1-2006

Swarm Intelligence: Theoretical Proof that Empirical Techniques are Optimal

Dmitry Iourinskiy

Scott A. Starks
The University of Texas at El Paso, sstarks@utep.edu

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Stephen F. Smith

Comments:
UTEP-CS-02-02c.
Short version published in the *Proceedings of the 2002 World Automation Congress*, Orlando, Florida, June 9-13, 2002, Paper ISSCI005; full paper published in A. Abraham, C. Grosan, and V. Ramos (Eds.), *Stigmergic Optimization*, Springer-Verlag, Berlin-Heidelberg, 2006, pp. 281-295.

Swarm Intelligence: Theoretical Proof That Empirical Techniques are Optimal

Dmitriy Iourinskiy[1], Scott A. Starks[2], Vladik Kreinovich[2], and Stephen F. Smith[3]

[1] School of Computing Science, Middlesex University, North London Business Park, Oakleigh Road South, London N11 1QS, UK d.iourinski@mdx.ac.uk
[2] NASA Pan-American Center for Earth and Environmental Studies, University of Texas at El Paso, El Paso, TX 79968, USA sstarks@utep.edu, vladik@utep.edu
[3] Robotics Institute, Carnegie Mellon University, 500 Forbes Ave., Pittsburgh, PA 15213, USA sfs@cs.cmu.edu

Summary. A natural way to distribute tasks between autonomous agents is to use *swarm intelligence* techniques, which simulate the way social insects (such as wasps) distribute tasks between themselves. In this paper, we theoretically prove that the corresponding successful biologically inspired formulas are indeed statistically optimal (in some reasonable sense).

Key words: swarm intelligence, autonomous agents, optimality proof

1 Introduction

1.1 What Is Swarm Intelligence

In many real-life situations, we have a large number of tasks, and a large number of autonomous agents which can solve these tasks. The problem is how to best match agents and tasks. This problem is typical:

- in manufacturing, where we have several machines capable of performing multiple tasks;
- in robotics, when we need to coordinate the actions of several autonomous robots;
- in computing, when several parallel computers are available, etc.

In general, if we want an optimal matching, then this problem is difficult to solve. For example, it is known that the problem of optimal manufacturing scheduling is NP-hard; see, e.g., [19]. Since we cannot have an optimal solution, we must look for heuristic solutions to such problems.

One of the natural sources of such heuristics is biology, specifically, the biology of insects. Insects are usually small, so it is difficult for an individual insect to perform complex tasks. Instead, they swarm together and perform tasks in collaboration. Since the existing social insects are the result of billions of years of survival-of-the-fittest evolution, we expect that all the features of their collaboration have been perfected to being almost optimal. Thus, it is reasonable to copy the way social insects interact. The resulting multi-agent systems are called *swarm intelligence* [8, 24].

1.2 What Formulas Are Used in the Existing Swarm Intelligence Systems

The biological observations led researchers to the following model for the insect collaboration: We have several classes of tasks. Each task T of type t is characterized by its degree of relevance $R_t(T)$; in biology, this degree of relevance is called a *stimulus*.

In principle, each agent can perform each task; in this sense, the agents are *universal*. However, different agents have different abilities with respect to different tasks. If an agent is not very skilled in a certain type of tasks, then this agent picks tasks of this type only when they are extremely important, i.e., when the stimulus is very high. If an agent is reasonable skilled in tasks of certain type, then this agent will also pick such tasks when the corresponding stimulus is much lower. This behavior can be characterized by assigning, to each agent A and to each type of tasks t, a *threshold* $\theta_t(A)$:

- if the stimulus $R_t(T)$ corresponding to a task T is much smaller than the threshold, then the agent will not take this task;
- if the stimulus is much larger than the threshold ($R_t(T) \gg \theta_t(A)$), then the agent will take this task.

In other words, whether the agent takes the task or not depends on the ratio $r \stackrel{\text{def}}{=} R_t(T)/\theta_t(A)$: if $r \ll 1$, the agent does not take the task; if $r \gg 1$, the agent takes the task.

When the ratio is close to 1 (i.e., when the stimulus is of the same order of magnitude as the threshold), then the same insect sometimes takes the task, sometimes does not. The frequency (probability) P with which an insect picks the task increases with the ratio r. From the biological observations, it was determined that the dependence of the probability P on the ratio r has the following form:

$$P(r) = \frac{r^2}{1 + r^2}. \tag{1}$$

In other words, the probability P of an agent A to pick the task T of type t is equal to:

$$P = \frac{R_t(T)^2}{R_t(T)^2 + \theta_t(A)^2}. \tag{2}$$

This formula was proposed in the 1990s in [9, 10, 39]. Since then, it has been used in the existing swarm intelligence systems, and it has led to reasonable results [1, 2, 8, 13, 14, 15, 16, 24, 31, 32].

