Invited Paper

An explanation of ordinal optimization: Soft computing for hard problems

Yu-Chi Ho

Harvard University, Pierce Hall, G-12J, 29 Oxford Street, Cambridge, MA 02138, USA

Received 1 January 1998; accepted 15 March 1998

Abstract

This tutorial explains the fundamentals of ordinal optimization, a tool for computationally intensive system performance evaluation and optimization problems. We argue its inclusion as a complementary tool in the arsenal of soft computing techniques. © 1999 Elsevier Science Inc. All rights reserved.

1. Introduction

It can be argued that Optimization in the general sense of making things Better is the principal driving force behind all of prescriptive scientific and engineering endeavor, be it operations research, control theory, or engineering design. It is also true that the real world is full of complex decision and optimization problems that we cannot solve. While the literature on optimization and decision making is huge, much of the concrete analytical results and success stories are associated with what may be called Real Variable Based methods. The idea of successive approximation to an optimum (say, minimum) by sequential improvements based on local information is often captured by the metaphor “skiing downhill in a fog”. The concepts of gradient (slope), curvature (valley), and trajectories of steepest descent (fall line) all require the notion of derivatives/gradient and are based on the existence of a more or less smooth response surface. There exist various first and second order algorithms

1 Corresponding author. Fax: 1-617 496 6404; e-mail: ho@arcadia.harvard.edu

0020-0255/99/$19.00 © 1999 Elsevier Science Inc. All rights reserved.
PII: S 0 0 2 0 - 0 2 5 5 ( 9 8 ) 1 0 0 5 6 - 7
of feasible directions for the iterative determination of the optimum (minimum) of an arbitrary multi-dimensional response or performance surface. A considerable number of major success stories exist in this genre including the Nobel prize winning work on linear programming. It is not necessary to repeat or even reference these here.

On the other hand, we submit that the reason many real world optimization problems remain unsolved is partly due to the changing nature of the problem domain in recent years which makes calculus or real variable based method less applicable. For example, a large number of human-made system problems, such as manufacturing automation, communication networks, computer performances, and/or general resource allocation problems, involve combinatorics rather than real analysis, symbols rather than variables, discrete instead of continuous choices, and synthesizing a configuration rather than proportioning the design parameters. Such problems are often characterized by the lack of structure, presence of large uncertainties, and enormously large search space. Optimization for such problem seem to call for a general search of the performance terrain or response surface as opposed to the "skiing downhill in a fog" metaphor of real variable based performance optimization.2

Arguments for change can also be made on the technological front. Sequential algorithms were often dictated as a result of the limited memory and centralized control of earlier generations of computers. With the advent of modern general purpose parallel computing and essentially unlimited size of virtual memory, distributed and parallel procedures on a network of machines can work hand-in-glove with the Search Based method of performance evaluation. It is one of the theses of this paper to argue for such a complementary approach to optimization.

If we accept the need for a search based method as a complement to the more established real variable based analytical techniques, then we can next argue that to quickly narrow the search for optimum performance to a "good enough" subset in the design universe is more important than to estimate accurately the values of the system performance during the initial stages of the process of optimization. We should compare order first and estimate value second, i.e., ordinal optimization comes before cardinal optimization. Colloquial expressions, such as "ballpark estimate", "80/20 solution" and "forest vs. trees", state the same sentiment. Furthermore, we shall argue that our preoccupation with the "best" may be only an ideal that will always be unattainable or not cost effective. Real world solutions to real world problems will involve

---

2 We hasten to add that we fully realize the distinction we make here is not absolutely black and white. A continuum of problems types exist. Similarly, there is a spectrum of the nature of optimization variables or search space ranging from continuous to integer to discrete to combinatorial to symbolic.
compromise for the "good enough". The purpose of this paper is to establish the distinct advantages of the softer approach of ordinal optimization for the search based type of problems, to analyze some of its general properties, and to show the many orders of magnitude improvement in computational efficiency that is possible under this mindset, and finally to advocate its use as an additional tool in the arsenal of soft computing and a complement to the more traditional cardinal approach to optimization and engineering design.

2. Problem statement and solution approach

2.1. Fundamental limitations of simulation based performance evaluation and optimization

Imagine we have a complex system. The performance of this system is conceptually given by \( J(\theta) \) where \( J \) is the expected performance measure and \( \theta \) are the system design parameters which may be continuous, discrete, combinatorial or even symbolic. A general problem of stochastic optimization can be defined as

\[
\min_{\theta \in \Theta} J(\theta) = E[L(\theta, \xi)]
\]

where \( \Theta \), the search space, is an arbitrary, huge, structureless but finite set; \( \theta \) the design alternatives, \( J \), the performance criterion which is the expectation of \( L \), the sample performance, as a functional of \( \theta \) and \( \xi \), the randomness in the systems.

However, there are some basic limitations related to search-based methods which require the use of simulation to estimate performances \( J(\theta) \) as in Eq. (1). There are two aspects to this problem: the "stochastic" part and the "optimization" part.

