

Online Appendix to: Industrial Strength COMPASS: A Comprehensive Algorithm and Software for Optimization via Simulation

JIE XU and BARRY L. NELSON

Northwestern University

and

L. JEFF HONG

The Hong Kong University of Science and Technology

This online supplement documents the various algorithms that constitute Industrial Strength COMPASS. The source code is available from www.ISCompass.net.

A. NICHING GA

The purpose of the NGA is to find promising subregions of the solution space Θ . Each subregion has a *center*, which is the sample best solution in that subregion, and a handful of other solutions around the center. In the local phase, COMPASS is initialized with the center solution as the current sample best, and the surrounding solutions as the active solutions defining the most promising area.

To be consistent with the GA literature, we use the term “individual” to mean a feasible solution and its “fitness” to mean the estimated expected value of the solution. The NGA described here is designed for minimization problems with integer-ordered decision variables.

A.1 The NGA Framework in ISC

In this section, we outline the NGA framework we use in ISC. Subsequent sections elaborate on important steps in the outline.

A.2 Randomly Sampling Solutions

ISC can use either algorithm RMD [Hong and Nelson 2006] or algorithm MIX-D [Pichitlamken and Nelson 2003] to randomly sample integer solutions from the appropriate feasible space. RMD, which is the default algorithm, guarantees (nearly) uniform sampling of solutions provided the space is connected in the sense that any solution can be reached from any other solution by a sequence

© 2010 ACM 1049-3301/2010/01-ART3 \$10.00

DOI 10.1145/1667072.1667075 <http://doi.acm.org/10.1145/1667072.1667075>

ACM Transactions on Modeling and Computer Simulation, Vol. 20, No. 1, Article 3, Publication date: January 2010.

Algorithm 1. THE NGA FRAMEWORK IN ISC.

Step 0. Randomly sample m_G solutions from Θ . Assign n_0 replications to evaluate each solution (Section A.2).

Step 1. Identify niche radius r and number of niches q and dynamically construct niches (Section A.3).

Step 2. Apply niche-based transition rule. If conditions are met, stop the NGA (Section A.12).

Step 3. Apply fitness sharing (Section A.4).

Step 4. Apply grouping procedure using shared fitness values. Note that here more replications may be assigned (Section A.6).

Step 5. Calculate group selection probabilities for each solution (Section A.6).

Step 6. Select $m_G/2$ parents via Stochastic Universal Sampling, and the other $m_G/2$ parents via a mating restriction scheme (Sections A.5–A.7).

Step 7. Do arithmetical crossover for each pair of parents (Section A.8).

Step 8. Apply mutation operator to each solution generated by crossover (Section A.9).

Step 10. Evaluate new solutions by assigning enough replications to each solution.

Step 11. Replace old population with new population and conserve niche centers (Section A.10).

Step 12. Apply transition rules to decide if the NGA should stop (Section A.12). If not, go back to step 1.

of coordinate direction moves. MIX-D provides the same guarantees for any convex feasible region. RMD is the default in ISC because it is substantially faster.

A.3 Identifying Niche Radius and Number of Niches

Key input parameters for the NGA are the number of niches to form, q , and the niche radius, r , which defines how close a solution must be to a niche center to be clustered in the same niche. Because our algorithm is intended to be general purpose, we have no way to know these values in advance, nor should we expect the user to be able to provide them. To solve this problem, we form a rough response surface model over Θ from which we extract estimates of q and r .

To explain the approach, we define for any subset of solutions $I \subseteq \Theta$ and particular solution $\mathbf{x}^* \in I$ the region

$$\mathcal{C}(\mathbf{x}^*) = \{\mathbf{x} : \mathbf{x} \in \Theta \text{ and } \|\mathbf{x} - \mathbf{x}^*\| \leq \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{y} \in I \text{ and } \mathbf{y} \neq \mathbf{x}^*\}$$

which is essentially a COMPASS most promising area centered at \mathbf{x}^* . Further, let

$$\mathcal{A}(\mathbf{x}^*) = \{j \in I : \mathbf{x}_j \text{ defines an active (nonredundant) constraint in } \mathcal{C}(\mathbf{x}^*)\}$$

which is determined via constraint pruning.

Algorithm 2. NICHE RADIUS IDENTIFICATION ON ITERATION k

Step 1. Reindex the solutions in the current population so that $\bar{G}_k(\mathbf{x}_1) \leq \bar{G}_k(\mathbf{x}_2) \leq \dots \leq \bar{G}_k(\mathbf{x}_{m_G})$. Let $I = \{1, 2, \dots, m_G\}$, $L = \emptyset$.

Step 2. For each $i \in I$, do

(1) If $\bar{G}_k(\mathbf{x}_i) \leq \bar{G}_k(\mathbf{x}_j), \forall j \in \mathcal{A}(\mathbf{x}_i)$, then $L = L \cup \{i\}$, $I = I \setminus \{\mathbf{x}_i\}$.

(2) $I = I \setminus \{i\}$. Go to Step 1.

Step 3. Return $r = \min_{i,j \in L, i \neq j} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|$ and $q = |L|$.

