

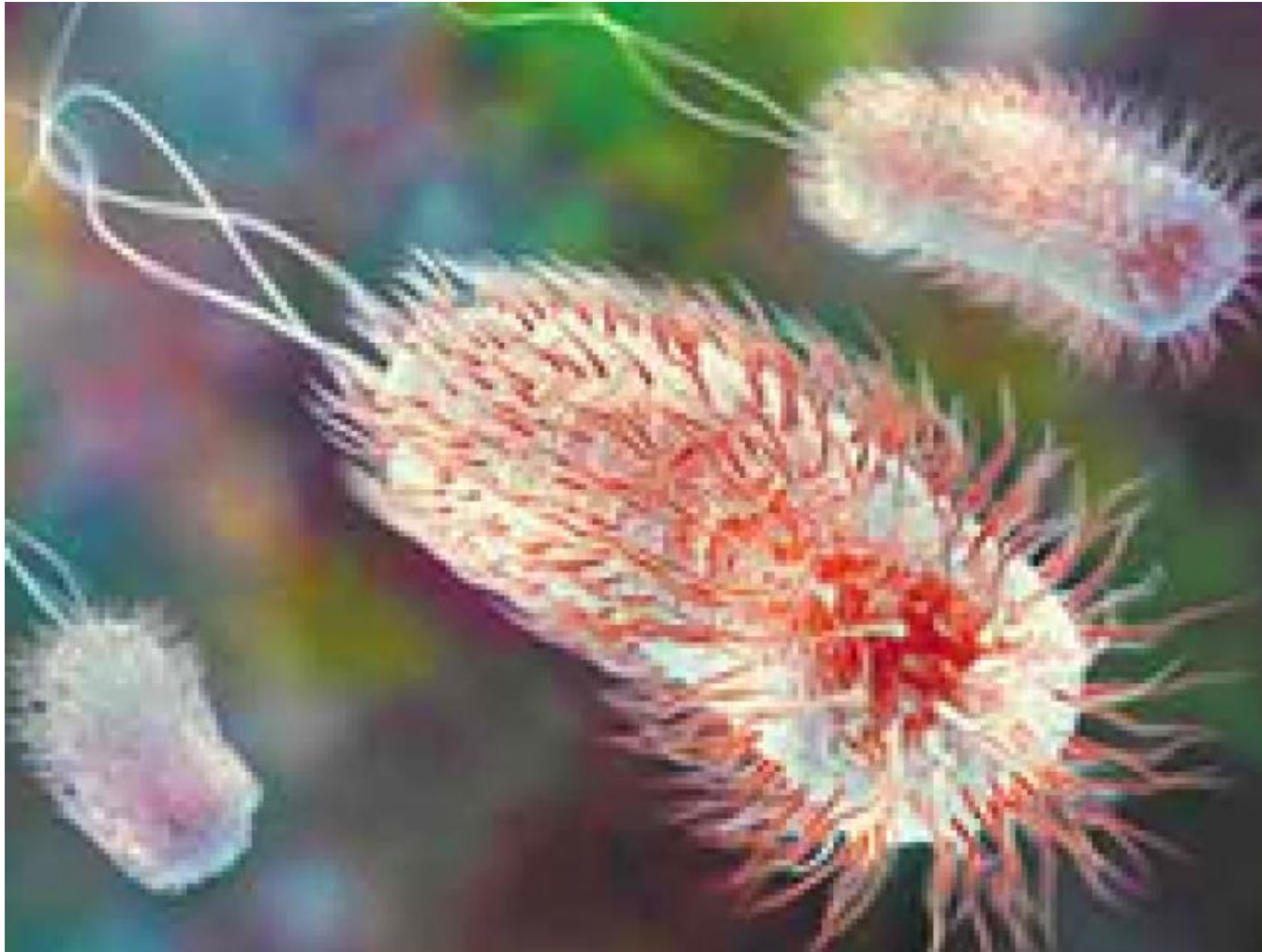
Computación Bioinspirada

Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe

Bacterial Foraging Optimization Algorithm

- Es una técnica de optimización bioinspirada para encontrar la solución óptima de un problema.
- En el año 2002, el profesor K. M. Passino propuso una técnica de optimización conocida como Bacterial Foraging Optimization Algorithm (BFO).
- El cual es basado en el comportamiento social de búsqueda de la bacteria E. Coli que se encuentra en el intestino humano.

Bacterial Foraging Optimization Algorithm



Bacterial Foraging Optimization Algorithm

- Bacterias buscan los nutrientes intentando maximizar la energía obtenida por unidad de tiempo.
- La maximización de tal función proporciona fuentes de nutrientes para sobrevivir y tiempo adicional para otras actividades importantes.

Bacterial Foraging Optimization Algorithm

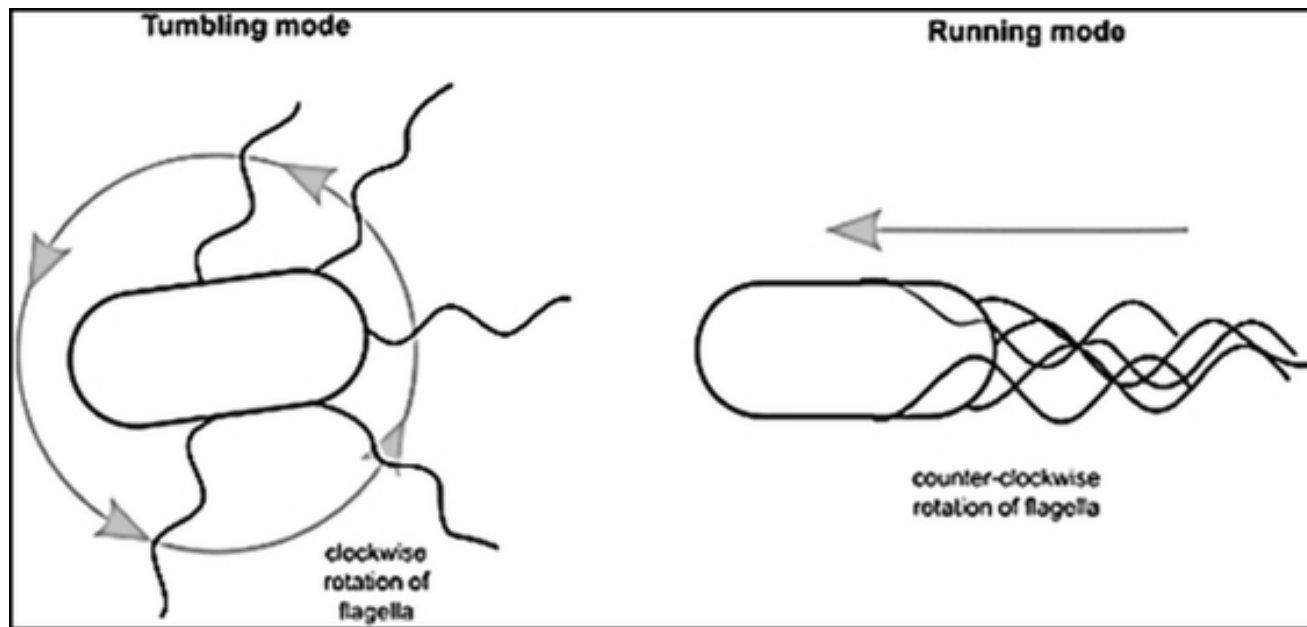
- Las bacterias perciben la dirección hacia los alimentos basándose en los gradientes de las sustancias químicas en su entorno.
- De manera similar, las bacterias secretan sustancias químicas que atraen y repelen al medio ambiente y pueden percibirse entre sí de manera similar.

Bacterial Foraging Optimization Algorithm

- Las bacterias perciben la dirección hacia los alimentos basándose en los gradientes de las sustancias químicas en su entorno.
- De manera similar, las bacterias secretan sustancias químicas que atraen y repelen al medio ambiente y pueden percibirse entre sí de manera similar.

Bacterial Foraging Optimization Algorithm

- Usando mecanismos de locomoción (como los flagelos), las bacterias pueden moverse en su entorno, a veces de forma caótica (dando vueltas y vueltas - Trumble), y otras veces de forma dirigida, lo que se conoce como nadar (Swim).



Bacterial Foraging Optimization Algorithm

- Las células bacterianas son tratadas como agentes en un ambiente, usando su percepción de la comida y otras células como motivación para moverse, y el movimiento estocástico como movimiento para reubicarse.
- Dependiendo de las interacciones célula-célula, las células pueden encontrar una fuente de alimento, y/o pueden repeler agresivamente o ignorarse mutuamente.

Bacterial Foraging Optimization Algorithm

- La estrategia de procesamiento de información del algoritmo es permitir que las células se agrupen estocástica y colectivamente hacia la solución óptima.
- Esto se logra a través de tres procesos en una población de células:

Bacterial Foraging Optimization Algorithm

- 1) “Quimiotaxis” donde el costo de las células es reducido por la proximidad a otras células y las células se mueven a lo largo de la superficie del costo manipulado de una en una.
- 2) “Reproducción” donde sólo aquellas células que se desempeñaron bien a lo largo de su vida pueden contribuir a la siguiente generación,
- 3) “Dispersión de la eliminación” donde las células son descartadas y se insertan nuevas células aleatorias con una baja probabilidad.

Bacterial Foraging Optimization Algorithm

```
Input:  $Problem_{size}, Cells_{num}, N_{ed}, N_{re}, N_c, N_s, Step_{size}, d_{attract}, w_{attract}, h_{repellant}, w_{repellant}, P_{ed}$   
Output:  $Cell_{best}$   
Population  $\leftarrow$  InitializePopulation( $Cells_{num}, Problem_{size}$ )  
For ( $l = 0$  To  $N_{ed}$ )  
  For ( $k = 0$  To  $N_{re}$ )  
    For ( $j = 0$  To  $N_c$ )  
      ChemotaxisAndSwim(Population,  $Problem_{size}, Cells_{num}, N_s, Step_{size}, d_{attract}, w_{attract}, h_{repellant}, w_{repellant}$ )  
      For ( $Cell \in$  Population)  
        If ( $Cost(Cell) \leq Cost(Cell_{best})$ )  
           $Cell_{best} \leftarrow Cell$   
        End  
      End  
    End  
  End  
  SortByCellHealth(Population)  
  Selected  $\leftarrow$  SelectByCellHealth(Population,  $\frac{Cells_{num}}{2}$ )  
  Population  $\leftarrow$  Selected  
  Population  $\leftarrow$  Selected  
End  
For ( $Cell \in$  Population)  
  If ( $Rand() \leq P_{ed}$ )  
     $Cell \leftarrow$  CreateCellAtRandomLocation()  
  End  
End  
End  
Return ( $Cell_{best}$ )
```

Bacterial Foraging Optimization Algorithm

```
Input: Population,  $Problem_{size}$ ,  $Cells_{num}$ ,  $N_s$ ,  $Step_{size}$ ,  $d_{attract}$ ,  $w_{attract}$ ,  $h_{repellant}$ ,  $w_{repellant}$   
For (Cell  $\in$  Population)  
     $Cell_{fitness} \leftarrow Cost(Cell) + Interaction(Cell, Population, d_{attract}, w_{attract}, h_{repellant}, w_{repellant})$   
     $Cell_{health} \leftarrow Cell_{fitness}$   
     $Cell' \leftarrow \emptyset$   
    For ( $i = 0$  To  $N_s$ )  
         $RandomStepDirection \leftarrow CreateStep(Problem_{size})$   
         $Cell' \leftarrow TakeStep(RandomStepDirection, Step_{size})$   
         $Cell'_{fitness} \leftarrow Cost(Cell') + Interaction(Cell', Population, d_{attract}, w_{attract}, h_{repellant}, w_{repellant})$   
        If ( $Cell'_{fitness} > Cell_{fitness}$ )  
             $i \leftarrow N_s$   
        Else  
             $cell \leftarrow Cell'$   
             $Cell_{health} \leftarrow Cell_{health} + Cell'_{fitness}$   
    End  
End  
End
```

Pseudocode for the ChemotaxisAndSwim function.

$$g(cell_k) = \sum_{i=1}^S \left[-d_{attr} \times \exp \left(-w_{attr} \times \sum_{m=1}^P (cell_m^k - other_m^i)^2 \right) \right] + \sum_{i=1}^S \left[h_{repel} \times \exp \left(-w_{repel} \times \sum_{m=1}^P (cell_m^k - other_m^i)^2 \right) \right]$$

Bacterial Foraging Optimization Algorithm

- Es común tener un gran número de iteraciones de quimiotaxis, y un pequeño número de las otras iteraciones.
- Los coeficientes de atracción y repulsión por defecto son:

$$\begin{aligned}d_{attract} &= 0.1, \\w_{attract} &= 0.2, \\h_{repellant} &= d_{attract}, \\w_{repellant} &= 10.\end{aligned}$$

$$g(cell_k) = \sum_{i=1}^S \left[-d_{attr} \times \exp \left(-w_{attr} \times \sum_{m=1}^P (cell_m^k - other_m^i)^2 \right) \right] + \sum_{i=1}^S \left[h_{repel} \times \exp \left(-w_{repel} \times \sum_{m=1}^P (cell_m^k - other_m^i)^2 \right) \right]$$

BFO: Ejemplo

Problema			
Minimizar	$f(x_1, x_2) = x_1^2 + x_2^2$	LimInf _i	LimSup _i
	$-5 \leq x_1 \leq 5$	-5	5
	$-5 \leq x_2 \leq 5$	-5	5
	$x_1, x_2 \in \mathbb{R}$		

BFO: Ejemplo

Tamaño de la Población (Cells_num): 4
Tamaño de Variación (step_size): 0.1
Número de Etapas de Eliminación y Dispersión (Ned): 2
Número de Etapas de Reproducción (Nre): 2
Número de Etapas de Quimiotaxis (Nc): 20
Número de Etapas de Nado (Ns): 5
Probabilidad de Eliminación y Dispersión (Ped): 0.25
d_attract: 0.1
w_attract: 0.2
h_repellant: 0.1
w_repellant: 10

BFO: Ejemplo

Población Inicial

```
{:vector=>[1.9529305935698744, -2.785701225919346]}  
{:vector=>[1.009050335503109, 3.238440968195672]}  
{:vector=>[-0.1488286369963676, -2.2541844938691966]}  
{:vector=>[2.2841602199438595, 3.0119361164171004]}
```


BFO: Ejemplo

```

Ned = 0
Nre = 0
{:vector=>[1.9529305935698744, -2.785701225919346], :cost=>11.574069223389728, :inter=>-0.03924023787365902, :fitness=>11.534828985516068}
Swim 0
{:vector=>[2.0361976216012017, -2.797838389548896], :cost=>11.97400040824795, :inter=>-0.04314253469348518, :fitness=>11.930857873554464}
--End Swim
{:vector=>[1.009050335503109, 3.238440968195672], :cost=>11.505682484067057, :inter=>-0.0717444246094891, :fitness=>11.433938059457567}
Swim 0
{:vector=>[1.0007601739992293, 3.1799036429021745], :cost=>11.113308104005489, :inter=>-0.0751708888164508, :fitness=>11.038137215189037}
Swim 1
{:vector=>[0.9061297748718227, 3.148347494771177], :cost=>10.733163116741206, :inter=>-0.08513978197825645, :fitness=>10.64802333476295}
Swim 2
{:vector=>[0.8344669533084264, 3.089076127314028], :cost=>10.238726416505282, :inter=>-0.10592173339062814, :fitness=>10.132804683114653}
Swim 3
{:vector=>[0.8888902197683856, 2.999915021370004], :cost=>9.789615958241281, :inter=>-0.11775118229252249, :fitness=>9.671864775948759}
Swim 4
{:vector=>[0.9081214109538046, 2.9412922325914517], :cost=>9.475884494535535, :inter=>-0.12958230789498465, :fitness=>9.34630218664055}
--End Swim
{:vector=>[-0.1488286369963676, -2.2541844938691966], :cost=>5.103497695590523, :inter=>-0.03936624593158314, :fitness=>5.06413144965894}
Swim 0
{:vector=>[-0.09327990776662326, -2.2645037351172443], :cost=>5.136678307552901, :inter=>-0.044379702503157745, :fitness=>5.092298605049743}
--End Swim
{:vector=>[2.2841602199438595, 3.0119361164171004], :cost=>14.289147079751706, :inter=>-0.0717393538118826, :fitness=>14.217407725939823}
Swim 0
{:vector=>[2.3446150107976553, 3.0042690022090857], :cost=>14.522851786492064, :inter=>-0.07303588604338129, :fitness=>14.449815900448682}
--End Swim
>> chemo=0, f=5.06413144965894, cost=5.103497695590523

