Exploring the Evolutionary Details of a Feasible-Infeasible Two-Population GA

Steven Orla Kimbrough[1], Ming Lu[1], and David Harlan Wood[2]

[1] University of Pennsylvania
{kimbrough, milu}@wharton.upenn.edu
[2] University of Delaware
{wood}@cis.udel.edu

Abstract. A two-population Genetic Algorithm for constrained optimization is exercised and analyzed. One population consists of feasible candidate solutions evolving toward optimality. Their infeasible but promising offspring are transferred to a second, infeasible population. Four striking features are illustrated by executing challenge problems from the literature. First, both populations evolve essentially optimal solutions. Second, both populations actively exchange offspring. Third, beneficial genetic materials may originate in either population, and typically diffuse into both populations. Fourth, optimization vs. constraint tradeoffs are revealed by the infeasible population.

1 Introduction

Constrained optimization problems (COPs) are ubiquitous. It is well known that people are poor at finding good solutions to even quite small constrained optimization problems. What is more, modern problems often contain tens of thousands of decision variables and hence are beyond direct comprehension by humans. Automated computational and algorithmic support is necessary. In spite of enormous progress in algorithmic theory and computerization, the need for solutions to COPs far outstrips the state-of-the-art in exact solution technique. And the situation is getting worse, because new applications for complex COPs continue to arise in bioinformatics, logistics, data mining, and other areas. The difficulty is fundamental; with few exceptions COPs are known to be NP-hard. So, approximating metaheuristics would seem to be a profitable approach.

Genetic algorithms (GAs) have much to recommend them as metaheuristics for constrained optimization, and are used routinely to attempt to maximize or minimize an objective function of a number of decision variables. Because GAs are general–purpose procedures they can be used even in computationally challenging cases in which the objective function and/or constraints are nonlinear, and/or the decision variables are integers or mixtures of integers and reals. When the objective function is constrained, however, use of GAs is problematic. Considerable attention has been paid to incorporating constraints (see [1–5] for excellent reviews), but no consensus approach has emerged. A natural—and the most often used—approach is to penalize a solution 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). This is the general form of a maximization constrained optimization problem:

$$\max_{x_i} z = Z(\boldsymbol{x}), \text{subject to } E(\boldsymbol{x}) \geq \boldsymbol{a}, F(\boldsymbol{x}) \leq \boldsymbol{b}, G(\boldsymbol{x}) = \boldsymbol{c}, x_i \in \mathcal{S}_i \qquad (1)$$

Here, $Z(\boldsymbol{x})$ is the objective function value produced by the candidate solution $\boldsymbol{x}$.[3] E, F and G each yield zero or more constraint inequalities or equalities. As functions, Z, E, F, G can be any functions at all (on $\boldsymbol{x}$). $\mathcal{S}_i$ is the set of permitted values for the x_i (the components of the vector $\boldsymbol{x}$), which are called the *decision variables* for the problem. $\mathcal{S}_i$ may include reals, integers, or a mixture. The purpose of a penalty function formulation is to produce a representation of the problem that can be directly and naturally encoded as a GA. To indicate a penalty function representation, let $\boldsymbol{x}$ be a (candidate) solution to a maximization constrained optimization problem. Its absolute fitness, $W(\boldsymbol{x})$, in the presence of penalties for constraint violation is $Z(\boldsymbol{x}) - P(\boldsymbol{x})$ and the COP is defined as:

$$\max_{x_i} z = W(\boldsymbol{x}) = Z(\boldsymbol{x}) - P(\boldsymbol{x}), \qquad (2)$$

where $P(\boldsymbol{x})$ is some total penalty associated with constraint violations at $\boldsymbol{x}$. Problems representable as in expression (2) are directly and naturally encoded as GAs. 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 population. This may be undesirable, since infeasible solutions may carry valuable information and may be useful in searching for optimal values.

