Optimizing Global-Local Search Hybrids

David E. Goldberg
Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801 USA

Siegfried Voessner
Department of Engineering-Economic Systems
and Operations Research
Stanford University
Stanford, California 94305 USA

Abstract

This paper develops a framework for optimizing global-local hybrids of search or optimization procedures. The paper starts by idealizing the search problem as a search by a global algorithm G for either (1) acceptable *targets*—solutions that meet a specified criterion—or for (2) *basins of attraction* that then lead to acceptable targets under a specified local search algorithm L. The paper continues by abstracting two sets of parameters—probabilities of successfully hitting targets and basins and time-to-criterion coefficients—and writing equations to account for the total time of search and for the reliability in reaching an acceptable solution. A two-basin optimality criterion is derived and applied to important representative problems. Continuations and extensions of the work are suggested, but the theory appears to be useful immediately in better understanding the economy of effective hybridization.

1 INTRODUCTION

Many industrial-strength genetic and evolutionary algorithms (GEAs) are explicit hybrids, combining the underlying GEA with one or more problem-specific local search procedures. While a good argument can be made that all GEAs are themselves implicit hybrids because they combine the actions of selection on the one hand with one or more variation operators, the usual motivation for hybridization in optimization practice is the achievement of increased *efficiency*. That is, practitioners seek adequate solution quality in minimum time or they seek maximum quality in specified time, although it is rare that practitioners have been even this precise in stating their goals. Moreover, when any theory has been adopted in the formulation of hybrid practice, it has been a micro-level theory, mainly useful for detailed operator design. There still remains a need to better understand efficient hybrids at the macro- or systems level.

This paper considers a systems-level theoretical framework for creating efficient hybrids of different optimization procedures. Although the paper is motivated by the practice of genetic and evolutionary algorithms, the framework is fairly general and should work well in other settings.

The paper starts with a brief review of hybridization in GEA practice and theory. It continues by establishing the idealization of optimization search methods and space that will enable us to construct a systems-level theory. Thereafter, the time accounting and reliability conditions are written and solved, yielding either maximum reliability in fixed time or minimum time to a specified reliability. The framework of optimizing optimization hybrids is then applied to a number of representative situations. The paper concludes with some comments on how these abstractions might be verified, used, and extended in real hybrid settings.

2 GEA HYBRIDS-A REVIEW

It is part of the folklore of GEAs that hybrids often improve the efficiency of search (Ibaraki, 1997). Smith (1985) and Grefenstette, Gopal, Rosmaita, and Van Gucht (1985) presented early hybrids algorithms in relatively small prototype systems, and Powell, Tong, and Skolnick (1989) were among the first to incorporate hybridization techniques in their commercially viable system for design of a gas turbine engine. Davis has perhaps been the foremost advocate of GEA hybridization (Davis, 1991), to the point where today, it is rare that the serious GEA application is undertaken without some kind of GEA combined with some specialized search method (Goldberg, 1994).

Along the way, theoreticians have made useful contributions to the state of hybrid knowledge. The distinction made between Baldwinian and Lamarckian learning by Hinton and Nowlan (1987) is particularly important and suggests that a local search method appended to a GEA can have a useful effect without backsubstituting the genotype corresponding to the termination point of the local search algorithm. Orvosh and Davis (1993) support this point of view with their empirically derived *rule of 20* that suggests that a Lamarckian step—if used at all—should only be used one in twenty trials in order that population diversity not be overly disturbed.

Another theoretical thread picked up in the literature is that of adaptive frequency of operator use. A number of practitioners have viewed the probabilistic mixture of different operators as something that should be adapted to achieved various search goals. This was picked up early and often in the literature of evolution strategies (see Bäck & Schwefel, 1995, for a good survey) and the same theme has been picked up elsewhere (Davis, 1989; Shaefer, 1987) fairly early in the genetic algorithm tradition. More recently, this theme has been mapped to the k-armed bandit problem by Lobo and Goldberg (1997) who recast the problem in stochastic automaton form and perform empirical testing on a simple test problem.

Whitley (1995) formulated the precise difference equation and Markov chain models for a genetic algorithm hybridized with some specified number of steps of local search in either Lamarckian or Baldwinian updates. The formulation is clever and exact; however, as anyone who has tried to apply results from exact difference or Markov chain equations knows, the sledding is tough and usually thwarts the kind of systems-level understanding we are seeking here.