```

$$\begin{aligned}
 &Step_{size} \quad g(cell_k) = \sum_{i=1}^S \left[-d_{attr} \times \exp \left(-w_{attr} \times \sum_{m=1}^P (cell_m^k - other_m^i)^2 \right) \right] + \\
 &\in [-1, 1] \quad \sum_{i=1}^S \left[h_{repel} \times \exp \left(-w_{repel} \times \sum_{m=1}^P (cell_m^k - other_m^i)^2 \right) \right]
 \end{aligned}$$

BFO: Ejemplo

```
{:vector=>[1.9529305935698744, -2.785701225919346], :cost=>11.574069223389728, :inter=>-0.03924023787365902, :fitness=>11.534828985516068}
Swim 0
{:vector=>[2.0361976216012017, -2.797838389548896], :cost=>11.97400040824795, :inter=>-0.04314253469348518, :fitness=>11.930857873554464}
```

$$g(cell_k) = \sum_{i=1}^S \left[-d_{attr} \times \exp \left(-w_{attr} \times \sum_{m=1}^P (cell_m^k - other_m^i)^2 \right) \right] + \sum_{i=1}^S \left[h_{repel} \times \exp \left(-w_{repel} \times \sum_{m=1}^P (cell_m^k - other_m^i)^2 \right) \right]$$

x1	1.95293059356987	1.00905033550311	-0.148828636996368	2.28416021994386	
x2	-2.78570122591935	3.23844096819567	-2.2541844938692	3.0119361164171	
d_attract	0.1				
w_attract	0.2				
h_repellant	0.1				
w_repellant	10				
cost	11.5740692233897				
	0	0.890909941568184	4.4173918632704	0.10971306538785	
	0	36.2902891749169	0.28251003644927	33.612598753254	
	0	37.1811991164851	4.69990189971967	33.7223118186419	
attract	-0.1	-5.8949765066E-05	-0.039063549957374	-0.000117738151219	-0.139240237873659
repellant	0.1	3.342750263E-163	3.877799895933E-22	3.514476158E-148	0.1
inter	-0.039240237873659				
fitness	11.5348289855161				

BFO: Ejemplo

```
{:vector=>[1.9529305935698744, -2.785701225919346], :cost=>11.574069223389728, :inter=>-0.039295141252271265, :fitness=>11.534774082137456, :sum_nutrients=>11.534828985516068}
Swim 0
{:vector=>[1.9812834510490889, -2.7463875099654373], :cost=>11.468128468295141, :inter=>-0.04097305917712529, :fitness=>11.427155409118017}
Swim 1
{:vector=>[1.974596675378077, -2.687666210826082], :cost=>11.122581691230383, :inter=>-0.0487638082771696, :fitness=>11.073817882953215}
Swim 2
{:vector=>[1.9819280512275608, -2.7313379909660735], :cost=>11.388246021137263, :inter=>-0.04244710912718096, :fitness=>11.345798912010082}
--End Swim
{:vector=>[0.9081214109538046, 2.9412922325914517], :cost=>9.475884494535535, :inter=>-0.06888260795182369, :fitness=>9.40700188658371, :sum_nutrients=>62.27107025511352}
Swim 0
{:vector=>[1.0043086195191269, 3.0087055097209094], :cost=>10.060944647465371, :inter=>-0.0850785947916026, :fitness=>9.97586605267377}
--End Swim
{:vector=>[-0.1488286369963676, -2.2541844938691966], :cost=>5.103497695590523, :inter=>-0.039544699446463494, :fitness=>5.063952996144059, :sum_nutrients=>5.06413144965894}
Swim 0
{:vector=>[-0.12780290824995003, -2.1773407628481993], :cost=>4.757146380917524, :inter=>-0.04565994910311662, :fitness=>4.711486431814407}
Swim 1
{:vector=>[-0.1947537835477802, -2.237783206086397], :cost=>5.04560271364849, :inter=>-0.040223009347163385, :fitness=>5.005379704301327}
--End Swim
{:vector=>[2.2841602199438595, 3.0119361164171004], :cost=>14.289147079751706, :inter=>-0.06864418026072458, :fitness=>14.220502899490981, :sum_nutrients=>14.217407725939823}
Swim 0
{:vector=>[2.346871169193362, 3.0707011206644794], :cost=>14.93700965724111, :inter=>-0.07305212867033044, :fitness=>14.86395752857078}
--End Swim
>> chemo=1, f=4.711486431814407, cost=4.757146380917524
```

BFO: Ejemplo

```
{:vector=>[1.622064829299831, -1.8347746387070163], :cost=>5.997492285293951, :inter=>-0.05547778143671819, :fitness=>5.942014503857234, :sum_nutrients=>5.94315428870008}
Swim 0
{:vector=>[1.711384400639848, -1.8095118749566328], :cost=>6.2031697923624805, :inter=>-0.060967076476180215, :fitness=>6.142202715886301}
--End Swim
{:vector=>[0.814648264624062, 1.9757898862036454], :cost=>4.56739746947961, :inter=>-0.08302688349879823, :fitness=>4.484370585980812, :sum_nutrients=>14.604118454588239}
Swim 0
{:vector=>[0.8258965447120441, 1.87596380642107], :cost=>4.201345305569123, :inter=>-0.09432683058482157, :fitness=>4.107018474984301}
Swim 1
{:vector=>[0.7562551653172678, 1.8435228038662093], :cost=>3.970498203443778, :inter=>-0.10145455566361493, :fitness=>3.869043647780163}
Swim 2
{:vector=>[0.7141613736437474, 1.921215367978914], :cost=>4.201094957763078, :inter=>-0.09231727545622233, :fitness=>4.108777682306855}
--End Swim
{:vector=>[-0.2656036801963266, -1.3185713775826395], :cost=>1.8091757927140122, :inter=>-0.0584378809024802, :fitness=>1.750737911811532, :sum_nutrients=>5.470841932499995}
Swim 0
{:vector=>[-0.16868145483389874, -1.3738129737281306], :cost=>1.91581551998861, :inter=>-0.07361926107866204, :fitness=>1.8421962589099479}
--End Swim
{:vector=>[2.1650320786197503, 2.112521174273263], :cost=>9.150109613205442, :inter=>-0.0762721756989534, :fitness=>9.073837437506489, :sum_nutrients=>27.97690521996529}
Swim 0
{:vector=>[2.2234143060774434, 2.1603193930702878], :cost=>9.610551056545415, :inter=>-0.0785984817870298, :fitness=>9.531952574758385}
--End Swim
>> chemo=19, f=1.750737911811532, cost=1.8091757927140122
```

BFO: Ejemplo

```
>> chemo=19, f=1.750737911811532, cost=1.8091757927140122
> best fitness=1.750737911811532, cost=1.8091757927140122
Células Antes de la Reproducción
{:vector=>[1.622064829299831, -1.8347746387070163], :cost=>5.997492285293951, :inter=>-0.05547778143671819, :fitness=>5.942014503857234, :sum_nutrients=>5.942014503857234}
{:vector=>[0.7562551653172678, 1.8435228038662093], :cost=>3.970498203443778, :inter=>-0.10145455566361493, :fitness=>3.869043647780163, :sum_nutrients=>12.460432708745277}
{:vector=>[-0.2656036801963266, -1.3185713775826395], :cost=>1.8091757927140122, :inter=>-0.0584378809024802, :fitness=>1.750737911811532, :sum_nutrients=>1.750737911811532}
{:vector=>[2.1650320786197503, 2.112521174273263], :cost=>9.150109613205442, :inter=>-0.0762721756989534, :fitness=>9.073837437506489, :sum_nutrients=>9.073837437506489}
Células Después de la Reproducción
{:vector=>[-0.2656036801963266, -1.3185713775826395], :cost=>1.8091757927140122, :inter=>-0.0584378809024802, :fitness=>1.750737911811532, :sum_nutrients=>1.750737911811532}
{:vector=>[1.622064829299831, -1.8347746387070163], :cost=>5.997492285293951, :inter=>-0.05547778143671819, :fitness=>5.942014503857234, :sum_nutrients=>5.942014503857234}
{:vector=>[-0.2656036801963266, -1.3185713775826395], :cost=>1.8091757927140122, :inter=>-0.0584378809024802, :fitness=>1.750737911811532, :sum_nutrients=>1.750737911811532}
{:vector=>[1.622064829299831, -1.8347746387070163], :cost=>5.997492285293951, :inter=>-0.05547778143671819, :fitness=>5.942014503857234, :sum_nutrients=>5.942014503857234}
--End Nre
Nre = 1
```


BFO: Ejemplo

```
Nre = 1
{:vector=>[-0.2656036801963266, -1.3185713775826395], :cost=>1.8091757927140122, :inter=>-0.09297826208780785, :fitness=>1.7161975306262043, :sum_nutrients=>1.750737911811532}
Swim 0
{:vector=>[-0.3423589993002233, -1.3238278598815871], :cost=>1.8697298870005132, :inter=>-0.09899206867556778, :fitness=>1.7707378183249454}
--End Swim
{:vector=>[1.622064829299831, -1.8347746387070163], :cost=>5.997492285293951, :inter=>-0.09297826208780785, :fitness=>5.9045140232061435, :sum_nutrients=>5.942014503857234}
Swim 0
{:vector=>[1.542453254824351, -1.9327003813462527], :cost=>6.114492807374185, :inter=>-0.1252660803650742, :fitness=>5.98922672700911}
--End Swim
{:vector=>[-0.2656036801963266, -1.3185713775826395], :cost=>1.8091757927140122, :inter=>-0.09297826208780785, :fitness=>1.7161975306262043, :sum_nutrients=>1.7161975306262043}
Swim 0
{:vector=>[-0.2644579978060946, -1.4086323136712975], :cost=>2.0541830277225612, :inter=>-0.10991142747354196, :fitness=>1.9442716002490192}
--End Swim
{:vector=>[1.622064829299831, -1.8347746387070163], :cost=>5.997492285293951, :inter=>-0.09297826208780785, :fitness=>5.9045140232061435, :sum_nutrients=>5.9045140232061435}
Swim 0
{:vector=>[1.5460664937194584, -1.8122870887792681], :cost=>5.674706095158015, :inter=>-0.11072919198531678, :fitness=>5.563976903172699}
Swim 1
{:vector=>[1.4856623471927715, -1.7669994656554584], :cost=>5.32947972149301, :inter=>-0.14451708350258352, :fitness=>5.184962637990427}
Swim 2
{:vector=>[1.584331837373191, -1.7609365621006186], :cost=>5.611004946657057, :inter=>-0.11001633721182238, :fitness=>5.500988609445234}
--End Swim
>> chemo=0, f=1.7161975306262043, cost=1.8091757927140122
{:vector=>[-0.2656036801963266, -1.3185713775826395], :cost=>1.8091757927140122, :inter=>-0.0985058973833087, :fitness=>1.7106698953307036, :sum_nutrients=>1.7161975306262043}
Swim 0
{:vector=>[-0.19860996475164144, -1.3635484110668417], :cost=>1.8987101874215568, :inter=>-0.11652645853729712, :fitness=>1.7821837288842597}
--End Swim
{:vector=>[1.622064829299831, -1.8347746387070163], :cost=>5.997492285293951, :inter=>-0.11322003749969484, :fitness=>5.884272247794256, :sum_nutrients=>5.9045140232061435}
Swim 0
{:vector=>[1.6149427962015446, -1.9053753122226529], :cost=>6.238495315430836, :inter=>-0.12626478746123782, :fitness=>6.112230527969598}
--End Swim
{:vector=>[-0.2656036801963266, -1.3185713775826395], :cost=>1.8091757927140122, :inter=>-0.0985058973833087, :fitness=>1.7106698953307036, :sum_nutrients=>1.7106698953307036}
Swim 0
{:vector=>[-0.17573497095832508, -1.228326204706816], :cost=>1.5396680451871743, :inter=>-0.1323061649604391, :fitness=>1.4073618802267351}
Swim 1
{:vector=>[-0.2690921283835515, -1.2831027811620836], :cost=>1.7187633205838637, :inter=>-0.10003796509843679, :fitness=>1.6187253554854268}
--End Swim
{:vector=>[1.4856623471927715, -1.7669994656554584], :cost=>5.32947972149301, :inter=>-0.12427530809069653, :fitness=>5.205204413402313, :sum_nutrients=>16.65345356436927}
Swim 0
{:vector=>[1.583162780306171, -1.6986080794308536], :cost=>5.391673796454539, :inter=>-0.12879784642053765, :fitness=>5.262875950034001}
--End Swim
>> chemo=1, f=1.4073618802267351, cost=1.5396680451871743
```

BFO: Ejemplo

```
--End Swim
>> chemo=19, f=-0.14072909363381203, cost=0.09787174085710094
> best fitness=-0.14072909363381203, cost=0.09787174085710094
Células Antes de la Reproducción
{:vector=>[-0.1839050899582404, -0.25324699322466676], :cost=>0.0979551216898829, :inter=>-0.24239637789192342, :fitness=>-0.1444412562020405, :sum_nutrients=>-0.3595125679010528}
{:vector=>[0.6340806152644392, -0.5398300397910464], :cost=>0.6934746985149325, :inter=>-0.30641061490277627, :fitness=>0.3870640836121562, :sum_nutrients=>3.759617852784816}
{:vector=>[-0.25285906718724277, -0.8715987037932177], :cost=>0.8236220083128198, :inter=>-0.23063113805144492, :fitness=>0.5929908702613749, :sum_nutrients=>0.5929908702613749}
{:vector=>[1.1552247907475715, -1.201679370101219], :cost=>2.7785776256846324, :inter=>-0.21897740371212365, :fitness=>2.559600221972509, :sum_nutrients=>2.559600221972509}
Células Después de la Reproducción
{:vector=>[-0.1839050899582404, -0.25324699322466676], :cost=>0.0979551216898829, :inter=>-0.24239637789192342, :fitness=>-0.1444412562020405, :sum_nutrients=>-0.3595125679010528}
{:vector=>[-0.25285906718724277, -0.8715987037932177], :cost=>0.8236220083128198, :inter=>-0.23063113805144492, :fitness=>0.5929908702613749, :sum_nutrients=>0.5929908702613749}
{:vector=>[-0.1839050899582404, -0.25324699322466676], :cost=>0.0979551216898829, :inter=>-0.24239637789192342, :fitness=>-0.1444412562020405, :sum_nutrients=>-0.3595125679010528}
{:vector=>[-0.25285906718724277, -0.8715987037932177], :cost=>0.8236220083128198, :inter=>-0.23063113805144492, :fitness=>0.5929908702613749, :sum_nutrients=>0.5929908702613749}
--End Nre
Células Antes de la Eliminación
{:vector=>[-0.1839050899582404, -0.25324699322466676], :cost=>0.0979551216898829, :inter=>-0.24239637789192342, :fitness=>-0.1444412562020405, :sum_nutrients=>-0.3595125679010528}
{:vector=>[-0.25285906718724277, -0.8715987037932177], :cost=>0.8236220083128198, :inter=>-0.23063113805144492, :fitness=>0.5929908702613749, :sum_nutrients=>0.5929908702613749}
{:vector=>[-0.1839050899582404, -0.25324699322466676], :cost=>0.0979551216898829, :inter=>-0.24239637789192342, :fitness=>-0.1444412562020405, :sum_nutrients=>-0.3595125679010528}
{:vector=>[-0.25285906718724277, -0.8715987037932177], :cost=>0.8236220083128198, :inter=>-0.23063113805144492, :fitness=>0.5929908702613749, :sum_nutrients=>0.5929908702613749}
Células Después de la Eliminación
{:vector=>[-0.1839050899582404, -0.25324699322466676], :cost=>0.0979551216898829, :inter=>-0.24239637789192342, :fitness=>-0.1444412562020405, :sum_nutrients=>-0.3595125679010528}
{:vector=>[-0.25285906718724277, -0.8715987037932177], :cost=>0.8236220083128198, :inter=>-0.23063113805144492, :fitness=>0.5929908702613749, :sum_nutrients=>0.5929908702613749}
{:vector=>[-0.1839050899582404, -0.25324699322466676], :cost=>0.0979551216898829, :inter=>-0.24239637789192342, :fitness=>-0.1444412562020405, :sum_nutrients=>-0.3595125679010528}
{:vector=>[-0.25285906718724277, -0.8715987037932177], :cost=>0.8236220083128198, :inter=>-0.23063113805144492, :fitness=>0.5929908702613749, :sum_nutrients=>0.5929908702613749}
--End Ned
Ned = 1
Nre = 0
```