The niche centers are the solutions in L , and for each solution $\mathbf{x}_i \in L$ its niches are those solutions no more than r distant from \mathbf{x}_i . Solutions that do not fall within a niche, if there are any, form a *nonniched* group. No solution will be in more than one niche.

A.4 Fitness Sharing

The NGA explores multiple regions in the solution space in parallel through the so-called fitness sharing mechanism. “Fitness” is a term used in the GA literature to mean the value of the objective function for a solution (e.g., Miller and Shaw [1995]). Here fitness is the sample average of all of the observations obtained on a solution. The basic idea is that if a niche is populated with too many solutions, these solutions should be given less chance to reproduce than they would have had in an ordinary GA, thus allowing solutions in less populated niches to have higher probabilities of being selected to generate new solutions. This can be done by discounting the fitness value of an individual by some factor that is dependent on how many solutions are in its niche. The discounted fitness values are then used in the NGA’s selection, mating, crossover, and mutation steps.

If a niche has n individuals including the niche center, then the shared fitness value f_{sh} of an individual within that niche with fitness value f is calculated as the discounted fitness value $f_{sh} = f/m$ if f is negative, and $f_{sh} = fm$ if f is positive, where m is the discounting factor, and we choose to let $m = n$. In ISC, the fitness value f of solution \mathbf{x}_i is its sample mean $G_k(\mathbf{x}_i)$.

If an individual \mathbf{x}_i is within the nonniched group, its shared fitness is calculated similarly, except that the discounting factor m is now calculated by using a triangular sharing function, $m = \sum_{j=1}^N sh(\|\mathbf{x}_i - \mathbf{x}_j\|)$, and

$$sh(\|\mathbf{x}_i - \mathbf{x}_j\|) = \begin{cases} 1 - \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{r} & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < r, \\ 0 & \text{otherwise.} \end{cases}$$

In ISC, since the NGA operates in a noisy environment, the selection step uses a statistical grouping procedure (see Section A.6) that incorporates sample variance information. Therefore, the fitness sharing step should also modify the sample variances to reflect the fact that the fitness value is now the discounted sample mean. We choose a very simple implementation: If the sample variance of solution \mathbf{x}_i is $S^2(\mathbf{x}_i)$, then the sample variance used in the grouping procedure is $S^2(\mathbf{x}_i)/m$ if $f_{sh} = f/m$, or $S^2(\mathbf{x}_i)m$ if $f_{sh} = fm$.

A.5 Selection

A selection method has two components: assigning selection probabilities to individuals, and selecting individuals based on those probabilities.

In the NGA implemented for ISC, we combine linear ranking with Stochastic Universal Sampling (SUS) [Baker 1987], which is a version of stratified

sampling. According to Sareni and Krahenbuhl [1998], SUS is the best for NGA among several selection methods.

A linear ranking scheme assigns the i th-best solution a selection probability

$$s_i = \frac{1}{m_G} \left(\eta - 2(\eta - 1) \left(\frac{i - 1}{m_G - 1} \right) \right),$$

where η is a constant between 1 and 2. The larger η is, the more likely a better solution is to be selected (in GA terminology, the higher the selection pressure). The worst solution has a selection probability of $p_1 \equiv s_{m_G} = (2 - \eta)/m_G$ and thus as long as $\eta < 2$, the worst solution has a positive probability to be selected as a parent for producing new solutions, as required in Appendix A of the published paper. For a DOvS problem, we suggest choosing a relatively low value for η , such as 1.5.

Given selection probabilities for each individual, SUS constructs a roulette wheel where the area for each individual is proportional to its selection probability. Then the roulette wheel is spun once, and the pointer will rest on some individual which is then selected as a parent. Other individuals are selected by advancing the pointer at a regular spacing until it wraps back to its starting point. A modification here is that instead of selecting m_G parents, only $m_G/2$ parents are selected through SUS because the second parent needs to be selected by a mating selection scheme, which will be introduced shortly. So the pointer is advanced by a spacing of $2/m_G$.

A.6 GA Selection in the Presence of Noise

To cope with the stochastic aspect of DOvS problems, we adopted the selection procedure described in Boesel [1999]. The idea is to define a metric to characterize the variability in the assigned selection probability, \hat{s}_i , in a stochastic environment as a squared deviation from the desired probability s_i . The expected sum of squared deviations due to misranking is defined as

$$E(\text{SSD}) = E \left\{ \sum_{i=1}^N (s_i - \hat{s}_i)^2 \right\}.$$

It is proved in Boesel [1999] that if N individuals can be divided into g groups, and that all individuals in group i are known to be superior to all solutions in group $i + 1$, and within group i all within-group orderings are equally likely to be observed, then $E(\text{SSD})$ is minimized when all members of each group are assigned their group's average selection probability. That is, $m_j^{(i)}$, the selection probability assigned to an individual j in group i of size N_i , is given by

$$m_j^{(i)} = \frac{1}{N_i} \sum_{j=1}^{N_i} s_j^{(i)}, \quad (6)$$

where $s_j^{(i)}$ is the selection probability of individual j in group i , as calculated by the deterministic NGA selection scheme.

In ISC, we use the grouping procedure devised by Boesel [1999]. The algorithm groups solutions that are not statistically significantly better than each other into one group. If there are too few groups formed in the first round, the algorithm identifies one group that is easiest to break into more groups, adds replications to solutions within this group, and processes this group again.

