisions for comparable computational effort than GENOCOP, a high-quality GA solver engine for constrained optimization problems [16, 17]; has produced excellent decisions for problems that cannot be handled by GENOCOP [14, 17], and has produced better decisions than GAMS solvers ([14] for a case that GAMS could only solve a relaxed version of; and personal communication for a case in which GAMS does produce a good decision, but the FI-2Pop GA does better).

Given these (and other) results, it is fair to conclude that the FI-2Pop GA has something going for it on constrained optimization problems and definitely merits further investigation. Piling up results from new, wider, more difficult tests is, of course, necessary and always welcome. Complementing this is the need to understand, to have insight into, when and why the FI-2Pop GA should work well and indeed to understand how it works. This latter goal, of understanding the algorithm better, is the focus of this paper.

* * *

The remainder of the paper is organized as follows. In §2 we introduce and briefly review the topic of constraint handling in GAs. §3 presents the feasible-infeasible two-population (FI-2Pop) GA and discusses attendant concepts and intuitions. §4 describes Yuan, a small but challenging nonlinear, mixed integer test problem for constrained optimization.[2] GAMS solves Yuan to (apparent) optimality and so does the FI-2Pop GA. GENOCOP, however, cannot handle Yuan. In §5.1 we present our distance mapping technique, appropriated from multivariate statistics, and in §5.2 we discuss results of one representative run of the Yuan problem with the FI-2Pop GA. Finally, we conclude with a discussion in §6.

2 Constraint Handling in Genetic Algorithms

Considerable attention has been paid to representing constraints (see [1, 3, 7, 9, 20, 21, 22, 23, 25, 26, 27] for excellent reviews and treatments; there is even a dedicated Web site [8]), but no consensus approach has emerged. A natural—and the most often used—approach is to penalize the candidate in proportion to the size of its constraint violations. A feasible solution is not penalized; an infeasible solution is penalized as a function of the magnitude of the violation(s) of the constraint(s). Putting this more formally, here is the general form of a maximization constrained optimization problem.

$$\max_{x_i} Z(\vec{x}), \textit{subject to } E(\vec{x}) \geq \vec{a}, F(\vec{x}) \leq \vec{b}, G(\vec{x}) = \vec{c}, x_i \in \mathcal{S}_i \tag{1}$$

Here, $Z(\vec{x})$ is the objective function value produced by the candidate solution $\vec{x}$.[3] E, F and G each yield zero or more constraint inequalities or equalities. Taken as functions, in the general case Z, E, F, G can be any functions at all (on $\vec{x}$) and in particular need not be linear. $\mathcal{S}_i$ is the set of permitted values for the x_i (the components of the vector $\vec{x}$), and are called the *decision variables*

[2]The Yuan problem appears in a book of challenge problems for nonlinear constrained optimization [10, page 266]. Subsequent to collecting the data reported here, we discovered that Yuan is well-solved by a GA with a 'death penalty' regime for infeasible solutions. Apparently, and surprisingly, Yuan is a GA-easy problem. This does not invalidate the findings reported here, which patterns we observe in many other problems.

[3]By *decision* or *candidate solution* or just *solution* we mean any instance of $\vec{x}$. The members of $\vec{x}$ are called the *decision variables* for the problem; any solution method seeks to find optimal values for these variables. Some candidate solutions are feasible and some not. An optimal solution is a candidate solution, which is feasible and for which no feasible candidate solution yields a better value of $Z(\vec{x})$.

for the problem. $\mathcal{S}_i$ may include reals, integers, or mixtures. Problems of the form of expression (1) are not directly translatable into the linear encodings normally used for EP (including GA) solutions. The purpose of a penalty function formulation is to produce a representation of the problem that can be directly and naturally encoded as a, EP/GA. To indicate a penalty function representation, let $\vec{x}$ be a (candidate) solution to a maximization constrained optimization problem. Its absolute fitness, $W(\vec{x})$, in the presence of penalties for constraint violation is $Z(\vec{x}) - P(\vec{x})$ and the COP is transformed to:

$$\max_{x_i} W(\vec{x}) = Z(\vec{x}) - P(\vec{x}) \tag{2}$$

where $P(\vec{x})$ is the total penalty (if any) associated with constraint violations by $\vec{x}$. Problems representable as in expression (2) are directly and naturally encoded as EPs. If an EP finds a solution $\vec{x}$ that is feasible ($P(\vec{x}) = 0$) and has a high value for $W(\vec{x})$, then we may congratulate ourselves on the successful application of this EP metaheuristic.

Typically, and by design, the penalty imposed on an infeasible solution will severely reduce the net fitness of the solution in question, leading to quick elimination of the solution from the EP population. This may be undesirable and it may be responsible for the generally recognized weak performance of EPs for constrained optimization problems.

3 The Feasible-Infeasible Two-Population GA and Intuitions on Its Workings

The essentials of the Feasible-Infeasible Two-Population (FI-2Pop) genetic algorithm for constrained optimization are easily described. Two populations are created at initialization and maintained throughout the run of the GA. The feasible population contains only feasible decisions (aka: solutions) and the infeasible population contains only infeasible decisions. When individuals are created, either at initialization or as a result of applying the genetic operators (we use mutation and crossover), they are tested for feasibility and placed in the appropriate population. Individuals in the feasible population are evaluated on fitness with respect only to their objective function values. Individuals in the infeasible population are evaluated on fitness with respect only to the sum of their constraint violations. Otherwise, within each population we operate a rather standard GA. Our implementation was designed for comparison with GENOCOP [24], a standard and excellent GA solver for constrained optimization problems. Details of our algorithm are given in Appendix A.[4]

The FI-2Pop GA was conceived as a principled response to the known limitations of using penalty functions on constraints (described above), and to the received wisdom that "The ideal way of

[4]We note that there are other varieties of two-population GAs. SGGA maintains two violation-penalized populations, using different penalty functions, and crossbreeds between them [18, 19]. GENOCOP III is based on repair and maintains two populations; both are feasible throughout. See [20] for a review of both systems. Chu & Beasley [5, 6] have explored a single-population GA in which each solution receives two fitness scores, one on the objective function (or 'fitness'), one on infeasibility (or 'unfitness'). Parents are selected for breeding based only their 'fitness' values; individuals are removed from the population in order of their 'unfitness' values. Yuchi & Kim [30] report success with a two-population scheme in which a portion of the infeasible solutions are probabilistically accepted for breeding each generation, based on their objective function values. Systematic comparison of these approaches must await future research.

handling constraints is to have the search space limited to the feasible space" [20]. We wish to challenge that wisdom. Penalty functions (on constraint violations) are problematic for a number of reasons. First, and most obviously, there is no known principled way to set the size of the penalties. Completely eliminating infeasible decisions is generally recognized to produce poor performance. Very high penalties are tantamount to eliminating infeasible decisions. Very low penalties lead to misdirecting the GA's search by seeding the population with infeasible decisions that score well on the objective function. A middling degree of penalization would seem to be called for, but on what scale is it to be measured? No principled answer has been forthcoming and penalty-function approaches are generally thought to perform less well than systems such as GENOCOP, which use repair and other devices.

Second, and more fundamentally, penalty function approaches conflate objective function evaluation and constraint violation evaluation. If these could be separated, arguably or at least intuitively, with specialization (on one or the other) it would be possible to do better at each. The FI-2Pop GA can be seen as achieving this, at least approximately. Feasible decisions are evaluated only with regard to the objective function, infeasible only with regard to the constraints. In a penalized regime, an infeasible decision is subjected to selection with regard to both its objective value and its constraints, thereby reducing the effect of selection for feasibility.

Third, optimal solutions to constrained problems are either in the interior of the feasible region or they are on or near the boundary of the feasible region. Selection on the infeasible population in the FI-2Pop GA will drive the population to, and eventually over, the boundary. If the optimal solution is on or near the boundary, this represents an additional source of exploration. The infeasible population will tend to probe the boundary of the feasible region.[5] Feasible children of infeasible parents will resemble them (as all children do), and hence be near by. Because the infeasible parents were selected without regard to their objective functions, mutation and crossover will tend to create feasible solutions that presumably are somewhat different than those already in the main population. These solutions either succeed or fail; in either case they contribute to an exploration of the feasible region. Thus, if the optimal solution is on or near the boundary, one would expect that the infeasible population would contribute to finding it. On the other hand, if the optimal solution is in the interior, away from the boundary, the infeasible population may not be useful, but the result is merely a slower, but otherwise unimpeded GA on the feasible population. The quality of the decisions found should not be affected.