The "stochastic" aspect has to do with the problem of performing numerical expectation since the functional \( L(x(t; \theta, \xi)) \) is available only in the form of a complex calculation via simulation. The standard approach is to estimate \( E[L(\theta, \xi)] \) by sampling, i.e.,

\[
E[L(\theta, \xi)] \approx \frac{1}{N} \sum_{i=1}^{N} L(\theta, \xi_i) = J(\theta)_{\text{est}},
\]

where \( \xi_i \) represents the \( i \)th sample of the randomness. The ultimate accuracy (confidence interval) of this estimate cannot be improved upon faster than

\[3\] Of course, here one is mindful of the business dictum, used by Mrs. Fields for her enormously successful cookie shops, which says "Good enough never is". However, this can be reconciled via the frame of reference of the optimization problem.
1/(N)^{1/2}, the result of averaging i.i.d. noise. This rate can be quite adequate or even very good for many problems but not good enough for the class of complex simulation based performance evaluation problems mentioned above. Here each sample of \( L(\theta, \xi) \) requires one simulation run which for the type of complex discrete event dynamic systems (DEDS) we are interested in here can take seconds or minutes even on the fastest computer not to mention the setup cost of programming the simulation. Millions or billions of required samples of \( L(\theta, \xi) \) for different values of \( \theta \) may become an infeasible burden. Yet there is no getting away from this fundamental limit of 1/(N)^{1/2}, i.e. one order of magnitude increase in accuracy requires two orders of magnitude increase in sampling cost.

The second limitation has to do with the "optimization" part. Traditionally, analysis again plays an important part in numerical optimization whether it is convergence, gradient computation, or general hill climbing. Again these tools, such as perturbation analysis (PA), take advantage of the structure and real variable nature of the problem to devise effective algorithms for optimization. However, when we exhaust such advantages and the remaining problem becomes structureless and \( \Theta \) is totally arbitrary as above, we encounter the limitation introduced by combinatorial explosion which we shall call the NP-hard Limitation.\(^4\) In such cases, a more or less blind search is the only alternative. While it is often easy to prove probabilistic convergence of random search, it is very hard to improve the search efficiency. An example will illustrate this:

Suppose \( \Theta \) is of size 10^{10}, which is small by combinatorial standards. We are allowed to take 10,000 samples of \( J(\theta) \) uniformly in \( \Theta \). What is the probability that at least one of these sample will belong in the top 50, top 500, or top 5000 of the designs of \( \Theta \)? Simple calculation yields

\[
\text{Prob(at least one of 10,000 sample is in top} - k\text{)} = 1 - \left( 1 - \frac{k}{10^{10}} \right)^{10,000}
\]

which turns out to be 0.00004999, 0.0004998, and 0.004988 respectively, i.e., forget it! This means that for a simulation based optimization problem of any complexity, a simple random search is not an effective approach – too much computation for too little return.\(^5\)

\(^4\) For example, in the case of a single input single output static decision rule, the number of possibilities is the size of the output possibilities raised to the power of the size of the input possibilities. This is an extremely large number.

\(^5\) To make matters worse, the No-Free-Lunch Theorem of Wolpert and MaCready, (IEEE Journal on Evolutionary Computation, Vol1 #1, 1997) says that without structural information no algorithm can do better on the average than blind search.
These two difficulties, the $1/(N)^{1/2}$ and the NP-Hard Limitation, are basic for general stochastic optimization problems of the type we are interested in. Each one can induce an insurmountable computational burden. To make matters worse, the effect of these limitations is multiplicative rather than additive. No amount of theoretical analysis can help once we are faced with these limitations. This is where ordinal optimization comes in.

2.2. The basic ideas of ordinal optimization (OO)

The idea of OO is to affect a strategic change of goals.\(^6\) It is based on two tenets:

(a) "Order" converges exponentially fast while "value" converges at rate $1/(N)^{1/2}$: In other words, it is much easier to determine whether or not alternative $A$ is better than $B$ than to determine $A - B = ?$. This is intuitively reasonable if we think of the problem of holding one package in each hand and trying to determine which package weighs more vs. to estimate the difference in weight between the two packages. More quantitatively, suppose we observe $A$ and $B$ through i.i.d. zero mean additive normal noises, the chance of misordering $A$ and $B$ is roughly proportional to the area under the overlapping tail of the two normal density functions centered at $A$ and $B$ as illustrated below.

![Diagram of normal distributions overlapping]

This notion can be made more precise. If $A$ and $B$ plus noise are the performance value estimates of two simulation experiments, then under the best conditions the estimates cannot converge to the true values faster than $1/(N)^{1/2}$ where "$N$" is the length of the simulation experiments. But the relative order of $A$ vs. $B$ converges exponentially fast [1,2].

Furthermore, value or cardinal estimation may not always be necessary. When human beings are faced with complex and difficult decision problems involving marriage, career, life-and-death medical treatments, we do not attempt to evaluate the expected utility of the alternatives as decision theory

\(^6\) This is a euphemism for retreat.
dictates. We simply try to determine which is the better alternative without necessarily knowing the detailed consequence of what each alternative entails, i.e., we emphasize the choice (order) rather than estimating the utility (value) of the choices.

(b) *Goal softening eases the computational burden of finding the optimum:* Traditional optimization determines a single alternative in $\Theta$ and hopes that it coincides with the true optimum. The metaphor of hitting one speeding bullet with another comes to mind. Instead if we are willing to settle for the set of "good enough" alternatives with high probabilities, e.g., any of the top-$n\%$ of the choices 95% of the time, then this probability improves dramatically with the size of these sets (i.e., $n$). It is like trying to hit a truck with a shotgun.

