Fuzzy Control of Stochastic Global Optimization Algorithms and Very Fast Simulated Reannealing

H i m e A g u i a r e O l i v e i r a J u n i o r
Electronic and Control Engineer , Independent Researcher
Rio de Janeiro , RJ , BRAZIL
h i m e @ engineer.com

Abstract
This paper presents a fuzzy control approach for improving convergence time in stochastic global minimization algorithms .We show concrete results when the method is applied to an efficient algorithm based on ideas related to Simulated Annealing.

1. Introduction

The task of global minimization of numerical functions has paramount importance in several areas of Knowledge . It appears in fields like Engineering , Finance , Management , Medicine , etc..

In practical cases , the function to be minimized shows itself in the form of a cost measure that varies with several parameters and is subject to certain constraints , imposed by its environment . When the objective function is "well-behaved" , there are several methods to find points in which it attains the minimum value , satisfying the constraints.

Problems start to rise when the given function presents several local minima, each one having its own attraction basin , making , typically , the final result to depend on the starting point . Unfortunately , most real problems originate very complex objective functions that are nonlinear , discontinuous , multi-modal , high-dimensional , etc. .

To solve such a class of problems , stochastic methods seem to be a good (sometimes , the only) way to go . Genetic algorithms and simulated annealing are among the most popular approaches to stochastic global optimization .

The problem in that case is related to speed of convergence and , in the genetic approach , warranty of the ability to reach a global optimum , under general conditions . Pure annealing methods , by the other side , have results assuring its convergence to a global minimum with probability 1 , but the performance presented by most implementations is not very encouraging . Despite this , researchers have found ways to overcome the limitations of original annealing schemes , leading to VFSR (Very Fast Simulated Re-annealing) , that is a sophisticated and really effective global optimization method. VFSR is particularly well suited to applications involving neuro-fuzzy systems and neural network training, taking into account its superior performance and simplicity.

ASA (Adaptive Simulated Annealing) is an implementation of VFSR , that brings us the benefits of being publicly available , parameterizable and well-maintained.

Besides , ASA shows itself as an alternative to GAs , according to the published benchmarks , that demonstrate its good quality .

Unfortunately , stochastic global optimization algorithms share a few bad characteristics like , for example , large periods of poor improvement in their way to a global extremum . In SA implementations , that is mainly due to the "cooling" schedule , whose speed is limited by the characteristics of probability density functions (PDFs) used to generate new candidate points . In this manner , if we choose to employ the so called Boltzmann Annealing (BA) , the "temperature" has to be lowered at a maximum rate of T(k) = T(0) / ln(k) . In case of Fast Annealing (FA) , the schedule becomes T(k) = T(0) / k , if assurance of convergence with probability 1 is to be maintained , resulting in a faster schedule. ASA has an even better default scheme , given by

$$T_i(k) = T_i(0) \times \exp(-C_i k^{\frac{1}{D}})$$
$$(\, C_i = \text{user - defined parameter} \,)$$

, thanks to its improved generating distribution . Note that subscripts indicate independent evolution of temperatures for each parameter dimension . In addition , it's possible for the ASA user to take advantage of Simulated Quenching (SQ) , resulting in

$$T_i(k) = T_i(0) \times \exp(-C_i k^{\frac{Q_i}{D}})$$
$$(\, Q_i = \text{quenching parameter} \,)$$

If we set quenching parameters to values greater than 1 , there is a gain in speed but the convergence to a global optimum is no longer assured (see [1]).

Such a procedure could be used for higher-dimensional parameter spaces , when computational resources are scarce.

Despite (or because) all that features , there is much tuning to be done, from the user's viewpoint ("Nonlinear systems are typically not typical" , as says Lester Ingber – the creator of ASA).

In the sequel , we describe a well-succeeded approach to accelerate ASA algorithm using a simple Mamdani fuzzy controller that dynamically adjusts certain user parameters related to quenching . It's shown that , by increasing the algorithm's perception of slow convergence , it's possible to speed it up significantly and to reduce enormously (perhaps eliminate) the user task of parameter tuning . That is done without changing the original ASA code .

2. General structure of Simulated Annealing algorithms

SA algorithms are based on the ideas introduced by N. Metropolis and others ,widely known as Monte Carlo importance-sampling techniques.

The method uses three fundamental components , that have great impact on the final implementation :

- A probability density function g(.) , used in the generation of new candidate points .
- A PDF a(.) , used in the acceptance/rejection of new generated points .
- A schedule T(.) , that determines how the temperatures will vary during the execution of the algorithm , that is , their dynamical profile .