The foregoing contributions have been important to hybrid practice, but the detailed or micro-level analysis common across these studies leaves us without a macro-level framework of hybrid efficiency, a matter to be taken up in the next section.

3 THE PROBLEM, 2 SEARCHERS, AND THEIR HYBRID

To build a macro-level model of optimization hybrids that will allow us to address the quality-efficiency tradeoff requires that we abstract critical features of three things: (1) the search problem to be solved, (2) the searchers to be combined, and (3) the hybrid scheme to coordinate the searchers. This section constructs that framework, which draws heavily on ideas developed elsewhere (Goldberg, 1991).

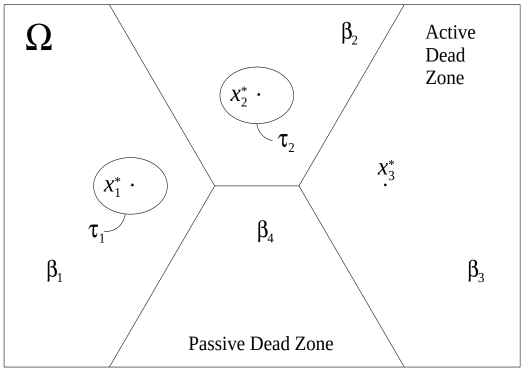

Figure 1: A two-dimensional sketch of an idealized search space depicts target islands τ_i, basins of attraction under local method L to those targets, β_i, and two types of *dead zone*, active and passive.

Consider an *idealized hybrid algorithm* (IHA or simply H) operating over a solution space Ω on a minimization problem $\phi : \Omega \to R$ (maximization may be accommodated with appropriate reversal of inequality and sign in what follows). The hybrid H works by coordinating the activities of a global method G and a local method L as follows. Each iteration, G is invoked once (taking unit time, without loss of generality) to generate some new candidate solution and this is followed by multiple invocations of L consuming no more than an allowable time $\lambda_a : 0 \leq \lambda_a \leq \lambda_{max}$. This process proceeds until either an allowable time T_a is exceeded or a solution *accuracy* target value $\phi \leq \phi_\tau$ is reached.

This straightforward outline of H contains our first clues as to how we must view the solution space Ω if we are to make some progress on the efficiency question. Given that we wish solutions better than some target value (ostensibly within some $\Delta\phi$ of the global, $\phi_\tau = \phi^* + \Delta\phi$), we first identify the level set with value ϕ_τ and better, subdivide the level set into one or more discrete island targets τ_i as depicted in figure 1, and then consider how G and L might combine to lead us to one of the targets.

The first thing to understand is that G may be successful all on its own. Calling the union of the targets the global region, $R_G = \bigcup_i \tau_i$, we denote the probability of hitting R_G in a single invocation of the global searcher P_G. We do not consider the calculation of P_G in detail except to note that with G taken as uniform random search, P_G may be calculated by summing the areas (hypervolumes) of the targets and dividing by the total area (hypervolume) of the space. With random search done according to some other probability distribution, this calculation becomes somewhat more complex, requiring us to integrate the distribution over the target region, and with a more involved global searcher, the calculation may become even more difficult still. In fact, for searchers other than random search, these probability parameters may not be stationary; however, here we assume that they are or that they vary slowly enough that constant values provide a useful approximation.

Outside of R_G, G needs help from L, but to describe the cost of this help and its interaction with the invocation of L, we need to further idealize the search space with two families of parameters. Recognizing that local search is a dynamic system, we know that it usually works by iterating from some starting position to some solution. We simplify things somewhat by assuming that in the usual case L converges toward a fixed-point solution x_i. We define the local time-to-criterion (TTC) values λ_i as the average number of time units required to get to the target starting from within the basin of attraction β_i. In figure 1, the basins are shown as tessellation polygons, but of course, more general basin geometries should be expected to occur. The key thing is that we've identified the local time constant values λ_i to help quantify times of arrival. In reality, different starting points within a basin may result in different arrival times to the basinic solution, and a more accurate formulation would treat arrival times probabilistically, but here we keep things simple and interpret the local TTC constants as mean values over the basin.

The other parameter needed in our analysis is suggested by the basins β_i. In difficult problems, hitting the targets directly is unlikely, but the chances of hitting one of the basins and converging with L to a target is quite good. To quantify this, we say that the probability of hitting the basin β_i (exclusive of the target τ_i) with an invocation of G is P_i. Again, as with P_G, it is fairly straightforward to imagine a calculation of the P_i when G is chosen as random search with some fixed and known probability distribution.

