Fitness Function Design for Genetic Algorithms in Cost Evaluation Based Problems.

CONFERENCE PAPER · JANUARY 1996

DOI: 10.1109/ICEC.1996.542362 · Source: DBLP

Agos C Rosa
University of Lisbon

Fitness Function Design for Genetic Algorithms in Cost Evaluation Based Problems

José Allen Lima
Systems and Robotics Institute
Instituto Superior Técnico
Av. Rovisco Pais, Torre Norte 6.21,
1096 Lisbon Codex
jalima@isr.ist.utl.pt

Nuno Gracias
Systems and Robotics Institute
Instituto Superior Técnico
Av. Rovisco Pais, Torre Norte 6.21,
1096 Lisbon Codex
ngracias@isr.ist.utl.pt

Henrique Pereira
Systems and Robotics Institute
Escola Superior de Tecnologia- IPS
Av. Rovisco Pais, Torre Norte 6.21,
1096 Lisbon Codex
hpereira@isr.ist.utl.pt

Agostinho Rosa
Systems and Robotics Institute
Instituto Superior Técnico
Av. Rovisco Pais, Torre Norte 6.21,
1096 Lisbon Codex
acrosa@isr.ist.utl.pt

Abstract-**This work presents a class of scaling functions for Genetic Algorithms. These functions imply individual performance to be expressed as a set of costs.**
Two basic functions are obtained. Both are based in exponential functions and contain a selectivity parameter assuring an adjustable degree of discernment between individuals. The first is translation invariant while the second is both translation and scale invariant.
Three examples were used to compare these scaling functions with linear scaling. A integer linear programming, a best path finding problem and a best path finding with deceiving characteristics.
In all examples, exponential based functions achieved better results than linear scaling.

I. Introduction

The choice of a fitness function can be a difficult step in Genetic Algorithms (GA). Different fitness functions promote different GA behaviour [6].
The use of proportional selection [4,7], linear scaling functions tend to produce low selective GA with a slow convergence. In order to minimise these effects it is common to scale the function.
Usually a fitness function consists of two functions [3]:

$$f(x) = g(h(x))$$

Where h is the objective function and g is a scaling function.
This work presents a different approach to the design of scaling functions. It is assumed that the objective function returns a cost or a cost vector.
Based on certain conditions and on proportional selection, two kinds of scaling functions are obtained. One is time invariant, thus called static selectivity scaling function. The other is time variant to be automatically adjusted in the GA progress. It was called dynamic selectivity scaling function. Both are based the exponential function.
A comparison between these functions and linear scaling was done with three examples:

- Integer Linear Programming (ILP)
- Best Path Finding (BPF)
- Best Path Finding with Obstacles (BPFWO)

The third example uses obstacles, arranged in an alley configuration, to act like deceiving elements in the scaling function.

II Background

The improvement of scaling functions is an important issue in GA optimisation. Goldberg considered three classes of scaling functions [4,9] :

- Linear Scaling
- Sigma Truncation
- Power Law Scaling

The second is an extension of Linear Scaling. Others extensions of Linear Scaling were also presented. John Grefenstette's GENESIS system used the *Windowing Algorithm* as default fitness technique [2,6]. Bramlette used dynamic scaling method, *Dynamic Range*, to assure a constant competitive pressure [1].
Hyperbolic scaling functions are also common. Songwu Lu and Tamer Basar achieved good results with hyperbolic scaling functions in a systems identification problem [8].
Maza and Tidor [10, 11] used the Boltzmann scaling which uses an exponential function. They defined some properties for the scaling functions: translation and scaling invariance. It was also shown that linear scaling is scale invariant but not translation invariant and, on the other hand, the Boltzmann scaling was translation invariant but not scale invariant.
Nuno Gracias *et. al* made a comparison between linear scaling, hyperbolic and exponential scaling functions for the Travelling Salesperson problem [5] and showed that latest achieved better results.

III Approach

This work presents a different method for scaling function design, which are deduced from a given set of conditions it must obey.