The basic approach is to generate a starting point , chosen according to convenient criteria , and to set the initial temperature so that the space state could be "sufficiently" explored . After that , new points are iteratively generated according to the generating PDF g (.) and probabilistically accepted or rejected , as dictated by PDF a(.) . If acceptance occurs , the candidate point becomes the current base point . During the run , temperatures are lowered and that reduces the probability of acceptance of new generated points with higher cost values than that of the current point . However , there is a non-zero probability of going "uphill", giving the opportunity to escape from local minima .

3. Main features of ASA/VFSR

As was said before , ASA , that is a practical realization of VFSR , is based upon the concept of simulated annealing , possessing in addition a great number of positive features . Among them we find :

. *Re-annealing* – it is the dynamical re-scaling of parameter temperatures , adapting generating distributions for each dimension according to the sensitivities shown in a given search direction . In a few words , if the cost function doesn't show significant variations when we vary one given parameter , it may be worth to extend the search interval for that particular dimension and vice-versa .

. *Quenching facilities* – as we cited before , ASA code has several user settable parameters related to quenching that allow us to improve the convergence speed . So , it's possible to tailor parameter and cost temperatures evolution by changing selected quenching factors in an easy and clean manner .

. *High level of parameterization* – ASA is coded in such a way that we can alter virtually any building block without significant effort . This way , it's possible to change generation/acceptance processes behavior , stopping criteria , starting point generation , log file detail level , etc. .

ASA was designed to find global minima belonging to a given compact subset of n-dimensional Euclidean space . It generates points component-wise , according to

$x_{i+1} = x_i + \Delta x_i$,

with $\Delta x_i = y_i (B_i - A_i)$,

$[A_i, B_i]$ = i - th dimension parameter range ,

$y_i \in [-1,1]$ is given by

$y_i = \mathrm{sgn}(u_i - 1/2) T_i [(1 + 1/T_i)^{|2u_i - 1|} - 1]$ where

$u_i \in [0,1]$ is generated from uniform distribution ,

T_i = current temperature relative to dimension i .

The compactness of the search space is not a severe limitation in practice , and in the absence of prior information about the possible location of global minima , it suffices to choose a sufficiently large hyper-rectangular domain .

4. Fuzzy quenching control of ASA

As we said before , by using the so-called simulated quenching we can improve the efficiency of the annealing schedule , assuming the risk of reaching a non-global minimum . In certain cases , however , we have no choice , as is the case for domains with very large number of dimensions , for instance .

To solve this problem , a fuzzy controller was designed . The approach is simple : we consider ASA as a MISO (Multiple Input Single Output) dynamical system and "close the loop", by sampling ASA's output (current cost function value) and acting on its inputs (a subset of settable parameters related to quenching) according to a fuzzy law (quenching controller) that does nothing more than emulate human reasoning about the underlying process . So , by the use of an intelligent controller we can speed up and slow down the temperature schedule , in addition to being able to take evasive actions in case of premature convergence .

We faced two main obstacles to get to our target :

1 – How the sampled outputs (cost function values) could tell us the present status of the progressing run ?
2 – How do we change ASA inputs in order to leave undesirable situations (permanence near non-global minima / slow progress) ?

The first question was handled thanks to the concept of *sub-energy* function , used in the TRUST method (see [2]) .

The sub-energy function is given by

$$SE(x, x_0) = \log(1/[1+\exp(-(f(x)-f(x_0))-a)])$$
where a is a real constant , and x_0 is the current "base point".

The base point is the best minimum point found so far . So , the function SE behaves qualitatively like the original f(.) when the search visits "better" points than the current minimum and tends to be flat in "worse" points . Thus , it's possible to assess when the search is located above , near or under the current minimum point by the inspection of values assumed by the sub-energy function . Such a detection process results in approximate conclusions like

The search is NEAR the current minimum
or
The search is VERY FAR from the current minimum

leading naturally to a fuzzy modeling opportunity .

The second question above is related to the consequent parts of the fuzzy rule base , in which we have to place corrective actions to keep the search progressing toward the global minimum . That was done by varying quenching degrees for generating and acceptance PDFs . The implementation used individual quenching factors for each dimension and one cost quenching factor .