In sum, by separating objective function evaluation and feasibility evaluation the FI-2Pop GA circumvents the problem of misdirecting the genetic search by infelicitous compromise. Further, the feasible population serves as a reservoir of genetic information that probes the boundary region of the feasible region, and contributes variation to the feasible population. And variation is the meat of selection.

We admit these are intuitive arguments, however plausible at some level. But they are testable. Let us see if there are data that corroborate them. We now turn to a discussion of our case study.

[5] Or boundaries. Nothing in this argument requires a connected boundary.

4 Problem: Yuan

Yuan, the model for our case study, is discussed in [10, page 266] and was originated in [29]. The model is nonlinear (quadratic and logarithmic) in the objective function and quadratic in the constraints. Moreover, it is a mixed-integer model, with three continuous variables and four binary variables. Genocop III is unable to handle models of this sort [24]. Yuan's formulation is as follows.

Objective function:

$$\begin{aligned}\min_{\mathbf{x},\mathbf{y}} z = \quad & (y_1 - 1)^2 + (y_2 - 2)^2 + (y_3 - 1)^2 \\ & - \ln(y_4 + 1) + (x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 \end{aligned} \tag{3}$$

Constraints:

$$\begin{aligned} y_1 + y_2 + y_3 + x_1 + x_2 + x_3 &\leq 5 && (4)\\ y_3^2 + x_1^2 + x_2^2 + x_3^2 &\leq 5.5 && (5)\\ y_1 + x_1 &\leq 1.2 && (6)\\ y_2 + x_2 &\leq 1.8 && (7)\\ y_3 + x_3 &\leq 2.5 && (8)\\ y_4 + x_1 &\leq 1.2 && (9)\\ y_2^2 + x_2^2 &\leq 1.64 && (10)\\ y_3^2 + x_3^2 &\leq 4.25 && (11)\\ y_2^2 + x_3^2 &\leq 4.64 && (12) \end{aligned}$$

$$\begin{aligned} x_1, x_2, x_3 &\geq 0 && (13)\\ y_1, y_2, y_3, y_4 &\in \{0, 1\} && (14) \end{aligned}$$

As reported by [10, page 266] and [29], at optimality the value of the objective function is $z^* = 4.5796$, with the decision variables set at $\mathbf{x}^* = (0.2, 0.8, 1.908)^T$ and $\mathbf{y}^* = (1, 1, 0, 1)^T$.

We describe now results from one quite typical run of the FI-2Pop GA on the Yuan problem.[6] At the end of 5,000 generations of alternating feasible and infeasible genetic exploration (equivalent to 10,000 generations in an ordinary GA), z^+, the objective function value of the best solution found, is 4.579588292413069. The variable settings in this solution are $\mathbf{y}^T = (1, 1, 0, 1)$, $x_1 = 0.199998178908325$, $x_2 = 0.799999776184869$, $x_3 = 1.90787728616851$.

Table 1 presents selected summary data from the feasible population, and Table 2 presents corresponding data from the infeasible population. Table 3 is a statistical summary, comparing the feasible and infeasible populations during this (typical) run.

[6] The FI-2Pop GA quite reliably solves Yuan, at least to a close approximation, regardless of random seed.

Gen	avg(z)	med(z)	best(z)	Out
0	12.69753170008833	12.398377718036665	7.298187351684621	0
1	10.525030685285538	7.630487622288854	7.298187351684621	9
⋮	⋮	⋮	⋮	⋮
132	5.36305664951297	4.610611805470927	4.610611805470927	6
133	4.99901681120808	4.610611805470927	4.610611805470927	7
134	4.811046486045267	4.610611805470927	4.6036328412286585	4
135	4.963130520739801	4.610611805470927	4.6036328412286585	6
136	5.265386452268696	5.0796213239433605	4.6036328412286585	6
⋮	⋮	⋮	⋮	⋮
4995	4.681263607203966	4.579599345059113	4.579591950631037	5
4996	4.607388863208869	4.579603549120526	4.579591799618665	4
4997	4.627371038278473	4.579599499443619	4.579589330419115	3
4998	4.7334919080753135	4.579603549120526	4.579588292413069	1
4999	4.599615403703765	4.579597549265095	4.579588292413069	2

Table 1: By-generation summary of the feasible population in a typical run of Yuan. Gen=generation. avg(z)=average objective function value by generation. med(z)=median objective function value by generation. best(z)=best solution found so far. Out=number of infeasible progeny produced by generation.

4.1 Evaluation of the Feasible Population

The feasible population rather quickly finds solutions within 0.5% of optimal by generation 136. Table 3 shows a subsequent slow, steady improvement in z^+, on the average. The variance of the feasible population appears to stabilize (Table 3, column 7), but cannot go to zero because of mutation and continuing immigration. In any event, the two-population GA again evidences excellent performance on this challenge problem from the literature, which is nonlinear in both the objective function and the constraints, and has a mixture of integer and real (floating point) decision variables.

4.2 Evolution of the Infeasible Population

As seen in Table 3, the average infeasibility of solutions in the infeasible population becomes closer to 0 as the run progresses. In the run under display here, the average infeasible solution moved towards feasibility from a distance of 0.2005 during generations 900–999 to a distance of 0.0126 during generations 4900–4999.

The infeasible solutions are not subjected to selection with regard to the z (objective function) values. Most interestingly, the z values in the infeasible population nevertheless clearly move towards z^+ (or z^*) as the run progresses. In the end, the best infeasible z values are not far from the optimal value, z^*. Compare the rightmost two columns of Table 3. Note that there is an overall reduction in the variance in z in the infeasible population as the generations progress. This is in

Gen	avg(z)	med(z)	avgInfeas	Out
0	8.143224744257274	8.368698923454343	-1.847537037871579	0
1	8.397975429018375	9.022887244519183	-1.1174324950527328	1
⋮	⋮	⋮	⋮	⋮
132	6.962531847852802	6.519229712067366	-0.24845267499703316	5
133	7.267970477305616	6.8234062690754875	-0.12999438680511116	2
134	7.706505082745875	6.038927899546921	-0.08853998581161633	1
135	6.521347185201272	5.257514914110913	-0.1733602611829584	5
136	7.175989168243694	7.516553449635433	-0.14180623232077735	6
⋮	⋮	⋮	⋮	⋮
4995	5.71510513474602	4.926231680014258	-3.16018619839209E-6	0
4996	5.516678970813462	4.926229526679555	-0.015957607821762917	0
4997	5.890533806672197	4.926189743129405	-0.0159577549401262	1
4998	6.115080232697109	5.07961534652323	-2.3668425574097095E-6	2
4999	5.742802940813199	4.926184475169487	-0.01595877557775823	0

Table 2: By-generation summary of the infeasible population in a typical run of Yuan

contrast to the variance in z for the feasible population, which does not appear to decline during the last 1000 generations.

4.3 Mixing between the Feasible and Infeasible Populations

The infeasible population can produce offspring that are feasible. Although this might seem to be unlikely, our example data show the infeasible population was steadily producing feasible solutions. See the InF→Fea column in Table 3. We also note some indication of modestly declining productivity as the run progressed.

Column Fea→InF in Table 3 shows the feasible population produced infeasible offspring at roughly double the rate InF→Fea of the infeasible population. Fea→InF , however, may be declining more rapidly than InF→Fea.[7]

These data support a clear and consistent picture of what is happening in the two-population GA data (in this typical run for this problem, but in our experience it is representative). Selection is driving the feasible population closer and closer to the boundary of the feasible region. Not only is there improvement in z^+ over time; there is steady but fluctuating improvement in the average z each generation. Similarly, selection is driving the infeasible population closer to the boundary separating the feasible and infeasible regions.

[7]Note that we are using nonuniform mutation on the floating point decision variables and that the radius of mutation declines as the number of generations increases.