A. Framework

Let $\vec{X}$ be the population of a canonical GA with a fixed size of N elements. It is assumed that the scaling function can be expressed just by a non-negative cost parameter c. For each time step it is defined:

c_i	cost of i-th element
$f_i=f(c_i)$	fitness of i-th element
$p_i=p(c_i)$	probability of i-th element to be selected

For all population respective vectors can be defined in the set as $(\vec{X},\vec{C},\vec{F},\vec{P})$.

Let also be

$$c_{\min} = \min\{c_i : i = 0..N-1\} \quad (1)$$

$$\bar{c} = \frac{\sum_{i=0}^{N-1} c_i}{N} \quad (2)$$

$$<\vec{F}> = \sum_{i=0}^{N-1} f_i \quad (3)$$

$$p_i = p(f(c_i)) = \frac{f(c_i)}{<\vec{F}>} \quad (4)$$

Definition 1: [10] *A selection procedure is translation invariant exactly when:*

$$\vec{P}(\vec{F}(\vec{C}(\vec{X}))) = \vec{P}(\vec{F}(\vec{C}(\vec{X}) + \vec{C}_0)) \quad (5)$$

where $\vec{C}_0$ *is a vector of identical elements* c_o.

Definition 2: [10] *A selection procedure is scale invariant exactly when:*

$$\vec{P}(\vec{F}(\vec{C}(\vec{X}))) = \vec{P}(\vec{F}(k \cdot \vec{C}(\vec{X}))) \quad (6)$$

where k is a positive scalar.

Definition 3 : *A selection procedure has fixed selectivity if, for two elements of* $\vec{X}$ *with a fixed cost difference,* Δc, *verifies:*

$$\frac{p(f(c+\Delta c))}{p(f(c))} = k \quad (7)$$

for any non negatives values of c.

Observation 1 : *If a selection procedure has a fixed selectivity, then it is translation invariant if the roulette wheel is used.*

Proof:

Note that:

$$p(f(c+\Delta c)) = k \cdot p(f(c)) \Leftrightarrow f(c+\Delta c) = k \cdot f(c) \quad (8)$$

So,

$$\frac{\vec{F}(\vec{C}(\vec{X}) + \Delta\vec{C})}{<\vec{F}(\vec{C}(\vec{X}) + \Delta\vec{C})>} = \frac{k \cdot \vec{F}(\vec{C}(\vec{X}))}{<k \cdot \vec{F}(\vec{C}(\vec{X}))>} =$$

$$= \frac{\vec{F}(\vec{C}(\vec{X}))}{<\vec{F}(\vec{C}(\vec{X}))>}$$

B. Static Selectivity Scaling

For an element s_i of $\vec{X}$ with cost c , let Δc be a positive disturbance. It is assumed that Δc is small.

The desired fitness function must verify :

$$p(f(c-\Delta c)) = (\varepsilon+1)p(f(c)) \quad \text{and} \quad \varepsilon > 0 \quad (9)$$

$$\frac{df}{dc} < 0 \quad \forall c > 0 \quad (10)$$

$$f(c) > 0 \quad \forall c > 0 \quad (11)$$

The (9) expression assures translation invariance and fixed selectivity for a fixed Δc. On the other hand, (10) and (11) assure that f is monotonous and the validation of (4).

Both Δc and ε can be considered designing constants in the determination of a fitness function that always assures a probability gain (1+ε) when cost is decreased by a fixed value Δc. This is independent of c values. If Δc represents the c minimal step, than (1+ε) will be the minimal probability step on the roulette wheel.

This function must verify conditions (9), (10) and (11). The first step is to replace *p(f(c))* by (4) in (9):

$$\frac{f(c-\Delta c)}{F} = (\varepsilon+1)\frac{f(c)}{F} \quad (12)$$

$$f(c-\Delta c) = (\varepsilon+1) \cdot f(c) \quad (13)$$

By making a first order approximation it is obtained:

$$f(c-\Delta c) \cong f(c) - \frac{df}{dc}(c) \cdot \Delta c \quad (14)$$

Being Δc constant and applying expression (14) in (13) gives:

$$\frac{df}{dc}(c) = -\frac{\varepsilon}{\Delta c} f(c) \qquad (15)$$

Solving this homogeneous linear differential equation, *f* can be expressed as:

$$f(c) = K \cdot e^{-\frac{\varepsilon}{\Delta c}c} \qquad (16)$$

The *K* depends on boundary conditions.σ is defined as:

$$\sigma = -\frac{\varepsilon}{\Delta c} \qquad (17)$$

Parameter σ can be interpreted as a discernment measure, i.e., a selectivity parameter. The higher σ is, the higher the probability gain will be for the same Δc. Note that this gain only depends on Δc, and not on *c* like others scaling functions. This means that this function preserves its discernment characteristic for all *c* values.
Being $\varepsilon > 0$ and $\Delta c > 0$, for positive values of *K*, conditions (10) and (11) are always satisfied. *K* can be any positive number, but as it doesn't affect selection, it is chosen *K*=1.
The static selectivity scaling function can now be expressed as:

$$f(c) = e^{-\sigma \cdot c} \qquad (18)$$

C. Dynamic selectivity scaling function

Result (18) showed a scaling function with a constant selectivity independent of *c* values. This property can complies with condition (9). Nonetheless, in this function, the fitness of some relevant elements of $\vec{X}$ cannot be predicted.
In this section is presented a function with a fixed relationship, between best and average cost of $\vec{X}$. This can be achieved by adding two more conditions:

$$f(c_{\min}) = 1 \qquad (19)$$
$$f(\bar{c}) = \phi \qquad \phi < 1 \qquad (20)$$

The fitness function must verify (9), (10), (11), (19) and (20).
Applying (19) and (20) to expression (16):

$$K \cdot e^{-\sigma \cdot c_{\min}} = 1 \qquad (21)$$
$$K \cdot e^{-\sigma \cdot \bar{c}} = \phi \qquad (22)$$

K can be obtained directly from (21):

$$K = e^{-\sigma \cdot c_{\min}} \qquad (23)$$

ϕ can be obtained by applying (23) in (22):

$$\phi = e^{-\sigma \cdot (\bar{c} - c_{\min})} \qquad (24)$$

After some algebraic manipulation σ can be expressed as:

$$\sigma = -\frac{\ln(\phi)}{\bar{c} - c_{\min}} \qquad (25)$$

The dynamic selectivity scaling function is defined by:

$$f(c) = e^{\frac{\ln(\phi)}{\bar{c} - c_{min}}(c - c_{min})} \qquad (26)$$

This function has the same properties as (18), assuring a fixed relation between best cost and the average cost.

Observation 2: *The dynamic selectivity scaling function is both scale invariance and translation invariance.*
Proof:
Let $\vec{C}'$ be a transformed cost by the equation

$$\vec{C}' = \alpha \cdot \vec{C} + \vec{C}_0 \qquad (27)$$

The dynamic parameters will then be

$$c'_{\min} = \alpha \cdot c_{\min} + c_0$$
$$\bar{c}' = \alpha \cdot \bar{c} + c_0$$

Thus,

$$f(c') = e^{\ln(\phi)\frac{c' - c'_{\min}}{\bar{c}' - c'_{\min}}} = e^{\ln(\phi)\frac{\alpha \cdot c + c_0 - \alpha \cdot c_{\min} - c_0}{\alpha \cdot \bar{c} + c_0 - \alpha \cdot c_{\min} - c_0}}$$
$$= e^{\ln(\phi)\frac{c - c_{\min}}{\bar{c} - c_{\min}}} = f(c)$$

D. Generalised selectivity scaling functions

Functions (18) and (26) depended only on single cost parameter. In this section they are generalised for vector of *M* cost parameters.
Let C be a $[M \times 1]$ cost vector of one element of $\vec{X}$:

$$C = \begin{bmatrix} c_0 \\ \cdot \\ \cdot \\ c_{M-1} \end{bmatrix} \qquad (28)$$

and $f(C)$ a scalar function. It is wanted that, for any c_i, the following conditions are verified:

$$\frac{\partial f}{\partial c_i} < 0 \qquad \forall i = 0..M-1 \qquad (29)$$

$$p(f(c_0,\ldots,c_i - \Delta c_i,\ldots,c_{M-1})) = (1+\varepsilon_i)\cdot p(f(c_0,\ldots,c_i,\ldots,c_{M-1})), \ \ \varepsilon_i > 0, \ \ \forall i = 0..M-1 \qquad (30)$$