The fuzzy controller's rule-base contains rules like

- IF AveSub IS NEAR ZERO THEN increase Quenching
- IF AveSub IS NEAR current minimum THEN increase Quenching
- IF StdDevSub IS ZERO THEN decrease Quenching

where :

- AveSub is a linguistic variable corresponding to the crisp average of last 100 sub-energy values.
- StdDevSub is a linguistic variable corresponding to the crisp standard deviation of last 100 sub-energy values.

Having outlined the structure of the whole scheme , it's time to show some practical results obtained from the optimization of some difficult functions.

5. Results

We will present four test cases in which multi-modal functions were submitted to three methods : ASA , fuzzy controlled ASA and a well known and effective floating point GA.

ASA and fuzzy ASA have exactly the same parameters; the only difference being the activation of the fuzzy controller , located in an external module that is called from within the cost function . The controller was kept unchanged across the runs , evidencing its independence relative to objective function characteristics and/or dimensionality .

The GA has the following settings :

- Population size – 75
- Elitism – ON
- Initial population identical to ASA starting point
- 3 crossover operators
- 5 mutation operators

The test functions are :

Function 1:

Domain: $\{ x \in R^3 : x_i \in [-10000, 10000] \}$

$$f(x) = x_1^2(2+\sin(120x_2)) + x_2^2(2+\sin(220x_1)) + x_3^2(2+\sin(50x_1))$$

Global minimum at (0,0,0).

Minimum value = 0.

Function 2:

Domain: $\{ x \in R^4 : x_i \in [-10, 10] \}$

$$f(x) = 100(x_2 - x_1)^2 + (1-x_1)^2 + 90(x_1 - x_3^2)^2 + (1-x_3)^2 + 10.1((x_2-1)^2 + (x_4-1)^2) + 19.8(x_2-1)(x_4-1)$$

Global minimum at (1,1,1,1).

Minimum value = 0.

Function 3:

Domain: $\{ x \in R^{50} : x_i \in [-10, 10] \}$

$$f(x) = f_1(x) + f_2(x) + f_3(x)$$

$$f_1(x) = \sum_{i=1}^{50} (i x_i^2)$$

$$f_2(x) = \sum_{i=1}^{50} (x_{i-1} + 5\sin x_i + x_{i+1}^2)^2$$

$$f_3(x) = \sum_{i=1}^{50} \ln^2(1 + | i\sin^2 x_{i-1} + 2x_i + 3x_{i+1} |)$$

with $x_0 = x_{50}$ and $x_{51} = x_1$

Global minimum at $0 \in R^{50}$.

Minimum value $= 0$.

Function 4:

Domain: The same as function 3.

$$f(x) = f_1(x) + f_2(x) + f_3(x)$$

$$f_1(x) = \sum_{i=1}^{50} (i x_i^2)$$

$$f_2(x) = \sum_{i=1}^{50} i\sin^2(x_{i-1}\sin x_i - x_i + \sin x_{i+1})$$

$$f_3(x) = \sum_{i=1}^{50} i\ln(1 + i(x_{i-1}^2 - 2x_i + 3x_{i+1} - \cos x_i + 1)^2)$$

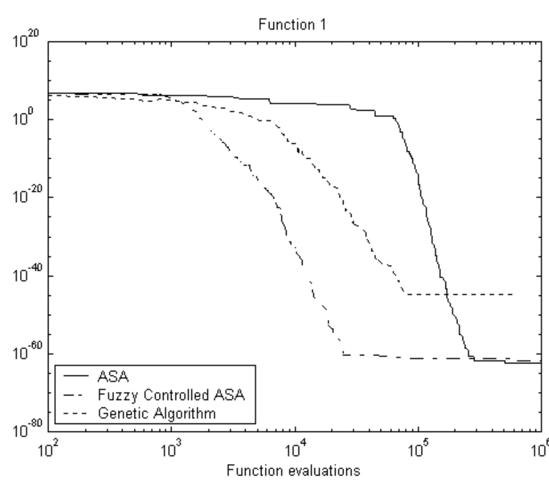
with $x_0 = x_{50}$ and $x_{51} = x_1$

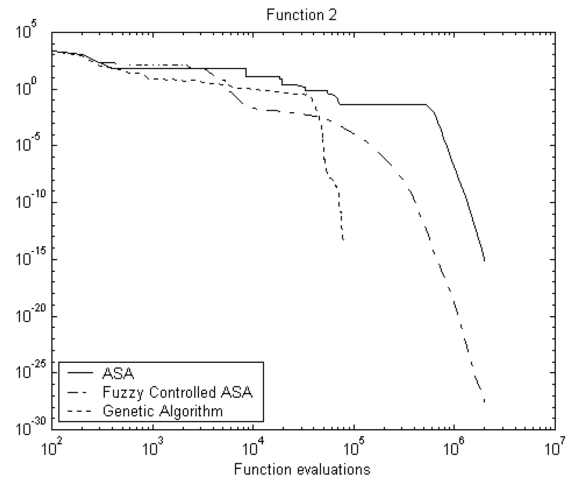
Global minimum at $0 \in R^{50}$.