One additional feature of the space should be mentioned before optimizing our idealized hybrid. Suppose instead of landing in the global zone or one of the tractable basins, G places the search at points in the space that do not lead to the global zone under local search. There are two cases to consider. Suppose G lands in a basin such as β_3 in which local search leads to a solution that does not meet criterion ($\phi > \phi_\tau$). Or worse, suppose G goes to a basin (β_4) where L fails to converge in $\lambda \le \lambda_{max}$ time units. We call both of these types of regions *dead zones*. The first, a *type I dead zone* is distinguished in that L converges to a solution, but the solution is inadequate. The second, a *type II dead zone*, is distinguished in that there is no indication of convergence and if the solution were permitted to continue, it would consume all remaining computational time. Although dead zones are not places we wish G to land, it is useful to calculate the probability of hitting the dead zone (region R_D) as follows:

$$P_D = 1 - P_G - \sum_i P_i \tag{1}$$

In words, the dead zone is what's left over when the global zone and tractable basins are removed, and the probability of hitting the dead zone is the complement of hitting the union of the global zone and tractable basins. Note that we only count as dead zone those basins that can never reach an acceptable solution under the permitted variation in λ_a. In optimizing the division of computation between global and local search, we may choose to ignore some of the tractable basins, because L will take too long on them, but this is exactly the decision we are trying to highlight (and make).

With these definitions, we are now in a position to consider time and reliability together, thereby optimizing our hybrid, a matter taken up in the next section.

4 TIME AND RELIABILITY

With the idealization of the last section under our belts, we examine the key relationships between the parameters of the last section and overall solution time and reliability. These together with appropriate optimization conditions will allow us to decide how to allocate our time wisely between local and global search. We start by accounting for the division of time between L and G and then calculate the probability of meeting criterion.

Accounting for time is straightforward. The key decision we make in setting up the hybrid is determining the split between local and global search. Calling the allowable local time constant λ_a, and the average local time constant $\bar{\lambda}$, we may write the solution time T consumed in n global-local iterations as follows:

$$T = (1 + \bar{\lambda})n. \tag{2}$$

Here, without loss of generality, global search is assumed to occupy unit time, and the cost of L is measured relative to that value. The relationship between

λ_a and $\bar{\lambda}$ depends on the rules of H and will be explored in a moment, but we now turn to the calculation of the reliability relationship.

Our goal is to have the hybrid H converge to one of the targets, but with a limit on the time that local search can operate, only those targets with local time-to-criterion constant values less than the allowable can be counted as successes. Thus, the probability of a successful hybrid iteration may be determined by summing the probability of hitting the global zone initially and the probability values of all those targets with TTC constants less than the allowable. Calling this probability P_{λ_a}, we calculate as follows:

$$P_{\lambda_a} = P_G + \sum_{i:\tau_i \neq \emptyset, \lambda_i \leq \lambda_a} P_i \qquad (3)$$

In words, the probability of success at level λ_a is the probability of hitting the global zone with G plus the sum of probabilities of G hitting any basin that leads to a target in time less than the allowable setting.

With the single trial calculation in hand, the success probability P_s of at least one success in n global-local iterations is given by elementary probability as

$$P_s = 1 - (1 - P_{\lambda_a})^n \qquad (4)$$

Once the allowable limit λ_a is chosen, the reliability and time conditions can be interrelated, but this requires one auxiliary relationship that depends on the rules of H.

In some hybrids, we might imagine that L is run λ_a time units regardless of the convergence status of the local searcher. In such cases, $\bar{\lambda} = \lambda_a$ and the calculations proceed quite conveniently. In those cases where λ_a is treated as an upper limit on the run time of L, $\bar{\lambda}$ needs to be calculated as follows:

$$\bar{\lambda} = \sum_{i:\tau_i \neq \emptyset, \lambda_i \leq \lambda_a} P_i \lambda_i + \sum_{i:\tau_i \neq \emptyset, \lambda_a < \lambda_i \leq \lambda_{max}} P_i \lambda_a + P_D \lambda_a \qquad (5)$$

This latter calculation is somewhat cumbersome, and the case of $\bar{\lambda} = \lambda_a$ is used in the remainder.

5 TWO WAYS TO OPTIMIZE

The primary decision to make in efficient hybridization is how much to spend in global search versus local search. In the formulation herein, this requires us to choose the allowable time spent in local search λ_a; however, we have two ways to go about this business. Given that we would like to achieve a solution of specified *accuracy*, we may either maximize the probability of achieving the solution—maximize the *reliability*—under a fixed time or we may minimize the time to achieve a solution of *specified* reliability. In this section, we formulate these two optimization problems using the equation of the previous section.