It is assumed that $f(C)$ can be decomposed in:

$$f(C) = \prod_{i=0}^{M-1} f_i(c_i) \qquad (31)$$

$$\frac{\partial f_i}{\partial c_j} = 0 \quad \text{if } i \neq j \qquad (32)$$

The function can be obtained by taking steps (14) to (17). The selectivity factors, σ_j , can be arranged in a selectivity vector:

$$\Sigma = \begin{bmatrix} \sigma_0 \\ . \\ . \\ \sigma_{M-1} \end{bmatrix}, \qquad \Sigma \in \Re^M \qquad (33)$$

The generalised static selectivity scaling function can now be expressed as:

$$f(C) = e^{-\Sigma^T \cdot C} \qquad (34)$$

Defining C_{min} as a vector, the i-th element is the minimal i-th cost of all the members of population $\vec{X}$. Let the dynamic selectivity vector, Ψ, be defined as:

$$\Psi = \begin{bmatrix} \frac{\ln(\phi_0)}{\bar{c}_0 - c_{0_{\min}}} \\ . \\ . \\ \frac{\ln(\phi_{M-1})}{\bar{c}_{M-1} - c_{M-1_{\min}}} \end{bmatrix} \qquad (35)$$

The generalised dynamic selectivity scaling function can now be expressed as :

$$f(C) = e^{\Psi^T \cdot (C - C_{\min})} \qquad (36)$$

Note that (36) does not assure that the best element of $\vec{X}$ has unitary fitness. This only happens when $C_{bestof[\vec{X}]} = C_{\min}$, *i.e.*, the best element has the minimal cost in every value of the cost vector, which, generally, may not be true. Nonetheless (36) still verifies (29), (31) and (32). And even if condition (19) is only verified for a hypothetical-element, the dynamic characteristics of (36) are still valid.

IV. Experimental Method

The operators used in these examples were:

- Crossover
- Mutation
- Proportional Reproduction

All simulations used populations of 40 strings. For each case, and to have some statistical results, 500 runs of the GA were made.

A. The Integer Linear Programming Problem

This combinatorial problem may be stated mathematically as:

$$\min \sum_{j=0}^{N-1} a_j \cdot x_j \qquad (37)$$

subject to:

$$\sum_{j=0}^{N-1} b_j \cdot x_j \geq B, \ \ x_j \text{ is an integer.} \qquad (38)$$

In this example x was restricted between 0 and 9. And N was set to 10, resulting in a10^{10} dimention search space.

B. The Best Path Finding Problem

In this problem a two dimensional grid is used. The grid has a starting point and a goal point. It can also have obstacles. Each string is defined by 20 instruction sequence. The available instructions are: Up, Down, Left, Right and Nop[1]. The distance to goal is the cost used to evaluate strings.

C. The Best Path Finding Problem With Obstacles

A more difficult problem can be created by adding obstacles in an alley shape to the previous one. This can be seen in figure 1. The cost is increased by a penalty for each obstacle crossed.

[1] Nop stands for No Operation.

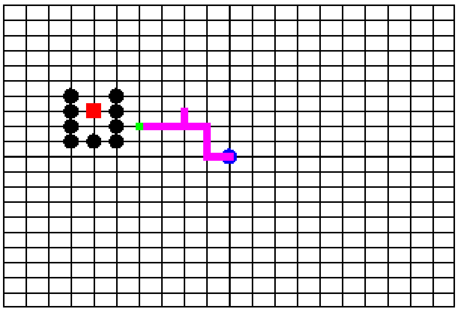

Figure 1 - The deceiving problem. Goal is inside an alley

(Goal is the square, the starting point is on centre.)