Generations	Infeasibility	InF→Fea	Fea→InF	z^+	medz_{InF}	$\sigma^2 z_{\text{Fea}}$	$\sigma^2 z_{\text{InF}}$
0–99	-0.2824	3.5400	7.3000	5.222503	7.123	2.302	6.839
900–999	-0.2005	3.4100	6.6200	4.594130	6.577	0.840	8.928
1900–1999	-0.0453	3.3100	6.4000	4.581232	9.468	1.015	7.713
2900–2999	-0.0858	3.0400	6.4800	4.579938	5.926	0.426	3.302
3900–3999	-0.0501	2.7000	6.3300	4.579845	5.103	0.251	1.775
4900–4999	-0.0126	3.2900	4.8200	4.579653	5.245	0.253	0.948

Table 3: Yuan Results: Averages over 100 generations. Infeasibility=$-1\cdot$sum of absolute violations of constraints (averaged over each solution for 100 generations). InF→Fea=number of feasible offspring from the infeasible population (by generation, averaged over 100 generations). Fea→InF =number of infeasible offspring from the infeasible population (by generation, averaged over 100 generations). z^+=best solution found in the feasible population (by generation, averaged over 100 generations). medz_{InF}=median objective function value in the infeasible population (by generation, averaged over 100 generations). $\sigma^2 z_{\text{Fea}}$=variance of objective function values in the feasible population (averaged over all solutions in 100 generations). $\sigma^2 z_{\text{InF}}$=variance of objective function values in the infeasible population (averaged over all solutions in 100 generations).

x_1	x_2	x_3	y_1	y_2	y_3	y_4
0.019176639369240966	0.12188533340384183	1.5120185756147242	1	1	0	1

Table 4: Gen=20, z=8.010270704398303, sumSlack= 10.515981932347778. First appearance of (1, 1, 0, 1) in the feasible population.)

4.4 Genetic Benefits of Population Immigration

The ebb and flow of offspring from one population to the other has two benefits. First, useful genetic materials are less likely to be lost than in a one-population GA. Second, these materials tend to diffuse into both populations.

Detailed examination of the run reveals that alleles and patterns of alleles (building blocks) will often arise in one population, quickly move to the other, and then be maintained indefinitely in both populations. Consider, for example, the pattern $\mathbf{y} = (1, 1, 0, 1)^T$ which appears in z^*. As shown in Table 4, this pattern first appears in the feasible population in generation 20. It is lost to the feasible population, but reappears beginning in generation 26 of the infeasible population and is present in generally increasing numbers for the remainder of the run.

As shown in Table 5, this pattern actually appears first in the infeasible population. It is then lost, but reappears sporadically. Starting in generation 29 it maintains its presence in most of the remaining generations of the infeasible population. The floating point x_i variables evidence a similar dynamic. All of this supports the hypothesis that the infeasible population is actively exploring valuable building blocks—think of Holland's schemata [11]—under a relaxed but increasingly tightening feasibility requirement.

x_1	x_2	x_3	y_1	y_2	y_3	y_4
1.0255729805767875	0.556585764572967	1.5120185756147242	1	1	0	1

Table 5: Gen=26, z=5.605040171124676, sumInfeas= -1.7453232819180533. First appearance of (1, 1, 0, 1) in the infeasible population.)

4.5 Potential Tradeoffs Revealed in Infeasible Population

At generation 236, there appears in the infeasible population a solution with $z = 4.47002605609438$, which is much better (smaller) than z^+, the best feasible solution found during the run, and z^*. This infeasible solution is at: $x_1 = 0.195462908809646, x_2 = 0.795752247026746, x_3 = 1.96768190221611$, $y_1 = y_2 = y_4 = 1, y_3 = 0$. All the variable values in this solution are close to their corresondents in z^+ (and z^*), except x_3. Further, only one constraint is violated, ($y_2^2 + x_3^2 \leq 4.64$), which comes in at 4.871772068.

This is potentially valuable information. Constraints may often be relaxed, for a price. This infeasible solution at generation 236 provides a specific measure of the *shadow price* for constraint ($y_2^2 + x_3^2 \leq 4.64$). If the cost of relaxing this constraint enough to make this solution feasible is less than the benefit achieved by improving (here, reducing) z^+, then the decision maker has been presented with an opportunity discovered by the algorithm. Such opportunities often occur in practice. Hundreds if not thousands of specific potential such opportunities are apparent from the data in the infeasible populations of this typical run.[8]

5 Distance Tracing: Yuan

5.1 Distance Metric

Much can potentially be learned about the operation of the FI-2Pop GA by viewing decisions (aka: solutions, i.e., settings of the decision variables) as points in decision space and by measuring distances between decisions or populations of decisions. Given two decisions, $\vec{x}$ and $\vec{y}$, quite a few distance measures have appeared in the literature, but the (square of the) Euclidean norm is natural and robust

$$(\vec{x} - \vec{y}) \cdot (\vec{x} - \vec{y})^T \tag{15}$$

and it is what we have used in our investigations. We have focused on comparing populations, and have represented the centroids of the populations as vectors of mean allele values, $\vec{\mu_x}$ and $\vec{\mu_y}$.

Two complications intervene. First, as in Yuan, decision variables may be integers, and may thus have non-integral means. We simply treat all decision variables as reals for the purpose of computing population centroids. Second, distance metrics are sensitive to scale. It is appropriate, even mandatory, to convert all decision variables to a common scale for proper comparison. Further, it is desirable (for treating each decision variable equally) to remove linear associations between

[8] We note that this concept explored, but in the context of a conventional single-population GA [13, 12, 4].

decision variables. To this end, we rescale our data with an inverse covariance matrix obtained from a large sample of both the feasible and infeasible populations. Specifically, we sample both populations every 25 generations, combine the results, and calculate, C^{-1} the inverse covariance matrix on the sample. Retaining the (square of the) Euclidean norm, our distance metric is

$$d(\mu_x, \mu_y) = (\vec{\mu}_x - \vec{\mu}_y)C^{-1}(\vec{\mu}_x - \vec{\mu}_y)^T \tag{16}$$

We shall now discuss some of what may be learned using this distance metric.

5.2 Plots with the Metric

Figure 1 presents two dimensions of information about the evolution of the system as it confronts the Yuan problem. The density plot displays the distances between the centroids of the feasible (ordinate, vertical dimension) and infeasible (abscissa, horizontal dimension) populations for the first 200 generations of the run. Darker cells indicate smaller distances, lighter cells, larger distances. There are two lighter-colored bands, one running horizontally, one vertically (and somewhat less pronounced). The darker squarish area near the origin, in the southwest corner, evidences very little pattern, indicating no very clear trends during the first 25 or so generations. The horizontal lighter-colored band indicates that after about 30 generations every infeasible population is comparatively far away from the feasible populations of the first 25 generations or so. A similar interpretation in stronger form attends the horizontal band. North of the horizontal band and east of the vertical band lies most of the territory during the first 200 generations. This area is remarkably uniform. It does not get discernibly darker as it moves east, indicating little if any movement of the two populations towards each other. Most interestingly, however, the area is replete with vertical stripes, suggesting a function of one variable, and in particular suggesting that after roughly 30 generations every feasible population is equidistant from every infeasible population (older than roughly 30 generations). Put more carefully, if we fix on any infeasible population (older than roughly 30 generations), then *every* feasible population (older than roughly 30 generations) is equally distant from it. Recall that distances are for population centroids. What this suggests is that quickly the centroid of the feasible population settles down and remains fairly stable, while the centroid of the infeasible population is moving about.

This impression is strengthened by Figure 2.