5.1 MINIMUM TIME

Minimizing the time subject to a given reliability is most easily done by recasting the problem in terms of the probability of not reaching criterion. We will call this the *probabilistic error* or simply the *error* and use the symbol α to denote the quantity. Assuming a specified allowable error α_a, the reliability condition may be rewritten as $\alpha_a = (1 - P_{\lambda_a})^n$. Taking the logarithm and rearranging for n yields

$$n = \frac{\ln \alpha_a}{\ln(1 - P_{\lambda_a})}. \qquad (6)$$

We recognize that n is restricted to the positive integers, but for the minimum time formulation this condition will not be of much consequence. We substitute this relationship into the time accounting equation, minimize, and obtain the following:

$$\min \quad (\lambda_a + 1)\frac{\ln \alpha_a}{\ln(1 - P_{\lambda_a})} \qquad (7)$$

Under known basin and target probabilities as well as local search time constants, this condition may be used to determine the optimal allowable local time constant, λ_a^*.

5.2 MAXIMUM RELIABILITY

Maximizing the reliability (minimizing the error) results in a similar condition with different specified coefficients and constraints. Assuming an allowable total time T_a and rearranging the time relation for n yields $n = \frac{T_a}{\lambda_a + 1}$. Although n must again be a positive integer, our main concern is that we be able to perform at least one global-local iteration. This condition suggests that the maximum allowable time value be set as

$$\lambda_{max} \leq T_a - 1. \qquad (8)$$

Substituting the expression for n into the reliability condition and minimizing the error yields the condition as follows:

$$\min \quad (1 - P_{\lambda_a})^{\frac{T_a}{\lambda_a + 1}} \qquad (9)$$

Comparison of equations 7 and the logarithm of equation 9 shows that they are equivalent after allowing for different constants ($\ln \alpha_a$ versus T_a) and their differing signs ($\ln \alpha$ is negative, and T_a is positive). Although the integer constraint on n exists for both, in practice it places a greater restriction on the reliability formulation, because we may often be called to impose the single iteration constraint (equation 8)

6 SOLUTION IN TWO BASINS

The conditions of the last section are fairly general, but to use them in practice, we consider the case of two basins in some detail. Although a two-basin solution sounds restrictive to the point of impracticality, the condition may be applied in the general discrete case with proper interpretation. Moreover, some rearrangement of the two-basin optimality condition gives us important physical insight into the tradeoff between global and local search.

Consider two basins with basin probabilities and local time constants (P_1, λ_1) and (P_2, λ_2) with $\lambda_1 < \lambda_2$. With only two basins there are only two possibilities to consider, either $\lambda_a^* = \lambda_1$ or $\lambda_a^* = \lambda_2$ and it is unclear ahead of time which should be preferred because the more expensive alternative (λ_2) may be fruitful if the cumulative probability of success $(P_1 + P_2)$ results in higher reliability (lower error).

Minimizing the error (equation 9) with fixed T_a suggests that we should prefer basin two if

$$(1 - P_1')^{T_a/\lambda_1'} \geq (1 - P_2')^{T_a/\lambda_2'} \qquad (10)$$

where $\lambda_i' = \lambda_i + 1$ and $P_i' = \sum_{j=1}^{i} P_j$. Some algebra yields

$$(1 - P_1')^{\lambda_2'/\lambda_1'} \geq 1 - P_2' \qquad (11)$$

For small P_1' this may be approximated by the condition

$$P_2' \geq \frac{\lambda_2'}{\lambda_1'} P_1' \qquad (12)$$

In words, it is worthwhile spending more on local search if the cumulative probability of success increases sufficiently. For small success probabilities that growth must increase at least as quickly as the inverse ratio of the global-local TTC constants.

Although the derivations of this section apply to two basins, the condition may be used in the general discrete case by first ordering all basins in λ_i order from low to high. Local optimality can be checked by moving through the list and successively checking the ith and $(i+1)$th basins. Global optimality can be determined by choosing the local optimum with least error.

Another approach is to consider a continuous probability distribution as a function of the total cost λ'. The solution to this formulation is identical to that published by Nakano, Davidor and Yamada (1994) in the context of genetic algorithm population size optimization convenient, and those interested should consult the original manuscript.