V. Results

For each example, 500 runs of the GA with a population of 40 strings and for 100 generations were made for the three scaling functions.

The tuning of the scaling functions parameters were done empirically for each example[2].

In the ILP example a mutation rate of 2% and a crossover rate of 80% was used. The fitness functions were:

Linear	$100-5\cdot c$
Static Selectivity	$e^{-0.1\cdot c}$
Dynamic Selectivity	$e^{\ln(0.1)\frac{c-c_{\min}}{\bar{c}-c_{\min}}}$

Table 1 - Fitness functions for ILP problem

The cost value, *c*, is given by

$$c=\sum_{j=0}^{N-1} a_j \cdot x_j \quad \text{if} \sum_{j=0}^{N-1} b_j \cdot x_j \geq B,$$

otherwise $c=\sum_{j=0}^{N-1} 10\cdot a_j$ (maximum cost)

Figure 2 shows the **average** of all best results for the 500 runs of the GA

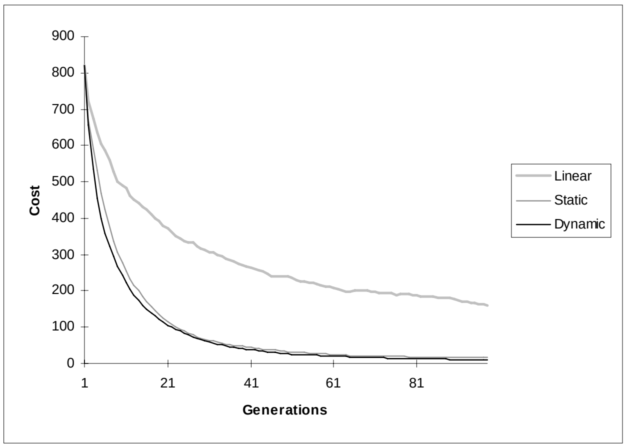

Figure 2 - ILP (Best of population)

A histogram of the average population in the 100th generation for the IPL problem can be seen in figure 3.

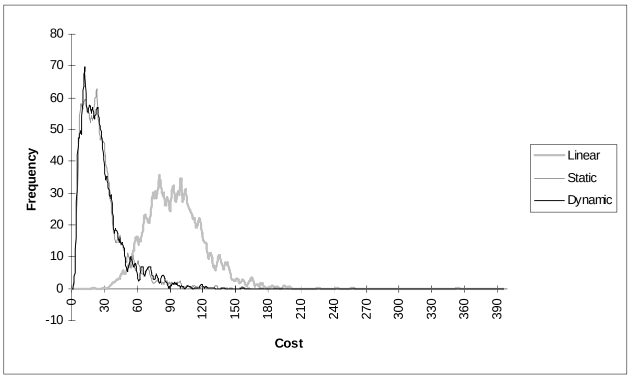

Figure 3 - IPL: Histogram of average population after 100 generations

In the BPF problems the mutations rates were 1% for the BPF and 10% for the BPF with obstacles. Both had a crossover rate of 80%.The fitness functions used were:

Linear	$20-4\cdot c$
Static Selectivity	$e^{-3\cdot c}$
Dynamic Selectivity	$e^{\ln(0.005)\frac{c-c_{\min}}{\bar{c}-c_{\min}}}$

Table 2

The cost, *c*, is just the distance to goal, increased by 2 for each obstacle bumped (in the BPFWO case). Figures 4 and 5 show the average of best of population for these two problems. Note the deceiveness in this last example.

[2] After several tests, the ones that achieved better results were chosen.