BFO: Ejemplo

```
>> chemo=19, f=-0.12324837757224955, cost=0.0018639220957609225
> best fitness=-0.12198151855712416, cost=0.0008427674061680323
Células Antes de la Reproducción
{:vector=>[0.24770425064539273, -0.05877218510004388], :cost=>0.06481156552922937, :inter=>-0.1854909352959739, :fitness=>-0.12067936976674454, :sum_nutrients=>-0.12067936976674454}
{:vector=>[-0.20225989669641842, 0.2508768354151019], :cost=>0.10384825235954198, :inter=>-0.24785441642388426, :fitness=>-0.14400616406434227, :sum_nutrients=>-0.14400616406434227}
{:vector=>[0.05982917586120043, 0.009896832027907043], :cost=>0.003677477568419055, :inter=>-0.14869967632793005, :fitness=>-0.145022198759511, :sum_nutrients=>-0.28267028145756234}
{:vector=>[-0.08938867317107906, -0.19479109675022288], :cost=>0.045933906264440674, :inter=>-0.19800510522775877, :fitness=>-0.1520711989633181, :sum_nutrients=>-0.1520711989633181}
Células Después de la Reproducción
{:vector=>[0.05982917586120043, 0.009896832027907043], :cost=>0.003677477568419055, :inter=>-0.14869967632793005, :fitness=>-0.145022198759511, :sum_nutrients=>-0.28267028145756234}
{:vector=>[-0.08938867317107906, -0.19479109675022288], :cost=>0.045933906264440674, :inter=>-0.19800510522775877, :fitness=>-0.1520711989633181, :sum_nutrients=>-0.1520711989633181}
{:vector=>[0.05982917586120043, 0.009896832027907043], :cost=>0.003677477568419055, :inter=>-0.14869967632793005, :fitness=>-0.145022198759511, :sum_nutrients=>-0.28267028145756234}
{:vector=>[-0.08938867317107906, -0.19479109675022288], :cost=>0.045933906264440674, :inter=>-0.19800510522775877, :fitness=>-0.1520711989633181, :sum_nutrients=>-0.1520711989633181}
--End Nre
Células Antes de la Eliminación
{:vector=>[0.05982917586120043, 0.009896832027907043], :cost=>0.003677477568419055, :inter=>-0.14869967632793005, :fitness=>-0.145022198759511, :sum_nutrients=>-0.28267028145756234}
{:vector=>[-0.08938867317107906, -0.19479109675022288], :cost=>0.045933906264440674, :inter=>-0.19800510522775877, :fitness=>-0.1520711989633181, :sum_nutrients=>-0.1520711989633181}
{:vector=>[0.05982917586120043, 0.009896832027907043], :cost=>0.003677477568419055, :inter=>-0.14869967632793005, :fitness=>-0.145022198759511, :sum_nutrients=>-0.28267028145756234}
{:vector=>[-0.08938867317107906, -0.19479109675022288], :cost=>0.045933906264440674, :inter=>-0.19800510522775877, :fitness=>-0.1520711989633181, :sum_nutrients=>-0.1520711989633181}
Células Después de la Eliminación
{:vector=>[-4.851258991747443, 3.0865453403251966], :cost=>0.003677477568419055, :inter=>-0.14869967632793005, :fitness=>-0.145022198759511, :sum_nutrients=>-0.28267028145756234}
{:vector=>[-0.08938867317107906, -0.19479109675022288], :cost=>0.045933906264440674, :inter=>-0.19800510522775877, :fitness=>-0.1520711989633181, :sum_nutrients=>-0.1520711989633181}
{:vector=>[-4.851258991747443, 3.0865453403251966], :cost=>0.003677477568419055, :inter=>-0.14869967632793005, :fitness=>-0.145022198759511, :sum_nutrients=>-0.28267028145756234}
{:vector=>[-0.08938867317107906, -0.19479109675022288], :cost=>0.045933906264440674, :inter=>-0.19800510522775877, :fitness=>-0.1520711989633181, :sum_nutrients=>-0.1520711989633181}
--End Ned
Solución Final: c=0.0008427674061680323, v=[0.0018108574962621901, -0.028973922780601583]
```


Bacterial Evolutionary Algorithm

608

IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 7, NO. 5, OCTOBER 1999

Fuzzy System Parameters Discovery by Bacterial Evolutionary Algorithm

Norberto Eiji Nawa and Takeshi Furuhashi

Sistemas Fuzzy

- Lógica Fuzzy
- Lógica: Es la ciencia que tiene por objetivo el estudio de las leyes de raciocinio.
- Lógica Fuzzy: Es la ciencia que se preocupa con los principios formales de raciocinio aproximado.

Sistemas Fuzzy

- Busca modelar la forma imprecisa de raciocinio que tiene un papel fundamental en la habilidad humana de tomar decisiones.
- La lógica Fuzzy es una herramienta capaz de capturar información vaga, imprecisa e incompleta, descritas en el lenguaje natural, y convertirlas a un formato numérico.

Sistemas Fuzzy

- La lógica fuzzy provee los fundamentos para efectuar el raciocinio aproximado, con proposiciones imprecisas, usando la teoría de conjuntos fuzzy como herramienta principal.
- La teoría de conjuntos fuzzy, es una extensión de la teoría de conjuntos clásicos. Propuesta por Lofti Zadeh en 1965.



Sistemas Fuzzy

- Los conjuntos clásicos, solo permiten dos opciones, la pertenencia o no pertenencia de un elemento.
- Los conjuntos fuzzy permite pertenencias parciales de un elemento en uno o mas conjuntos.

Sistemas Fuzzy

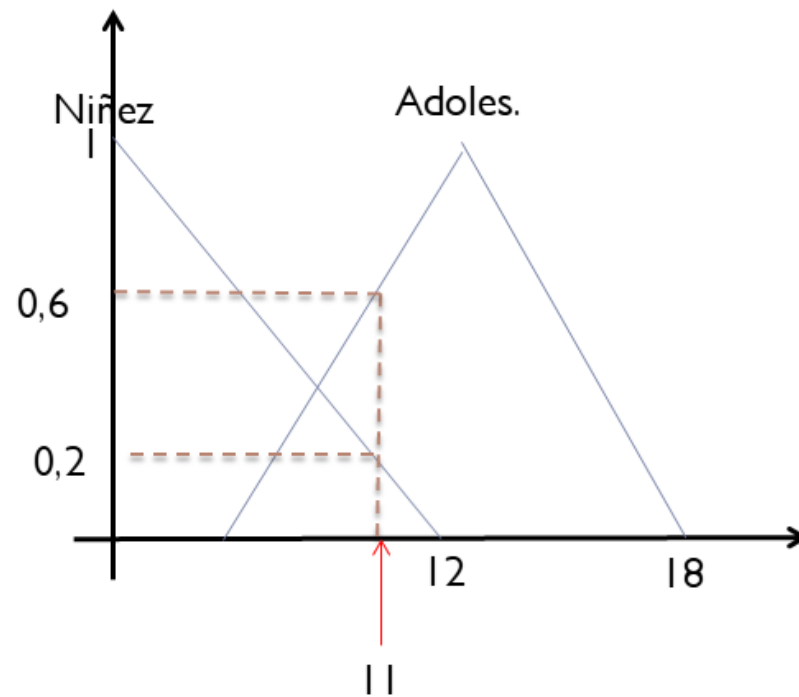
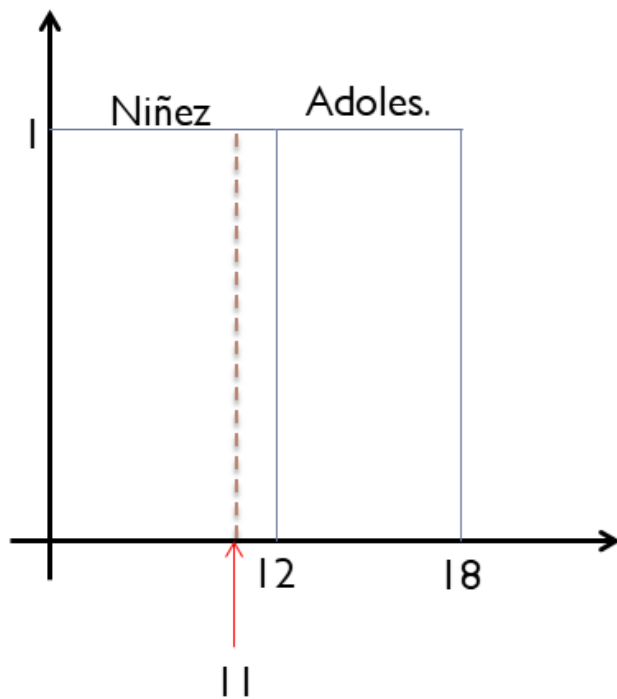
- Lógica clásica: Existen dos estados posibles para una puerta:
 - Puerta abierta (1)
 - Puerta cerrada (0)
- Lógica fuzzy: Establece que una puerta no tiene por que estar necesariamente abierta o cerrada, existen infinitos estados:
 - Puerta abierta (1)
 - Puerta bastante abierta (0,8)
 - Puerta media abierta (0,5)
 - Puerta casi cerrada (0,1)
 - Puerta cerrada (0)



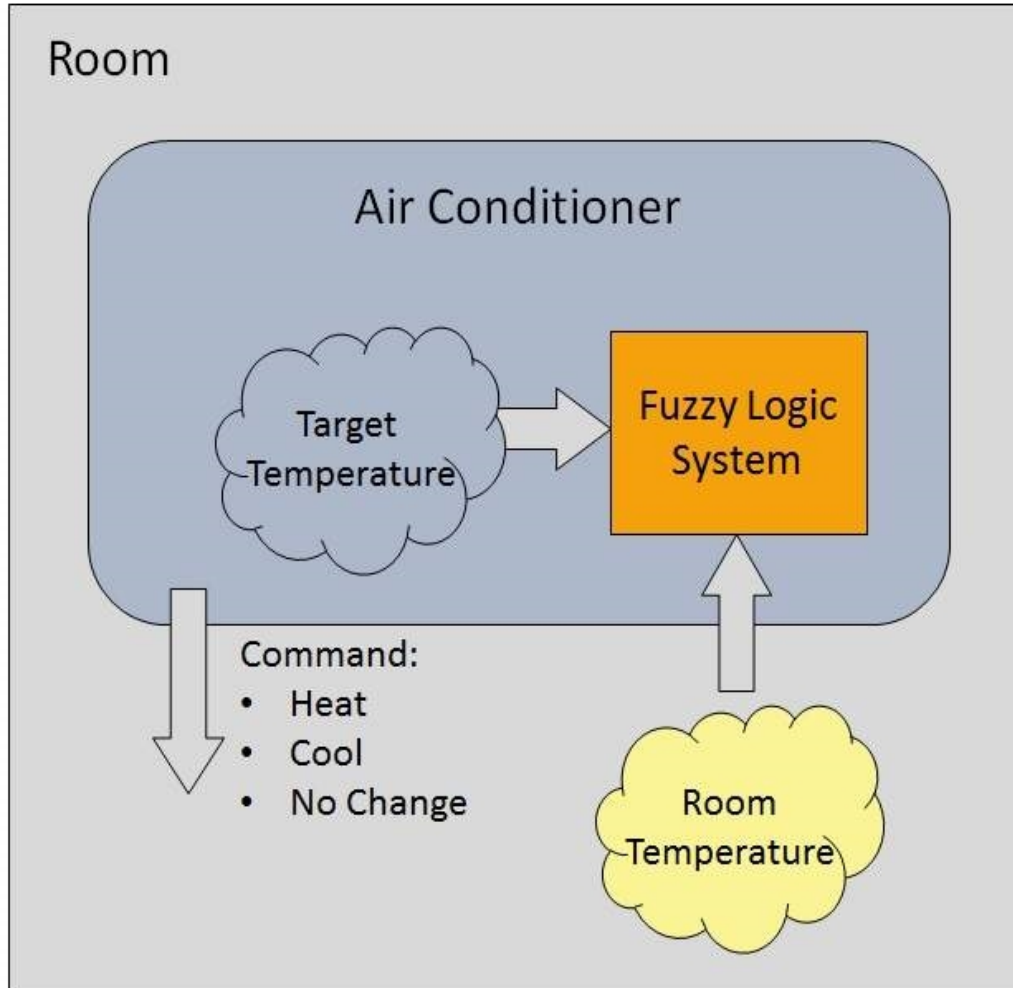
Sistemas Fuzzy

Conjuntos *Fuzzy* - Ejemplo

- ▶ Ejemplo: Edad de una persona: 11 años

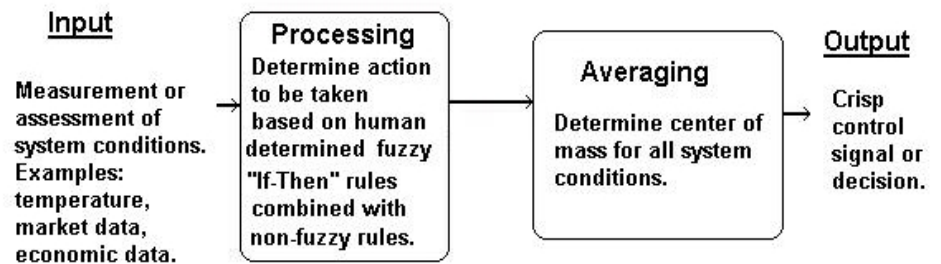


Sistemas Fuzzy



Fuzzy logic application

- household appliances
- animation systems
- industrial automation
- chemical industry
- aerospace
- robotics
- mining and metal processing
- transportation

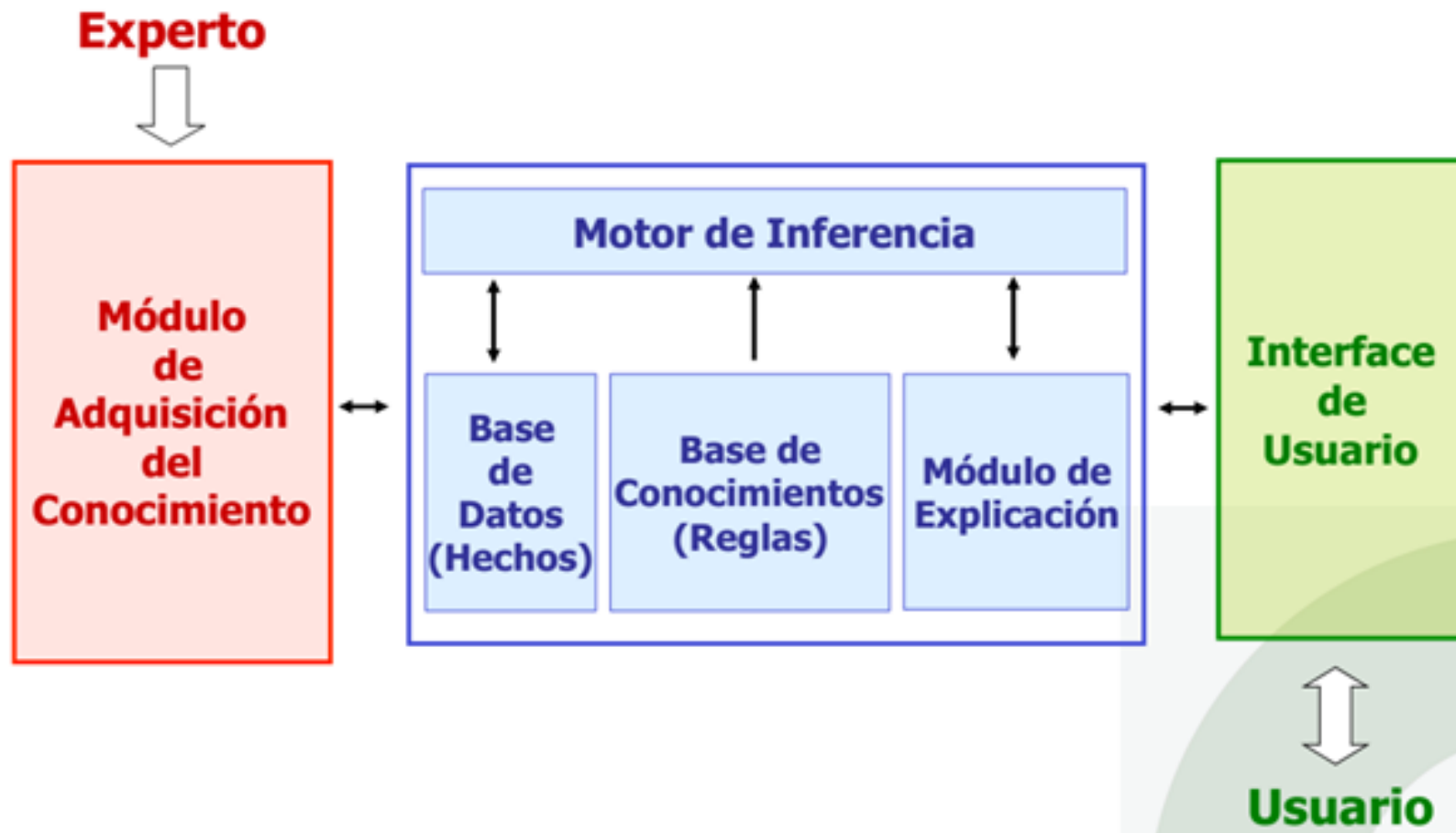


The Fuzzy Logic Control-Analysis Method

Sistemas Fuzzy

- Un tipo de esos sistemas fuzzy, son los sistemas fuzzy basados en reglas (SFBR).
- Son una extensión de los sistemas basados en reglas (SBR) de la lógica clásica.
- Es una de las metodologías mas utilizadas en sistemas expertos.