We focus here on a feasible-infeasible two-population (FI-2Pop) GA deployed for constrained optimization problems (COPs). In this way of organizing an evolutionary process, two populations of solutions are maintained and evolved: the feasible population (consisting only of feasible solutions) and the infeasible population (consisting only of infeasible solutions). Genetic operations are applied to the two populations sequentially. Offspring are placed in one of the two populations, depending only on whether they are feasible or not. Thus, feasible parents may produce infeasible children and infeasible parents may produce feasible children. It is evident from prior work, [6–8], and from new results reported here, that the FI-2Pop GA has considerable merits. We wish to understand why. To this end, there are three main issues we wish to address: (a) What are the characteristic properties of this search process? (b) How does the infeasible population contribute to the search process? and (c) How might information in the infeasible population be exploited for decision making? Our approach will be an empirical one, in which we follow the details of the evolutionary process under the FI-2Pop

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

GA. We have been impressed by analogous studies of biological systems, e.g., [9–11], and we note that some of these scientists have turned to digital agents and computational systems in order to examine the microstructure of evolutionary processes, e.g. [12]. What works in nature should work in management science.

We next discuss the first (of four) optimization problems, newly treated here with the FI-2Pop GA. Following that, we report briefly on the remaining three challenge problems, each of which is nonlinear and can be solved by Genocop III, a standard and generally excellent GA solver [13]. See Table 2 for a comparison of the two-population GA solutions to those obtained by Genocop III. In general agreement with previously-published results [7], the two-population GA outperforms Genocop III both in best solution found and in the standard deviation (and mean) of the best solution found, across a sample of runs.[4]

2 Four Problems

Yuan, our first problem, is discussed in [16, page 266] and was originated in [17]. 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. The constraints for all four of our models are inequalities. Genocop III is unable to handle models of this sort [13]. As reported by [16, page 266] and [17], at optimality the (minimizing) 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 two-population GA on the Yuan problem. 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 $\mathbf{y}^T = (1, 1, 0, 1)$, $x_1 = 0.199998178908325$, $x_2 = 0.799999776184869$, $x_3 = 1.90787728616851$.

Feasible Population Converges as Variance Minimizes. The feasible population finds solutions within 0.5% of optimal by generation 136. Table 1 shows a subsequent slow, steady improvement in z^+, on the average. The variance of the feasible population appears to stabilize (Table 1, column 7), but cannot go to zero because of mutation and continuing immigration. In any event, the two-population GA evidences excellent performance on this problem.

[4] There is insufficient space to describe fully our FI-2Pop GA. For details see [14], also [8]. In brief, all parameters were set *ex ante*, were not tuned for any of the problems, and were set identically for all problems. We used fitness-proportional selection of parents, single-point crossover (probability 0.4), and mutation (at probability 0.4, non-uniform mutation for floating point alleles, with degrees (b in Michalewicz's formula [15, pages 103, 111]) equal to 2). We note that the results reported in Table 2 for the two-population GA relied on uncollected random number seeds determined by reading the system clock. For close comparison with GENOCOP, the feasible and infeasible populations were sized at 50 each. A run of the FI-2Pop GA consisted of 5,000 feasible generations and 5,000 infeasible generations, alternating. The comparison runs of GENOCOP were for 10,000 generations on populations of 50.

Generations	Violation	InF→Fea	Fea→InF	z^+	medz_{InF}	$\sigma^2 z_{\mathrm{Fea}}$	$\sigma^2 z_{\mathrm{InF}}$
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 1. Yuan Results: Averages over 100 generations. Violation=$-1\cdot$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 feasible population (by generation, averaged over 100 generations). z^+=best objective function value 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). $\sigma^2 z_{\mathrm{Fea}}$=variance of objective function values in the feasible population (averaged over all solutions in 100 generations). $\sigma^2 z_{\mathrm{InF}}$=variance of objective function values in the infeasible population (averaged over all solutions in 100 generations).

Infeasible Population Converges towards Feasibility as Its Variation Decreases. As seen in Table 1, the average infeasibility (Violation, column 2 of Table 1) of solutions in the infeasible population becomes closer to 0 as the run progresses. In the run under display here, the average infeasible solution reduced its Violation by more than an order of magnitude during the run.

The infeasible solutions are not subjected to selection with regard to the z (objective function) values. Most interestingly, the z values for the infeasible population nevertheless clearly move towards z^+ (or z^*) as the run progresses. In the end, the best of these z values are not far from the optimal value, z^*. Compare the rightmost two columns of Table 1. Note that there is an overall reduction in the variance in z for the infeasible population as the generations progress. This is in contrast to the variance in z for the feasible population, which appears to stabilize during the last 1000 generations.

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 1. We also note some indication of modestly declining migration as the run progressed. Column Fea→InF in Table 1 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 .