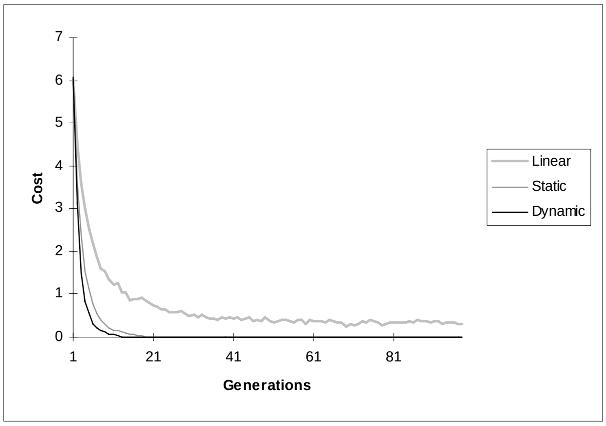

Figure 4 - BPF (Best of population)

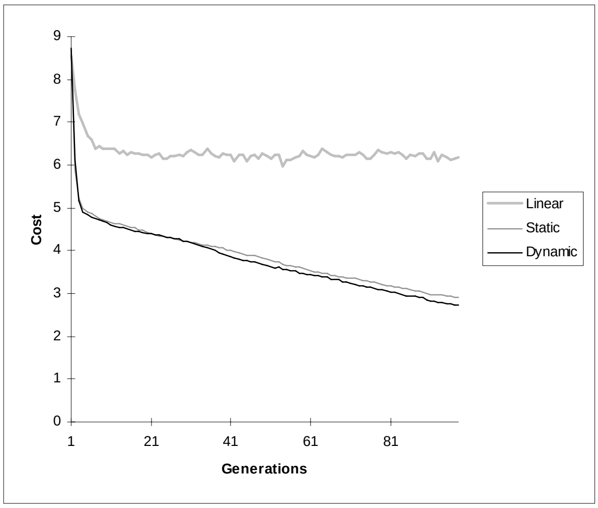

Figure 5 - BPFWO (Best of Population)

VI. Conclusions

In this paper a class of scaling functions were presented. Both are translation invariant, being the dynamic also scaling invariant.

On the tests presented, exponential based scaling functions achieved better results than linear scaling function. Convergence was always faster and, for the same generation, exponential scaling presented better solutions than linear scaling.

Dynamic scaling achieved slightly better results than the static scaling. On the other hand, the dynamic scaling requires more computation time. This suggests that for large populations the static scaling may be more suitable.

The results of the three examples suggest that exponential scaling functions have good potentialities and should be tested on other kind of problems.

References

[1] Bramlette, M. (1991). *Initialisation, Mutation and Selection Methods in Genetic Algorithms for Function Optimisation.* Proc. of the 4th International Conference on Genetic Algorithms, University of California, San Diego.

[2] Davis, L. (1991) (Ed.). *Handbook of Genetic Algorithms.* Van Nostrand Reinhold , New York.

[3] De Jong, K (1975). *Analysis of the Behaviour of a Class of Genetic Adaptive Systems.* Doctoral dissertation, Department Computer and Communication Sciences, University of Michigan, Ann Arbor.

[4] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimisation and Machine Learning.* Reading, Mass.: Addison-Wesley.

[5] Gracias, N., J. Lima, H. Pereira (1995) *Solving the Travelling Salesman Problem with Genetic Algorithms.* Proc. on the 1st Workshop on Biomedical Engineering and Genetic Algorithms, A. Rosa (Ed.), Instituto Superior Técnico,Lisbon.

[6] Grefenstette, J., J. Baker (1989). *How Genetic Algorithms Work: A critical Look at Implicit Parallelism.* Proc. of the 3rd International Conference on Genetic Algorithms. George Mason University, California.

[7] Holland, J. (1975). *Adaptation in Natural and Artificial System.* Ann Harbour: University of Michigan Press.

[8] Lu, S., T. Basar (1995). *Genetic Algorithms-Based Identification.* IEEE Trans. on Systems, Man and Cybernetics.

[9] Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag.

[10] Maza, M., B. Tidor (1993). *An Analysis of Selection Procedures with Particular Attention to Proportional Selection and Boltzmann Selection.* Proc. of the 5th International Conference on Genetic Algorithms. S. Forrest (Ed) University of Illinois, Urbana-Champaign

[11] Maza, M., B. Tidor (1992). *Increased Flexibility in Genetic Algorithms: The use of Boltzmann Selective Pressure to Control Propagation.* In Proc. of the ORCA CSTS Conference: *Computer Science and Operational Research: New Developments in Their Interfaces*, pages 10-19.