Let us formalize this notion a bit. Suppose we define a subset of the design space $\Theta$ as the "good enough" subset, $G$, which could be the top-$g$ designs or top $- n(\equiv g/N)\%$ of the designs of the sampled set of $N$ designs. We denote the size of the subset $G$ as $|G| \equiv g$. We now pick out (blindly or by some rule) another subset, called the *selected subset*, $S$ with the same cardinality, i.e., $|S| = |G| = g$. \(^7\)

We now ask the question what is the probability that among the set $S$ we have at least $k(\leq g)$ of the member of $G$, i.e., $P(|G \cap S| \geq k) = \pi$. In other words, how many of the truly "good" designs are in the selected set? This question is relevant since our rule for picking the set $S$ clearly affects how many good designs we get in $S$. In the case of blind picking we expect the overlap to be smaller than that if we pick based on some estimated order of the alternatives.

We define $P(|G \cap S| \geq k)$ as the *alignment probability* which is a measure of the goodness of our selection rules. Fig. 1 above illustrates the concept. For

\(^7\) Note $|S|$ and $|G|$ need not be equal in which case a slightly more general form of Eq. (4) to follow results.
perspective, we have also indicated the traditional optimization view where the subsets $G$ and $S$ are both singletons.

For the case of blind picking which can be considered as observing the relative orders through noise having infinite variance (which is saying that any design can be considered as the good enough design and vice versa), this alignment probability is given by [3].

$$P(|G \cap S| \geq k) = \sum_{i=k}^{g} \binom{\frac{k}{i}}{\frac{N-g}{g-i}} \frac{\frac{N}{g}}{g-i}.$$  \hspace{1cm} (4)

For example, if $N = 200$, $g = 12$, and $k = 1$, then $P \equiv P(|G \subset S| \geq k) = 0.5$ or $N = 200, g = 32, k = 1 \Rightarrow P \approx 1$. Not bad for total ignorance! You can calculated $P$ for other values $N$, $g$, and $k$ easily, one example of which for $N = 1000$ is illustrated in Fig. 2.

Once again this shows that by blindly picking some 50–130 choices out of 1000 we can get quite high values for $P$. Remember these curves are for blind picks. On the other hand, if we don't pick blindly, i.e., the variance of the estimation error is not infinite and estimated good designs have a better chance at being actual good designs, then we should definitely do better. Suppose now we no longer pick designs blindly, but pick according to their estimated performance, however approximate they may be, e.g., the selected subset $S$ can be the estimated top-$g$ or top-$n(=g/N)$% of the designs. It turns out we no longer can determine $P$ in closed form. But $P$ can be easily and experimentally calculated via a simple Monte Carlo experiment as a function of the noise variance $\sigma^2$, $N$, $g$, and $k$, i.e., the response surface $P(|G \cap S| \geq k; \sigma^2, N)$. For example, we can assume a particular form for $J(\theta)$, e.g., $J(\theta^i) = i$ in which case the best design is $\theta^1$ (we are minimizing) and the worst design is $\theta^N$ and the ordered performances are linearly increasing. For any finite i.i.d estimation noise $w$, we can implement $\hat{J}(\theta) = J(\theta) + w$ and directly check the alignment between the observed and true order.

More generally, this alignment probability can be made precise and quantitative [4,5] and shown to be exponentially increasing with respect to the size of $G$ and $S$. Consequently, by combining the two basic ideas (a) and (b) of OO in this section, several orders of magnitude of easing of the computational burden can be achieved. For a list of experimental evidences and success stories of the applications of OO to complex problems, real or made up, consult Refs. [3,6–11, 21–24].

---

8 This can be characterized as the Horse Race selection rule where we estimate the performance of ALL designs and pick the estimated top-$g$ designs.
2.3. A spreadsheet demonstration

A simple and instructive demonstration of the ideas of ordinal optimization can be easily carried out and the alignment probability \( P(|G \leq S| \geq k; N, \sigma^2) \) determined as in Fig. 3.

Column 1 models the \( N (=200) \) in this demo) alternatives and the true order 1 through \( N \). Column 2 shows the linearly increasing ordered performance values \( J(\theta) \) from 1 through \( N (=200) \). These values must be increasing by definition (assuming we are minimizing). The rate of increase with respect to the noise variance \( \sigma^2 \) essentially determines the estimation or approximation error of \( \hat{J}(\theta) \). This is shown by the random noise generated in column 3 which in this case has a large range \( U(0,100) \). Column 4 displays the corrupted (or estimated) performances \( \hat{J}(\theta) \). When we sort on the column 4 we can directly observe the alignment in column 1, i.e., how many numbers 1 through \( g \) are in the top-g rows. Try this and you will be surprised! It takes less than two minutes to setup on EXCEL or Lotus 1-2-3.  

2.4. Elements of OO

Now consider a thought experiment. We evaluate \( J(\theta) \) for all possible values of \( \theta \in \Theta \). Now imagine we plot all these \( J(\theta) \) values in a histogram and get

\[ \text{Fig. 2. } P \text{ vs } g \text{ parametrized by } k = 1, 2, \ldots, 10 \text{ for } N = 1000. \]
<table>
<thead>
<tr>
<th>Design # = θ</th>
<th>J(θ)</th>
<th>Noise( w \in U(0,W))</th>
<th>Observed J(θ) + w</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000000</td>
<td>83.03771</td>
<td>84.03771</td>
</tr>
<tr>
<td>2</td>
<td>2.000000</td>
<td>12.50149</td>
<td>14.50149</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>199</td>
<td>199.00000</td>
<td>30.93567</td>
<td>229.93567</td>
</tr>
<tr>
<td>200</td>
<td>200.00000</td>
<td>1.05834</td>
<td>201.05834</td>
</tr>
</tbody>
</table>

Fig. 3. Spread sheet implementation of generic experiment.

what is called the Performance Density Function (PDF) or its integral the Performance Distribution Function as illustrated in Fig. 4.