Notice that the sample average $\bar{G}(\mathbf{x}_i)$ and sample variance $S^2(\mathbf{x}_i)$ are shared fitness values and sample variances. We use n_i to denote the number of replications that have been assigned to \mathbf{x}_i .

Algorithm 3. GROUPING PROCEDURE FOR NGA

Step 1. Given current population $(\bar{G}(\mathbf{x}_i), S^2(\mathbf{x}_i), n_i), i = 1, 2, \dots, m_G$, let $\bar{n} = \frac{1}{m_G} \sum_{i=1}^{m_G} n_i$.

Step 2. Sort and reindex so that $\bar{G}(\mathbf{x}_1) \geq \bar{G}(\mathbf{x}_2) \geq \dots \geq \bar{G}(\mathbf{x}_{m_G})$.

Step 3. Let

$$S^2 = \frac{1}{m_G} \sum_{i=1}^{m_G} S^2(\mathbf{x}_i)$$

$$R = Q_{m_G, \sum_{i=1}^{m_G} (n_i - 1)}^{1-\alpha_G} S / \sqrt{\bar{n}},$$

where $Q_{m_G, \sum_{i=1}^{m_G} (n_i - 1)}^{1-\alpha_G}$ is the upper $1 - \alpha_G$ quantile of the studentized range distribution [e.g., Miller 1981] with m_G d.f. and $\sum_{i=1}^{m_G} (n_i - 1)$ samples.

Step 4.

- (1) $g = 1, i = 1$
- (2) while $i < m_G$ do
 - (a) $\mathcal{G}_g = \{i\}$, bottom = i , $i = i + 1$
 - (b) while $\bar{G}(\mathbf{x}_{\text{bottom}}) - \bar{G}(\mathbf{x}_i) < R$ and $i < m_G$ do
 - $\mathcal{G}_g = \mathcal{G}_g \cup \{i\}$, $i = i + 1$
 - (c) end while
 - (d) $g = g + 1$
- (3) end while

Step 5. If the number of groups formed $g < g_m$, where g_m is the minimum number of groups to obtain, let

$$\hat{R}_i = \max_{j \in \mathcal{G}_i} \bar{G}(\mathbf{x}_j) - \min_{j \in \mathcal{G}_i} \bar{G}(\mathbf{x}_j), i = 1, 2, \dots, g,$$

$$\hat{m}_i = |\mathcal{G}_i|.$$

- (1) For any group with $|\hat{R}_i| < \delta_G$, do nothing further with \mathcal{G}_i .
- (2) For groups with $|\hat{R}_i| \geq \delta_G$, do the following
 - (a) $\hat{R} = \max \hat{R}_i$, $\hat{\ell} = \arg \max_{i=1,2,\dots,g} \hat{R}_i$.
 - (a) Set

$$\hat{n} = \left\lceil \frac{Q^2 S^2}{\hat{R}^2} \right\rceil,$$

where

$$Q = Q_{\hat{m}_{\hat{\ell}}, \sum_{j \in \mathcal{G}_{\hat{\ell}}} (n_j - 1)}^{1-\alpha_G},$$

$$S^2 = \frac{1}{\hat{m}_{\hat{\ell}}} \sum_{j \in G_{\hat{\ell}}} S_j^2.$$

- (b) Obtain $\max(\hat{n} - n_j, 0)$ more observations for $j \in G_{\hat{\ell}}$.
 - (c) Apply fitness sharing.
 - (d) Go to step 1 but with group $\hat{\ell}$ as the population, and update $g_m \leftarrow g_m - g + 1$.
-

One issue with the grouping procedure is how many groups we want to form at each iteration. Boesel [1999, pages 66–68] pointed out that for linear ranking, each additional group formed provides less benefit and we may use as few as 3 groups.

A.7 Mating Restriction

The purpose of a mating restriction scheme is to maintain stable growth of niches during the evolution process by selecting individuals that are similar enough to each other to participate in crossover.

The mating restriction mechanism implemented here is proposed by Miller and Shaw [1995] and is known as dynamic inbreeding. Given the first parent \mathbf{x}_i selected by the usual GA selection scheme, the inbreeding scheme works as follows.

Algorithm 4. DYNAMIC INBREEDING

Step 1. Uniformly randomly sample m individuals from the population.

Step 2. From those individuals, select the best one that is within the same niche group as \mathbf{x}_i . If there is no such an individual, select the one that is closest to \mathbf{x}_i .

A.8 Crossover

Michalewicz [1996] described several crossover operators for GAs that optimize real-valued variables. Our experiments suggested that the arithmetical operator works best. Given two parents \mathbf{x}_i and \mathbf{x}_j , the arithmetical crossover operator produces two offspring \mathbf{x}'_i and \mathbf{x}'_j by generating a number β from $U(0, 1)$ distribution, and letting $\mathbf{x}'_i = \beta\mathbf{x}_i + (1 - \beta)\mathbf{x}_j$, and $\mathbf{x}'_j = (1 - \beta)\mathbf{x}_i + \beta\mathbf{x}_j$. Since this is an integer-ordered optimization problem, we need to round $\mathbf{x}'_i, \mathbf{x}'_j$ to integers.