Sistemas Fuzzy



SFBR

- Se basa en reglas determinísticas, es decir, reglas con formato SI -> ENTONCES. En el antecedente generalmente son utilizados los operadores lógicos.
- Están divididos en la base de conocimiento (variables y la reglas) y el motor de inferencia que obtiene las conclusiones.

SFBR - Ejemplo SBR

- Sistema experto para la detección de cáncer.
- Variables: Edad y Peso
- SI Edad < 12 Y Peso < 20 ENTONCES Cáncer
- SI Edad < 12 Y Peso > 20 ENTONCES No Cáncer
- SI Edad > 12 Y Edad < 28 Y Peso > 40 ENTONCES No Cáncer
-

SFBR - Ejemplo SBR

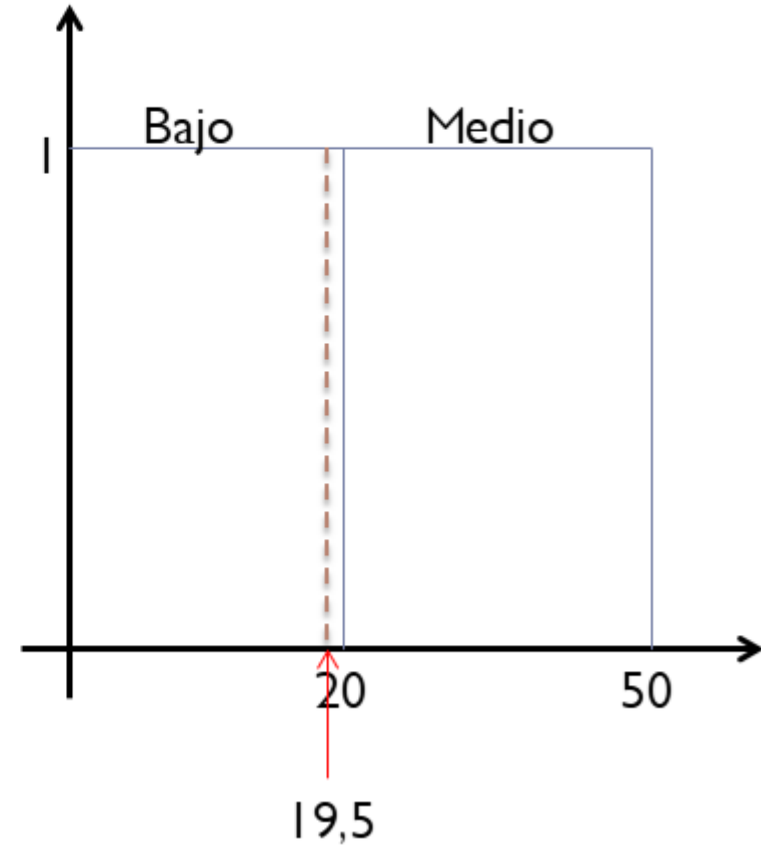
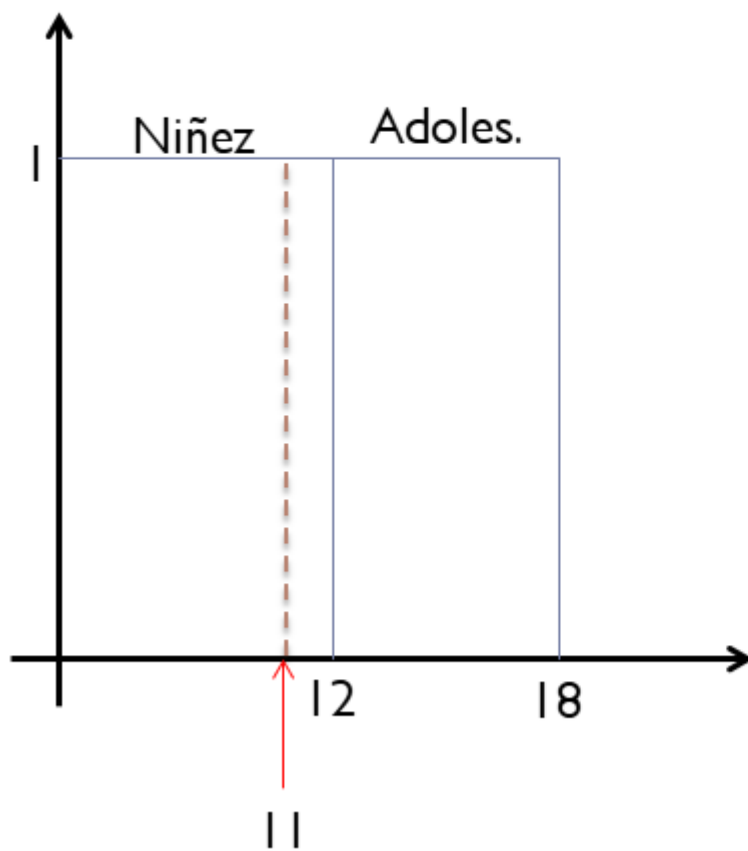
- Ejemplo: Sistema experto para la detección de cáncer.
- Variables: Edad y Peso
- SI Edad ES Niño Y Peso ES Bajo ENTONCES Cáncer
- SI Edad ES Niño Y Peso ES Medio ENTONCES No Cáncer
- SI Edad ES Adolescente Y Peso ES Medio ENTONCES No Cáncer
-

SFBR - Ejemplo SBR

- Ejemplo: Sistema experto para la detección de cáncer.
- Variables: Edad y Peso
- SI Edad ES Niñez Y Peso ES Bajo ENTONCES Cáncer
- SI Edad ES Niñez Peso ES Medio ENTONCES No Cáncer
- SI Edad ES Adolescente Y Peso ES Medio ENTONCES No Cáncer
-

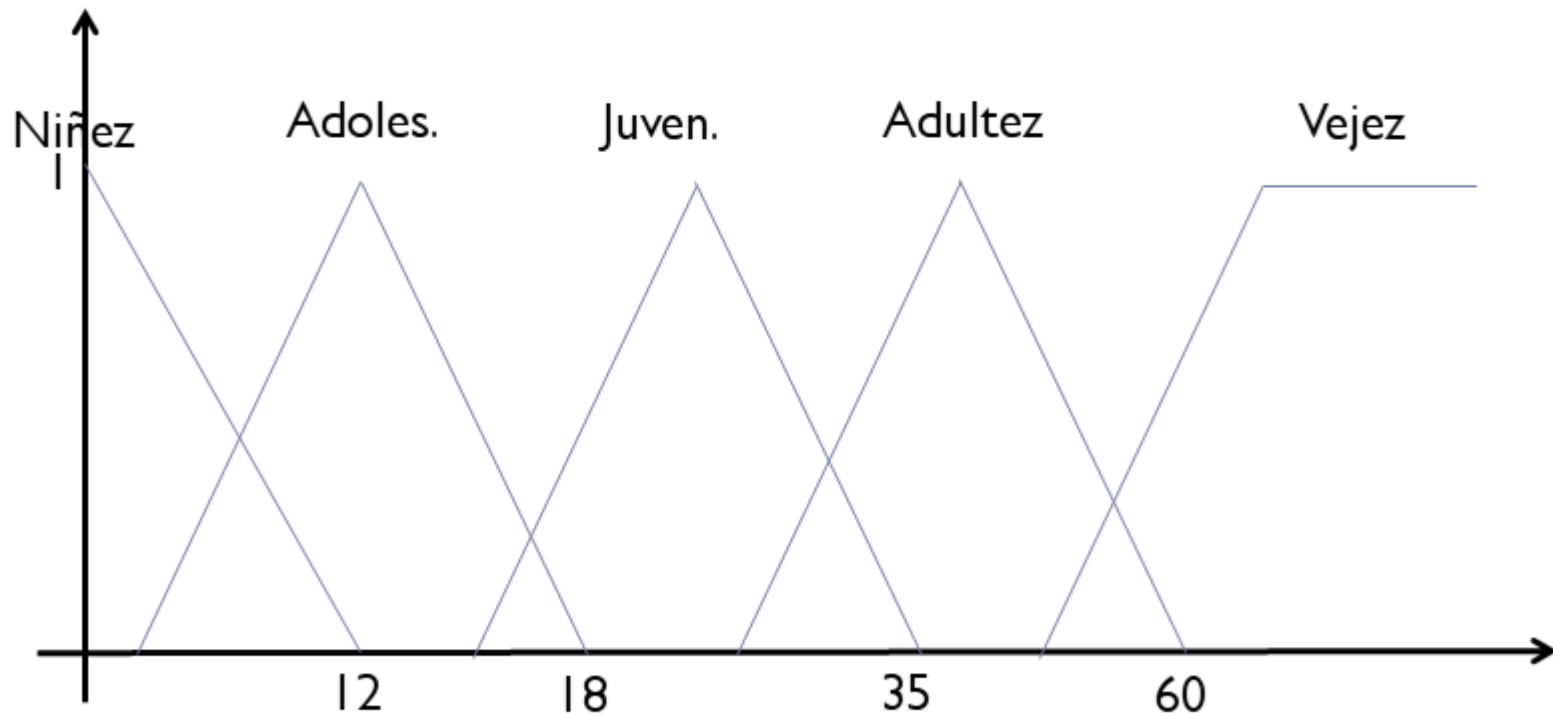
SFBR - Ejemplo SBR

- SI Edad ES Niñez Y Peso ES Bajo ENTONCES Cáncer



- También se utilizan variables lingüísticas y términos lingüísticos.
- Términos lingüísticos vinculados a un conjunto fuzzy.
- Edad: Niñez, Adolescencia, Juventud, Adultez y Vejez.
- Peso: Bajo, Medio y Alto

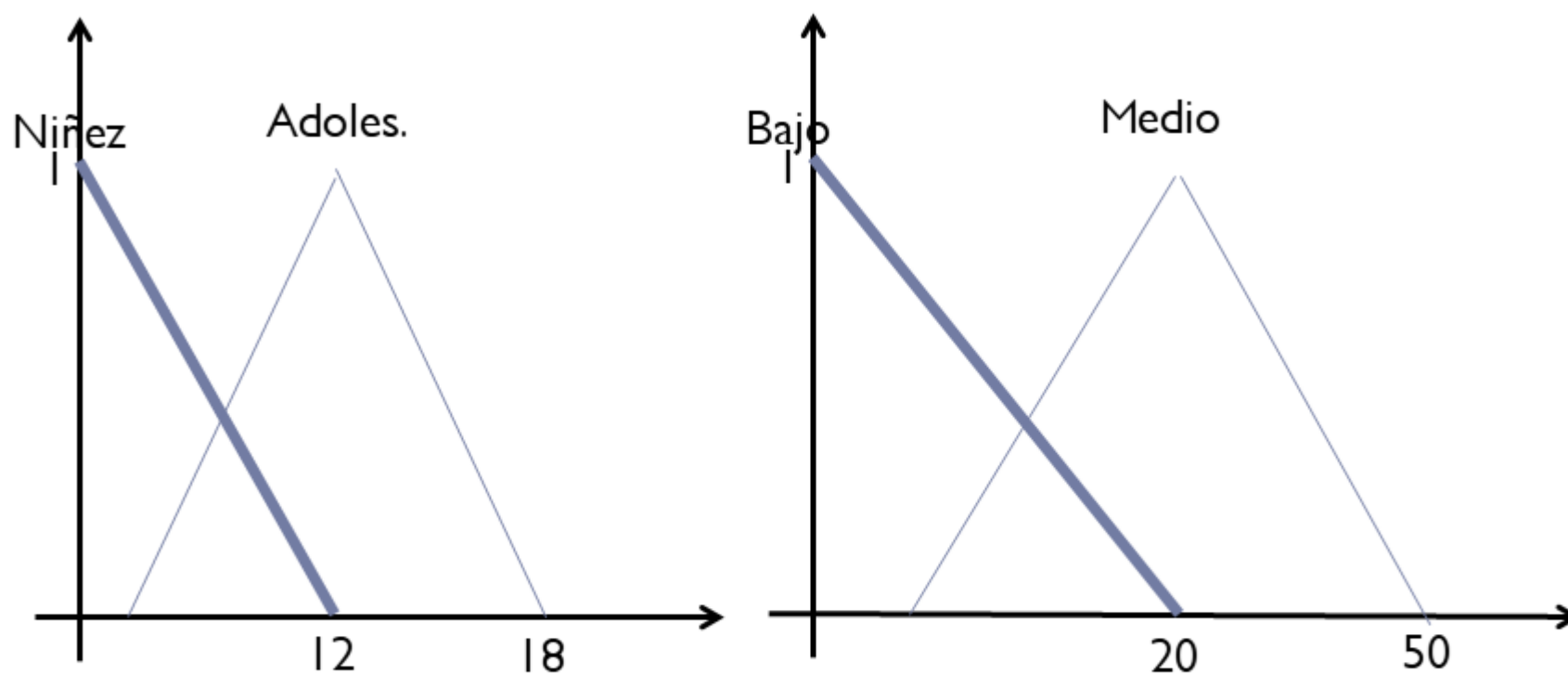
- ▶ Edad: Niñez, Adolescencia, Juventud, Adultez y Vejez.



- ▶ Peso: Bajo, Medio y Alto

SFBR

1. SI Edad ES N Y Peso ES B ENTONCES C

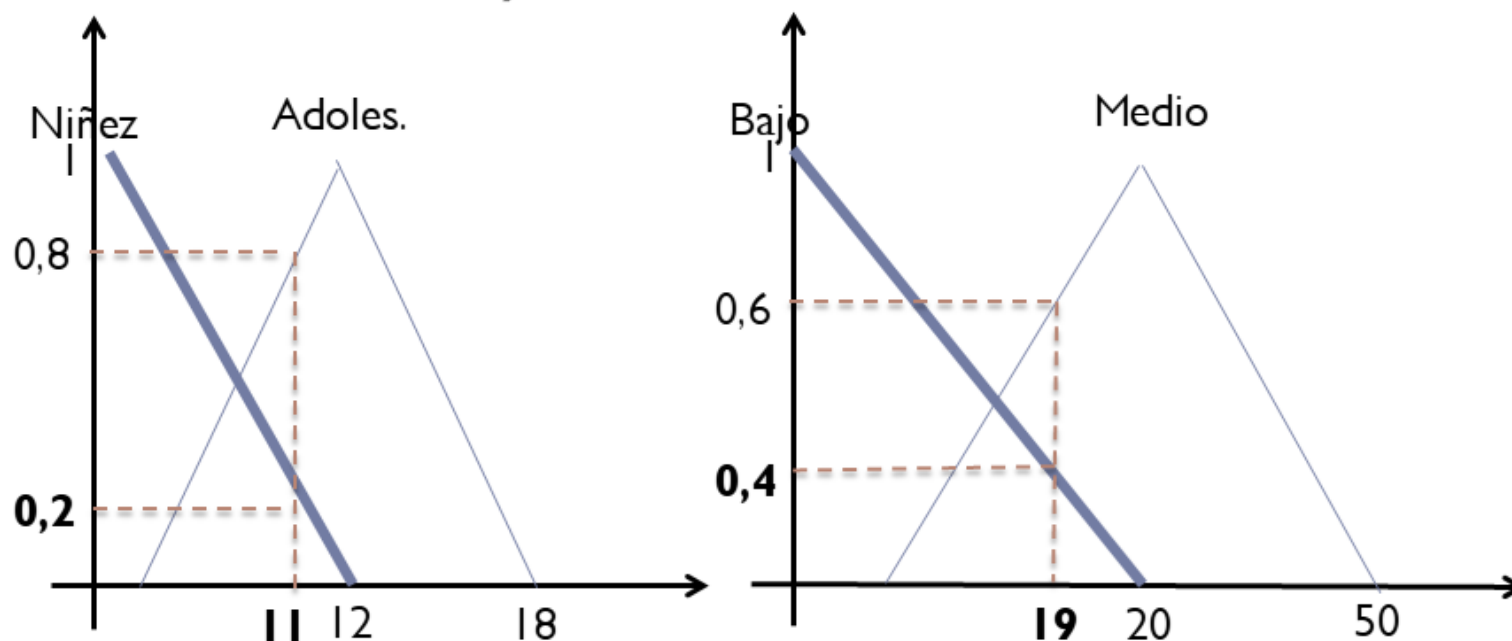


2. SI Edad ES N Y Peso ES M ENTONCES NC

3. ...