A related concept of PDF is simply the Ordered Performance Curve (OPC) which we have already encountered in Section 2.3 for the spread sheet demo. This is the a plot of the performance value against the #1 design, #2 design, and so forth to the #N design in that order. By definition, it must be monotonically increasing. It can also be thought of as the performance distribution function (integral of the PDF) of Fig. 4 turned sideways with the design # range 1 through N and performance value normalized to (0,1).

Now we assume

**Assumption 1.** We can uniformly sample the Θ space and in principle get \( J(\theta^i) \) which represent the \( i \)th sample from \( \Theta^{10} \), for \( i = 1, 2, 3, \ldots, N \).

---

10 In practice, of course we can only sample estimates of \( J(\theta) \). Also, how to sample uniformly from \( \Theta \) may not be easy or obvious in some cases. These problems will be discussed separately later.
Fact 1. $N$ need not be very large (say $< 1000$) before we have a very representative set of performances.

The basic idea behind this is statistical. For example, we don’t need to interview all 100 million voters to determine the outcome of a presidential election. In fact, most TV networks only interview a couple of thousands voters. Similarly, suppose we take 1000 samples uniformly from any distribution and ask the question what is the probability that these 1000 samples are all from the lower 99.5% of the search space?¹¹ This is easily given by $(0.995)^{1000} = 0.0066$, a very small probability. Another question would be what is the probability that at least five of these 1000 samples are in the top 5% of the search space? This is given by

$$\text{Prob(at least five top 5% designs among 1000 samples)} = 1 - \sum_{i=0}^{4} \binom{1000}{i} (0.95)^{1000-i} (0.05)^i \cong 1 - 1.82 \times 10^{-17} \cong 1.$$  

Other examples can be similarly calculated to illustrate the point that for our purposes we don’t have to sample more than a couple of thousand designs to have representative coverage of all possible performances. This is because we

¹¹ To be precise, suppose there are altogether 1 million possible designs in the search space $\Theta$. The lower 99.5% means design #5001th through #1,000,000th with #1 as the best design, #2 as the second best, and so on. Note begin in the top 0.5% of the search space does not necessarily mean we are within 0.5% of the best performance value $J$. One can always have a problem in which the $J$s are [10 million, 2.0, 1.999999, 1.999998, ..., 1.0]. Such pathological needle-in-a-haystack problems are inherently hard. Short of exhaustive search, nothing will work.
are primarily interested in the subset of "good enough" designs or alternatives and in locating one of these. So long as we are assured that enough "good" alternatives are present in the sampled set, we can henceforth work with the sampled set of \( N \) and treat them as if they were the total population. More precisely, whatever probability we calculate using the \( N \) samples as the entire population is really a conditional probability, \( P(\cdot/N) \). The unconditional probability with respect to the population is given by

\[
P = \sum P(\cdot/N)P(N).
\]  

(5)

All we are saying is that the difference between \( P \) and \( P(\cdot/N) \) is minimal for any \( N \geq 1000 \), or \( P(N) \) is essentially uniform over all possible realization of the \( N \) samples when \( N \geq 1000 \). Finally, the reader may object on the grounds that to be able to narrow things down to say 1% or even 0.1% may not mean very much since 1% of infinity is still infinity. In the space of combinatorial choices which easily has cardinality in zillions, restricting the sampling to a few thousand choices may not be sufficient. This is a legitimate objection which we shall deal with in Section 3.

However this is not the end of this story because in practice for all except the simplest system we cannot calculate \( J(\theta) \) in closed form and must estimate it via simulation or approximation. For complex problems even the task of evaluating \( J \) for \( N(\sim1000) \) designs may still be too time consuming to be feasible.

**Fact 2.** *Even for these couple of thousand designs \( J(\theta^i) \), for \( i = 1, 2, 3, \ldots, N \approx 1000 \), we only need to estimate their performance very approximately in order to narrow down the "good" design among them.*

Now consider the Ordered Performance Curve (OPC) of these \( N \) Samples. By definition it must be monotonically increasing. A linearly increasing OPC from 1 through \( N \) means that the performance density function is uniform, i.e., good and bad designs are equally distributed in \( \Theta \), a reasonable supposition when we don't know any better. Note \( P \) is implicitly dependent on the way we determine \( S \). We have discussed blindly picking (BP) \( S \) and now are simply picking the estimated top-g as in a horse race (HR). It should be clear that other rules for determining the members of \( S \) based on other selection rules or heuristics are possible. This is in fact the role of heuristics, namely to bias in favor of good outcomes.  

More discussion on this later.

In any case, the point is that for any given way of picking \( S \), e.g., choose the estimated top-g designs, we can calculate once and for all the value of \( P(|G \cap S| \geq k; \sigma^2, N) \) for all possible problems. We have computed these alignment probability curves and found that [4].

\[ ^{12} \text{In fact one distinct advantage of most heuristic rules compared to the horse race (HR) rule is that they may not require estimation of all } N \text{ designs, however approximately, to determine } S. \]
Fact 3. $P(|G \cap S| \geq k; \sigma^2, N)$ can be in the range 0.90–0.95 even for very large $\sigma^2$ and $N \approx 1000$.