One consequence of this rounding is that there is a nonzero probability that a parent will be generated as an offspring. Since the feasible solution region Θ is bounded, let $\xi > 0$ be the largest absolute value of a coordinate of any feasible solution \mathbf{x}_j . Then with a probability $p_2 = 1/(4\xi) > 0$, we have $\beta \leq 1/(4\xi)$, in which case $\mathbf{x}'_i = \beta\mathbf{x}_i + (1 - \beta)\mathbf{x}_j$ will be rounded to \mathbf{x}_j (and similarly \mathbf{x}'_j will be rounded to \mathbf{x}_i). The probability that a parent is generated as an offspring is thus at least p_2 , which satisfies the requirement in our proof of the global convergence of NGA in Appendix A of the published paper [Michalewicz 1996].

The arithmetical operator automatically handles convex constraints. But after rounding offspring to integers, it is possible that the resulting solutions are infeasible. In that case, we simply keep the parents. Another drawback with the arithmetical operator is that there is no way to generate offspring outside

of the convex combination of its two parents. To alleviate this problem, we use a relatively large mutation probability p_3 .

A.9 Mutation

Two mutation operators are implemented: uniform and nonuniform [Michalewicz 1996]. When a solution is chosen to undergo a mutation according to the mutation probability p_3 , a coordinate of this solution is randomly chosen with a probability of $p_4 = 1/d$ (where d is the dimension of \mathbf{x}), and its value is perturbed as follows.

- Uniform Mutation.* A line segment is drawn along the chosen coordinate direction, which hits boundaries of the feasible region at both ends. Then the new value of that coordinate is randomly chosen on that line segment. Since the feasible solution region Θ is bounded, we may assume that the largest absolute value a coordinate of any feasible solution \mathbf{x}_j can have is $\xi > 0$, and thus each solution along that coordinate will be selected with a probability at least $p_5 = 1/2\xi > 0$, thus satisfying the requirement in Appendix A of the published paper.
- Nonuniform Mutation.* The basic idea is that as evolution continues, more fine-tuning capability may be desirable to make the NGA more aggressive. Therefore, the mutated coordinate of \mathbf{x} , say x'_i , is determined as

$$x'_i = \begin{cases} x_i + \Delta(k, \text{right}(i) - x_i), & \text{with probability } 0.5, \\ x_i - \Delta(k, x_i - \text{left}(i)), & \text{with probability } 0.5, \end{cases}$$

where k is the generation number, $\text{right}(i)$ and $\text{left}(i)$ are the end points of the line segment along the i th coordinate direction to the boundary of the feasible region, and $\Delta(k, y) = y \times U(0, 1) \times (1 - k/K)^b$. Here K is the maximum number of generations allowed and b is a parameter controlling the degree of nonuniformity.

Preliminary experiments showed that nonuniform mutation was better suited for our purpose. In the framework of ISC, however, it is not clear how to set K for nonuniform mutation because the NGA can be terminated at any generation. We simply chose K very large and if it does happen that the number of generations k exceeds K , we use a small number in place of $1 - k/K$. Therefore, in ISC the function $\Delta(k, y)$ for nonuniform mutation is

$$\Delta(k, y) = y \times U(0, 1) \times \max \left\{ 0.005, \left(1 - \frac{k}{K} \right)^b \right\}.$$

A careful reader will notice that nonuniform mutation invalidates the NGA global convergence proof, so uniform mutation can be used to strictly adhere to the framework.

A.10 Replacement of Old Generation

Although it is not an essential component of the NGA, we choose to implement *elitism* as an option to speed up convergence and conserve old niche centers so

Table II. Control Parameters for the NGA

Parameter	Functionality	Value
m_G	Population size	50
η	Linear ranking coefficient	$1 \leq \eta \leq 2, 1.5$
m	Mating restriction parameter (Algorithm 4)	10
p_3	Mutation probability	0.328
b	Nonuniform mutation parameter, the bigger b , the larger the non-uniformity	1.5
K	Used in non-uniform mutation	50
T_G	Maximum number of generations without continuous improvement for transition to local phase to occur	3
g_m	Minimum number of groups to form in Algorithm 3	3
δ_G	Minimum difference in fitness values to detect in Algorithm 3	User specified (default δ_C)
α_G	Grouping quantile parameter in Algorithm 3	0.1
n_0	Initial number of replications	User specified (default 5)
Δn	Incremental number of replications	$\log_{10}(\text{number of iterations})$

as to stabilize the growth of niches; “elitism” means that the best individuals, in particular, niche centers, are always preserved through evolutions. Elitism is routinely used in GAs to speed up convergence. However, a consequence of using elitism is that it invalidates the global convergence property of the NGA, as proved in Appendix A of the published paper. If this is a concern, then the user can turn elitism off and thus preserve the global convergence property of the NGA. In our experience elitism helps, and since we transition from the global phase long before anything like global convergence occurs, this is a useful practical compromise.

Algorithm 5. GENERATION UPDATE

Step 1. If elitism is turned on, do step 2.

Step 2. Niche center conservation:

(1) Mark all individuals in the new population as unprocessed.

(2) For each old niche center \mathbf{x}_i , scan the unprocessed individuals in the new population. If the worst individual that is within the niche radius r from \mathbf{x}_i is inferior to \mathbf{x}_i , then replace that individual with \mathbf{x}_i .