Entrada: Edad: 11 años y Peso: 19 kg

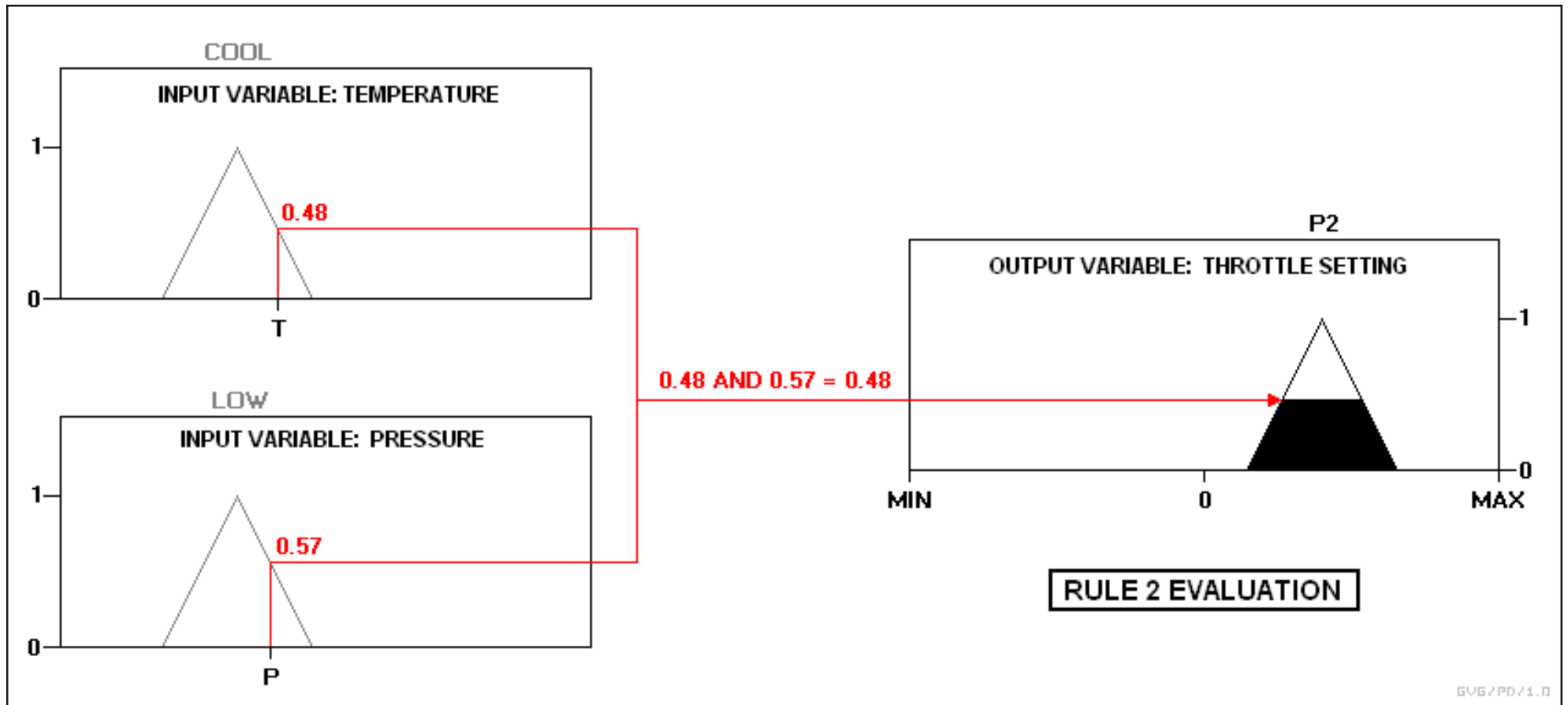
1. SI Edad ES N y Peso ES B ENTONCES Cáncer



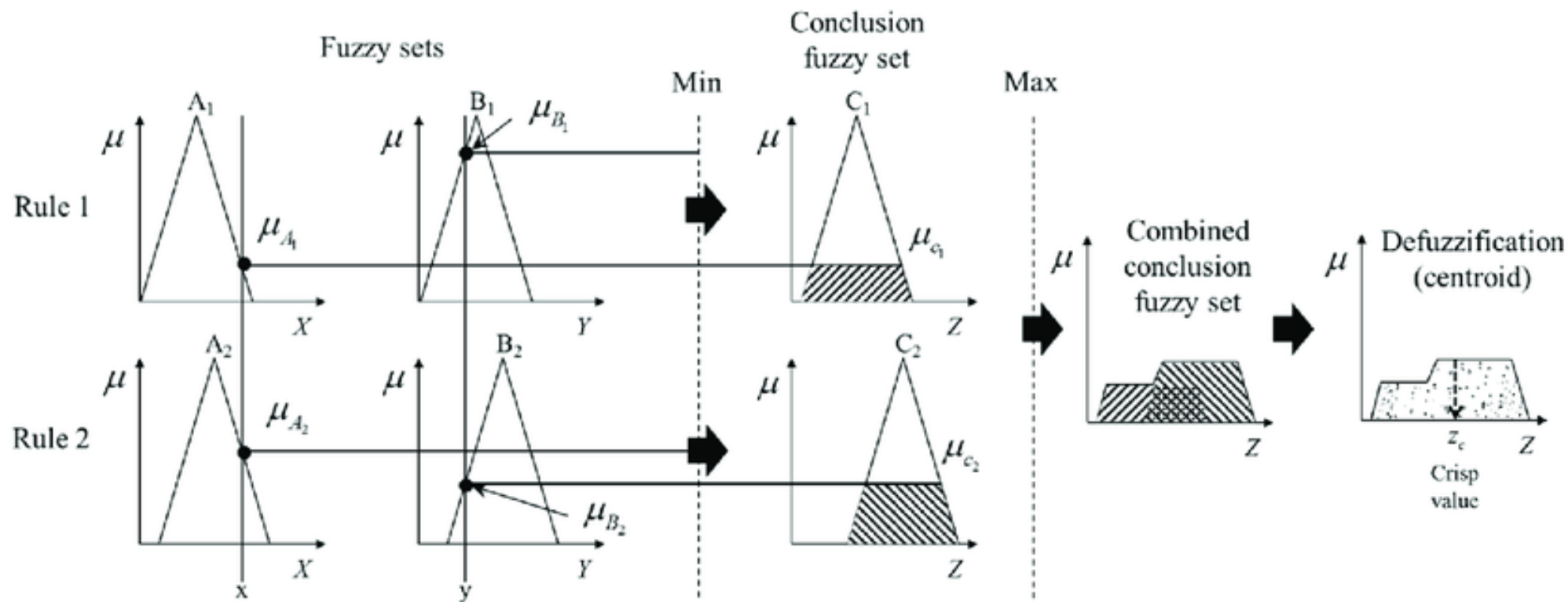
2. SI Edad ES N Y Peso ES M ENTONCES No Cáncer

3. ...

SFBR

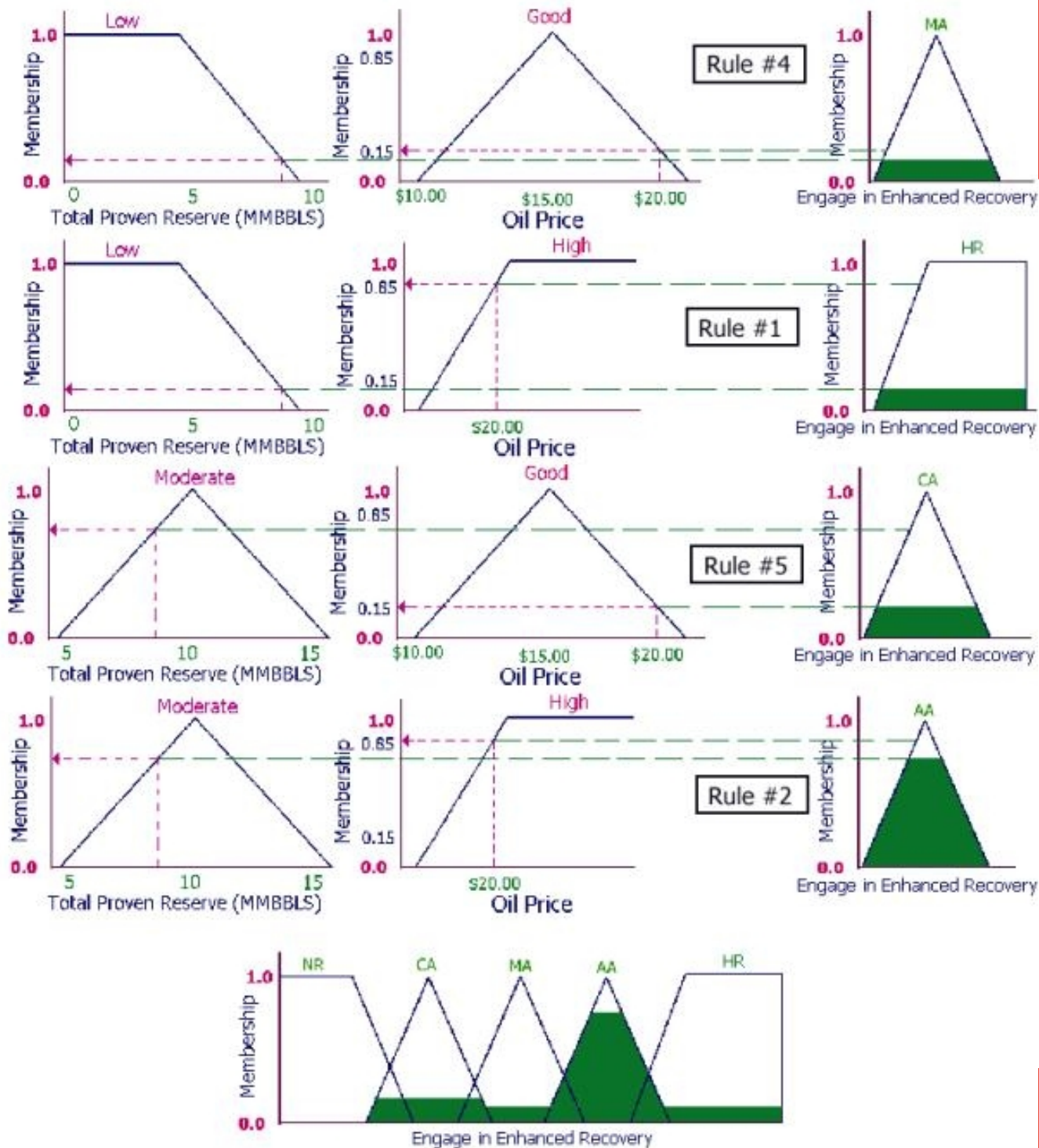


SFBR

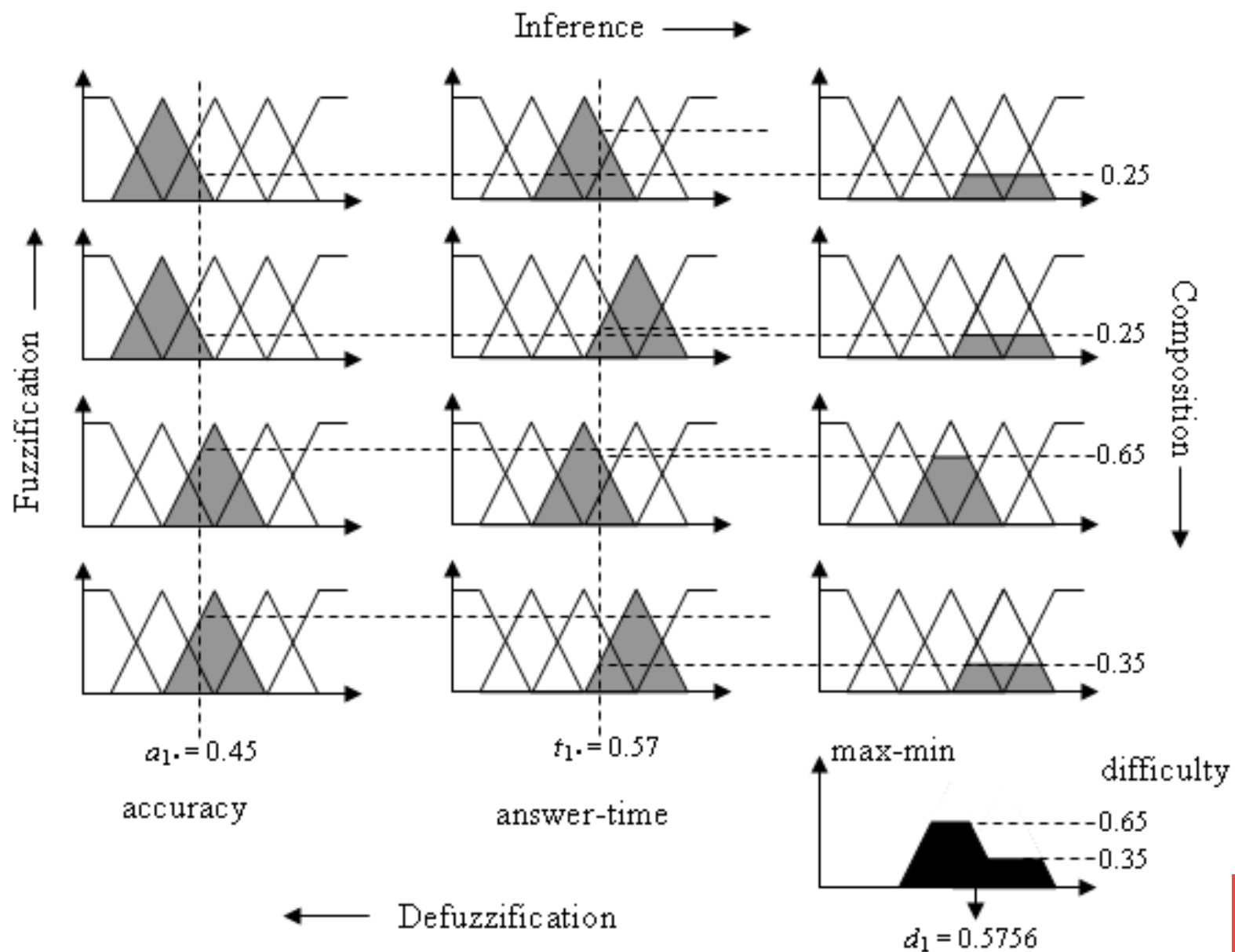


Rule 1. If 'x' is ' A_1 ' and 'y' is ' B_1 ' then 'Z' is ' C_1 '
 Rule 2. If 'x' is ' A_2 ' and 'y' is ' B_2 ' then 'Z' is ' C_2 '