Fact 3 is a more explicit statement of Fact 2 which we shall continue to justify. However, before continuing, it should be clear how $P(|G \cap S| \geq k; \sigma^2, N)$ can be used very effectively to help us narrow down the search for good designs. Consider the conceptual flow chart below in Fig. 5. We call this flowchart a Soft Optimization Shell or SOS. It is soft optimization because we do not insist on getting the “best” but only the “good enough”. In fact it is this tradeoff that enables us to get high values for $P$. SOS is shell because we can plug in whatever computer model for the system under investigation ranging from back-of-the-envelope formula to complex simulation programs.

In the case of a simulation program we can vary the length of the simulation experiment to control the parameter $\sigma^2$ which provides added flexibility. It is also clear how a parallel simulation language that can run a set of simulation experiments over the $N$ designs simultaneously on either sequential or parallel computers can be very useful in this SOS environment. The point is that we can have a hierarchy of models ranging from the simplest to the most complex. At each level of the hierarchy we can quantify the degree and confidence of the approximation (in terms of the alignment probability described above). We can also creatively allocate our computing resources over different levels of this hierarchy between “quickly surveying the landscape” to “lavish attention on particular details”. The possibility of exploiting the synergistic interaction of what human and machine each doing their best is endless in Fig. 5 (see also Section 3 and [12]). Furthermore, the possibility of learning and adaptation also presents itself in this connection. For example, Genetic Algorithm (GA) works on the principle that we evaluate the fitness (performance) of a population of designs and use the idea of reproduction, cross-breeding, and mutation to improve the population. OO and the horse race selection here should work hand-in-glove with GA in the SOS environment since it enables and justifies a much speedier evaluation of the fitnesses, a fact always assumed in applications of GA. But without OO, many of the hard problems mentioned in the introduction will not be amenable to GA treatment. We have only scratched the surface here.

Now we go back to justifying and explaining Facts 2 and 3. If one has thought a bit about the quantity $P \equiv P(|G \cap S| \geq K; \sigma^2, N)$, then one may raise the following legitimate question at this point. While it is clear that in the case of blind picking, $BP, (= \sigma^2 \to \infty)$ $P$ is independent of the underlying problem, surely this cannot be so for finite $\sigma^2$. However, we shall argue that this dependence is not tightly coupled and can be categorized into a few cases with $P$ pre-calculated (or accurately estimated) for each case. To do this, we shall recall the concept of Ordered Performance Curves (OPC). This is the plot
Choose a crude model of the system under study. This can be a simplified analytical model or a very short simulation with large variance. This essentially sets the value $\sigma^2$. Also define what is $G$, e.g., the top-g

Choose a selection rule

other selection rule for determining $S$

Horse Race Selection Rule
Pick $N$, $g$ and use the model to estimate all $N$ as well as the top-$g$ designs. This determines the set $S$

$P(I \cap S \geq k; \sigma^2, N)$

$\geq 0.9$

There are good designs in $S$ with high probability. We can go on to more refined models for performance evaluation

$< 0.9$

Iterate on the model, e.g., by making a simulation expt with smaller variance

Iterate on the selection rule for $S$

iterate on $N$, $g$, the search domain for the parameters

Fig. 5. Flow chart for SOS.
of the performance value against the #1 design, #2 design, and so forth to the #N design in that order. By definition, it must be monotonically increasing (assuming we are minimizing the objective with the best design having the smallest J value). We illustrate five varieties of possible performance density curves in Fig. 6(a)–(e) and their corresponding OPCs in Fig. 7(a)–(e) below.

While one can argue that there is an infinite number of possible performance density curves besides the five illustrated here, we submit that the five categories capture the major features: 1. lots of average performances, 2. all performances are evenly distributed, 3. lots of good performances, 4. lots of bad performances, 5. performances are either good or bad but little middle ground. Categories 3 and 5 represent problems where a good design will be relatively easy to find. Category 4 and to a lesser extent category 1 have a paucity of good

---

![Fig. 6.](image)

Fig. 6. (a) Normal N(0,1); (b) Uniform U(0,\sqrt{12}); (c) Exponential E(\lambda)(\lambda = 1); (d) Negative exponential NE(\lambda)(\lambda = 1); (e) Inverse Normal.
Fig. 7. (a) OPC Corresponding to Fig. 6(a). (b) OPC Corresponding to Fig. 6(b). (c) OPC Corresponding to Fig. 6(c). (d) OPC Corresponding to Fig. 6(d). (e) OPC Corresponding to Fig. 6(e).

designs. Note also when the slopes near the optimum are flat which means noise can more easily corrupt the performance orders of the designs, using Eq. (4) and the blind choice selection rule will furnish us with lower bounds on $P$. Finally, SOS is expected to be used mostly in the initial search or narrowing down of choices. The estimation errors (equivalently the $\sigma^2$) tend to be large in such cases where the different categories do not cause much of a difference in the value of $P$ since in the limit of $\sigma^2 \to \infty$ any difference disappears. Thus it is our thesis we only need to pre-calculate $P \equiv P(|G \cap S| \geq k; s^2, N)$ for the case of Fig. 6(b) and 7b, the uniformly distributed case. The linearly increasing OPC then reflects the favorable bias of our selection rule: "pick the estimated Top-g in a horse race". Furthermore, the parameter $\sigma^2$ gives us another degree of flexibility. A large $\sigma^2$ will approximate the blind choice rule (Or Fig. 7(c) and 7e) while a small $\sigma^2$ increases the favorable selection bias (or Fig. 7(a) and Fig. 7(d)). A set of normalized surfaces for $P$ as a function of $\sigma^2$, g, and $N$ for all problem types of Fig. 6(a)–6(e) have been calculated [4]. A number of experiments of applying the horse race rule and $P$ thus calculated to fairly
complex problems support our thesis \(^{13}\) [see earlier reference on success stories of application].