(3) If there is no such individual, replace the worst unprocessed individual in the new population with \mathbf{x}_i .

(4) Mark the modified individual as processed and go back to 2.

A.11 GA Control Parameters

Table II lists control parameters for the NGA in ISC.

A.12 Transition Rules

In practice, we terminate the NGA when conditions indicate that we have uncovered fertile regions for seeking locally optimal solutions. If the budget for the NGA is exhausted, we also terminate it and move on to the local search phase.

A.12.1 Budget Rule. If the user specifies a budget for the NGA in terms of number of replications, then at the beginning of each iteration the NGA checks whether that budget is exceeded. If it is, then the NGA is terminated; otherwise the NGA proceeds to generate a new population. Notice that this is a *soft* budget rule because as long as there are some replications available at the beginning of an iteration, the NGA will continue.

A.12.2 Niche Rule. If at any time there is only 1 niche, then go to the local phase. The philosophy is that when there is only one niche it suggests that there is likely only one local minimum and thus running local search alone is sufficient.

A.12.3 Improvement Rule. If the NGA does not achieve any improvement in T_G generations, that is, no apparently better solutions are found in any of T_G consecutive generations, then go to the local phase. This is similar to a regular GA stopping criterion except that we may want to set T_G smaller, say $T_G = 3$.

A.12.4 Dominance Rule. Let q be the number of niches and let \mathbf{x}_{ij} be the j^{th} solution in niche i , $i = 1, 2, \dots, q$, $j = 1, 2, \dots, q_i$. Further let n_{ij} be the number of observations collected for \mathbf{x}_{ij} , let $n_i = \sum_{j=1}^{q_i} n_{ij}$, and let $v_i = n_i - q_i$.

If the solutions within one niche dominate all other niches (on the whole, not just the best individual), then we exploit it by going to the local phase.

The model used here is that the ℓ^{th} replication of solution \mathbf{x}_{ij} gives a fitness value $Y_{ij\ell} = \mu_{ij} + \varepsilon_{ij\ell}$, where $\varepsilon_{ij\ell}$'s are i.i.d. and are distributed as $N(0, \sigma_t^2)$. We assume equal variance within a niche population. We denote by $t_{v,\beta}$ the β quantile of the t distribution with v degrees of freedom. We further define

$$\begin{aligned} SSE_i &= \sum_{j=1}^{q_i} \sum_{\ell=1}^{n_{ij}} (Y_{ij\ell} - \bar{Y}_{ij\cdot})^2 \\ &= \sum_{j=1}^{q_i} \sum_{\ell=1}^{n_{ij}} Y_{ij\ell}^2 - \sum_{j=1}^{q_i} n_{ij} \bar{Y}_{ij\cdot}^2 \end{aligned}$$

where $\bar{Y}_{ij\cdot} = \sum_{\ell=1}^{n_{ij}} Y_{ij\ell} / n_{ij}$, and

$$\begin{aligned} \mu_i &= \frac{1}{q_i} \sum_{j=1}^{q_i} \mu_{ij}, \\ \hat{\mu}_i &= \frac{1}{n_i} \sum_{j=1}^{q_i} \sum_{\ell=1}^{n_{ij}} Y_{ij\ell}. \end{aligned}$$

Let $I = \{i : \hat{\mu}_i \geq \hat{\mu}_j - \omega_{ij}, \forall j \neq i\}$, where

$$\omega = \sqrt{\left(t_{v_i, \beta}^2 \frac{SSE_i}{v_i n_i} + t_{v_j, \beta}^2 \frac{SSE_j}{v_j n_j}\right)},$$

$$\beta = (1 - \alpha)^{1/(q-1)}.$$

Thus, the set I contains niches that are statistically better than others in terms of average quality of individuals within those niches. If $|I| = 1$, then we believe that there is a dominant niche and thus stop the NGA to move on to explore that niche with the local search algorithm.

A.13 Setting Up COMPASS After the NGA Finishes

Upon finishing the global phase, we have at hand a list of niche centers together with surrounding solutions. For each niche center, starting from the best one, we run COMPASS using that center as the starting point and surrounding solutions as active solutions.

B. COMPASS IN ISC

B.1 Generalized COMPASS Constraint Placement

The idea behind generalized constraint placement is to approximate the decrease in objective value as linear along the line connecting a solution \mathbf{x}_i to the current sample best $\hat{\mathbf{x}}^*$, and to make the constraint hyperplane as close to $\hat{\mathbf{x}}^*$ as possible subject to the estimated difference being statistically significant, but no more than half way. The generalized COMPASS constraint is

$$(\hat{\mathbf{x}}^* - \mathbf{x}_i)(\mathbf{x} - (\beta \hat{\mathbf{x}}^* + (1 - \beta)\mathbf{x}_i)) \geq 0, \beta \in [0, 1/2]. \quad (7)$$

Notice that when $\beta = 0$, the constraint simplifies to $(\hat{\mathbf{x}}^* - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i) \geq 0$, which means that $\hat{\mathbf{x}}^*$ and \mathbf{x} are on the same side with respect to \mathbf{x}_i geometrically, and thus nothing is cut off. If $\beta = 1/2$, then it is the usual COMPASS constraint.