1.3 Formulation of the Problem

The idea that a probability P should depend on the ratio r is very convincing. However, the specific dependence of P on r (as described by the formula (1)) is rather ad hoc. Since this formula is successful, it is reasonable to try to find a justification for its use.

In this paper, we provide such a justification.

2 Main Idea

Since we want to design an *intelligent* system, we should allow agents to learn, i.e., to use their experience to correct their behavior. In the swarm intelligence model, at any given moment of time, the behavior of an agent A towards tasks of all possible types t is characterized by its thresholds $\theta_t(A)$. Thus, learning means changing the agent's thresholds, from the original values $\theta_t(A)$ to new values $\theta'_t(A)$. As a result, the probability

$$P = P(r) = P\left(\frac{R_t(A)}{\theta_t(A)}\right) \tag{3}$$

of an agent A taking the task T changes to a new value

$$P' = P(r') = P\left(\frac{R_t(A)}{\theta'_t(A)}\right). \tag{4}$$

The formula describing the transition from the original probabilities (3) to the new probabilities (4) can be further simplified if we denote the ratio of the old and the new thresholds by

$$\lambda = \frac{\theta_t(A)}{\theta'_t(A)}. \tag{5}$$

In terms of λ, we have $r' = \lambda \cdot r$, hence the new probability is equal to

$$P' = P(\lambda \cdot r). \tag{6}$$

From the statistical viewpoint (see, e.g., [21, 40, 42]), the optimal way of updating probabilities is by using the Bayes formula. Specifically, if we have n incompatible hypotheses $H_1, \ldots, H_n$ with initial probabilities

$$P_0(H_1), \ldots, P_0(H_n), \tag{7}$$

then, after observations E, we update the initial probabilities to the new values:

$$P(H_i \,|\, E) = \frac{P(E \,|\, H_i) \cdot P_0(H_i)}{P(E \,|\, H_1) \cdot P_0(H_1) + \ldots + P(E \,|\, H_n) \cdot P_0(H_n)}. \tag{8}$$

Thus, *an optimal function $P(r)$ can be determined as the one for which the transition from the old probabilities (3) to the new probabilities (4), (6) can be described by the (fractionally linear) Bayes formula (8).*

3 From the Main Idea to the Exact Formulas

Let us formalize the above condition. In our case, we have two hypotheses: the hypothesis H_1 that it is reasonable for an agent A to take a task of given type t, and the opposite hypothesis H_2 that it is not reasonable for the agent A to take such a task. Initially, the probability of the hypothesis H_1 is equal to P, and the probability of the opposite hypothesis H_2 is equal to $1 - P$. According to Bayes formula, after some experience E, the probability P should be updated to the following new value $P' = P(H_1 \,|\, E)$:

$$P' = \frac{P(E \,|\, H_1) \cdot P}{P(E \,|\, H_1) \cdot P + P(E \,|\, H_2) \cdot (1 - P)}. \tag{9}$$

If we denote $P(E \,|\, H_1)$ by a, $P(E \,|\, H_2)$ by b, and explicitly mention that the probability P depends on the ratio r, then the formula (9) takes the following form:

$$P' = \frac{a \cdot P(r)}{a \cdot P(r) + b \cdot (1 - P(r))}. \tag{10}$$

We want the expression (6) to be representable in this form (10). So, we arrive at the following definition:

4 First Result

Definition 1. *A monotonic function $P(r) : [0, \infty) \to [0, 1]$ is called* optimal *if, for every $\lambda > 0$, there exist values $a(\lambda)$ and $b(\lambda)$ for which*

$$P(\lambda \cdot r) = \frac{a(\lambda) \cdot P(r)}{a(\lambda) \cdot P(r) + b(\lambda) \cdot (1 - P(r))}. \tag{11}$$

Comment. In other words, we require that the 2-parametric family of functions $F = \left\{ \dfrac{a \cdot P(r)}{a \cdot P(r) + b} \right\}$ corresponding to Bayesian learning be *scale-invariant* under a "re-scaling" $r \to \lambda \cdot r$.

Theorem 1. *Every optimal function $P(r)$ has the form*

$$P(r) = \frac{r^\alpha}{r^\alpha + c} \tag{12}$$

for some real numbers α and c.

In other words, for the optimal function $P(r)$, we have

$$P = \frac{R_t(T)^\alpha}{R_t(T)^\alpha + c \cdot \theta_t(A)^\alpha}. \tag{13}$$

If we re-scale the threshold by calling $\theta' = c^{1/\alpha} \cdot \theta$ the new threshold, then the formula (13) simplifies into

$$P = \frac{R_t(T)^\alpha}{R_t(T)^\alpha + \theta_t(A)^\alpha}. \tag{14}$$

Thus, we show that formula (14) – which is a minor generalization of the original formula (2) – is indeed optimal.