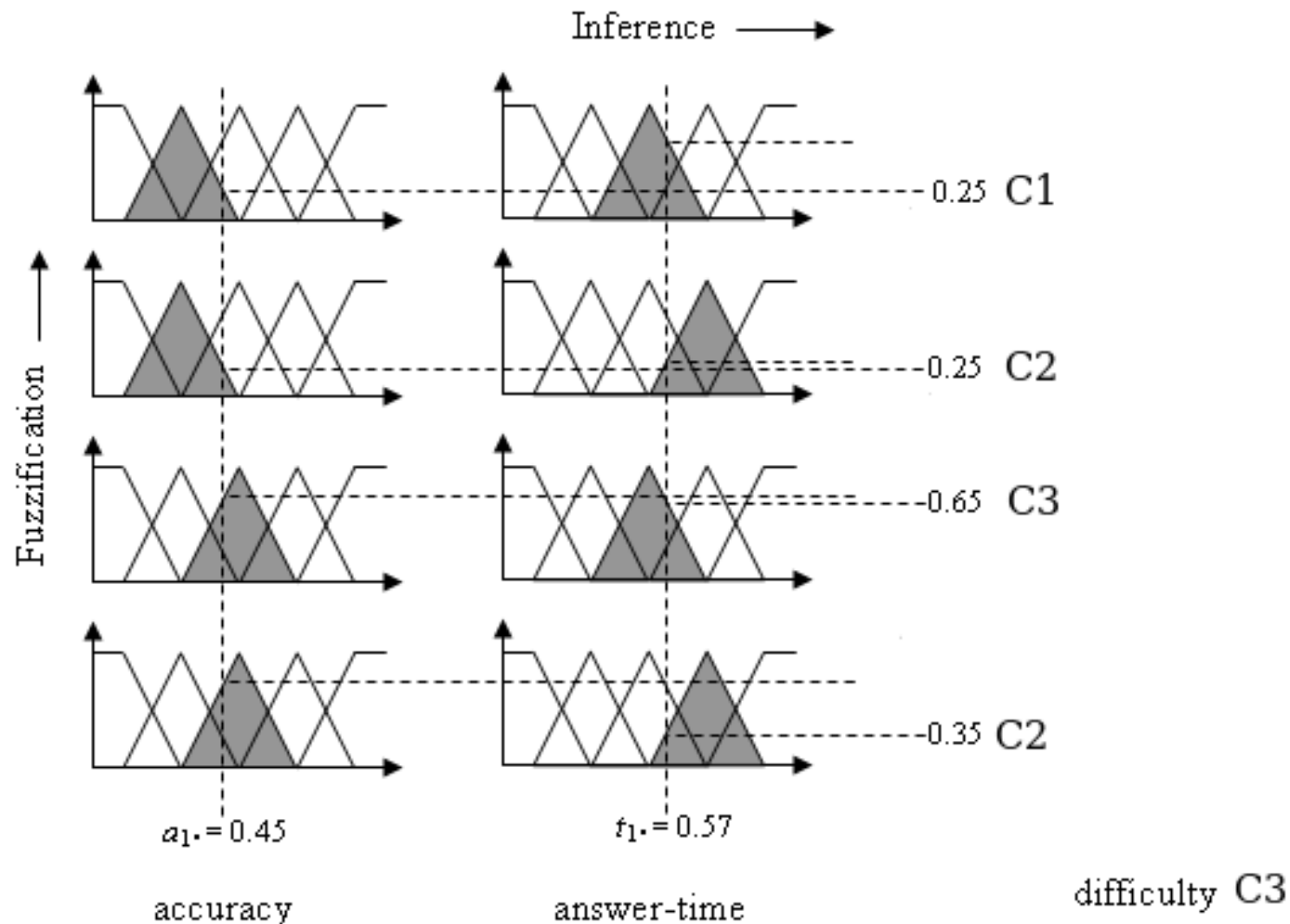
SFBR



SFBR

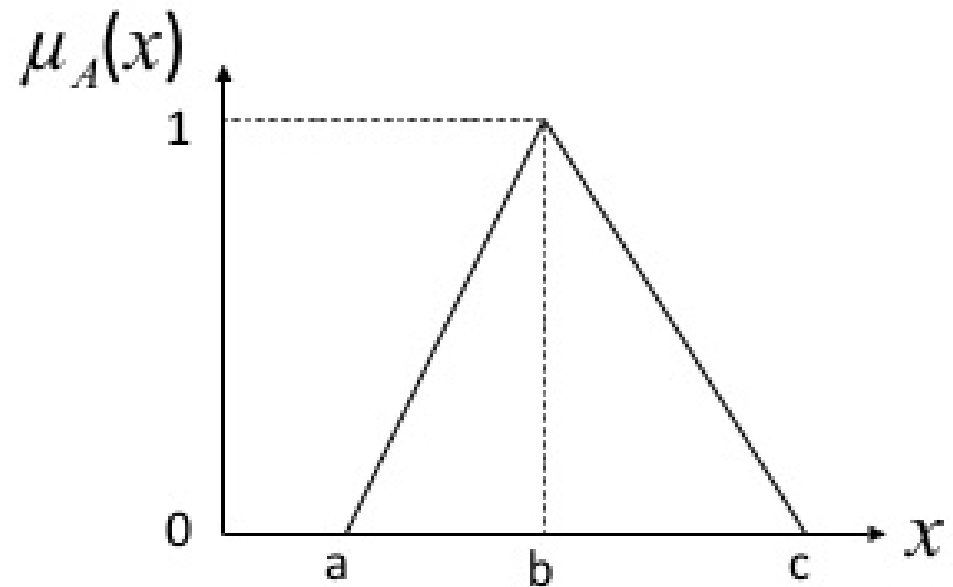


Sistemas Fuzzy de Clasificación Basados en Regla (SFCBR)



Sistemas Fuzzy de Clasificación Basados en Regla (SFCBR)

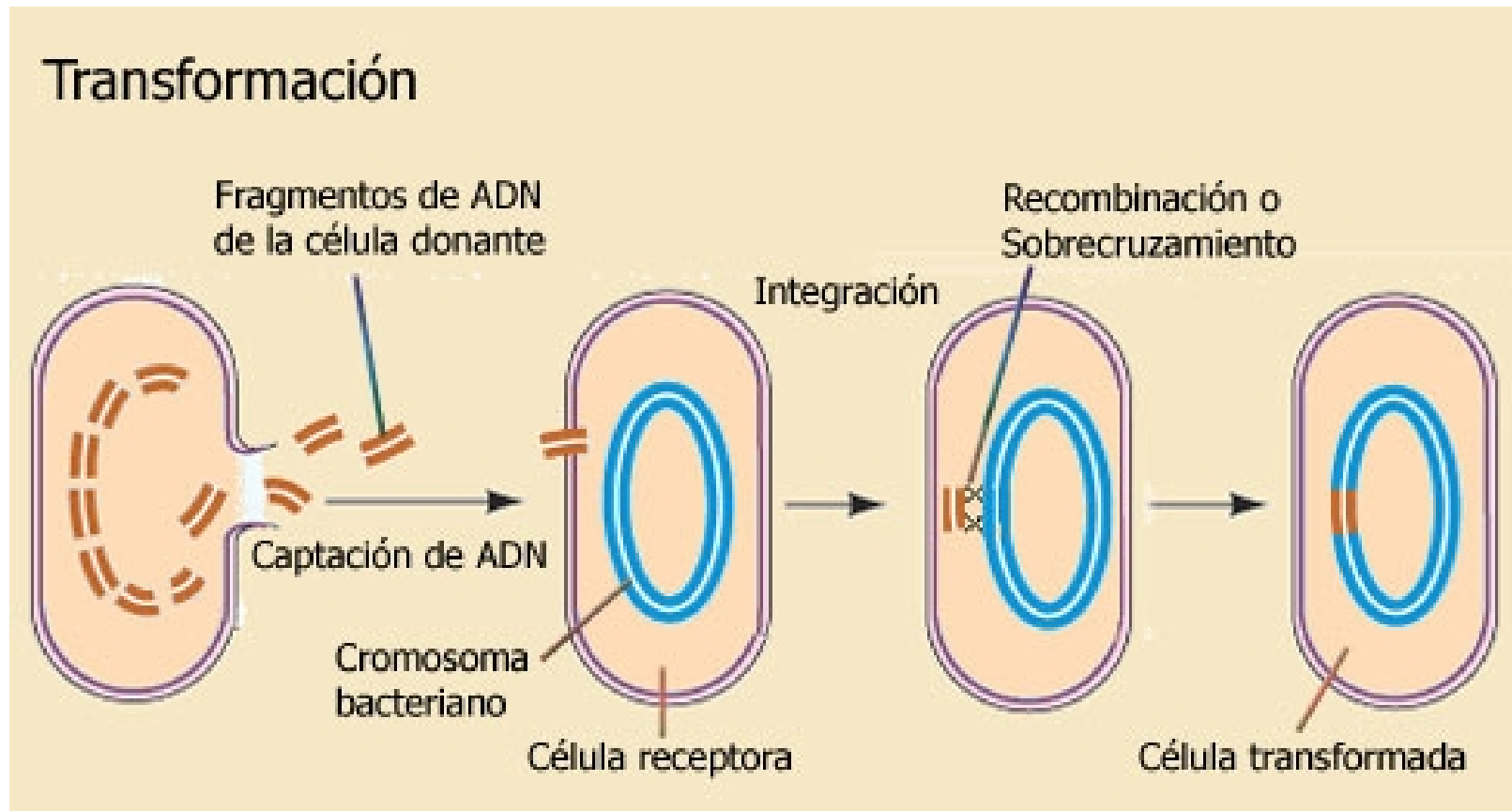
$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if } b \leq x \leq c \\ 0 & \text{if } x \geq c \end{cases}$$



Bacterial Evolutionary Algorithm

- Un proceso interesante de recombinación bacteriana se puede encontrar en la naturaleza: las bacterias pueden transferir ADN a las células receptoras a través del apareamiento.
- Este proceso se llama transducción.

Bacterial Evolutionary Algorithm



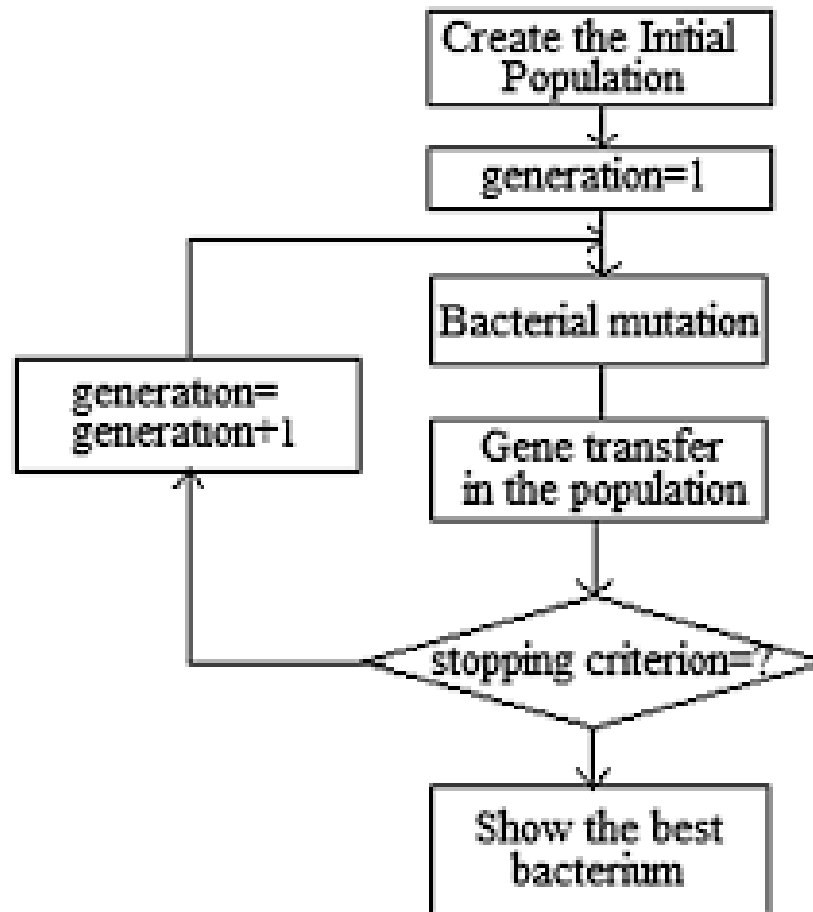
Bacterial Evolutionary Algorithm

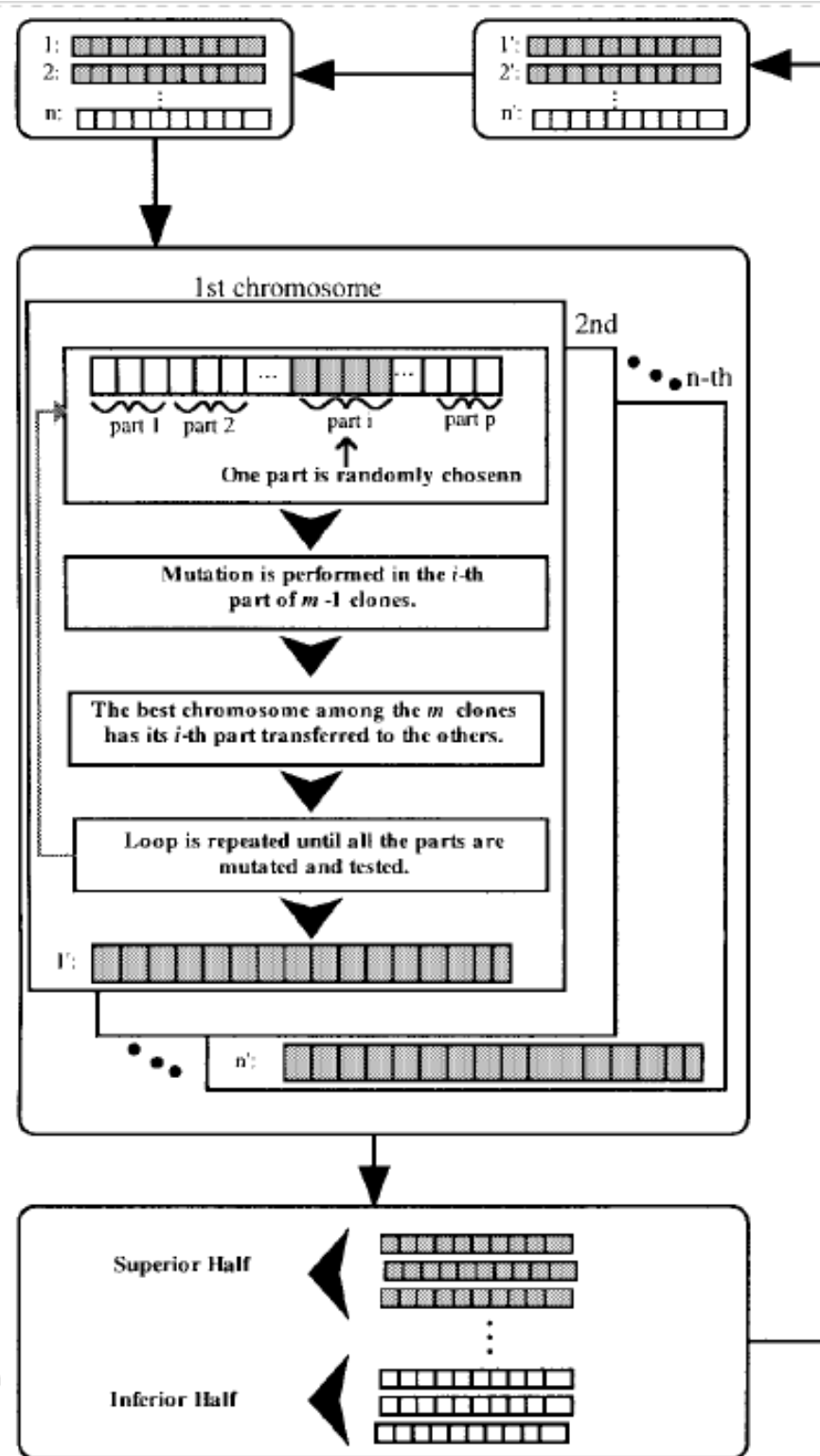
- Por transducción, es posible extender las características de una sola bacteria al resto de la población.
- Este mecanismo de recombinación genética configura un proceso de evolución microbiana. Las células masculinas transfieren directamente hebras de genes a las células femeninas. Después de eso, esas células femeninas adquieren características de las células masculinas y se transforman en células masculinas.