One problem with the generalized COMPASS constraint placement algorithm is that when objective values of active solutions are very close to the current sample best solution, β is very likely to be set to zero, cutting nothing off. Therefore, the algorithm would fail to declare a singleton and the stopping test procedure would not be invoked; that is, COMPASS would not terminate even upon hitting a local optimum. To circumvent this difficulty, we impose a lower bound ε on the value of β and then relate it to the dimension of the solution space. Intuitively, the higher the dimension is, the smaller β should be to avoid unnecessarily cutting off too many points. We used $\varepsilon = \eta^d$, with $\eta = 0.5$, where d is the dimension of a solution \mathbf{x} .

Assuming that we have independent sampling with unequal sample sizes and unequal variances, the smallest difference between $\tilde{G}(\hat{\mathbf{x}}^*)$ and $\tilde{G}(\mathbf{x}_i)$ we can detect is

$$t_{1-\alpha, \nu} S, \quad (8)$$

where the standard error of the point estimate $S \equiv \text{standard error}(\tilde{G}(\hat{\mathbf{x}}^*) - \tilde{G}(\mathbf{x}_i))$ and approximate degrees of freedom ν are given by Banks et al. [2004].

We have

$$S = \sqrt{\frac{S^2(\mathbf{x}_i)}{n_i} + \frac{S^2(\widehat{\mathbf{x}}^*)}{n}} \quad (9)$$

$$\nu = \frac{(S^2(\mathbf{x}_i)/n_i + S^2(\widehat{\mathbf{x}}^*)/n)^2}{[(S^2(\mathbf{x}_i)/n_i)^2/(n_i - 1)] + [(S^2(\widehat{\mathbf{x}}^*)/n)^2/(n - 1)]} \quad (10)$$

where $S^2(\mathbf{x}_i)$ and $S^2(\widehat{\mathbf{x}}^*)$, and n_i and n , are sample variances and number of observations, respectively, for \mathbf{x}_i and $\widehat{\mathbf{x}}^*$. The degrees of freedom ν is rounded to an integer in the preceding expression. We require that $n \geq 2$, $n_i \geq 2$, and recommend that $n \geq 5$, $n_i \geq 5$.

We will set $\beta \in [\varepsilon, 1/2]$ based on the statistical test, unless the Euclidean distance between $\widehat{\mathbf{x}}^*$ and \mathbf{x}_i is less than or equal to \sqrt{d} , in which case we set $\beta = 1/2$ and do not apply the test. Here \sqrt{d} is the distance to the farthest point in the unit hypercube.

Algorithm 6 gives the procedure to determine β .

Algorithm 6. COMPASS CONSTRAINT PLACEMENT

Step 0. If $\|\widehat{\mathbf{x}}^* - \mathbf{x}_i\| \leq \sqrt{d}$, set $\beta = 1/2$ and return; otherwise go to Step 1.

Step 1 Calculate S and ν by (9) and (10).

Step 2. If $|\bar{G}(\mathbf{x}_i) - \bar{G}(\widehat{\mathbf{x}}^*)| < t_{1-\alpha_p, \nu} S$, then let $\beta = \varepsilon$; otherwise, let

$$\beta = \min \left\{ 1 - \frac{t_{1-\alpha_p, \nu} S}{|\bar{G}(\mathbf{x}_i) - \bar{G}(\widehat{\mathbf{x}}^*)|}, \frac{1}{2} \right\}.$$

C. COMPASS SAMPLE ALLOCATION RULE

The base Sample Allocation Rule (SAR) for COMPASS is that every active solution has at least $n_0(\ln k)^{1.01}$ replications, where k is the number of COMPASS iterations; this is the rule used in Hong and Nelson [2006]. Thus, active solutions receive an infinite number of replications in the limit which guarantees the convergence of COMPASS.

However, as COMPASS progresses toward locally optimal solutions, it may be desirable to adaptively allocate additional simulation effort beyond that specified by the fixed schedule. Intuitively, any solution that is not statistically inferior to the current sample best should be allocated somewhat more replications than those that are not so that their status can become clearer. To accomplish this, we also implemented an adaptive SAR based on OCBA (see, for instance, Chen et al. [2000]). In COMPASS as implemented in ISC (Algorithm 8 in Section C.1 that follows) we apply the following OCBA heuristic in step 1 (an option that can be turned off by the user).

Again, let $S^2(\mathbf{x}_i)$ and $S^2(\widehat{\mathbf{x}}^*)$ denote sample variances for \mathbf{x}_i and $\widehat{\mathbf{x}}^*$. Let $\widehat{\delta}(\mathbf{x}) = \bar{G}(\mathbf{x}) - \bar{G}(\widehat{\mathbf{x}}^*)$. Assume that at each iteration we have ΔN additional replications to allocate to active solutions that define the most promising area. In our implementation, we fix ΔN to the size of the set of active solutions, denoted by \mathcal{A}_{k-1} . Note that these are *additional* replications.

Algorithm 7 describes the OCBA-based adaptive SAR.

Algorithm 7. ADAPTIVE SAR FOR COMPASS

Step 0. At iteration k , let $\Delta N = |\mathcal{A}_{k-1}|$.