3. Some additional issues in ordinal optimization

While in section II we outlined the main ideas of ordinal optimization as well as how it might be and have been applied, we do not wish to give the impression that all problems have been solved. The science and art of OO is just beginning. We treat here a host of secondary issues to bolster the case for OO and to suggest additional future research in this section.

3.1. Correlated, non-zero mean, and design-dependent estimation noises

Since the alignment probability \(P\) only deals with orders and not values, it is clear that any mean of the estimation noise, \(w\), cannot affect \(P\). We shall also argue that correlation among \(w_i, i = 1, 2, \ldots\) cannot hurt in general. First if the \(w_i\)'s are positively correlated then to a first approximation the correlation can only help again by similar reasoning as the case for the mean. If the noises are negatively correlated then the worst can happen is that we throw out half the samples \(^{14}\) but with the rest of the samples positively correlated which helps. More careful treatment of the case of correlated noise can be found in \([13,20]\).

OO has also been used experimentally with success on complex problems which are completely deterministic \([10]\). Here we are replacing a complex but deterministic model by a simpler model. The approximation error is represented by noise. The only rationalization of this replacement besides experimental evidence is the Kolmogorov's contention that complexity is equivalent to randomness \([14]\), i.e., we can represent

\[
J(\theta)_{\text{complex}} = J(\theta)_{\text{simple}} + \text{noise/error} \tag{6}
\]

where all the complexity of the model is hidden inside the noise/error term. While this is not unreasonable, there is no assurance that the noise term will be i.i.d. as required by our theory. The remedy for this is to regress out any deterministic component of the noise term in (6) through extra computations, i.e., we postulate

\(^{13}\) After allowing some knowledge about \(\sigma^2\). Alternatively, one can always use the \(N\) samples to estimate the shape of the OPC and select the pre-calculated \(P = P(\|G \ll S\| \geq k; \sigma^2, N)\) accordingly. We at present do not believe the SOS needs to be this sophisticated. After all SOS is used only to estimate probability and for separating the good from the bad.

\(^{14}\) Once \(N\) is larger than 500, doubling or halving the samples does not seem to affect \(P\) greatly. This is a statistical fact.
\[ J(\theta)_{\text{complex}} = J(\theta)_{\text{simple}} + w(\theta) + \text{noise/error} \]  

(7)

where the \(w(\theta)\) term in (7) is again of simple structure, say low order polynomials in \(\theta\), and the noise/error term is i.i.d. To determine \(w(\theta)\), we can perform linear regression which requires the computation of the true \(J(\theta)\) for a small number of \(\theta\)s. This cost is equivalent to the cost of supervised learning in the AI context. We have had some success here [10] but much more needs to be done [15].

3.2. How to improve the \(P\) values via goal softening.

There will be times when the value of \(P\) is not large enough (say <0.8) to be interesting. One’s first instinct is to increase the value of \(N\). However, this buys one relatively little once \(N\) is larger than a few hundred. Instead the value of \(P\) can be significantly increased if we are willing to soften our goals, i.e., increase the size of \(G\) and/or \(S\). Consider the example below

Performance distribution: Normal (0, \(\sigma_{\text{signal}}\))
Additive noise: \(N(0, \sigma_{\text{noise}})\)
# Of samples taken: \(N = 100\)
The selected set: The best of \(N\) samples, i.e., \(|S| = 1\)
The good enough set: \(G = \text{top 5\%}\)

A typical result is that for \(\sigma_{\text{signal}} = \sigma_{\text{noise}}\), i.e., signal to noise to ratio is one,

\[ \text{Prob}(|G \cap S| = 1) = 0.563. \]

Increasing \(N\) from 100 to 300 yields \(\text{Prob}(|G \cap S| = 1) = 0.696\). This is not particularly encouraging. However, by softening the question we can again improve the result. Suppose instead of one sampling experiment of 300, we perform three independent experiments of 100 samples each, select the best of each experiment and let these three best samples be the selected set, i.e., \(|S| = 3\). Now we can calculate the alignment probability as

\[ \text{Prob}(|G \cap S| = 1; |S| = 3) = 1 - (1 - \text{Prob}(|G \cap S| = 1; |S| = 1))^3 \]
\[ = 1 - (1 - 0.563)^3 = 0.917 \]

which is considerably more useful and interesting. This little exercise illustrates two important points. First is the power of softening the goal. Second, alignment probabilities for different choices of \(G\) and \(S\) can be built up from building blocks of elementary alignment probabilities. For another example, take the uniform performance distribution with \(\sigma_{\text{noise}}/\sigma_{\text{signal}} = 1\), we have
Prob(the observed best design from one experiment of 800 samples
is in fact a top-5% good design) = 0.384
≡ P(best).

However,

Prob(at least one top 5% good design among the observed best
design from 8 independent experiments of 100 samples each) = 0.945
≡ P(eight best).