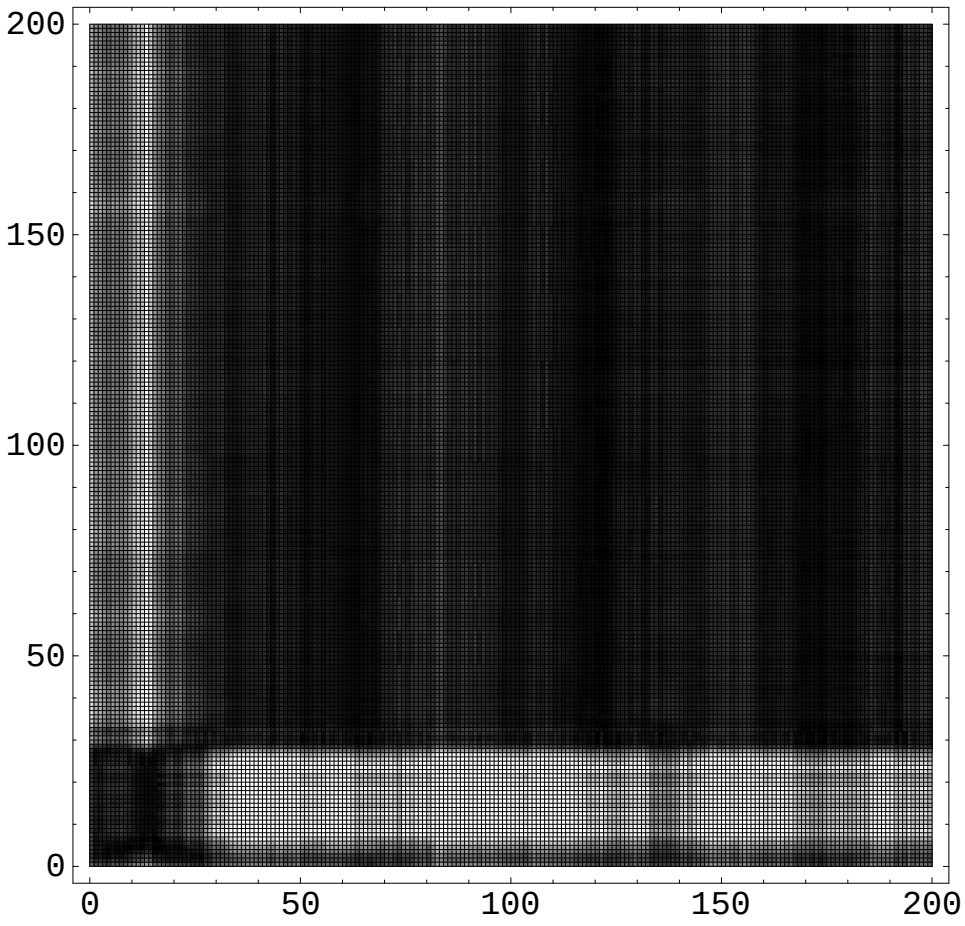

Figure 1: Yuan Plot. First 200 generations. Feasible population: abscissa. Infeasible population: ordinate. Light=further apart. Dark=closer. basicfirst200yuan.pdf

Figure 2 is like Figure 1, but instead of looking at the first 200 generations, it displays information for the entire run, sampled every 25 generations. Broadly speaking two facts are apparent. First, as the run progresses, the distances between the feasible and the infeasible population centroids tend to decline, although the process is not uniform. This is seen in Figure 2 by noting that the plot darkens from left to right. Second, and most strikingly, the vertical lines of common lightness mean that after the first 50 or so generations the distance between a given infeasible population and *any* of the feasible populations is roughly constant. This suggests that the feasible population quickly stabilizes, and the infeasible population gradually moves towards the boundary and hence the feasible population (remember: everything is in terms of centroids). Note in this regard that better and better decisions appear in the feasible population in the latter half of the run, suggesting a positive effect from the approach by the infeasible population to the boundary.

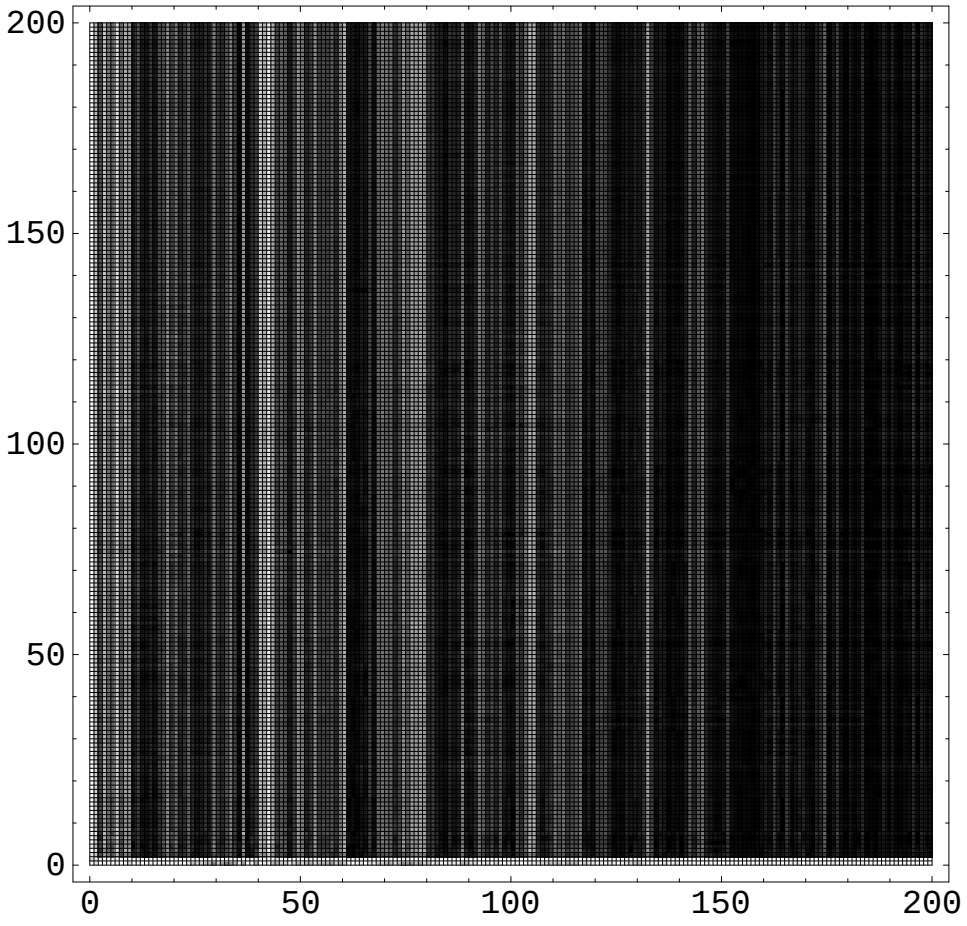

Figure 2: Yuan Plot. Every 25 generations. Feasible population: ordinate. Infeasible population: abscissa. Shading indicates magnitude of distance. Light=further apart. Dark=closer. basicevery25yuan.pdf

Figure 3 reinforces Figure 2 by plotting the distance between the feasible and infeasible populations (sampled every 25 generations) over the entire run. Clearly, there is a strong trend of diminishing distances between the centroids of the two populations. This is remarkable, given that the two populations are being selected with quite distinct fitness functions.

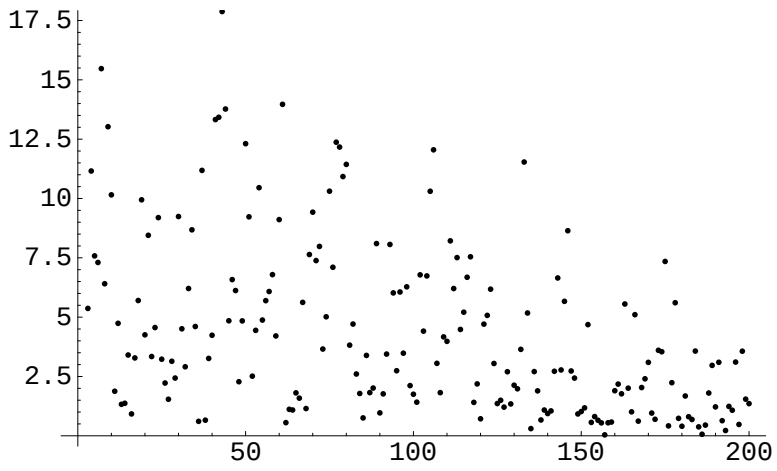

Figure 3: Yuan Plot. Distances between the feasible and infeasible populations, sampled every 25 generations. yuanfeainfeadistevery25.pdf

Figure 4 also reinforces Figure 2. It plots the distances from the feasible population to the infeasible population in the final generation (5000). Not much trend is apparent, indicating that the centroid of the feasible population moves very little.