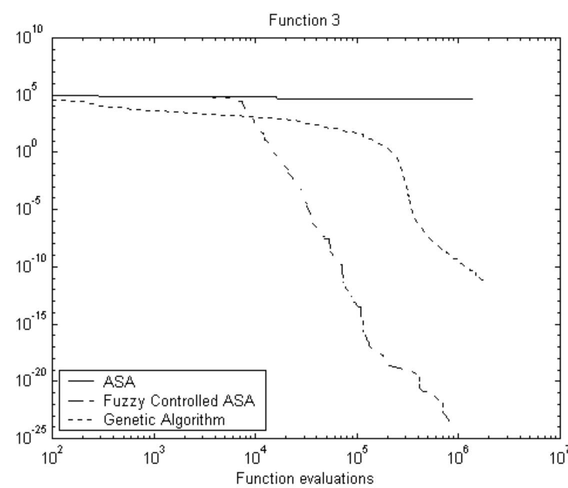
Minimum value $= 0$.

In the sequence , we show the evolution of the minimization processes . On the x-axis , we have the number of cost function evaluations and on the y-axis, the best (minimum) value found so far .

As can be seen from the graphs ,the new algorithm presents better performance in all cases when compared to original ASA . In higher-dimensional cases , it outperforms the FPGA too , presenting itself as an alternative solution to real world problems.

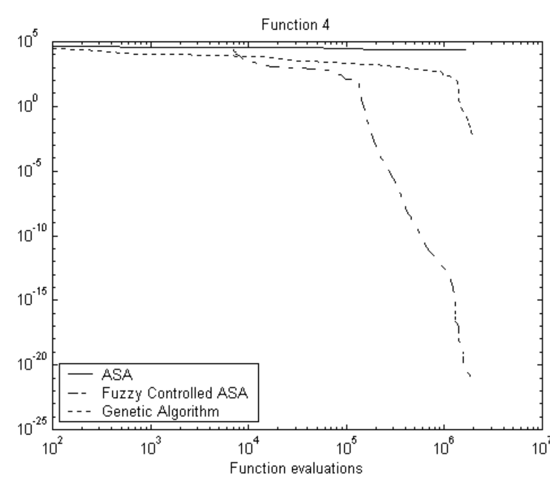






6. Conclusions

It was shown that VFSR/ASA performance can be improved by the application of fuzzy control techniques . Results showed also that fuzzy controlled



ASA can be faster than a general purpose FPGA in difficult minimization problems. It's important to note that starting points were the same for all three methods (in each run) and their location was chosen to be the "worst possible" and "very far" from basins of attraction of global minima .The graphs were constructed by averaging 50 runs for each test function / method combination .

We have used Fuzzy ASA in the training of ANNs , neuro-fuzzy systems and other "devices" , demanding difficult global minimization tasks . The practical results are very good and additional research is in progress .

7. References

[1] Ingber , L. , ASA : Lessons learned , Control and Cybernetics , invited paper to special issue 'Simulated Annealing applied to Combinatorial Optimization", 1995 . Available at www . ingber . com .

[2] Barhen,J. / Protopopescu ,V. / Reister.D., TRUST:A Deterministic Algorithm for Global Optimization , Science ,Vol. 276., May 1997, pp. 1094-1097.

[3] Pintér , J. , Global Optimization in Action , Kluwer Academic Publishers ., Dordrecht , The Netherlands, 1996.

[4] Rosen-, B. , Function Optimization based on Advanced Simulated Annealing, Available at www . ingber . com .

[5] Lee M.A. , Takagi H. , Dynamic Control of Genetic Algorithms using Fuzzy Logic Techniques , Proceedings of the 5th Int'l Conference on Genetic Algorithms (ICGA' 93) , Urbama Champaign,IL , USA , July 17-21,1993 , pp 76-83 .