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

problem	Best known	Genocop III		Two-population GA	
		Best of 10	Std.	Best of 10	Std.
Hesse (min)	-310	-306.972	16.185309	-309.9907	0.044785738
Pooling (max)	450	433.6981	37.02992564	444.157	15.08449889
Colville (min)	10122.69643	10126.6504	108.3765361	10122.8412	0.715469975

Table 2. Comparison of the Two-Population GA with Genocop III

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.[5]

Genetic Benefits of Immigration. The flow of offspring from one population to the other offers two benefits. First, maintaining infeasible solutions can preserve useful genetic materials, in contrast to a one-population GA. Second, useful genetic 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.

Information on Potentially Profitable Tradeoffs Automatically Generated. There is potentially valuable information among the infeasible solutions. Constraints may often be relaxed, for sufficient gain. These are opportunities where relaxing one or a few constraints 'legalizes' a known improvement. For example, at generation 236, there appears an infeasible solution for which $z = 4.47002605609438$, which is much better (smaller) than z^+, the best feasible solution found during the run, and z^*. This infeasible solution is at: $x_1 = 0.195462908809646, x_2 = 0.795752247026746, x_3 = 1.96768190221611, y_1 = y_2 = y_4 = 1, y_3 = 0$. The variable values in this solution, except x_3, are close to those of near-optimal solutions in z^+ (and z^*). Further, only one constraint is violated, ($y_2^2 + x_3^2 \leq 4.64$), which comes in at 4.871772068. 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 is not too much, the decision maker has been presented with an opportunity discovered by the algorithm. Many such opportunities occur in practice.[6]

Problem: Hesse. Our second problem is discussed in [16, page 24] and originated in [20]. It is quadratic in the objective function and in two constraints. The variables are continuous, and the best reported solution, $z^* = -310$, is at $\mathbf{x}^T = (5, 1, 5, 0, 5, 10)$. The two-population GA found its z^+ at -309.99141652481194.

---

[5] We use elite selection, which copies the best-of-generation solution into the next generation. Also, we use non-uniform mutation [15, pages 103, 111], in which the expected step size change from mutation of a floating point allele is a decreasing function of the number percentage of generations elapsed. This by itself will have the effect of reducing emigration between the populations as the run progresses.

[6] This concept has been explored in the context of a conventional GA [18, 19].

Generations	Violation	InF→Fea	Fea→InF	z^+	medz_{InF}	$\sigma^2 z_{\text{Fea}}$	$\sigma^2 z_{\text{InF}}$
0–99	-0.6650	1.5000	6.5400	-275.883824	-250.401	1284.434	13446.196
900–999	-0.4406	1.5500	6.5300	-292.164230	-288.285	331.403	3170.694
1900–1999	-0.2420	2.1000	6.6700	-306.323098	-241.333	230.956	5167.766
2900–2999	-0.1088	1.6900	6.3200	-306.606229	-284.745	43.058	413.583
3900–3999	-0.0342	1.9300	6.4900	-308.839796	-298.533	4.607	151.386
4900–4999	-0.0001	2.1400	5.1700	-309.969180	-309.930	0.000248	0.001351

Table 3. Hesse Results: Averages over 100 generations. See Table 1 for legend.

For this solution, x_1 =4.999949615359374, $x_3 = 5.0$, $x_4 = 4.3379483053785203E-5$, $x_2 = 1.0000502743019608$, $x_5 = 4.9999277165343345$, $x_6 = 10.0$.[7]

Table 3, for the Hesse problem, corresponds to Table 1 for the Yuan problem. The comments previously made about the Yuan problem and the patterns in evolution noticed there apply to the Hesse problem without much qualification. In this run of the Hesse problem we see an even stronger movement (than in Table 1) by the infeasible population towards feasibility. As shown in Table 3 the *average* infeasibility of the infeasible solutions during the last 100 generations of the run is -0.0001. During the last generation the average is $-3.99733965164017e-07$.

Mixing between the Feasible and Infeasible Populations. As with Yuan, notice that both populations continue to export throughout the run, the feasible population at a somewhat higher rate. Again, the median z values in the infeasible population approach z^+ even though selection in the infeasible population is not directed at z. Finally, although the overall pattern is similar, the z variances in Hesse decline over time much more rapidly than they do in Yuan.