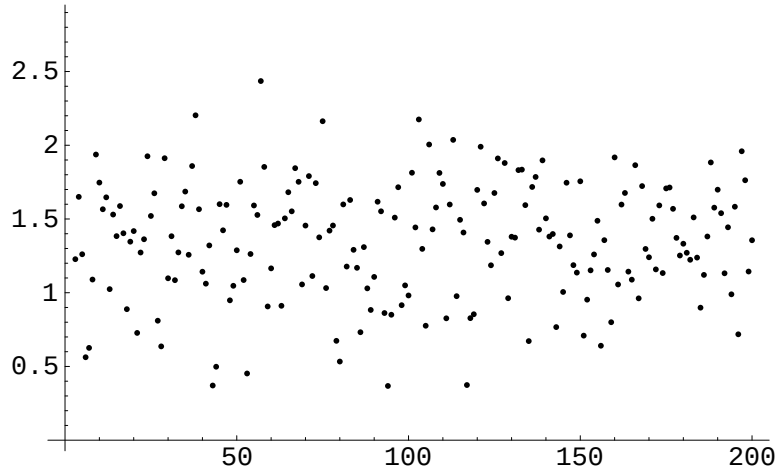

Figure 4: Yuan Plot. Distances between the feasible population and infeasible population at generation 5000 (=25*200), sampled every 25 generations. feasVsInfeas200.pdf

So far, we have established that for this run of Yuan the infeasible population is moving its centroid much more than the feasible population. But is the feasible population moving at all? Figure 5 is a density plot of the distances between the feasible populations (sampled every 25 generations) throughout the course of the run. The plot darkens as it moves from southwest (near the origin) to the northeast, indicating that later generations of the feasible population are closer to each other than they are to earlier generations, and that earlier generations are somewhat dispersed from each other.

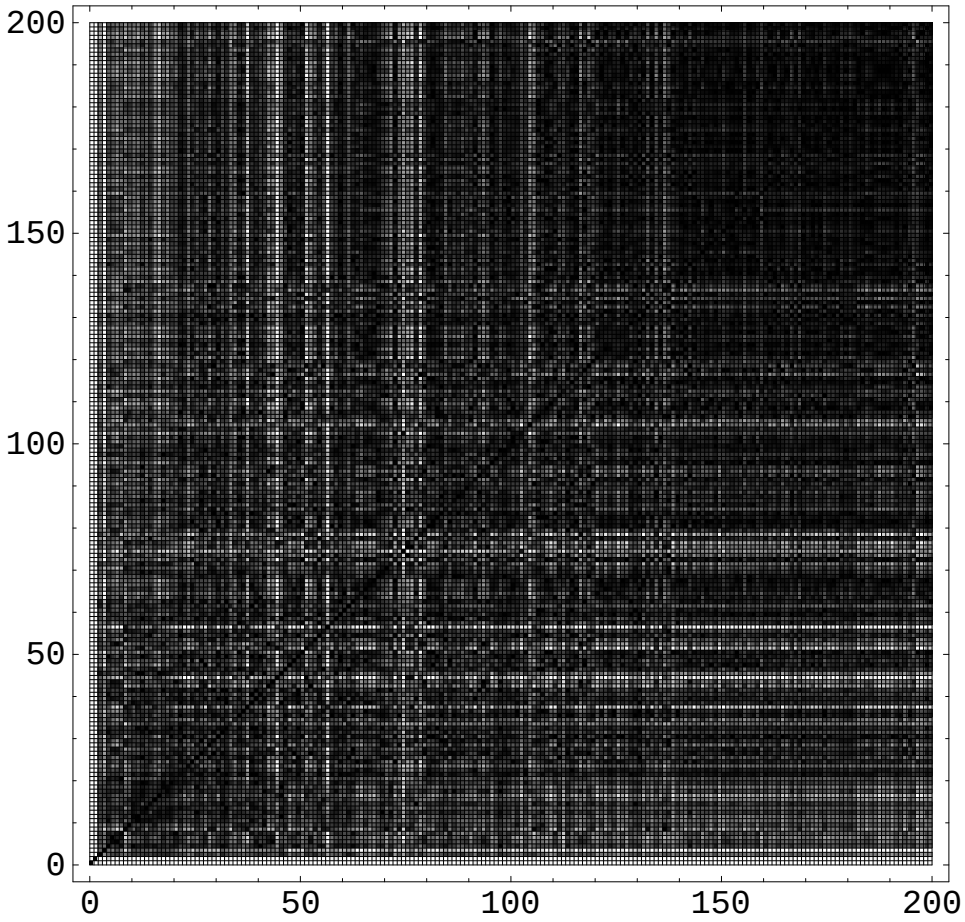

Figure 5: Yuan Plot. Every 25 generations. Feasible population: abscissa. Feasible population: ordinate. Light=further apart. Dark=closer. DensityPlotYuanFeaFeaEvery25.pdf

The general pattern of Figure 5 is confirmed by the instance of Figure 6, which shows the distances of the feasible populations from the feasible population at generation 5000 (=200*25).

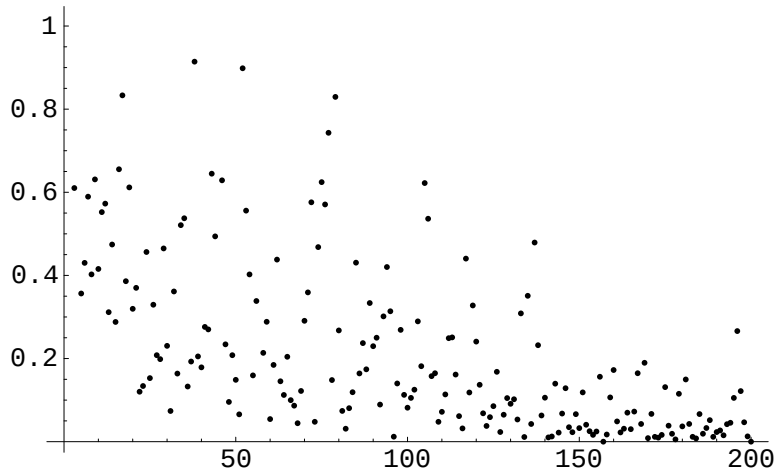

Figure 6: Yuan Plot. Every 25 generations. Feasible population: abscissa. Distance to Feasible population at generation 5000 (=200*25): ordinate. ListPlotYuanFeaFea200Every25.pdf

Figure 7 is the analog to Figure 5, but for the infeasible populations instead of the feasible populations. Notice that in the two figures the same general pattern obtains.