Step 1. Allocate 2 additional replications to $\widehat{\mathbf{x}}^*$.

Step 2. Let $\Delta N = \Delta N - 2$.

Step 3. Let $R = \sum_{\mathbf{x} \in \mathcal{A}_{k-1}} S^2(\mathbf{x}) / \widehat{\delta}(\mathbf{x})$.

Step 4. Set $\Delta N(\mathbf{x}) = \frac{S^2(\mathbf{x}) / \widehat{\delta}(\mathbf{x})}{R} \cdot \Delta N$. If $\Delta N(\mathbf{x}) < 1$, round down to 0; otherwise, round to the nearest integer.

C.1 COMPASS with Constraint Pruning

When the k^{th} iteration begins, we have:

- a Most Promising Area (MPA) C_k defined by
 - $\widehat{\mathbf{x}}_{k-1}^*$: The sample best solution from previous iteration, which is the user-specified starting point \mathbf{x}_0 when $k = 1$.
 - a set of active solutions, denoted by $\mathcal{A}(\widehat{\mathbf{x}}_{k-1}^*)$, plus the original problem constraints that define the constraints actually forming the MPA. For notational simplicity, we shall use \mathcal{A}_{k-1} instead of $\mathcal{A}(\widehat{\mathbf{x}}_{k-1}^*)$ in the following.
- a list of visited solutions \mathcal{V}_{k-1} , which is $\{\mathbf{x}_0\}$ and the active solutions provided by the NGA, \mathcal{A}_0 , when $k = 1$.
- a list of the sample best visited solutions, denoted as \mathcal{B}_{k-1} , in \mathcal{V}_{k-1} . The size of the list is usually set to the dimension of the solution space d ; $\mathcal{B}_{k-1} = \{\mathbf{x}_0\}$ for $k = 1$.

Algorithm 8. COMPASS WITH CONSTRAINT PRUNING

Step 0. Let $k = 0$, $\mathcal{V}_0 = \{\mathbf{x}_0\}$, and \mathcal{A}_0 be the active solutions provided by the NGA. Let $\mathcal{C}_0 = \Theta$.

Step 1. Let $k = k + 1$. Let $\mathcal{B}_{k-1} = \emptyset$. Uniformly and independently sample m_L solutions (duplicates allowed) from C_{k-1} (see Section A.2). Denote this set of solutions as S_k . Apply SAR to $\widehat{\mathbf{x}}_{k-1}^*$, \mathcal{A}_{k-1} and S_k , and let $\mathcal{V}_k = \mathcal{V}_{k-1} \cup S_k$.

Step 2. Identify \mathcal{A}_k by solving LP programs formed by $\widehat{\mathbf{x}}_k^*$ as the center, and S_k , \mathcal{A}_{k-1} , \mathcal{B}_{k-1} , and $\widehat{\mathbf{x}}_{k-1}^*$ (if $\widehat{\mathbf{x}}_{k-1}^* \neq \widehat{\mathbf{x}}_k^*$), as candidate active solutions, and thus obtain C_k .

Step 3. Check if C_k is a singleton. If not, go to step 1; else, apply stopping test procedure to $\widehat{\mathbf{x}}_k^* \cup \mathcal{A}_k$ using $\widehat{\mathbf{x}}_k^*$ as the standard. If $\widehat{\mathbf{x}}_k^*$ is still optimal, return $\widehat{\mathbf{x}}_k^*$ as the final result; else, let $\widehat{\mathbf{x}}$ be the new sample best, let $\widehat{\mathbf{x}}_{k-1}^* \leftarrow \widehat{\mathbf{x}}_k^*$, $\mathcal{A}_{k-1} \leftarrow \mathcal{A}_k$, $\widehat{\mathbf{x}}_k^* \leftarrow \widehat{\mathbf{x}}$, populate \mathcal{B}_{k-1} with the d sample best solutions in \mathcal{V}_k , and go to step 2.

C.2 COMPASS Control Parameters

Table III lists the control parameters for COMPASS.

D. CLEAN-UP

Once the COMPASS local search has exhausted all niches found by the NGA global search, or the budget has run out, ISC enters the clean-up phase with a

Table III. COMPASS Control Parameters

Parameter	Functionality	Default value
n_0	Initial number of replications assigned to a new solution	User specified (default 5)
m_L	Within the most promising area, randomly sample with replacement m_L solutions	5
N_k	The minimum number of replications assigned to a solution at iteration k	$n_0(\ln k)^{1.01}$
α_p	$1 - \alpha_p$ is the confidence level for generalized COMPASS constraint placement method	0.10
δ_L	Indifference zone parameter for the local optimality stopping test.	User specified (default δ_C)
α_L	$1 - \alpha_L$ is the confidence level for the local optimality stopping test.	User specified (default 0.05)
cp	Number of COMPASS iterations between constraint prunings	User specified (default 5)

collection of locally optimal solutions that we reindex as $\mathcal{L} = \{1, 2, \dots, |\mathcal{L}|\}$ for convenience.

We want to select the best of these solutions, or one within δ_C of the best, with high confidence, and also state a $\pm\delta_C$ confidence interval on the performance of the selected solution. The user specifies parameters δ_C and α_C , but α_C defaults to 0.05.