Bacterial Evolutionary Algorithm

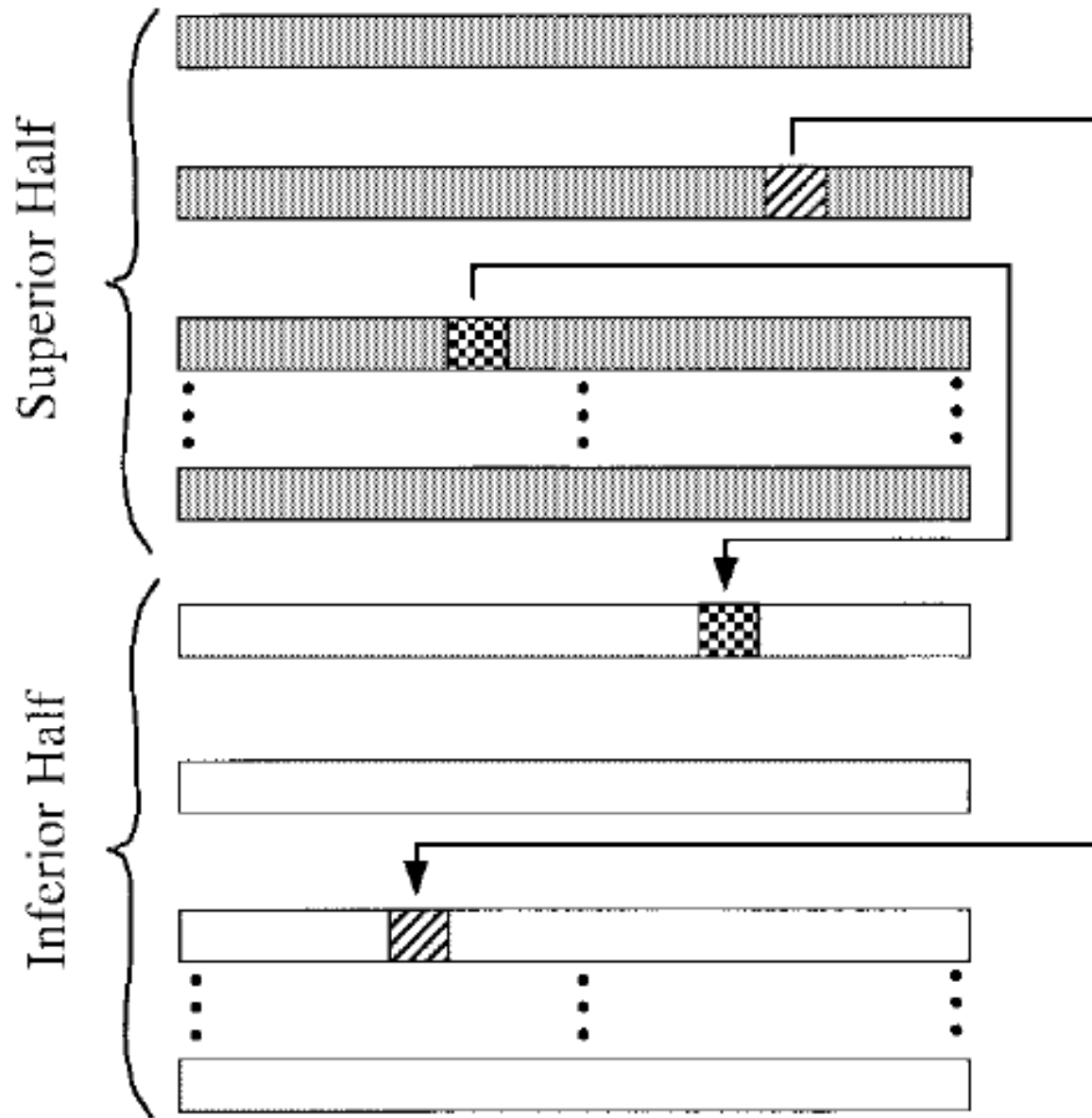
- De este modo, las características de una bacteria pueden propagarse a toda la población bacteriana.
- Los genes pueden ser transferidos de una sola bacteria (célula huésped) a otras (células receptoras) y eventualmente esto llevaría a un aumento en la velocidad de evolución de toda la población.
- Dos características definen al BEA: Mutación y la transferencias de información.

Bacterial Evolutionary Algorithm

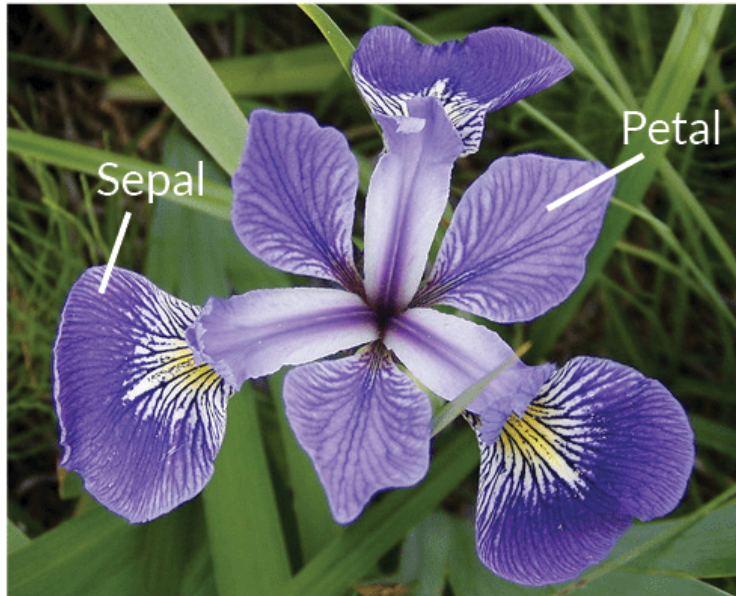




Bacterial Evolutionary Algorithm



BEA - Ejemplo



Iris Versicolor

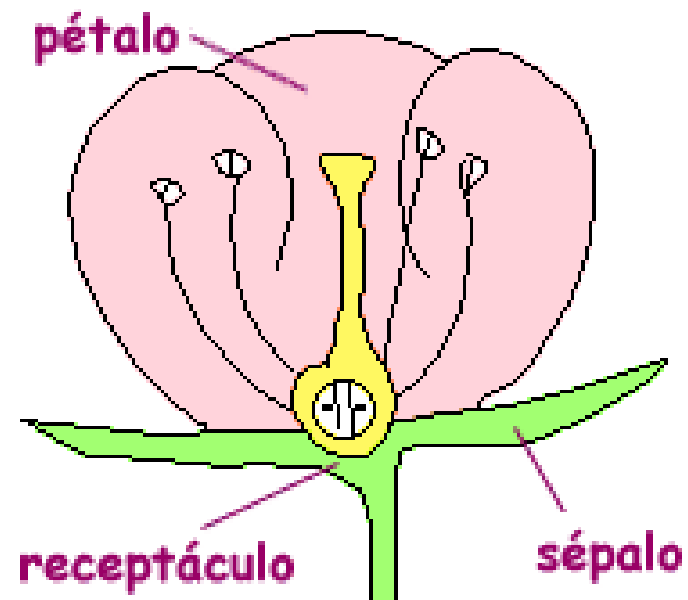


Iris Setosa

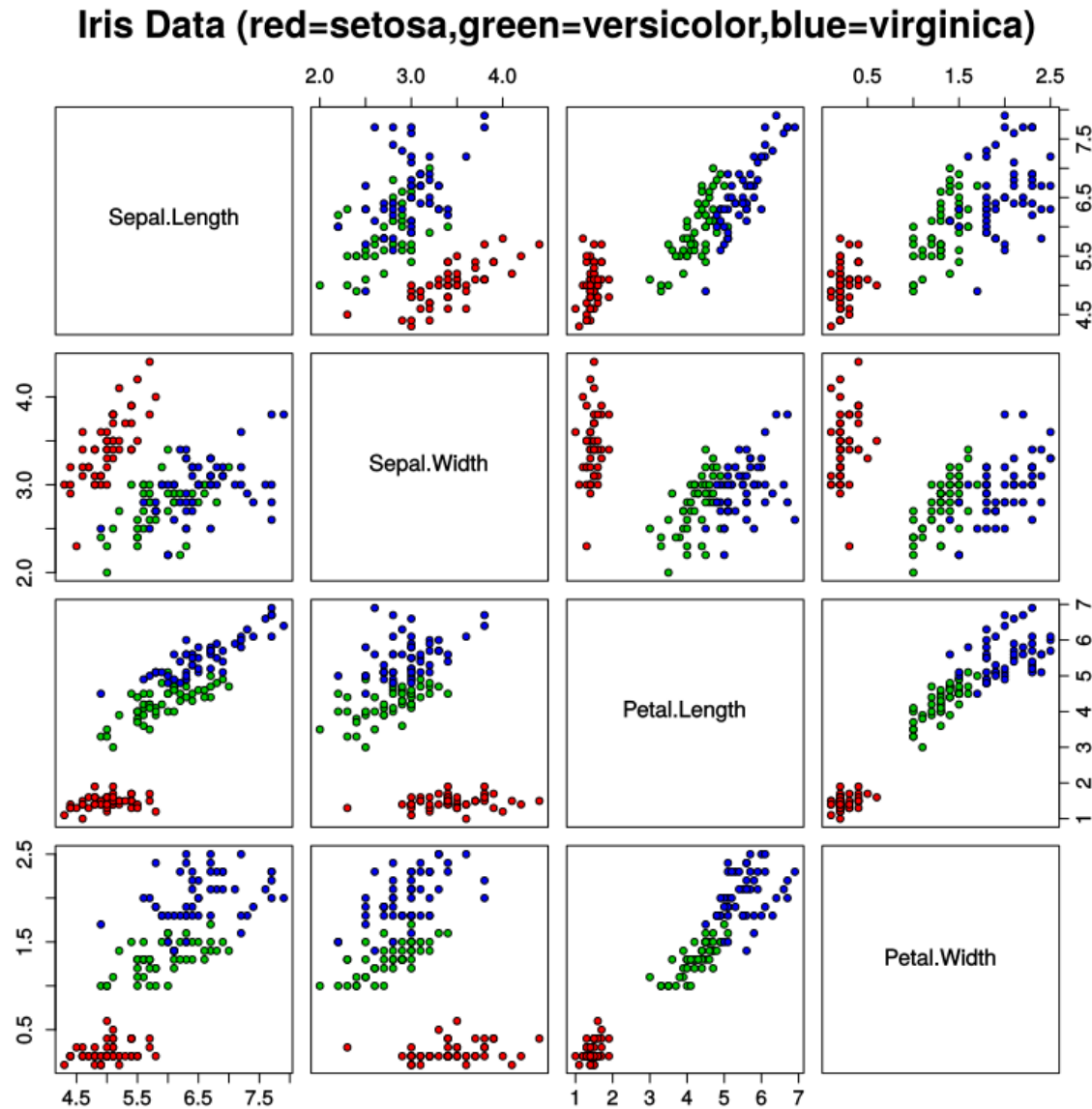


Iris Virginica

BEA - Ejemplo



BEA - Ejemplo

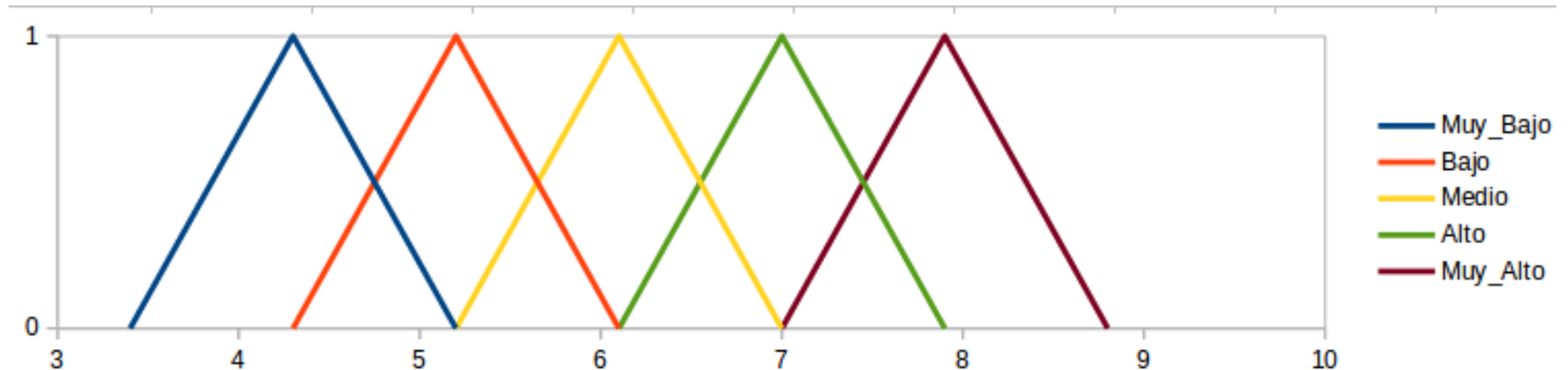


BEA - Ejemplo

- Tamaño de la Población: 10 SFCBR
- Cantidad de Reglas: 5 reglas por cada SFCBR
- Cantidad de Iteraciones: 10 Iteraciones
- Cantidad de Transferencias: 10 Transferencias

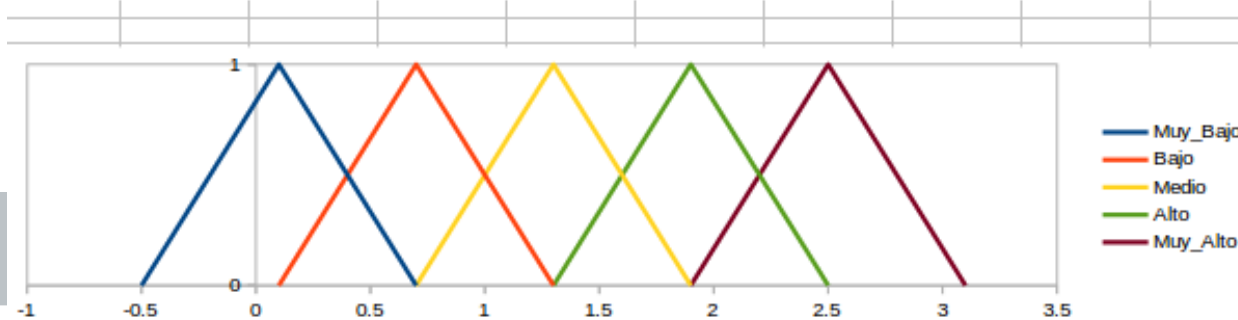
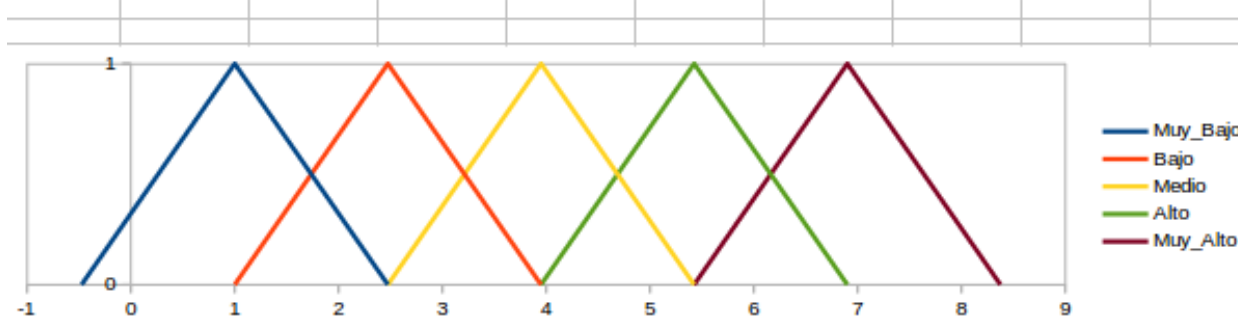
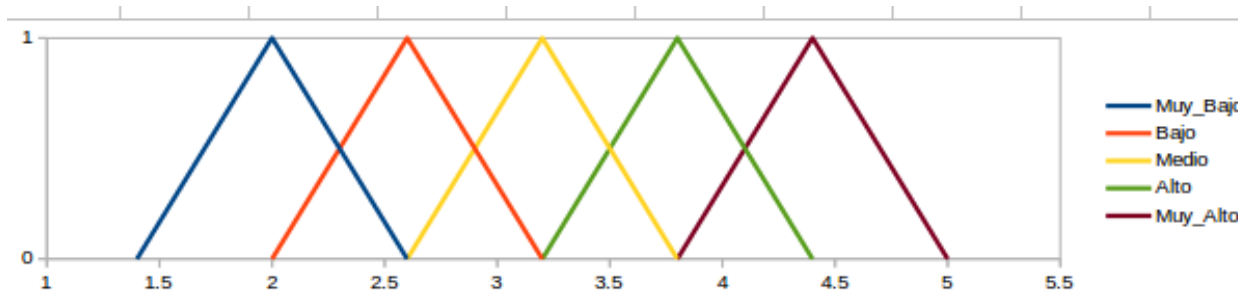
BEA - Ejemplo