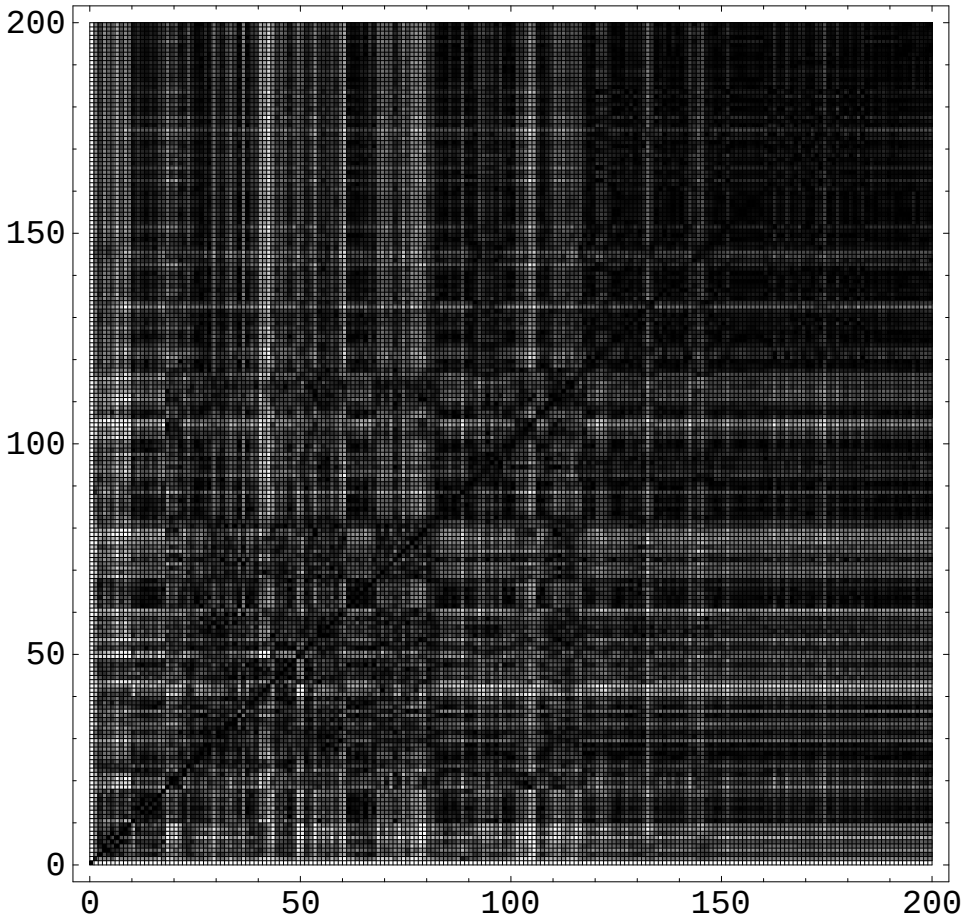

Figure 7: Yuan Plot. Every 25 generations. Infeasible population: abscissa. Infeasible population: ordinate. Light=further apart. Dark=closer. DensityPlotYuanInFeaInFeaEvery25.pdf

Figure 8 is the analog of Figure 6, but for the infeasible populations instead of the feasible populations. Again, the same general pattern obtains in the two figures. What is different is the scale on the ordinate, demonstrating that the centroid of the feasible populations has moved little compared to the movement of the centroid of the infeasible populations.

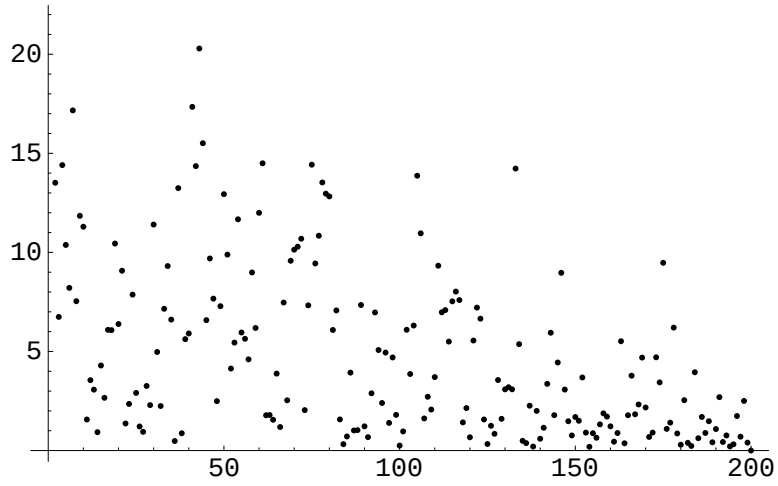

Figure 8: Yuan Plot. Every 25 generations. Infeasible population: abscissa. Distance to Infeasible population at generation 5000 (=200*25): ordinate. ListPlotYuanInFeaInFea200Every25.pdf

Now we divide the feasible population into incumbents (daughters of feasibles) and immigrants (sons of infeasibles). Note that in many generations there were no feasible children of the infeasible population that survived initial selection in the feasible population. In these generations the distance between the sons of the feasible and the sons of the infeasible were stipulated to be -10. The effect of this is evident in Figure 9.

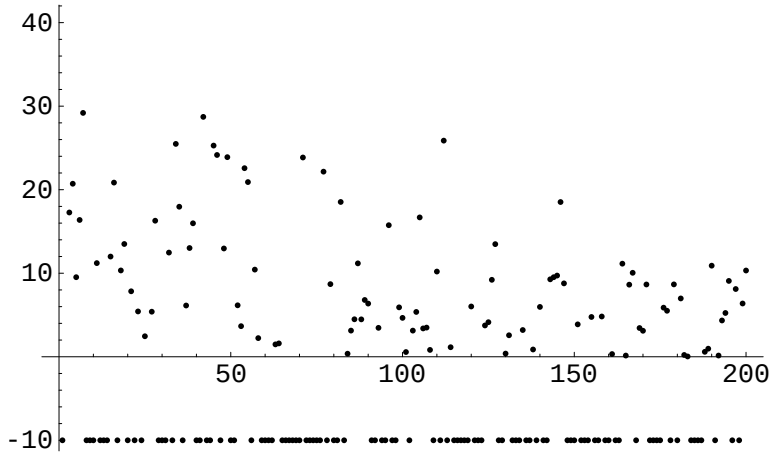

Figure 9: Yuan Plot. Every 25 generations. Feasible population, distances by generation. Feasible children of feasible parents: abscissa. Feasible children of infeasible parents: ordinate. ListPlot-FeaInctVSFeaImgtEvery25.pdf

It is important to note that the sons of the infeasibles showing up here have survived a competitive process in the feasible population. A feasible population (of 50) is created, then the infeasible population is processed. Any resulting feasible sons are put into the feasible population, which is then reduced to the standard size (here 50) by fitness-proportional selection (with 1 elitism). Thus, the feasible sons showing up have, probabilistically, competitive fitnesses in the feasible population. They are also, as the figure shows, atypical. In short immigrants from the infeasible population succeed in the feasible population and introduce considerable variation. A healthy situation indeed for a genetic system under selection.

Figure 10 is the analog for the infeasible population of Figure 9. Notice that the feasible population is more prolific in spinning off emigrants. (See also Table 3.) Note as well that, especially towards the end of the run, the immigrants in the infeasible population are closer to the incumbents than are the immigrants in the feasible population to their incumbents.

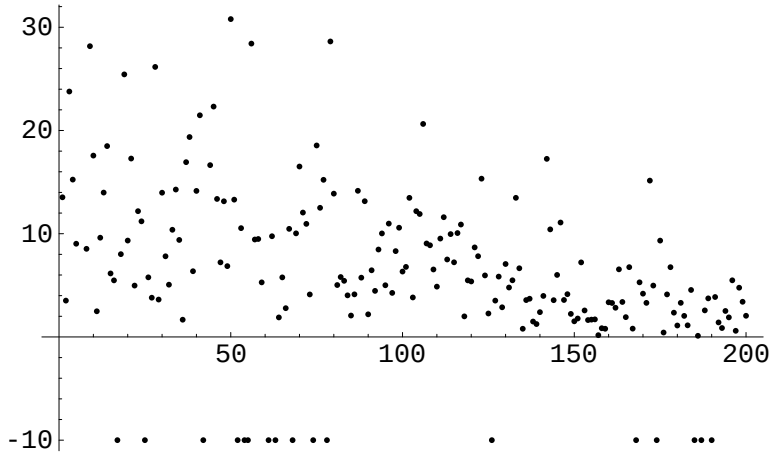

Figure 10: Yuan Plot. Every 25 generations. Infeasible population, distances by generation. Infeasible children of infeasible parents: abscissa. Infeasible children of feasible parents: ordinate. ListPlotInFeaInctVSInFeaImgtEvery25.pdf

Figures 11 and 12 present data from two new runs of the FI-2Pop GA on the Yuan model. The figures present distances, from the respective runs, between the centroid of the feasible population in generation 5000 and the centroids of the previous generations. In the run underlying Figure 11, at initialization the feasible population was set to empty. Thus, *all* subsequent decisions produced by the run are descended from infeasible ancestors. Similarly, in the run behind Figure 12 the infeasible population was made empty at initialization and in consequence all subsequent decisions have feasible ancestors. Qualitatively the patterns are quite similar to each other and to Figure 5, the corresponding plot with neither population empty at initialization. Note that the variance in Figure 12 is lower than in the other two cases, confirming that the infeasible population is a major source of variance in the feasible population. Further, $z^+ = 4.579595083691246$ for the run with empty feasible starts and 4.579583174633275 (slightly better than either of the two other runs) for the run with empty infeasible starts. Plots of (a) the distances between feasible and infeasible populations and (b) distances between the infeasible population in generation 5000 and all previous generations, yield qualitatively similar results to those in Figures Figures 11 and 12.