Information on Potentially Profitable Tradeoffs Automatically Generated. On the uses of the infeasible population for automatically identifying constraint reconsiderations, we note that at generation 20 there appears in the infeasible population the solution $\mathbf{x}^T = (5.30504506985872, 1.09360041810019, 4.03893111661352, 4.74281156868017, 4.86902458174934, 9.62832716285562)$ with z =-330.33892227141 and infeasibility of -0.42288930351706. Only two constraints are violated: one constraint is violated by amount -0.024243816 and the other is violated by amount -0.398645488, for a total violation of -0.422889304. Further, the sum of slacks in the constraints is 21.5538596235044, so there is considerable room to tighten other constraints, e.g., by selling capacity. For example, a third constraint comes in at 13.12158005, well above its minimum of 4. This leaves more than 9 units of slack capacity, for which there may be other uses. As in the Yuan problem, the infeasible population contains many solutions such as this one, which may be used by managers profitably.

Problem: Pooling. Our third problem is discussed in [16, pages 38–9] and was originated in [21]. It is multiplicative in both the objective function and the constraints. All variables are continuous. The reported optimal solution is

[7] This is a better solution than reported in Table 2. The random number seed used for this run, 2, just happens to produce a better result. The run is not atypical.

2-pop variable	2-pop value at z^+	z^* value
x_1	0.0030777554240600784	0
x_2	0.49999973531965175	0.5
x_3	2.4252327646072726E-27	0
x_4	100.61930243127915	100
x_5	4.06847097629025E-30	0
x_6	99.3805723645953	100

Table 4. Variable settings at z^+ for the Pooling run under discussion

Generations	Violation	InF→Fea	Fea→InF	z^+	medz_{InF}	$\sigma^2 z_{\text{Fea}}$	$\sigma^2 z_{\text{InF}}$
0–99	-11.6311	1.40	9.68	396.665205	26.172	12734.450	80033.132
900–999	-10.0925	1.83	11.32	439.451911	369.936	1341.593	17441.671
1900–1999	-5.3936	1.26	10.98	447.986872	417.019	379.998	3812.274
2900–2999	-2.4942	1.36	10.46	449.080888	432.793	120.450	927.995
3900–3999	-0.6229	1.56	10.09	449.090581	441.848	6.935	89.803
4900–4999	-0.0015	2.01	7.78	449.330277	449.273	0.002987	0.008043

Table 5. Pooling Results: Averages over 100 generations. See Table 1 for legend.

$z^* = 450$, at $q_{11} = 0$, $q_{21} = 0.5$, $q_{41} = 0.5$, $y_{11} = 0$, $y_{12} = 100$, $z_{31} = 0$, and $z_{32} = 100$.

We map the problem variables as follows: $y_{11} \mapsto x_3$, $y_{12} \mapsto x_4$, $q_{11} \mapsto x_1$, $q_{21} \mapsto x_2$, $q_{41} \mapsto 1 - x_1 - x_2$, $z_{31} \mapsto x_5$, $z_{32} \mapsto x_6$. This eliminates the equality constraint and yields a mathematically equivalent problem. In the run to be discussed here we obtained $z^+ = 449.3803716736962$ with a random seed of 1. (This random seed was not used in the data for Table 2, which used seeds based on the system clock.) Table 4 shows the value of $\mathbf{x}^{+T}$ and its correspondence to the settings of the original variables at z^*.

Information on Potentially Profitable Tradeoffs Automatically Generated. Again we see that important managerial information is algorithmically extractable from the infeasible population. We note that in generation 4187 of the infeasible population a solution with $z = 453.692728740051$ was found, but then immediately lost. It has an infeasibility of -0.948398970709817, with the decision variables set to $\mathbf{x}^T = (0.00751253130969082, 0.499665249936584, 3.06696077041692e-27, 100.552108603048, 4.6473502896556e-30, 99.4273030879242)$. In this solution, only one constraint is violated. If the right-hand side could cheaply be raised from 0 to at least 0.948398970709817, this solution would become profitable and feasible. Also, there is extra slack associated with this solution: 100.959097265662. At z^+ the slack is 100.49712706071. There are many other examples of possible tradeoffs in this run.