7 APPLICATION OF THE THEORY

Let us turn to applying the theory to a number of representative cases:

1. Choosing λ_a with $P_G = 0$, uniform λ_i
2. Choosing λ_a with $P_G > 0$, uniform λ_i
3. Analyzing change of λ_a^* under improvement in G
4. Analyzing change of λ_a^* under relaxation of criterion

In the remainder of this section, each of these is considered in turn.

7.1 CASE I: $P_G = 0$, UNIFORM λ_i

With little or no probability of G hitting the global zone, G cannot find a solution without L, and with uniform $\lambda_i = \lambda_0$ there is essentially no tradeoff. Either we can afford local search or not, and if we can, the correct setting for the allowable time to criterion parameter is $\lambda_a^* = \lambda_0$. Once local search is enabled, the only way to make a mistake is to hit the dead zone. In this case $P_D = 1 - \sum_i P_i$ and the optimal error may be written straight away:

$$\alpha^* = P_D^{T_a/(\lambda_0+1)} \qquad (13)$$

Figure 2 shows the optimal error as a function of the allowable time T_a.

The assumption of uniform λ is approximately met in local solvers that have rapid convergence rates that are approximately equal across many basins. The assumption of no global zone should be approximately met in difficult problems or in relatively easy problems where the success criterion is beyond the reach of global search.

7.2 CASE II: $P_G > 0$, UNIFORM λ_i

The first case was particularly simple because there was no tradeoff. We simply set the allowable time to criterion to the uniform value and let the algorithm (G+L) rip. With the introduction of a non-null global zone, the decision becomes more interesting. There is now a non-zero probability of hitting the target with G alone (the probability P_G). Thus, we must consider two possibilities. Either we set $\lambda_a = 0$ or we set λ_a

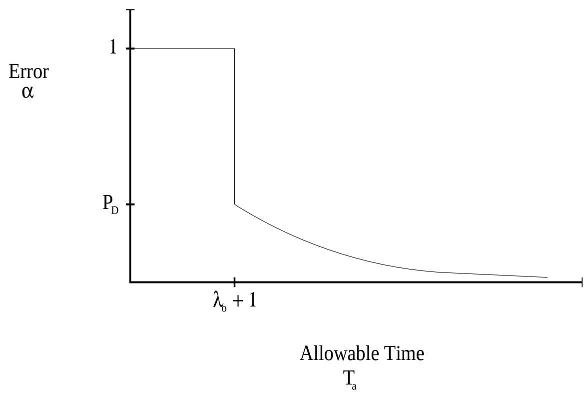

Figure 2: In case I with $P_G = 0$, the calculation is guaranteed to fail until there is sufficient allowable time to permit at least one single global-local iteration.

to the uniform value λ_0. Of course, this is a special case of our two-basin calculation earlier. The cumulative probability of hitting either the global zone or the basins leading to success is the complement of the dead zone probability. Thus, we should go with G alone when $(1 - P_G)^{T_a} < [1 - (1 - P_D)]^{T_a/\lambda_0'}$. Eliminating the allowable time and rearranging yields

$$P_D > (1 - P_G)^{\lambda_0'} \tag{14}$$

Thus for large global zones, large dead zones, or large uniform λ_0 values, we should choose G alone. When the situation is the efficient combination is to have G and L working together.

Another way to examine this case is to calculate the critical time to criterion value λ_c' where the error is the same with or without L. Setting $P_D = (1 - P_G)^{\lambda_c'}$ and solving for the critical value yields

$$\lambda_c' = \frac{\ln P_D}{\ln(1 - P_G)} \tag{15}$$

If the uniform TTC value λ_0' is greater than the critical value, go with G alone; otherwise use G and L together.

7.3 CASE III: IMPROVEMENT OF G

Suppose that we are using a global searcher G_1 and we have the opportunity to use an improved global search algorithm G_2. How should we expect the division between global and local search to change? Here we will assume the same model as in Case II. Thinking about this qualitatively is helpful. First, a better G should improve the probability of hitting the target with G alone. If the original global zone probability was P_G, imagine it being expanded by a factor $\chi \geq 1$. Second, an improved global searcher should also reduce the probability of hitting the dead zone, so we imagine it being reduced by a reduction coefficient ρ.