5 Proof of Theorem 1

It is known that many formulas in probability theory can be simplified if instead of the probability P, we consider the corresponding odds

$$O = \frac{P}{1-P}. \tag{15}$$

(If we know the odds O, then we can reconstruct the probability P as $P = O/(1+O)$.) The right-hand side of the formula (11) can be represented in terms of odds $O(r)$, if we divide both the numerator and the denominators by $1 - P(r)$. As a result, we get the following formula:

$$P(\lambda \cdot r) = \frac{a(\lambda) \cdot O(r)}{a(\lambda) \cdot O(r) + b(\lambda)}. \tag{16}$$

Based on this formula, we can compute the corresponding odds $O(\lambda \cdot r)$: first, we compute the value

$$1 - P(\lambda \cdot r) = \frac{b(\lambda)}{a(\lambda) \cdot O(r) + b(\lambda)}, \tag{17}$$

and then divide (16) by (17), resulting in:

$$O(\lambda \cdot r) = c(\lambda) \cdot O(r), \tag{18}$$

where we denoted $c(\lambda) = a(\lambda)/b(\lambda)$. It is known (see, e.g., [3, 29]) that all monotonic solutions of the functional equation (18) are of the form $O(r) = C \cdot r^{\alpha}$. Therefore, we can reconstruct the probability $P(r)$ as

$$P(r) = \frac{O(r)}{O(r)+1} = \frac{C \cdot r^{\alpha}}{C \cdot r^{\alpha}+1}. \tag{19}$$

Dividing both the numerator and the denominator of the right-hand side by C and denoting $c = 1/C$, we get the desired formula (12). Q.E.D.

6 From Informally "Optimal" to Formally Optimal Selections

6.1 In the Previous Section, We Used Informal "Optimality"

In the above text, we argued that if a selection of a probability function is optimal (in some reasonable sense), than it is natural to expect that this selection should be scale-invariant. We used this argument to justify the empirical selection of a probability function – or, to be more precise, the empirical selection of a 2-parametric family $F = \left\{ \dfrac{a \cdot P(r)}{a \cdot P(r) + b} \right\}$.

In this section, we will go one step further, and explain that the empirical selection is indeed optimal – in the precise mathematical sense of this word.

In these terms, the question is how to select, out of all possible families, the family which is optimal in some reasonable sense, i.e., which is optimal in the sense of some optimality criterion.

6.2 What is an Optimality Criterion?

When we say that some *optimality criterion* is given, we mean that, given two different families F and F', we can decide whether the first or the second one is better, or whether these families are equivalent w.r.t. the given criterion. In mathematical terms, this means that we have a *pre-ordering relation* $\preceq$ on the set of all possible families.

One way to approach the problem of choosing the "best" family F is to select *one* optimality criterion, and to find a family that is the best with respect to this criterion. The main drawback of this approach is that there can be different optimality criteria, and they can lead to different optimal solutions. It is, therefore, desirable not only to describe a family that is optimal relative to some criterion, but to describe *all* families that can be optimal relative

to different natural criteria[4]. In this section, we are planning to implement exactly this more ambitious task.

6.3 Examples of Optimality Criteria

Pre-ordering is the general formulation of optimization problems in general, not only of the problem of choosing a family F. In general optimization theory, in which we are comparing arbitrary *alternatives* a', a'', ..., from a given set A, the most frequent case of such a pre-ordering is when a *numerical criterion* is used, i.e., when a function $J : A \to R$ is given for which $a' \preceq a''$ iff $J(a') \leq J(a'')$.

Several natural numerical criteria can be proposed for choosing a function J. For example, we can take, as a criterion, the *average* computation time (average in the sense of some natural probability measure on the set of all problems).

Alternatively, we can fix a class of problems, and take the largest (worst-case) computation time for problems of this class as the desired (numerical) optimality criterion.

Many other criteria of this type can be (and have actually been) proposed. For such "worst-case" optimality criteria, it often happens that there are several different alternatives that perform equally well in the worst case, but whose performance differ drastically in the average cases. In this case, it makes sense, among all the alternatives with the optimal worst-case behavior, to choose the one for which the average behavior is the best possible. This very natural idea leads to the optimality criterion that is not described by one numerical optimality criterion $J(a)$: in this case, we need *two* functions: $J_1(a)$ describes the worst-case behavior, $J_2(a)$ describes the average-case behavior, and $a \preceq b$ iff either $J_1(a) < J_1(b)$, or $J_1(a) = J_1(b)$ and $J_2(a) \leq J_2(b)$.

We could further specify the described optimality criterion and end up with *one* natural criterion. However, as we have already mentioned, the goal of this chapter is not to find *one* family that is optimal relative to some criterion, but to describe *all* families that are optimal relative to some natural optimality criteria. In view of this goal, in the following text, we will not specify the criterion, but, vice versa, we will describe a very general class of *natural* optimality criteria.