Algorithm 9. CLEAN-UP PROCEDURE

Input. \mathcal{L} , the set of locally optimal solutions of size $|\mathcal{L}|$ and constants δ_C and α_C . For each $i \in \mathcal{L}$, $N_C^{(0)}(\mathbf{x}_i)$ is the number of observations, $\bar{G}(\mathbf{x}_i)$ is the sample mean, and $S^2(\mathbf{x}_i)$ is the sample variance.

Step 1. Let

$$\begin{aligned} \underline{n} &= \min_{i \in \mathcal{L}} N_C^{(0)}(\mathbf{x}_i), \\ t_i &= t_{(1-\alpha_C/2)^{1/(|\mathcal{L}|-1)}, N_C^{(0)}(\mathbf{x}_i)-1}, \\ h &= h(2, (1-\alpha_C/2)^{1/(|\mathcal{L}|-1)}, \underline{n}) \\ \omega_{i\ell} &= \left\{ \frac{t_i^2 S^2(\mathbf{x}_i)}{N_C^{(0)}(\mathbf{x}_i)} + \frac{t_\ell^2 S^2(\mathbf{x}_\ell)}{N_C^{(0)}(\mathbf{x}_\ell)} \right\}^{1/2}, \quad \forall i \neq \ell \end{aligned}$$

where $h(2, (1-\alpha_C/2)^{1/(|\mathcal{L}|-1)}, \underline{n})$ is Rinott's constant in the special case of 2 solutions, confidence level $(1-\alpha_C/2)^{1/(|\mathcal{L}|-1)}$ and \underline{n} degrees of freedom (see Boesel et al. [2003]).

Step 2. Let $I = \{i : \bar{G}(\mathbf{x}_i) \leq \bar{G}(\mathbf{x}_\ell) + \omega_{i\ell}, \forall \ell \neq i\}$.

Step 3. For all $i \in I$, compute $N_C(\mathbf{x}_i) = \max\{N_C^{(0)}(\mathbf{x}_i), \lceil h^2 S^2(\mathbf{x}_i)/\delta^2 \rceil\}$. Collect $N_C(\mathbf{x}_i) - N_C^{(0)}(\mathbf{x}_i)$ more observations and update $\bar{G}(\mathbf{x}_i)$.

Step 4. Let $B = \arg \min_{i \in I} \bar{G}(\mathbf{x}_i)$. Report \mathbf{x}_B as the best solution and claim $\bar{G}(\mathbf{x}_B) \pm \delta_C$ as the $(1-\alpha_C/2) \times 100\%$ confidence interval for $g(\mathbf{x}_B)$.

Table IV. Control Parameters used in Test Problems

	Multi-modal	Singular	Flowline	Inventory	High- d
n_0	5	5	5	20	5
δ_G	0.08	1	0.05	0.1	10
α_b	0.05	0.05	0.05	0.05	0.05
δ_b	0.08	1	0.05	20	10
α_L	0.05	0.05	0.05	0.05	0.05
δ_L	0.08	1	0.05	50	10
α_C	0.05	0.05	0.05	0.05	0.05
δ_C	0.08	1	0.05	20	10
cp	5	5	5	5	50

E. USER-SPECIFIED PARAMETERS FOR TEST PROBLEMS

In Table IV we list user-specified ISC control parameters using the notation introduced earlier. The only parameter that does not have a default is δ_C .

REFERENCES

- BAKER, J. E. 1987. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the 2nd International Conference on Genetic Algorithms*. 14–21.
- BANKS, J., CARSON, J. S., NELSON, B. L., AND NICOL, D. M. 2004. *Discrete-Event System Simulation*, 4th ed. Pearson Prentice Hall, Upper Saddle River, NJ.
- BOESEL, J. 1999. Search and selection for large-scale stochastic optimization. Doctoral dissertation, Department of IEMS, Northwestern University, Evanston, IL.
- BOESEL, J., NELSON, B. L., AND KIM, S.-H. 2003. Using ranking and selection to ‘clean up’ after simulation optimization. *Oper. Res.* 51, 814–825.
- CHEN, C.-H., LIN, J., YÜCESAN, E., AND CHICK, S. E. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discr. Event Dynam. Syst. Theory Appl.* 10, 251–270.
- HONG, L. J. AND NELSON, B. L. 2006. Discrete optimization via simulation using COMPASS. *Oper. Res.* 54, 115–129.
- MICHALEWICZ, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Algorithms* 3rd Ed. Springer, Berlin.
- MILLER, R. G. 1981. *Simultaneous Statistical Inference* 2nd Ed. Springer, Berlin.
- MILLER, B. L., AND SHAW, M. J. 1995. Genetic algorithms with dynamic niche sharing for multimodal function optimization. IlliGAL rep. No. 95010, Illinois Genetic Algorithms Laboratory, University of Illinois at Champaign-Urbana.
- PICHTLAMKEN, J., AND NELSON, B. L. 2003. A combined procedure for optimization via simulation. *ACM Trans. Model. Comput. Simul.* 13, 155–179.
- SARENI, B. AND KRAHENBUHL, L. 1998. Fitness sharing and niching methods revisited. *IEEE Trans. Evol. Comput.* 2, 97–106.