Using equation 15 permits the calculation of the ratio of the critical λ values in the improved G to that of the original:

$$\Lambda = \frac{\lambda_{c_2}'}{\lambda_{c_1}'} = \frac{\ln P_{D_2} / \ln P_{D_1}}{\ln(1 - P_{G_2}) / \ln(1 - P_{G_1})} \tag{16}$$

Since $P_{D_2} = \rho P_{D_1}$ and $P_{G_2} = \chi P_{G_1}$, Λ may be rewritten as follows

$$\Lambda = \frac{\gamma}{\delta} \tag{17}$$

where $\gamma = 1 + \ln\rho / \ln P_{D_1}$ and $\delta = \ln(1 - \chi P_{G_1}) / \ln(1 - P_{G_1})$. For solvable problems $P_{D_1} \neq 1$ and dead zone reduction ($\rho < 1$), $\gamma > 1$.

Interpreting δ is aided by recognizing that for small x, $\ln(1 - x) \approx -x$. Thus, $\delta \approx \chi$ and $\Lambda \approx \gamma/\chi$. Thus, the global zone enhancement and dead zone reduction tend to work against one another as should be expected. Global zone enhancement should tend to push the hybrid toward greater usage of G, whereas dead zone reduction under conditions of fixed R_G enlarge the probability of success under L + G.

7.4 CASE IV: RELAXED CRITERION

Suppose the user decides to relax the accuracy criterion. Again using the case II model, we recognize two effects. Reduction of the criterion makes it easier for global search to hit the target, and it also reduces the size of the dead zone. The former effect is fairly straightforward to envision, because it is easy to think of larger (relaxed) targets being easier for the global searcher to hit. The effect of criterion relaxation on the dead zone is less obvious, but we may reason as follows. A relaxation in criterion means that for certain type I dead zones under the previous criterion, the fixed points that previously were not sufficiently accurate to meet criterion will now pass muster. Also, certain type II (non-convergent) dead zones may also wander through function values that meet criterion. As a result the tendency under relaxation of criterion will be for the dead zone to reduce in size.

Notice that case IV is essentially the same as case III, and therefore the analysis of the ratio Λ may be used without modification.

8 2 EXPERIMENTS

This section presents the results of proof-of-principle experiments that support the foregoing theory. We

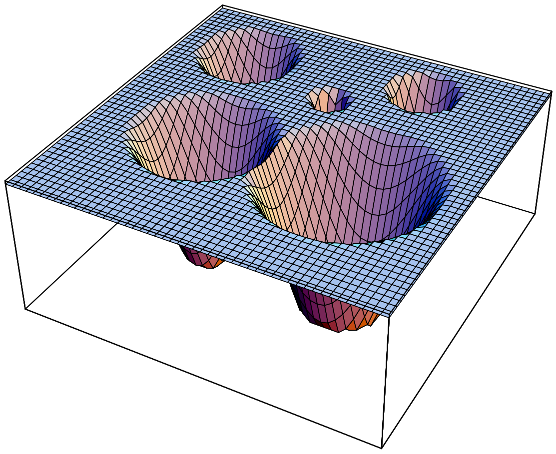

Figure 3: The testbed function $f(x,y)$ has five, quasi-concave basins of attraction as shown in this top view

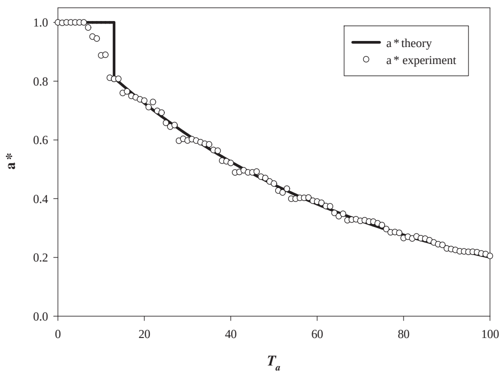

Figure 4: The probabilistic error α^* is shown as function of the allowable time T_a for both theory and experiment. The theory matches well as expected.

have chosen a simple two-dimensional function $f(x,y)$ to be minimized as our test bed. The two variables x and y are in the closed interval $[0,10]$, and the function consists of 5 quasi-concave basins (center $c_i = (cx_i, cy_i)$, radius r_i, depth d_i) and a surrounding area with function value

$$f(x,y) = \begin{cases} \frac{d_i}{r_i^2}\left(\bar{r}^2\right)\left(2 - \frac{\bar{r}^2}{r_i^2}\right) - d_i & \text{for } \bar{r}^2 \leq r_i^2 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where $\bar{x} = x - cx_i$, $\bar{y} = y - cy_i$, $\bar{r}^2 = \bar{x}^2 + \bar{y}^2$, and $c_i = \{(2.0, 8.0),\ (3.0, 4.0),\ (5.0, 7.0),\ (7.0, 8.5),\ (7.0, 4.0)\}$, $r_i = \{1.5, 2.0, 0.5, 1.0, 2.5\}$, $d_i = \{2.0, 3.0, 2.0, 4.0, 2.0\}$. Figure 3 displays the function. The global minimum is at $(7.0, 8.5)$ and has a value of -4.