So, let us formulate what "natural" means.

[4] In this phrase, the word "natural" is used informally. We basically want to say that from the purely mathematical viewpoint, there can be weird ("unnatural") optimality criteria. In our text, we will only consider criteria that satisfy some requirements that we would, from the common sense viewpoint, consider reasonable and natural.

6.4 What Optimality Criteria are Natural?

It is reasonable to require that the relative quality of two families does not change if we simply change the threshold, i.e., replace the function $P(r)$ with $P(\lambda \cdot r)$, and correspondingly, the family $F = \left\{ \dfrac{a \cdot P(r)}{a \cdot P(r) + b} \right\}$ with the "re-scaled" family $T_\lambda(F) \stackrel{\text{def}}{=} \left\{ \dfrac{a \cdot P(\lambda \cdot r)}{a \cdot P(\lambda \cdot r) + b} \right\}$.

There is one more reasonable requirement for a criterion, that is related with the following idea: If the criterion does not select a single optimal family, i.e., if it considers several different families equally good, then we can always use some other criterion to help select between these "equally good" ones, thus designing a two-step criterion. If this new criterion still does not select a unique family, we can continue this process until we arrive at a combination multi-step criterion for which there is only one optimal family. Therefore, we can always assume that our criterion is *final* in this sense.

Definition 2. *By an* optimality criterion*, we mean a pre-ordering (i.e., a transitive reflexive relation) $\preceq$ on the set A of all possible families. An optimality criterion $\preceq$ is called:*

- *scale-invariant if for all F, F', and $\lambda > 0$, $F \preceq F'$ implies $T_\lambda(F) \preceq T_\lambda(F')$;*
- *final if there exists one and only one family F that is preferable to all the others, i.e., for which $F' \preceq F$ for all $F' \neq F$.*

Theorem 2.

- *If a family F is optimal w.r.t. some scale-invariant final optimality criterion, then this family F is generated by $P(r) = r^\alpha$ for some $\alpha > 0$.*
- *For families corresponding to $P(r) = r^\alpha$, there exists a scale-invariant final optimality criterion for which the only optimal family is this family.*

Comment. In other words, if the optimality criterion satisfies the above-described natural properties, then *the optimal function is $P(r) = r^\alpha$.*

6.5 Proof of Theorem 2

We have already shown, in the proof of Theorem 1, that:

- for $P(r) = r^\alpha$, the corresponding family is scale-invariant, and
- vice versa, that if a family is scale-invariant, then it corresponds to $P(r) = r^\alpha$.

1°. To prove the first part of Theorem 2, we thus need to show that for every scale-invariant and final optimality criterion, the corresponding optimal family F_{opt} is scale-invariant, i.e., that $T_\lambda(F_{\text{opt}}) = F_{\text{opt}}$ for all $\lambda > 0$. Then, the result will follow from Theorem 1.

Indeed, the transformation T_λ is invertible, its inverse transformation is a scaling by $1/\lambda$: $T_\lambda^{-1} = T_{1/\lambda}$. Now, from the optimality of F_{opt}, we conclude that for every $F' \in A$, $T_\lambda^{-1}(F') \preceq F_{\text{opt}}$. From the invariance of the optimality criterion, we can now conclude that $F' \preceq T_\lambda(F_{\text{opt}})$. This is true for all $F' \in A$ and therefore, the family $T(F_{\text{opt}})$ is optimal.

But since the criterion is final, there is only one optimal indicator function; hence, $T_\lambda(F_{\text{opt}}) = F_{\text{opt}}$. So, the optimal family is indeed invariant and hence, due to Theorem 1, it corresponds to $P(r) = r^\alpha$. The first part is proven.

2°. Let us now prove the second part of Proposition 2. Let $P(r) = r^\alpha$, and let F_0 be the corresponding family. We will then define the optimality criterion as follows: $F \preceq F'$ iff F' is equal to this F_0.

Since the family F_0 is scale-invariant, thus the defined optimality criterion is also scale-invariant. It is also clearly final.

The family F_0 is clearly optimal w.r.t. this scale-invariant and final optimality criterion. The theorem is proven.

7 Discussion

Traditionally, the choice of a function $P(r)$ is done empirically, by comparing the results of different choices. Two related questions naturally arise:

- first, a *theoretical* question: how can we explain the empirical selection?
- second, a *practical* question: an empirical choice is made by using only finitely many functions; is this choice indeed the best – or there are other, even better functions $P(r)$, which we did not discover because we did not try them?

Our result answers both questions:

- first, we provide a theoretical explanation for the optimality of the empirical choice;
- thus, by proving that these empirical formulas are optimal not only in comparison with other functions that we have tried, but in comparison with all possible functions $P(r)$, we enable the practitioners not to waste time on trying different functions $P(r)$.