This again demonstrates the value of softening the optimization requirements which cannot be overemphasized. It is this softening of goals that makes the alignment probability high enough to be interesting and useful! In fact even in the limit of infinite noise, we can use Eq. (4) to make P nearly equal to one although the proper choice of small k and large |S| or |G| and still affect an 8:1 reduction from |N| to |G| (RE: Fig. 2 in Section 2. Conversely, if |G| and |S| are small, say equal to 1 or 2, then one can easily construct examples where P is rather low. But this is not a criticism of the approach but a criticism of traditional optimization where one takes a hard view and is trying to hit one bullet with another (the case of |G| = |S| = 1 Re: Fig. 1). Finally, if one is willing to do a bit of extra work by evaluating in detail the eight best designs from the eight experiments of 100 samples each, then one will have obtained a top-5% design with high probability of 0.945 instead of 0.384 (i.e., we have not really retreated to a softer goal but increased our confidence in getting a top-5% answer).

3.3. Relationship to other statistical and AI techniques

Readers who are familiar with the statistics may inquire whether or not we are reinventing the wheel with OO since it is obviously related to the Ranking and Selection literature in statistics [16]. Our answer is that we are quantifying and unifying various engineering heuristics under one umbrella. While there are connections between what we do and that of ranking and selection statistics, a considerable difference in emphasis remains. The principal difference has to do with the size of the population we work with. In ranking and selection statistics, we are mostly concerned with ranking among five or ten alternatives as in drug efficacy testing; while in engineering applications, the number of alternatives is in billions or zillions. Also in OO we studiously avoid cardinal notions while the R&S literature is still concerned with things such as the distance between the best and the rest, a cardinal notion. Finally, while questions such as “Is the observed order a maximum likelihood estimate of the actual order?” are most intriguing and elegant in R&S, they are of little concern in OO since
the probability of coincidence of the observed and actual order is so minuscule to be totally uninteresting. On the other hand, R&S and the Order Statistics literature are clearly related to our interest in OO. For example, the additive noise model described earlier in Eqs. (6) and (7) is known as the Thurstone Mosteller-Daniel model in R&S.

Two prominent tools in computational intelligence and optimization are *Fuzzy Sets* and *Genetic Algorithms* [17]. We submit ordinal optimization as described here is complementary to these techniques and they both fit into the SOS framework to create synergism.

In a nutshell, Ordinal Optimization or SOS enables the use of crude models, whether through shorter simulations or simpler constructs, to isolate quantitatively subsets of design space that contain good enough designs with high probability. We have so far simply taken the definition of "good enough" as anything in the top-\(g\) alternatives. But this is a classic crisp definition. What about the top-(\(g + 1\)) alternative? Similarly, high probability is defined as probability greater than 0.99. But what about probability equal to 0.95? Fuzzy set theory argues persuasively that definitions should not be so sharp but should be softened to correspond to human imprecision and vagueness. For example consider the below figure.]

---

### Classical vs. Fuzzy Definition of "Good Enough"

| #1 | #2 | ... | #g | #g+1 |...
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

### Classical vs. Fuzzy Definition of "High Probability"

| 0.99 | 0.95 | 0.9  
|------|------|------
| 1    | 1    | 1    |

---

By softening the definitions, we can easily demonstrate that additional speedup can be gained in simulation experiments. This will further enhance the effectiveness of SOS as envisioned.

Similarly, Genetic Algorithm (GA) utilizes the idea of evaluating the performances of a system over a population of alternatives. We have repeatedly
pointed out that in simulation the evaluation of performances is a time-consuming bottleneck, thus, making GA not applicable to a large class of problems. However, with OO we have a speedy means of isolating the good from the bad which will extend the applicability of GA to a large set of problems. On the other hand, GA offers the possibility of the iterative application of OO, and to adaptively search for good performance in a learning mode. Considerable synergism is possible. This leads to the next topic.

3.4. Adaptive learning in OO

OO offers the possibility of narrowing down the search. However, to be able to get within 1% of a search space of cardinality of say $10^{10}$ is still $10^8$ away from the optimal. This is of scant comfort. Thus, iterative application of OO is necessary very much in the spirit of hill climbing in traditional optimization. The difference is that we now iterate on successive search regions rather than successive points, i.e.,

$$(\text{Search Region})_{k+1} = f((\text{Search region})_k, \text{information from}_{\text{th search}}).$$ \hspace{1cm} (8)

A key fact is

**Fact 4.** suppose we observe the performance of two samples $\hat{J}(\theta^1)$ and $\hat{J}(\theta^2)$ then

$$\text{Prob} [\hat{J}(\theta^2) < J(\theta^1), \hat{J}(\theta^2) < \hat{J}(\theta^1)] > \text{Prob} [J(\theta^2) < J(\theta^1), \hat{J}(\theta^1) < \hat{J}(\theta^2)]$$

i.e., if one sample is observed or estimated to be better in performance than another, then it is more likely to be actually better.

We leave the simple proof of this fact to the reader. The implication of this fact is that we can utilize this fact to learn which search region or method of sampling is better than another by comparing the PDFs of the respective region/methods. This fact has been used successfully (via a manual implementation of SOS of Fig. 5) to obtain a solution to the well-known Witsenhausen problem that is 50% better than the original [18].