The global search algorithm G has been chosen as random search with a uniform probability distribution over the space. The local search algorithm L has been chosen as a standard quasi-Newton algorithm (Press, Teukolsky, Saul, Vetterling, & Flannery, 1992), which has for geometrically similar basins almost a uniform λ. L takes $\bar{\lambda} \approx \lambda_0 = 7.0539$ time units on average. The probability of hitting the dead zone $P_D = 0.5760$. Both experiments have been carried out using simulation - termination criterion was a maximum error of 0.01%.

8.1 EXPERIMENT I: $P_G \approx 0$, UNIFORM λ_i

By setting ϕ_τ to -3.99 the chance of hitting a target P_G is nearly zero (0.0001).

In figure 4 the experimental probabilistic error α_{sim} obtained from simulation is plotted as a function of T_a and is compared to the theoretical results (equation 13). The plot shows a good match between theory and experiment.

8.2 EXPERIMENT II: LARGE P_G

By lowering the threshold for an acceptable solution, the global zone increases. In this case it is possible to find an acceptable solution by using local search only. For large global zones (assuming that λ_0 cannot be changed) it is more efficient to go just with global search if $P_D > (1 - P_G)^{\lambda'_0}$. We will show an example where the right thing to do is to use G alone (in Experiment I we go with G + L since $P_D < (1 - P_G)^{\lambda'_0}$).

Setting ϕ_τ to -1 leads to: $P_G = 0.1490$. Equation (14) predicts that the probabilistic error α is smaller for all T_a by using G alone. To verify this, we try both ways: G and G+L and compare the results with our analytical models.

Figure 5 shows that the chance of not finding an acceptable optimum is significantly lower if we use G alone. The use of G+L combined, adds more computational overhead than benefit to the search in this case. We also observe that for the given example, our analytical models predict experimental α results adequately. well. Note in the theoretical curves that the integer stair step has not been plotted.

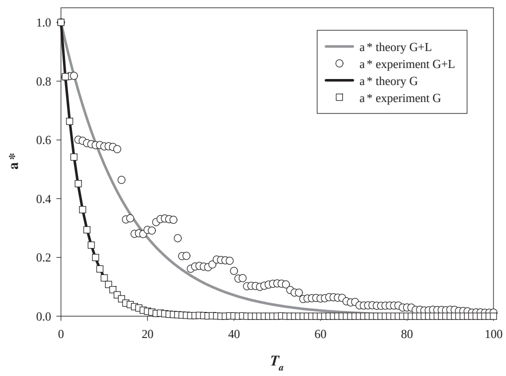

Figure 5: The error α for G and G+L as a function of the allowable time T_a shows good agreement between theory and experiment and also demonstrates the correctness of the optimality condition. In this case, G should be used alone, because it consistently gives lower error for the same allowable time.

9 EXTENSIONS

The foregoing analyses have attempted to take some useful steps toward a better theoretical understanding of the optimization of global-local hybrids. The effort is noted by its simplicity, its connection with common GEA practice, its stark juxtaposition of L and G, its ability to integrate the components of the hybrid, and its ability to address and answer the local-global efficiency decision.

On the other hand, the analysis raises a number of difficult questions. It assumes knowledge of parameters (λ_i, P_i) that depend in complex ways on the problem being solved and the searchers being used. Calculation of the parameters is not trivial even when the problem and searchers are well specified. Moreover, the parameters have been assumed to be constant, but they may not remain stationary, and even if they do, they may vary probabilistically. Nonetheless, the benefits of a simple analysis procedure that permit us to start asking and answering the right questions, outweighs further inaction in understanding efficient hybridization. With this in mind, a number of continuations and extensions of this work suggest themselves as follows:

1. Perform additional empirical investigation of the model proposed herein on both ideal and real problems and solvers.

2. Perform theoretical-empirical investigation of the calculation of P_i values for various Gs.

3. Perform theoretical-empirical investigation of λ_i values for different types of solvers and basins of attraction.

4. Consider on-line estimation and other means of optimizing hybrids in practice.

5. Consider various extensions to the model, including non-deterministic parameters, multiple solvers, and realistic nonstationarities.