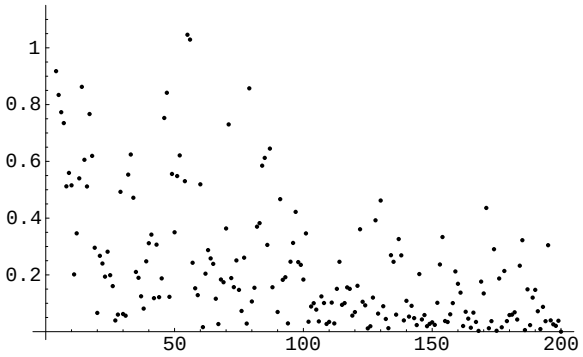

Figure 11: Yuan Plot. Every 25 generations, starting with an empty feasible population. Distances between the feasible population and feasible population at generation 5000 (=25*200), emptyfestartfeasVsFeas200.pdf

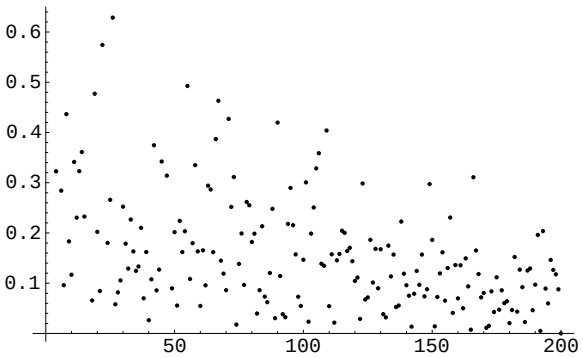

Figure 12: Yuan Plot. Every 25 generations, starting with an empty infeasible population. Distances between the feasible population and feasible population at generation 5000 (=25*200), emptyinfeastartfeaVsFea200.pdf

6 Discussion and Conclusion

We began this investigation with (i) the presupposition that the FI-2Pop GA has a strong record in solving constrained optimization problems and (ii) intuitions regarding why this might be so. The challenge, however, is to understand the FI-2Pop GA (as well as other metaheuristics). To this end we have here undertaken a detailed case study of a single problem, Yuan. In §4 we reported on various summary statistics and patterns found in the data for a single run. §5 describes and exploits a method introduced here—distance tracing with the inverse covariance transform—for describing behavior by populations of decisions in an evolution program. We can, in summary, make the following observations on this case:

1. The centroids of both the feasible and the infeasible populations move during the course of the GA's runs, with the infeasible population moving much more than the feasible.
2. The centroids of both populations move towards each other, and are quite close by the end of the runs.
3. The infeasible population is a continuing source of variation in the feasible population, throughout the runs.
4. Starting the FI-2Pop GA with either an empty feasible population or an empty infeasible population has little effect on the macro behavior of the system.
5. Both populations become more homogeneous (are reduced in variance), the feasible population much faster than the infeasible population.
6. The infeasible population is, on average, just barely infeasible towards the end of the run, indicating that its members are all close to the boundary between the feasible and infeasible regions.

This leads to the following description of what is happening as the FI-2Pop GA solves the Yuan problem. The feasible population rather quickly undergoes a significant reduction in variance, and concentrates in a region near or including z^+, the best solution it will find. After that, the centroid of the population moves steadily but slightly in magnitude, along with steady but modest improvements in the best decision found. Emigration from the feasible population is substantial ($\approx$15–25%) throughout the run. Because children will resemble their parents and because the feasible population is comparatively narrow in its range, the decisions it introduces into the infeasible population will also be near the region of concentration of the feasible population. This trend is reinforced by the GA's use of nonuniform mutation: mutation sizes (but not frequencies) are reduced for the real-valued variables as the run progresses.

The infeasible population is also an ongoing contributor to the feasible population and to its variance. Selection on the infeasible population drives it towards the boundary. Selection can have very little direct effect on reducing the variance of the infeasible population and can have no direct effect on the objective function values of the infeasible decisions. Yet, strong trends are observed on both variance and objective function values. This is due to immigration from the (increasingly concentrated) feasible population, which clusters nearer and nearer the feasible

population. Infeasible solutions comparatively far from the feasible population occasionally give rise to feasible descendants, resulting in the parent(s) being removed from the infeasible population. Thus, the infeasible population becomes increasingly concentrated (in the case of Yuan) near to the feasible population. The two populations continue to exchange solutions throughout the run. As the run progresses, the two populations are increasingly concentrated near to each other, resulting in a focused exploration in the region of z^+.

That is (much of) the story for Yuan and the FI-2Pop GA. The story is a happy one in this case, and it would appear that the behavior of the algorithm is felicitous, rather than merely fortuitous. Finding the neighborhood of the optimum (as it apparently does), and focusing exploration there surely has much to recommend it as a strategy. Yuan, however, is just a single case. To see what this may tell us in general, we need to consider the FI-2Pop GA in light of the exploration–exploitation dilemma for optimization and indeed for learning. The No Free Lunch theorems [25, 28] tell us that the dilemma is fundamental: there are no universally optimal heuristics for constrained optimization. Is there anything useful to say in general about the FI-2Pop GA, or any other metaheuristic for that matter? There is.

Consider standard GAs from the perspective of the exploration–exploitation dilemma. Selection is inherently exploitive, although stochastically selecting a population based on fitness admits a degree of exploration. The selection operator is restrained in its greed. Conversely, mutation and crossover are predominantly exploratory operators, admitting a degree of exploitation by being restricted to higher fitness individuals. The FI-2Pop GA adds to this mixture of factors. The FI-2Pop GA is a source for additional variance, in both populations. As such, it adds tilt towards exploration. Of course, it cannot be guaranteed that the FI-2Pop GA will add variance to the two populations. It can be expected, however, that feasible offspring from infeasible parents will typically be somewhat different from feasible offspring, since the respective parents have survived in two very different selection regimes. A similar argument applies to the infeasible population.

Of course, and to repeat, there is no guarantee in general that variance will in fact be added or that if added it will prove useful. If the FI-2Pop GA fails to add variance (fails to produce feasible offspring from the infeasible population, or the feasible offspring fail to survive or survive but are not different), the resulting system is in effect operating a death penalty regime on the feasible population. If the FI-2Pop GA adds variance, but the variance is not useful, the system becomes somewhat noisier. None of these failure conditions would appear to vitiate the search. They would tend to reduce the effectiveness of the search, but would not threaten it's ultimate success, given additional resources (more generations or larger populations).[9]

On the other hand, there will surely be many cases in which additional variance introduced by the FI-2Pop GA will be useful. This is of course an empirical matter, but we note that over-convergence is a well-known bugbear for GAs and that the FI-2Pop GA has in fact a strong track record. Our thesis is that this strong track record may be explained in part by the additional variance created by the FI-2Pop GA. In this regard we note the following points, all of which are illustrated by the Yuan case.

[9] We note that the pressure of selection is proportional to the standard deviation of the population, which is itself proportional to the square root of the size of the population. Thus, doubling (or halving) the size of the population will less than double (or halve) the rate of response to selection.

1. Immigrants into the feasible population will have ancestors selected (with varying degrees of rigor) for proximity to the boundary of the feasible region. Especially if created by mutation, the immigrants will have a tendency to locate near their ancestors, and hence near the boundary, but on the feasible side. This is a desirable property if the optimum decisions (or even many of the very good decisions) are on or near the boundary.

2. Earlier on in a run, immigrants into the feasible population will tend not to have, and hence tend not to resemble, feasible ancestors. This will tend to introduce a larger variance into the feasible population.

3. Later on in a run, many immigrants into the feasible population will have, and will resemble, feasible ancestors, and thus will introduce less absolute variance into the feasible population. The effect is to localize the added variance in a region of concentration of the feasible population. If indeed a region of concentration is the neighborhood of good decisions, the effect is to intensify the local search.

In conclusion, the FI-2Pop GA is seen, both from first principles and from the run data analyzed above, to introduce variance into the feasible population in ways not available to penalty function approaches. Whether this additional variance often proves useful must be determined by experience. To date, that experience is favorable and the FI-2Pop GA is a credible distinct option, perhaps as part of a suite of approaches, for solving constrained optimization problems with evolution programs (EPs). The distance mapping technique for studying the performance of EPs, introduced here, offers the prospect of insight into alternate methods, which insight will complement experience from benchmark testing.

References

[1] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York, NY, 1996.