Problem: Colville. Our fourth problem is discussed in [16, pages 92–3] and originated in [22]. It is multiplicative and quadratic in the objective function and multiplicative in the constraints. The variables are continuous. The reported

Generations	Violation	InF→Fea	Fea→InF	z^+	medz_{InF}	$\sigma^2 z_{\text{Fea}}$	$\sigma^2 z_{\text{InF}}$
0–99	-7.6055	2.5700	9.5700	10510.09	10105.056	439324.262	619808.371
900–999	-5.2850	2.0000	11.6800	10125.22	9900.066	53933.498	194832.518
1900–1999	-3.0104	1.7300	11.6800	10124.72	9962.886	17250.181	146940.068
2900–2999	-1.3644	1.9600	12.0600	10124.72	10059.269	3709.520	13635.974
3900–3999	-0.3630	1.6100	11.5900	10123.61	10093.688	184.869	1625.862
4900–4999	-0.0006	2.5600	8.7300	10123.47	10123.412	0.00364	0.03279

Table 6. Colville Results: Averages over 100 generations. See Table 1 for legend.

optimal value is $z^* = 10122.69643$[8] with $\mathbf{t}^{*T} = (78, 33, 29.998, 45, 36.7673)$. In the run we shall consider here, $z^+ = 10123.464905030838$ and $\mathbf{t}^{+T} = (78.00000321918812, 33.01144193119939, 30.001941327255274, 44.99999999924409, 36.75964862060901)$. The patterns to be seen in the summary data for Colville, in Table 6, are the familiar ones, given above for Yuan and Hesse.

Information on Potentially Profitable Tradeoffs Automatically Generated. Again, and there are many other examples in this run, we see that important managerial information about constraints is contained in the infeasible population. We observe that in generation 4908 of the infeasible population a solution with $z = 10122.6441615269$ was found and retained for some time. It has a low infeasibility, -0.000226063887148076, with the decision variables set to $\mathbf{t}^T = (78, 33.0114452272884, 30.0001612811051, 44.9999999992441, 36.7558397945725)$. Also, there is considerable slack associated with this solution: 2007.93820179402. At z^+ the slack is 2007.7787830732.

3 Discussion and Conclusion

We have newly examined four constrained optimization problems, recognized as difficult in the literature. We report results confirming—and extending to more difficult problems—previous work on the two-population GA: it works very well. We have discerned a number of characteristic properties of this search process, which are summarized in Tables 1, 3, 5, and 6. On four rather different problems a common, untuned FI-2Pop GA (parameters were set *ex ante*) produced excellent results in each case *and* produced similar patterns of behavior. Further, we picked these four problems for this study based only on their independent interest. Specifically: we are not withholding any unfavorable results on unreported cases.

Interpreting these results broadly, it appears that the FI-2Pop GA is successful because the infeasible population maintains genetic variation that would otherwise be destroyed. The genetic material so preserved is available, perhaps much later, to combine with newly-arisen improvements. Much, however, remains to be investigated. We note in this regard that there are other—rather different—

[8] We note an error in [16, page 93], where it is reported as 1.1436. Plugging in the variable values they give yields $z^* = 10122.69643$.

varieties of two-population GAs. SGGA maintains two violation-penalized populations, using different penalty functions, and crossbreeds between them [23, 24]. GENOCOP III is based on repair and maintains two populations; both are feasible throughout. See [25] for a review of both systems. Chu & Beasley [26, 27] 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 [28] 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.

References

1. Bäck, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, New York, NY (1996)
2. Bäck, T., Fogel, D., Michalewicz, Z., eds.: Advanced Algorithms and Operators. Institute of Physics Publishing, Bristol, UK (2000)
3. Coello, C.A.: 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.
4. Bäck, T., Fogel, D., Michalewicz, Z., eds.: Handbook of Evolutionary Computation. Institute of Physics Publishing, http://www.iop.org/Books/CIL/HEC/index.htm (1997–2003)
5. Coello, C.A.: 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.
6. Kimbrough, S.O., Lu, M., Wood, D.H., Wu, D.J.: Exploring a two-market genetic algorithm. In Langdon, W.B., Cantú-Paz, E., et al., eds.: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), San Francisco, CA, Morgan Kaufmann (2002) 415–21
7. Kimbrough, S.O., Lu, M., Wood, D.H., Wu, D.J.: Exploring a two-population genetic algorithm. In Cantú-Paz, E., et al., eds.: Genetic and Evolutionary Computation (GECCO 2003). LNCS 2723, Berlin, Germany, Springer (2003) 1148–1159
8. Kimbrough, S.O., Lu, M., Safavi, S.M.: Exploring a financial product model with a two-population genetic algorithm. In: Proceedings of the 2004 Congress on Evolutionary Computation, Piscataway, NJ, IEEE Neural Network Society, IEEE Service Center (June 19–23, 2004) 855–862
9. Cooper, T.F., Rozen, D.E., Lenski, R.E.: Parallel changes in gene expression after 20,000 generations of evolution in *e. coli.* Proceedings of the National Academy of Sciences **100** (2003) 1072–1077
10. Taddei, F., Radman, M., Maynard-Smith, J., Toupance, B., Gouyon, P.H., Goodelle, B.: Role of mutator alleles in adaptive evolution. Nature **387** (1997) 700–703
11. Lenski, R.E., Mongold, J.A., Sniegowski, P.D., Travisano, M., Vasi, F., Gerrish, P.J., Schmidt, T.M.: Evolution of competitive fitness in experimental populations