8 Extending the Optimality Result to a Broader Context

8.1 Formulation of a More General Problem

Swarm intelligence techniques are a class of methodology for solving global optimization problems. In this chapter, we have discussed how to optimally select techniques from this class. It is reasonable to consider this problem in a broader setting: how can we optimally select techniques for solving global

optimization problems – without necessarily restricting ourselves to swarm intelligence.

The need to make such a selection comes from the fact that, in general, the problem of finding the exact values x that minimize a given objective function $f(x)$ is computationally difficult (NP-hard); see, e.g., [41]. Crudely speaking, NP-hardness means that (provided that P≠NP) it is not possible to have an algorithm that solves *all* optimization problems in reasonable time. In other words, no matter how good is an algorithm for solving global optimization optimization problems, there will always be cases in which better results are possible.

Since we cannot hope for a single algorithm for global optimization, new algorithms are constantly designed, and the existing algorithms are constantly modified. As a result, we have a wide variety of different global optimization techniques and methods; see, e.g., [17, 20, 22, 30, 38]. In particular, there exist numerous heuristic and semi-heuristic techniques which – similar to swarm intelligence techniques – emulate the way optimization is done in nature; e.g., genetic algorithms simulate the biological evolution which, in general, leads to the birth and survival individuals and species which are best fit for a given environment; see, e.g., [28].

Because of this variety of different global optimization techniques, every time we have a new optimization problem, we must select the best technique for solving this problem. This selection problem is made even more complex by the fact that most techniques for solving global optimization problems have *parameters* that need to be adjusted to the problem or to the class of problems. For example, in gradient methods, we can select different step sizes.

When we have a *single* parameter (or few parameters) to choose, it is possible to empirically try many values and come up with an (almost) optimal value. Thus, in such situations, we can come up with optimal version of the corresponding technique. In other approaches, e.g., in swarm intelligence, instead of selecting the value of single *number*-valued parameter, we have select the auxiliary *function*. It is not practically possible to test all possible functions, so it is not easy to come up with an optimal version of the corresponding technique.

In this chapter, we described an indirect way of finding the optimal version of swarm intelligence techniques. It is desirable to consider a more general problem of selecting the best auxiliary function within a given global optimization technique – so that we would be able to either analytically solve the problem of finding the optimal auxiliary function, or at least reduce this problem to an easier-to-solve problem of finding a few numerical parameters.

8.2 Case Study: Optimal Selection of Penalty (Barrier) Functions

A well-known Lagrange multiplier method minimizes a function $f(x)$ under a constraint $g(x) = 0$ by reducing it to the un-constrained problem of optimizing a new objective function $f(x) + \lambda \cdot g(x)$. One of the known approaches

to solving a similar problem with a constraint $g(x) \geq 0$ is the *penalty* (*barrier*) method in which we reduce the original problem to the un-constrained problem of optimizing a new objective function $f(x) + \lambda \cdot g(x) + \mu \cdot P(g(x))$, for an appropriate (non-linear) penalty function $P(y)$. Traditionally, the most widely used penalty functions are $P(y) = y \cdot \ln(y)$ and $P(y) = y^2$. How can we select an optimal penalty function? Or, to be more precise, how can we select the optimal family $\{\lambda \cdot y + \mu \cdot P(y)\}$?

The first natural requirements is that the optimal penalty function $P(y)$ should be smooth. Smoothness is needed because smooth functions are easier to optimize, and we therefore want our techniques to preserve smoothness.

In solving a similar problem from swarm intelligence, we used the argument that the optimal expression should not change if we simply change the threshold and thus, re-scale the parameter r. For penalty functions, similarly, the measuring unit for measuring the quantity y is often a matter of arbitrary choice: we can use meters or feet to measure length, we can use pounds or kilograms to measure weight, etc. If a selection of the penalty function $P(y)$ is "optimal" (in some intuitive sense), then the results of using this penalty functions should not change if we simply change the measuring unit for measuring y – i.e., replace each value y with a new value $C \cdot y$, where C is the ratio of the corresponding units. Indeed, otherwise, if the "quality" of the resulting penalty method changes with this "re-scaling", we could change the unit and get a better penalty function $P(y)$ – which contradicts to our assumption that the selection of $P(y)$ is already optimal.

So, the "optimal" choices $P(y)$ can be determined from the requirement that the family $\{\lambda \cdot y + \mu \cdot P(y)\}$ be invariant under the corresponding re-scaling.

Definition 3. *A 2-parametric family of functions $F = \{\lambda \cdot y + \mu \cdot P(y)\}$ is called* scale-invariant *if for every $C > 0$, it coincides with the family $T_C(F) \stackrel{\text{def}}{=} \{\lambda \cdot C \cdot y + \mu \cdot P(C \cdot y)\}$.*

At first glance, scale-invariance is a reasonable but weak property. It turns out, however, that this seemingly weak property actually almost uniquely determines the optimal selection of penalty functions; see, e.g., [29].