[2] Thomas Bäck, David Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing, http://www.iop.org/Books/CIL/HEC/index.htm, 1997–2003.

[3] Thomas Bäck, David Fogel, and Zbigniew Michalewicz, editors. *Advanced Algorithms and Operators*. Institute of Physics Publishing, Bristol, UK, 2000.

[4] B. Branley, R. Fradin, S. O. Kimbrough, and T. Shafer. On heuristic mapping of decision surfaces for post-evaluation analysis. In Ralph H. Sprague, Jr., editor, *Proceedings of the Thirtieth Annual Hawaii International Conference on System Sciences*, Los Alamitos, CA, 1997. IEEE Press.

[5] P. C. Chu and J. E. Beasley. A genetic algorithm for the generalized assignment problem. *Computers and Operations Research*, 24(1):17–23, 1997.

[6] P. C. Chu and J. E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4(1):63–86, June 1998.

[7] Carlos Artemio Coello Coello. A survey of constraint handling techniques used with evolutionary algorithms. Technical report Lania-RI-99-04, Laboratorio Nacional de Informática Avanzada, Veracruz, México, 1999. http://www.lania.mx/~ccoello/constraint.html.

[8] Carlos Artemio Coello Coello. List of references on constraint-handling techniques used with evolutionary algorithms. World Wide Web, Accessed January 2003. http://www.cs.cinvestav.mx/~constraint/index.html.

[9] A. E. Eiben. Evolutionary algorithms and constraint satisfaction: Definitions, survey, methodology, and research directions. In Leila Kallel, Bart Naudts, and Alex Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, pages 13–30. Springer-Verlag, Berlin, Germany, 2001.

[10] Christodoulos A. Floudas, Panos M. Pardalos, and et al. *Handbook of Test Problems in Local and Global Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[11] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[12] S. O. Kimbrough and J. R. Oliver. On automating candle lighting analysis: Insight from search with genetic algorithms and approximate models. In Jay F. Nunamaker, Jr. and Ralph H. Sprague, Jr., editors, *Proceedings of the Twenty-Sixth Annual Hawaii International Conference on System Sciences, Volume III: Information Systems: Decision Support and Knowledge-Based Systems*, pages 536–544, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[13] S. O. Kimbrough, J. R. Oliver, and C. W. Pritchett. On post-evaluation analysis: Candlelighting and surrogate models. *Interfaces*, 23(7):17–28, May-June 1993.

[14] Steven O. Kimbrough, Ming Lu, and Soofi M. Safavi. Exploring a financial product model with a two-population genetic algorithm. In *Proceedings of the 2004 Congress on Evolutionary Computation*, pages 855–862, Piscataway, NJ, June 19–23, 2004. IEEE Neural Network Society, IEEE Service Center.

[15] Steven O. Kimbrough, Ming Lu, David Harlan Wood, and D. J. Wu. Exploring a two-market genetic algorithm. In W. B. Langdon, E. Cantú-Paz, and et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, pages 415–21, San Francisco, CA, 2002. Morgan Kaufmann.

[16] Steven O. Kimbrough, Ming Lu, David Harlan Wood, and D. J. Wu. Exploring a two-population genetic algorithm. In Erick Cantú-Paz and et al., editors, *Genetic and Evolutionary Computation (GECCO 2003)*, LNCS 2723, pages 1148–1159, Berlin, Germany, 2003. Springer.

[17] Steven Orla Kimbrough, Ming Lu, and David Harlan Wood. Exploring the evolutionary details of a feasible-infeasible two-population ga. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Birmingham, UK, 18-22 September 2004.

[18] R. Le Riche and R. T. Haftka. Improved genetic algorithm for minimum thickness composite laminate design. *Composites Engineering*, 3(1):121–139, 1995.

[19] R. Le Riche, C. Knopf-Lenoir, and R. T. Haftka. A segregated genetic algorithm for constrained structural optimization. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 558–565. Morgan Kaufmann, 1995.

[20] Z. Michalewicz, D. Dasgupta, R.G. Le Riche, and M. Schoenauer. Evolutionary algorithms for constrained engineering problems. *Computers & Industrial Engineering Journal*, 30(2):851–870, 1996.

[21] Zbigniew Michalewicz. A survey of constraint handling techniques in evolutionary computation methods. In *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pages 135–155, Cambridge, MA, 1995. MIT Press. http://www.coe.uncc.edu/~zbyszek/papers.html.

[22] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin, Germany, third edition, 1996.

[23] Zbigniew Michalewicz. Numerical optimization: Handling nonlinear constraints. In Bäck et al. [2], page G9.9.

[24] Zbigniew Michalewicz. Genocop – optimization via genetic algorithms. World Wide Web, Accessed January 2003. http://www.cs.sunysb.edu/∼algorith/implement/genocop/implement.shtml.

[25] Zbigniew Michalewicz and David B. Fogel. *How to Solve It: Modern Heuristics*. Springer, Berlin, Germany, 2000.

[26] Ruhul Sarker, Masoud Mohammadian, and Xin Yao, editors. *Evolutionary Optimization*. Kluwer, Boston, MA, 2002.

[27] Alice E. Smith and David W. Coit. Penalty functions. In Bäck et al. [2], page C5.2.

[28] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[29] X. Yuan, S. Zhang, and L. Pibouleau. A mixed-integer nonlinear-programming method for process design. *RAIRO - Recherche Opérationnelle-Operations Research*, 22(3):331–346, 1988.

[30] Ming Yuchi and Jong-Hwan Kim. Grouping-based evolutionary algorithm: Seeking balance between feasible and infeasible individuals of contrained optimization problems. In *Proceedings of the 2004 Congress on Evolutionary Computation*, pages 280–7, Piscataway, NJ, June 19–23, 2004. IEEE Neural Network Society, IEEE Service Center.

A Specifics of the Feasible-Infeasible Two-Population GA

We describe here key details of the two-population GA for constrained optimization used in the work reported in this paper. The algorithm is essentially that used in [16]. All key parameters were set *ex ante*, were not tuned for any of the problems, and were set identically for all problems. Two populations of solutions are created at initialization and maintained throughout a run: the *feasible population* consists of only feasible solutions to the constrained optimization problem; the *infeasible population* consists of only infeasible (constraint-violating) solutions. Each population is initialized to a size of 50 and, with a qualification described below, this size is maintained throughout the run. The number of generations was 5,000 for each population, corresponding to 10,000 generations in a normal GA. Floating point encoding, rather than binary encoding, is used for real-valued alleles.

Initialization proceeds one population at a time, beginning with the feasible population. A solution is randomly generated and its feasibility determined. If it is feasible the solution is placed into the feasible population; otherwise it is discarded. This continues until 50 feasible solutions are obtained or 5,000 attempts have been made. The algorithm proceeds even if fewer than 50 feasible solutions are found. The analogous process is conducted to initialize the infeasible population.

The two-population GA maintains 4 collections of solutions at each generation: the feasible population, the feasible pool, the infeasible population, and the infeasible pool. Creation of the next generation begins by processing the feasible population of the current generation. Fitness (as objective function value) is calculated for each member of the population and 50 new solutions are generated using the genetic operators (in order): fitness-proportional selection of parents, crossover (probability 0.4, single-point), and mutation (at probability 0.4, non-uniform mutation for floating point alleles, with degrees (b in Michalewicz's formula [22, pages 103, 111]) equal to 2). The feasibility of each of the 50 solutions is determined and each solution is placed in one of two pools, either a (next generation) feasible pool or an infeasible pool, as appropriate. The current generation infeasible pool is added to the just-created infeasible pool. (Thus, the 'infeasible population' at generation 0 is really the infeasible pool at generation 0.) Fitness (as sum of constraint violations) is then calculated for each member of the current infeasible pool. If necessary, the size of the infeasible pool is reduced to 50, based on fitness. The result is the next generation infeasible population. Using the next generation infeasible population, 50 new solutions are generated as before. Feasible results are placed in the next generation feasible pool. If necessary, the size of the feasible pool is reduced to 50, based on fitness. The result is the next generation feasible population. Infeasible results are placed in the next generation infeasible pool and the contents of the next generation

infeasible population are also placed into the next generation infeasible pool. This completes one generation. Processing continues until 5,000 generations have elapsed.