To drive home the point of the importance of knowledge and learning in optimization search we return to the example of uniform sampling of 10,000 points in a space of 10 billion illustrated in Section 2.1. The probability of finding anything good is not at all promising, e.g., to find at least one sample within the top-5000 with 10,000 samples is practically zero. But suppose somehow we have learned that the top-50, top-500, and top-5000 designs are restricted to subset of the search space of cardinality 1 million, then the sampling probability improves to 0.393, 0.993, and 1 respectively, i.e., we can hardly fail! Thus, using OO and Fact 4 above offer the possibility for us to learn quickly with crude models and to implement Eq. (8). Again we have only begun.
3.5. Other heuristics selection rules besides blind pick (BP) or horse race (HR) and other definitions of $G$.

The HR selection rule requires us to estimate, however approximately, all $N$ alternatives. This can still be time consuming. The blind pick BP rule does not have this disadvantage. There can be many other selection rules, e.g.,
1. tournament pair-wise elimination rules to select the best or 4-best (quarter finals in tennis tournaments),
2. round robin selection rule, e.g., baseball divisional, league, and world champions,
3. Greedy rule, e.g., in traveling salesman problem using random starting point always pick the next least-cost route to travel.

Every one of these rules will induce different alignment probabilities $P$ and have different properties. Note also that for heuristic selection rules that do not require the sampling of $N$ alternatives we can in principle compute $P \equiv P(|G \cap S| \geq k; \sigma^2, N)$ directly via Monte Carlo simulation. We submit one of the goals and outstanding problems of artificial intelligence and of ordinal optimization is to quantify such heuristic rules. We also believe our definition of the subsets $S$ and $G$ generalizes the concept of optimization where traditionally $S$ and $G$ are singletons and heuristic rules for picking $S$ are not quantified.

Similarly, other definitions of $G$ are possible. Currently, we have chosen a non-fuzzy approach vs. possibly a more reasonable and fuzzified approach to the definition of $G$ as explained in Section 3.3 above. Finally, with respect to different OPCs, we can redefine $G$ to be

where we have taken into account that for flat OPC a more liberal definition of $G$ can be accepted.
3.6. The problem of constraints

So far we have discussed the problem of OO without any constraints. Consider the problem of

\[(P-1) \quad \text{Min } \theta J(\theta) \equiv \text{Min } \theta E[L(\theta)] \text{ subject to } E[f(\theta)] \equiv F(\theta) \leq 0.\]

The main problem here is to evaluate \(E[f(\theta) \equiv F(\theta)]\) as difficult as evaluating \(J(\theta) \equiv E[L(\theta)]\). Both are cardinal notions that we are trying to avoid. But how do we get around this problem of constraints which is basically a cardinal notion?

One solution is that we simply assume away the problem. If we accept that instead of the best we are willing to settle for the good enough, then it seems consistent that instead of satisfying the constraints exactly we might be willing to settle for the "close enough". In many real life problems we indeed have such flexibilities. Or we can always set the constraint level tighter than usual and hope that "close enough" will in fact provide constraint satisfaction. In any case, once we are willing to soften the constraint, the ideas of penalty function and LaGrange multipliers come into play. In place of (P-1), we can consider

\[(P-2) \quad \text{Min}_\theta \{J(\theta) + \lambda \text{Step}[G(\theta)]^2\} \text{ for some } \lambda \gg 1\]

or

\[(P-3) \quad \text{Min}_\theta \{J(\theta) + \lambda G(\theta)\} \text{ for some } \lambda \geq 0 \text{ and } \lambda G(\theta) = 0.\]

In fact, (P-3) can also be viewed as the scalarization of a vector optimization problem with the twin criteria \(J(\theta)\) and \(G(\theta)\). The scalar criterion \(J(\theta) + \lambda G(\theta)\) maps out the pareto frontier as \(\lambda \geq 0\) varies. A great deal of theory on pareto optimality and vector criteria in cardinal optimization can be of used here. The point is with either (P-2) or (P-3) we are back to the unconstrained problem.

Of course, there will be times or problems where the constraints are really hard and we cannot settle for "close enough". All is not lost, several possibilities exist. First, we can still use (P-2 or -3) to determine "good enough" \(\theta\)'s. Once we determined the select set, \(S\), we can always directly verify which member of \(S\) indeed satisfies the constraints. Again the issue comes to playing with the value of \(\lambda\), an unavoidable problem when there are constraints. Secondly, it may be possible in some problems to actually substitute out the variables which are not free using the constraints directly. Finally, if the problems involve searching in the strategy space (see Section 3.3), then for some problems the constraints may be directly incorporated when the strategy space is defined. Once that is done, no more constraints are present. More details on constrained ordinal optimization can be found in [19].
4. Conclusion

As we have cautioned earlier in Section 3, the science and art of ordinal optimization is at its infancy. Much remains to be done. However, the subject does have the following distinguishing features:
1. It is immediately applicable to real and complex problems.
2. It promises orders of magnitude saving in computational cost.
3. It has already accumulated many success stories.

Acknowledgements

The work reported in this paper is supported in part by NSF grants EEC-94-02384, EID-92-12122, ARO contracts DAAA-03-92-G-0115, DAAH-04-95-0148, and AFOSR contract F49620-95-1-0131 and is the results of extensive discussions with the current and former members of the Harvard DEDS group, C.H. Chen, M. Deng, M. Larson, E. T.W. Lau, LooHay Lee, David Li, and N. Patsis who are or have been all engaged in Ph.D researches on various aspects of ordinal optimization.

References