Proposition 1. *If a family $\{\lambda \cdot y + \mu \cdot P(y)\}$ is scale-invariant, then this family corresponds to $P(y) = y^\alpha$ or to $P(y) = y \cdot \ln(y)$.*

8.3 Proof of Proposition 1

Since the family is scale-invariant, for every C, the re-scaled function $P(C \cdot y)$ must belong to the same family, i.e., there must exist $\lambda(C)$ and $\mu(C)$ for which

$$P(C \cdot y) = \lambda(C) \cdot y + \mu(C) \cdot P(y) \tag{20}$$

for all C and y.

Differentiating both sides of (20) by C and setting $C = 1$, we conclude that

$$y \cdot \frac{dP}{dy} = \lambda_0 \cdot y + \mu_0 \cdot P(y), \tag{21}$$

where $\lambda_0 \stackrel{\text{def}}{=} \frac{d\lambda(C)}{dC}_{|C=1}$ and $\mu_0 \stackrel{\text{def}}{=} \frac{d\mu(C)}{dC}_{|C=1}$. One can check that the only solutions to these equation are $P(y) = C_1 \cdot y + C_2 \cdot y^{\mu_0}$ (when $\mu_0 \neq 1$) and $P(y) = C_1 \cdot y + C_2 \cdot y \cdot \ln(y)$ (when $\mu_0 = 1$). Thus, the only scale-invariant families $\{\lambda \cdot y + \mu \cdot P(y)\}$ are families corresponding to $P(y) = y \cdot \ln(y)$ and $P(y) = y^\alpha$ for some real number α.

Thus, under any scale-invariant optimality criterion, the optimal penalty function must indeed take one of the desired forms. Q.E.D.

Comments.

- We can also show that for every scale-invariant final optimality criterion, the optimal family corresponds to $P(y) = y \cdot \ln(y)$ and $P(y) = y^\alpha$.
- This example also shows that we can go beyond theoretical justification of empirically best heuristic, towards finding new optimal heuristics: indeed, for penalty functions, instead of two-parameter families $\{\lambda \cdot y + \mu \cdot P(y)\}$, we can consider multiple-parameter families

  $$\{\lambda \cdot y + \mu_1 \cdot P_1(y) + \ldots + \mu_m \cdot P_m(y)\}$$

  for several functions $P_1(y), \ldots, P_m(y)$. In this case, the optimal functions have also been theoretically found: they are of the type

  $$P_i(y) = y^{\alpha_i} \cdot (\ln(y))^{p_i}$$

  for real (or complex) values α_i and non-negative integer values of p_i [29].

8.4 Other Examples

Similar symmetry-based techniques provide an explanation of several other empirically optimal techniques.

How to bisect a box. For example, many optimization algorithms are based on the branch-and-bound idea, where we subdivide the original domain into several smaller subdomains – and thus, reduce the original difficult-to-solve problem of optimizaing the objective function $f(x)$ over the entire domain to several easier-to-solve problems of optimizing $f(x)$ over smaller domains (usually, boxes). Some of these boxes must be further subdivided, etc. Two natural questions arise:

- which box should we select for bisection?
- which variable shall we use to bisect the selected box?

To answer both questions, several heuristic techniques have been proposed, and there has been an extensive empirical comparative analysis of these techniques. It turns out that for both questions, the symmetry-based approach enables us to theoretically justify the empirical selection:

- Until recently, for subdivision, a box B was selected for which the computed lower bound $\underline{f}(B)$ was the smallest possible. Recently (see, e.g, [11, 12]), it was shown that the optimization algorithms converge much faster if we select, instead, a box B with the largest possible value of the ratio
$$I_0 = \frac{\widetilde{f} - \underline{f}(B)}{\overline{f}(B) - \underline{f}(B)},$$
where $\widetilde{f}$ is a current upper bound on the actual global minimum. In [25], we give a symmetry-based theoretical justification for this empirical criterion. Namely, we consider all possible indicator functions $I(\underline{f}(B), \overline{f}(B), \widetilde{f})$, and we show that:
  - first, that the empirically best criterion I_0 is the only one that is *invariant* w.r.t. some reasonable symmetries – namely, shift and scaling; and
  - second, that this criterion is *optimal* in some (symmetry-related) reasonable sense.
- We can bisect a given box in n different ways, depending on which of n sides we decided to halve. So, the natural question appears: which side should we cut? i.e., where to bisect a given box? Historically the first idea was to cut the *longest* side (for which $x_i^U - x_i^L \to \max$). It was shown (in [33, 34]) that much better results are achieved if we choose a side i for which $|d_i|(x_i^U - x_i^L) \to \max$, where d_i is the known approximation for the partial derivative $\dfrac{\partial f}{\partial x_i}$. In [23], we consider arbitrary selection criteria, i.e., functions
$$S(f, d_1, \ldots, d_n, x_1^L, x_1^U, \ldots, x_n^L, x_n^U),$$
which map available information into an index $S \in \{1, 2, \ldots, n\}$, and we show that the empirically best box-splitting strategy is the only scale-invariant one – and is, thus, optimal under any scale-invariant final optimality criterion.