These steps are not easy, but item 1 is already underway, and others are likely to be undertaken as the practitioner demands more rational, efficient design of optimization hybrids.

10 CONCLUSIONS

This paper has constructed a systems-level theory of efficient global-local hybrid search, applied that theory to a number of base cases, and outlined a number of continuations and extensions to the work. By idealizing the hybrid as consisting of steps by a global solver G, followed by steps by a local solver L, and by idealizing a search space as consisting of basins of attraction that lead to acceptable targets, the framework is able to decompose the problem of hybrid search. In the framework, a single iteration results in either the global searcher hitting a target, in which case the job is done, or the global searcher hitting a potential basin of attraction, in which case local search leads us to a target (or on a wild goose chase). With this abstraction, the framework requires two sets of parameters, characteristic probability values of hitting targets and basins and time-to-criterion coefficients that quantify the length of time L expected before reaching acceptable solutions. Together, these two parameters are used with suitable equations accounting for time and reliability, and the result is a theory that permits the user to calculate an optimal balance of local and global search.

Hybrids have long been used in genetic and evolutionary algorithm practice, but much of this usage has been ad hoc and without benefit from a macro-level theory. The results of this paper should aid users in better understanding and choosing the proper balance between global and local solvers to help find solutions quickly, reliably, and accurately.

Acknowledgments

This paper was conceived while Professor Goldberg was on sabbatical at the Section on Medical Informatics at Stanford University, and he thanks Mark Musen, Russ Altman, and John Koza for the invitation to Stanford. He is also grateful to Fernando Lobo for a number of conversations that helped shape this work; the authors thank Martin Pelikan for a careful reading and also thank Jacob Borgerson for help with several of the figures.

Professor Goldberg's contribution to this paper was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-97-1-0050. Research funding for this project was also provided by a grant from the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the U. S. Army, or the U. S. Government.

References

Bäck, T. & Schwefel, H.-P. (1995). Evolution Strategies I: Variants and their computational implementation. In G. Winter, J. Périaux, M. Galán, & P. Cuesta, *Genetic algorithms in engineering and computer science* (pp. 111–126). Chichester: John Wiley.

Davis, L. (1989). Adapting operator probabilities in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms,* 60–69.

Davis, L. (1991). *The handbook of genetic algorithms.* New York: Van Nostrand Reinhold.

Goldberg, D. E. (1991). Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems, 5*, 139–167. (originally published as IlliGAL 90001)

Goldberg, D. E. (1994). Genetic and evolutionary algorithms come of age. *Communications of the ACM, 37*(3), 113–119.

Grefenstette, J., Gopal, R., Rosmaita, B, & Van Gucht, D. (1985). Genetic algorithms for the traveling salesman problem. *Proceedings of an International Conference on Genetic Algorithms and Their Application,* 160–165.

Hinton, G. E., & Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems, 1*, 495–502.

Ibaraki, T. (1997). Combinations with other optimization methods. In T. Bäck, D. B. Fogel, & Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation* (Section D3). New York: Oxford University Press.

Lobo, F., & Goldberg, D. E. (1997). Decision making in a hybrid genetic algorithm. *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation,* 121–125.

Nakano, R., Davidor, Y., & Yamada, T. (1994). Optimal population size under constant computation cost. *Parallel Problems Solving from Nature—PPSN III*, 130–138.

Orvosh, D, & Davis, L. (1993). Shall we repair? Genetic algorithms, combinatorial optimization, and feasibility constraints. *Proceedings of the Fifth International Conference on Genetic Algorithms*, 650.

Powell, D., Tong, S. S., & Skolnick, M. M. (1989). EnGENEous domain independent, machine learning for design optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, 151–159.

Press, W. H., Teukolsky, S. A., Saul, A, Vetterling, W. T., Flannery, B. P. (1992). *Numerical recipes in C* (2nd. ed.). Cambridge, MA: Cambridge University Press.

Shaefer, C. G. (1987). The ARGOT strategy: Adaptive representation genetic optimizer technique. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms,* 50–58.

Smith, D. (1985). Bin packing with adaptive search. *Proceedings of an International Conference on Genetic Algorithms and Their Application,*202–206.

Whitley, D. (1995). Modeling hybrid genetic algorithms. In G. Winter, J. Périaux, M. Galán, & P. Cuesta, *Genetic algorithms in engineering and computer science* (pp. 191–201). Chichester: John Wiley.