of E. coli: what makes one genotype a better competitor than another? Antonie van Leeuwenhoek **73** (1998) 35–47
12. Wilke, C.O., Wang, J., Ofria, C., Lenski, R.E., Adami, C.: Evolution of digital organisms at high mutation rate leads to survival of the flattest. Nature **412** (2003) 331–333
13. Michalewicz, Z.: Genocop – optimization via genetic algorithms. World Wide Web (Accessed January 2003) http://www.cs.sunysb.edu/∼algorith/implement/genocop/implement.shtml.
14. Kimbrough, S.O., Lu, M., Wood, D.H.: Exploring the evolutionary details of a feasible-infeasible two-population genetic algorithm in the context of constrained optimization. Working paper, University of Pennsylvania, Department of Operations and Information Management, Philadelphia, PA (2004) Posted at http://opim-sun.wharton.upenn.edu/˜sok/comprats/.
15. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Third edn. Springer, Berlin, Germany (1996)
16. Floudas, C.A., Pardalos, P.M., et al.: Handbook of Test Problems in Local and Global Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands (1999)
17. Yuan, X., Zhang, S., Pibouleau, L.: A mixed-integer nonlinear-programming method for process design. RAIRO - Recherche Opérationnelle-Operations Research **22** (1988) 331–346
18. Kimbrough, S.O., Oliver, J.R., Pritchett, C.W.: On post-evaluation analysis: Candle-lighting and surrogate models. Interfaces **23** (May-June 1993) 17–28
19. Branley, B., Fradin, R., Kimbrough, S.O., Shafer, T.: On heuristic mapping of decision surfaces for post-evaluation analysis. In Sprague, Jr., R.H., ed.: Proceedings of the Thirtieth Annual Hawaii International Conference on System Sciences, Los Alamitos, CA, IEEE Press (1997)
20. Hesse, R.: A heuristic search procedure for estimating a global solution of nonconvex programming problems. Operations Research **21** (1973) 1267
21. Ben-Tal, A., Eiger, G., Gershovitz, V.: Global minimization by reducing the duality gap. Mathematical Programming **63** (1994) 193–212
22. Rijckaert, M., Martens, X.: Comparision of generalized geometric programming algorithms. Journal of Optimization Theory and Applications. **26** (1978) 205
23. Le Riche, R., Haftka, R.T.: Improved genetic algorithm for minimum thickness composite laminate design. Composites Engineering **3** (1995) 121–139
24. Le Riche, R., Knopf-Lenoir, C., Haftka, R.T.: A segregated genetic algorithm for constrained structural optimization. In: Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann (1995) 558–565
25. Michalewicz, Z., Dasgupta, D., Le Riche, R., Schoenauer, M.: Evolutionary algorithms for constrained engineering problems. Computers & Industrial Engineering Journal **30** (1996) 851–870
26. Chu, P.C., Beasley, J.E.: A genetic algorithm for the generalized assignment problem. Computers and Operations Research **24** (1997) 17–23
27. Chu, P.C., Beasley, J.E.: A genetic algorithm for the multidimensional knapsack problem. Journal of Heuristics **4** (1998) 63–86
28. Yuchi, M., Kim, J.H.: Grouping-based evolutionary algorithm: Seeking balance between feasible and infeasible individuals of contrained optimization problems. In: Proceedings of the 2004 Congress on Evolutionary Computation, Piscataway, NJ, IEEE Neural Network Society, IEEE Service Center (June 19–23, 2004) 280–7