How to enlarge a box. Sometimes, it is beneficial to (slightly) enlarge the original (non-degenerate) box $[x^L, x^U]$ and thus improve the performance of the algorithm; the empirically efficient "epsilon-inflation" technique [35, 36]
$$[x_i^L, x_i^U] \to [(1+\varepsilon)x_i^L - \varepsilon \cdot x_i^U, (1+\varepsilon)x_i^U - \varepsilon \cdot x_i^L]$$
was proven to be the only shift- and scale-invariant technique and thus, the only one optimal under an arbitrary shift-invariant and scale-invariant optimality criterion [26] (see also [37]).

Convex-based techniques. Several algorithms for solving convex global optimization problems are based on the fact that for convex functions there exist efficient algorithms for finding the global minimum. There are numerous effective global optimization techniques that reduce the general global optimization problems to convex ones; see, e.g., [17, 38]. Empirically, among these techniques, the best are αBB method [4, 5, 17, 27] and its modifications recently proposed in [6, 7]. It turns out [18] that this empirical optimality can also be explained via shift- and scale-invariance.

Simulated annealing and genetic algorithms. By using shift-invariance, we can also explain why the probability proportional to $\exp(-\gamma \cdot f(x))$ is optimal in simulated annealing [29].

By using scale- and shift-invariance, we explain why exponential and power re-scalings of the objective function are optimal in genetic algorithms [29].

Acknowledgments

This work was supported in part by NASA under cooperative agreement NCC5-209 and grant NCC 2-1232, by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grants numbers F49620-95-1-0518 and F49620-00-1-0365, by grant No. W-00016 from the U.S.-Czech Science and Technology Joint Fund, by NSF grants CDA-9522207, ERA-0112968 and 9710940 Mexico/Conacyt, and by IEEE/ACM SC2001 Minority Serving Institutions Participation Grant.

Smith's work was sponsored in part by the Department of Defense Advanced Research Projects Agency and the U.S. Air Force Rome Research Laboratory under contract F30602-00-2-0503, by the National Aeronautics and Space Administration under Contract NCC2-1243 and by the CMU Robotics Institute.

The authors are thankful to the editors and to the anonymous referees for valuable suggestions.

References

1. Abraham A, Grosan C, Ramos V, Eds. (2006) Swarm Intelligence and Data Mining. Springer-Verlag, Berlin – Heidelberg
2. Abraham A, Grosan C, Ramos V, Eds. (2006) Stigmergic Optimization. Springer-Verlag, Berlin – Heidelberg
3. Aczel J (1966) Lectures on functional equations and their applications. Academic Press, New York, London
4. Adjiman CS, Androulakis I, Floudas CA (1998) A global optimization method, αBB, for general twice-differentiable constrained NLP II. Implementation and computational results. Computers and Chemical Engineering, 22:1159–1179

5. Adjiman CS, Dallwig S, Androulakis I, Floudas CA (1998) A global optimization method, αBB, for general twice-differentiable constrained NLP I. Theoretical aspects. Computers and Chemical Engineering, 22:1137–1158
6. Akrotirianakis IG, Floudas CA (2004) Computational experience with a new class of convex underestimators: box-constrained NLP problems. Journal of Global Optimization, 29:249–264
7. Akrotirianakis IG, Floudas CA (2006) A new class of improved convex underestimators for twice continuously differentiable constrained NLPs. Journal of Global Optimization, to appear.
8. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford
9. Bonabeau E, Sobkowski A, Theraulaz G, Jean-Louis Deneubourg J-L (1997) Adaptive Task Allocation Inspired by a Model of Division of Labor in Social Insects. In: Lundh D, Olsson B (eds), Bio Computation and Emegent Computing, World Scientific, Singapore, 36–45
10. Bonabeau E, Théraulaz G, Denebourg J-L (1996) Quantitative study of the fixed response threshold model for the regulation of division of labour in insect societies. Proc. Roy. Soc. B 263:1565–1569
11. Casado LG, García I (1998) New load balancing criterion for parallel interval global optimization algorithm, In: Proc. of the 16th IASTED International Conference, Garmisch-Partenkirchen, Germany, February 1998, 321–323
12. Casado LG, García I, Csendes T (2000) A new multisection technique in interval methods for global optimization. Computing, 65:263–269
13. Cicirello VA, Smith SF (2001) Ant colony control for autonomous decentralized shop floor routing. In: Proc. 5th Int'l Symposium on Autonomous Decentralized Systems ISADS'2001, IEEE Computer Society Press, March 2001.
14. Cicirello, VA, Smith SF (2001) Insect societies and manufacturing. In: Proc. of IJCAI'01 Workshop on AI and Manufacturing: New AI Paradigms and Manufacturing, August 2001.
15. Cicirello VA, Smith SF (2001) Wasp nests for self-configurable factories. In: Agents'2001, Proc. 5th Int'l Conference on Autonomous Agents, ACM Press, May-June 2001.
16. Cicirello VA, Smith SF (2001) Improved routing wasps for distributed factory control. In: Proc. of IJCAI'01 Workshop on AI and Manufacturing: New AI Paradigms and Manufacturing, August 2001.
17. Floudas CA (2000) Deterministic Global Optimization: Theory, Methods, and Applications. Kluwer, Dordrecht
18. Floudas CA, Kreinovich V (2006) Towards Optimal Techniques for Solving Global Optimization Problems: Symmetry-Based Approach. In: Torn A, Zilinskas, J (eds.), Models and Algorithms for Global Optimization, Springer, Dordrecht (to appear)
19. Garey M, Johnson, D (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco
20. Horst R, Pardalos PM, eds. (1995) Handbook of Global Optimization, Kluwer, Dordrecht
21. Jaynes ET, Bretthorst GL, ed. (2003) Probability Theory: The Logic of Science. Cambridge University Press, Cambridge, UK
22. Kearfott RB (1996) Rigorous Global Search: Continuous Problems. Kluwer, Dordrecht

23. Kearfott RB, Kreinovich V (1998) Where to Bisect a Box? A Theoretical Explanation of the Experimental Results. In: Alefeld G, Trejo RA (eds.), Interval Computations and its Applications to Reasoning Under Uncertainty, Knowledge Representation, and Control Theory. Proceedings of MEXICON'98, Workshop on Interval Computations, 4th World Congress on Expert Systems, México City, México
24. Kennedy J, Eberhart R, Shi Y (2001) Swarm Intelligence. Morgan Kaufmann, San Francisco, California
25. Kreinovich V, Csendes T (2001) Theoretical Justification of a Heuristic Subbox Selection Criterion. Central European Journal of Operations Research CEJOR, 9:255–265
26. Kreinovich V, Starks SA, Mayer G (1997) On a Theoretical Justification of The Choice of Epsilon-Inflation in PASCAL-XSC. Reliable Computing, 3:437–452
27. Maranas CD, Floudas CA (1994) Global minimal potential energy conformations for small molecules. Journal of Global Optimization, 4:135–170
28. Michalewicz Z (1996) Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin
29. Nguyen HT, Kreinovich V (1997) Applications of continuous mathematics in computer science. Kluwer, Dordrecht
30. Pinter JD (1996) Global Optimization in Action. Kluwer, Dordrecht
31. Ramos V, Merelo JJ (2002) Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning. In: Alba E, Herrera F, Merelo JJ, et al. (Eds.), Proc. of 1st Spanish Conference on Evolutionary and Bio-Inspired Algorithms AEB'02, Centro Univ. de Mèrida, Mérida, Spain, Feb. 6–8, 2002, pp. 284–293
32. Ramos V, Muge F, Pina P (2002) Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies. In: Ruiz-del-Solar A, Abraham A, Köppen M (Eds.), Frontiers in Artificial Intelligence and Applications, Soft Computing Systems – Design, Management and Applications, 2nd Int. Conf. on Hybrid Intelligent Systems, Santiago, Chile, Dec. 2002, IOS Press, 87:500–509
33. Ratz D (1992) Automatische Ergebnisverifikation bei globalen Optimierungsproblemen. Ph.D. dissertation, Universität Karlsruhe
34. Ratz D (1994) Box-Splitting Strategies for the Interval Gauss–Seidel Step in a Global Optimization Method. Computing, 53:337–354
35. Rump SM (1980) Kleine Fehlerschranken bei Matrixproblemen, Ph.D. dissertation, Universität Karlsruhe
36. Rump SM (1992) On the solution of interval linear systems. Computing, 47:337–353
37. Rump SM (1998) A Note on Epsilon-Inflation. Reliable Computing, 4(4):371–375
38. Tawarmalani M, Sahinidis NV (2002) Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications. Kluwer, Dordrecht
39. Theraulaz G, Bonabeau E, Deneubourg JL (1998) Reponse threshold reinforcement and division of labor in insect societies, Proceeeings of the Royal Society of London B 263(1393):327–335
40. Vapnik VN (2000) The Nature of Statistical Learning Theory. Springer-Verlag, New York

41. Vavasis SA (1991) Nonlinear Optimization: Complexity Issues. Oxford University Press, New York
42. Wadsworth HM (ed) (1990) Handbook of statistical methods for engineers and scientists. McGraw-Hill Publishing Co., New York