	Muy_Bajo			Bajo			Medio			Alto			Muy_Alto		
1	3.4	4.3	5.2	4.3	5.2	6.1	5.2	6.1	7	6.1	7	7.9	7	7.9	8.8
2	1.4	2	2.6	2	2.6	3.2	2.6	3.2	3.8	3.2	3.8	4.4	3.8	4.4	5
3	-0.475	1	2.475	1	2.475	3.95	2.475	3.95	5.425	3.95	5.425	6.9	5.425	6.9	8.375
4	-0.5	0.1	0.7	0.1	0.7	1.3	0.7	1.3	1.9	1.3	1.9	2.5	1.9	2.5	3.1



BEA - Ejemplo

	Muy_Bajo			Bajo			Medio			Alto			Muy_Alto		
1	3.4	4.3	5.2	4.3	5.2	6.1	5.2	6.1	7	6.1	7	7.9	7	7.9	8.8
2	1.4	2	2.6	2	2.6	3.2	2.6	3.2	3.8	3.2	3.8	4.4	3.8	4.4	5
3	-0.475	1	2.475	1	2.475	3.95	2.475	3.95	5.425	3.95	5.425	6.9	5.425	6.9	8.375
4	-0.5	0.1	0.7	0.1	0.7	1.3	0.7	1.3	1.9	1.3	1.9	2.5	1.9	2.5	3.1



BEA - Ejemplo

Población Inicial

[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación: 0.36666667

[1, 3, 3, 1, 1, 0, 3, 0, 2, 0, 4, 1, 4, 4, 2, 3, 3, 0, 2, 2, 3, 1, 3, 0, 1]

Tasa Clasificación: 0.36666667

[1, 3, 3, 0, 2, 1, 3, 1, 1, 0, 1, 4, 0, 1, 2, 0, 1, 3, 2, 0, 0, 4, 3, 0, 1]

Tasa Clasificación: 0.31666666

[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 1, 0, 2, 4, 1, 4, 2, 2, 0, 1, 3, 1, 1, 4, 2]

Tasa Clasificación: 0.33333334

[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 2, 3, 4, 0, 0, 2, 3, 3, 2, 2, 2, 4, 4, 2]

Tasa Clasificación: 0.425

[0, 1, 2, 0, 0, 2, 2, 1, 3, 1, 1, 1, 3, 1, 0, 2, 1, 0, 4, 2, 1, 4, 2, 1, 0]

Tasa Clasificación: 0.33333334

[1, 0, 3, 2, 2, 2, 2, 4, 2, 2, 1, 3, 2, 4, 2, 1, 3, 1, 3, 0, 3, 4, 1, 1, 2]

Tasa Clasificación: 0.36666667

[2, 0, 3, 0, 0, 1, 2, 0, 2, 1, 4, 2, 0, 0, 1, 2, 1, 2, 3, 1, 3, 0, 1, 2, 1]

Tasa Clasificación: 0.45

[2, 0, 0, 2, 1, 4, 1, 1, 3, 2, 0, 3, 2, 4, 1, 3, 0, 1, 4, 2, 1, 0, 2, 3, 2]

Tasa Clasificación: 0.35

[1, 3, 4, 3, 2, 2, 3, 2, 3, 2, 3, 4, 4, 0, 1, 3, 2, 2, 3, 1, 3, 0, 3, 4, 0]

Tasa Clasificación: 0.40833333

BEA - Ejemplo

[0, 2, 1, 2, 0, 3, 1, 4, 0, 0, 4, 4, 0, 3, 0, 2, 3, 0, 1, 0, 3, 2, 0, 3, 2]

Tasa Clasificación: 0.33333334

RULE 1: IF LS IS Muy_Bajo AND AS IS Medio AND LP IS Bajo AND AP IS Medio THEN Class IS Iris-setosa

RULE 2: IF LS IS Alto AND AS IS Bajo AND LP IS Muy_Alto AND AP IS Muy_Bajo THEN Class IS Iris-setosa

RULE 3: IF LS IS Muy_Alto AND AS IS Muy_Alto AND LP IS Muy_Bajo AND AP IS Alto THEN Class IS Iris-setosa

RULE 4: IF LS IS Medio AND AS IS Alto AND LP IS Muy_Bajo AND AP IS Bajo THEN Class IS Iris-setosa

RULE 5: IF LS IS Alto AND AS IS Medio AND LP IS Muy_Bajo AND AP IS Alto THEN Class IS Iris-versicolor

0 = Muy Bajo

1 = Bajo

2 = Medio

3 = Alto

4 = Muy Alto

0 = Iris-setosa

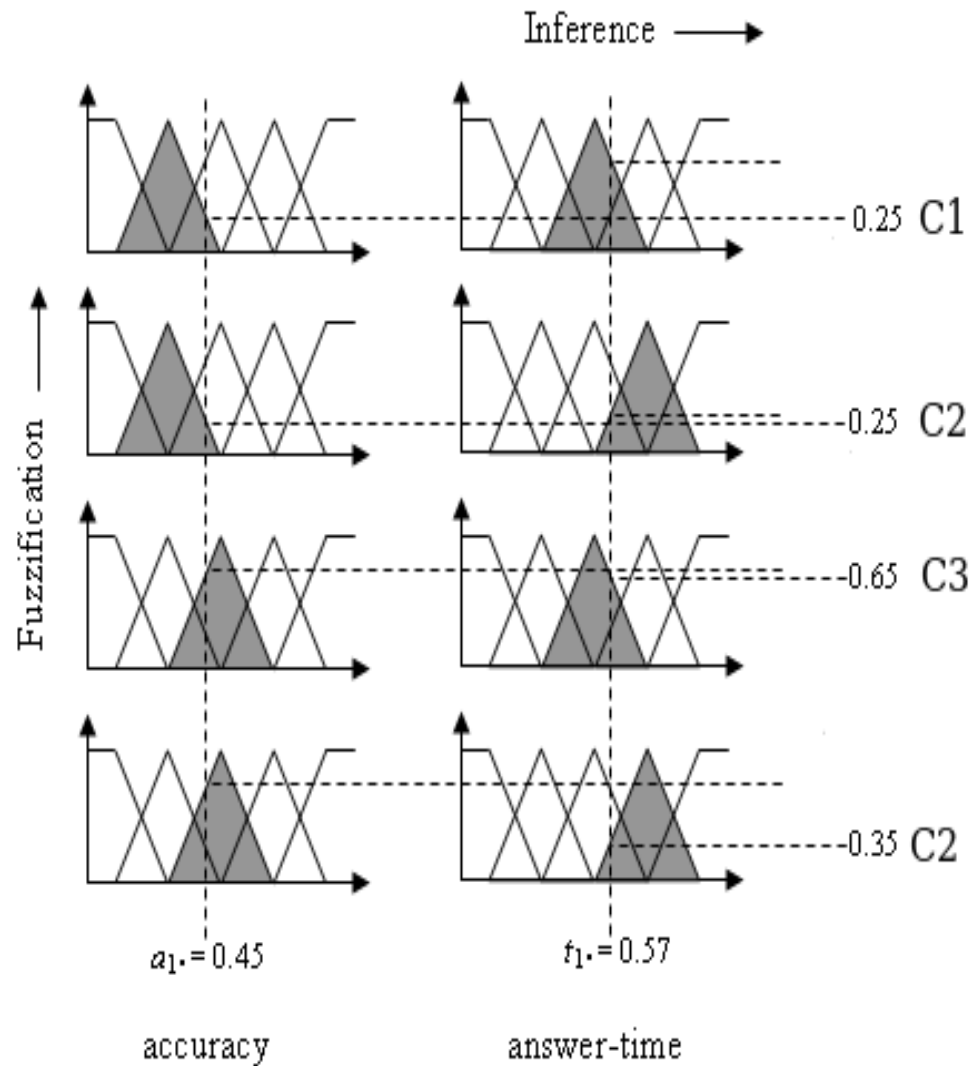
1 = Iris-virginica

2 = Iris-versicolor

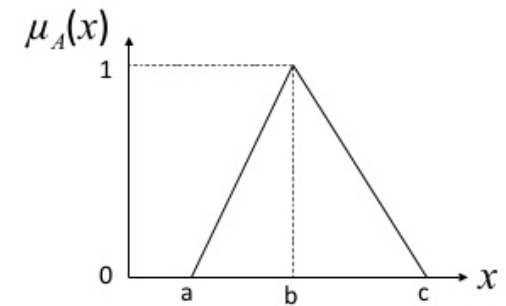
BEA - Ejemplo

```
@relation iris
@attribute SepalLength real [4.3, 7.9]
@attribute SepalWidth real [2.0, 4.4]
@attribute PetalLength real [1.0, 6.9]
@attribute PetalWidth real [0.1, 2.5]
@attribute Class {Iris-setosa, Iris-versicolor, Iris-virginica}
@inputs SepalLength, SepalWidth, PetalLength, PetalWidth
@outputs Class
@data
5.1, 3.5, 1.4, 0.2, Iris-setosa
4.9, 3.0, 1.4, 0.2, Iris-setosa
4.6, 3.1, 1.5, 0.2, Iris-setosa
5.0, 3.6, 1.4, 0.2, Iris-setosa
5.4, 3.9, 1.7, 0.4, Iris-setosa
5.0, 3.4, 1.5, 0.2, Iris-setosa
4.4, 2.9, 1.4, 0.2, Iris-setosa
4.8, 3.4, 1.6, 0.2, Iris-setosa
4.8, 3.0, 1.4, 0.1, Iris-setosa
4.3, 3.0, 1.1, 0.1, Iris-setosa
5.7, 4.4, 1.5, 0.4, Iris-setosa
5.4, 3.9, 1.3, 0.4, Iris-setosa
5.1, 3.5, 1.4, 0.3, Iris-setosa
5.4, 3.4, 1.7, 0.2, Iris-setosa
5.1, 3.7, 1.5, 0.4, Iris-setosa
4.6, 3.6, 1.0, 0.2, Iris-setosa
5.1, 3.3, 1.7, 0.5, Iris-setosa
4.8, 3.4, 1.9, 0.2, Iris-setosa
5.0, 3.0, 1.6, 0.2, Iris-setosa
5.0, 3.4, 1.6, 0.4, Iris-setosa
5.2, 3.5, 1.5, 0.2, Iris-setosa
5.2, 3.4, 1.4, 0.2, Iris-setosa
4.7, 3.2, 1.6, 0.2, Iris-setosa
4.8, 3.1, 1.6, 0.2, Iris-setosa
5.4, 3.4, 1.5, 0.4, Iris-setosa
5.2, 4.1, 1.5, 0.1, Iris-setosa
5.5, 4.2, 1.4, 0.2, Iris-setosa
```

BEA - Ejemplo



$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if } b \leq x \leq c \\ 0 & \text{if } x \geq c \end{cases}$$



BEA - Ejemplo

```
Iteración 1-----
Clonación
Cromosoma 1$$$$$$
[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]
Tasa Clasificación del Cromosoma: 0.36666667461395264
[19, 7, 8, 3, 23, 11, 21, 5, 12, 1, 4, 25, 9, 16, 20, 15, 17, 18, 22, 6, 14, 13, 10, 2, 24]
Posición: 19
[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 0, 0, 3, 3, 3, 3, 0]
Tasa Clasificación del Clon: 0.36666667461395264
[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 4, 0, 3, 3, 3, 3, 0]
Tasa Clasificación del Clon: 0.36666667461395264
[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 0, 0, 3, 3, 3, 3, 0]
Tasa Clasificación del Clon: 0.36666667461395264
[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 4, 0, 3, 3, 3, 3, 0]
Tasa Clasificación del Clon: 0.36666667461395264
[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 2, 0, 3, 3, 3, 3, 0]
Tasa Clasificación del Clon: 0.36666667461395264
Cromosoma en la Población
[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]
-----
```

BEA - Ejemplo

Posicion: 7

[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.36666667461395264

[1, 2, 4, 3, 1, 4, 1, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.36666667461395264

[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.36666667461395264

[1, 2, 4, 3, 1, 4, 4, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.3333333432674408

[1, 2, 4, 3, 1, 4, 0, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.3333333432674408

Cromosoma en la Población

[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

BEA - Ejemplo

Posicion: 3

[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.36666667461395264

[1, 2, 3, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.4166666567325592

[1, 2, 0, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.36666667461395264

[1, 2, 3, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.4166666567325592

[1, 2, 4, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

Tasa Clasificación del Clon: 0.36666667461395264

Cromosoma en la Población

[1, 2, 3, 3, 1, 4, 2, 4, 4, 2, 3, 1, 3, 0, 1, 4, 0, 0, 3, 0, 3, 3, 3, 3, 0]

BEA - Ejemplo

Cromosoma 2\$\$\$\$\$\$

[1, 3, 3, 1, 1, 0, 3, 0, 2, 0, 4, 1, 4, 4, 2, 3, 3, 0, 2, 2, 3, 1, 3, 0, 1]

Tasa Clasificación del Cromosoma: 0.36666667461395264

[15, 24, 18, 7, 25, 4, 12, 22, 13, 16, 8, 10, 20, 21, 11, 1, 2, 3, 17, 23, 9, 5, 6, 19, 14]

Posicion: 15

[1, 3, 3, 1, 1, 0, 3, 0, 2, 0, 4, 1, 4, 4, 0, 3, 3, 0, 2, 2, 3, 1, 3, 0, 1]

Tasa Clasificación del Clon: 0.3333333432674408

[1, 3, 3, 1, 1, 0, 3, 0, 2, 0, 4, 1, 4, 4, 2, 3, 3, 0, 2, 2, 3, 1, 3, 0, 1]

Tasa Clasificación del Clon: 0.36666667461395264

[1, 3, 3, 1, 1, 0, 3, 0, 2, 0, 4, 1, 4, 4, 0, 3, 3, 0, 2, 2, 3, 1, 3, 0, 1]

Tasa Clasificación del Clon: 0.3333333432674408

[1, 3, 3, 1, 1, 0, 3, 0, 2, 0, 4, 1, 4, 4, 2, 3, 3, 0, 2, 2, 3, 1, 3, 0, 1]

Tasa Clasificación del Clon: 0.36666667461395264

[1, 3, 3, 1, 1, 0, 3, 0, 2, 0, 4, 1, 4, 4, 1, 3, 3, 0, 2, 2, 3, 1, 3, 0, 1]

Tasa Clasificación del Clon: 0.3333333432674408

Cromosoma en la Población

[1, 3, 3, 1, 1, 0, 3, 0, 2, 0, 4, 1, 4, 4, 2, 3, 3, 0, 2, 2, 3, 1, 3, 0, 1]

BEA - Ejemplo

```
*****
Cromosoma 3$$$$$$
[1, 3, 3, 0, 1, 1, 3, 1, 1, 0, 1, 4, 0, 1, 0, 0, 1, 3, 2, 2, 0, 4, 3, 0, 1]
Tasa Clasificación del Cromosoma: 0.34166666865348816
[14, 7, 6, 1, 21, 5, 22, 19, 17, 25, 23, 20, 16, 8, 10, 9, 2, 24, 18, 15, 13, 11, 4, 3, 12]
Posicion: 14
[1, 3, 3, 0, 1, 1, 3, 1, 1, 0, 1, 4, 0, 3, 0, 0, 1, 3, 2, 2, 0, 4, 3, 0, 1]
Tasa Clasificación del Clon: 0.34166666865348816
[1, 3, 3, 0, 1, 1, 3, 1, 1, 0, 1, 4, 0, 4, 0, 0, 1, 3, 2, 2, 0, 4, 3, 0, 1]
Tasa Clasificación del Clon: 0.34166666865348816
[1, 3, 3, 0, 1, 1, 3, 1, 1, 0, 1, 4, 0, 0, 0, 0, 1, 3, 2, 2, 0, 4, 3, 0, 1]
Tasa Clasificación del Clon: 0.34166666865348816
[1, 3, 3, 0, 1, 1, 3, 1, 1, 0, 1, 4, 0, 1, 0, 0, 1, 3, 2, 2, 0, 4, 3, 0, 1]
Tasa Clasificación del Clon: 0.34166666865348816
[1, 3, 3, 0, 1, 1, 3, 1, 1, 0, 1, 4, 0, 2, 0, 0, 1, 3, 2, 2, 0, 4, 3, 0, 1]
Tasa Clasificación del Clon: 0.34166666865348816
Cromosoma en la Población
[1, 3, 3, 0, 1, 1, 3, 1, 1, 0, 1, 4, 0, 1, 0, 0, 1, 3, 2, 2, 0, 4, 3, 0, 1]
-----
```

BEA - Ejemplo

```
*****
Cromosoma 4$$$$$$
[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 1, 2, 2, 2, 1, 4, 2, 2, 0, 1, 3, 2, 2, 3, 2]
Tasa Clasificación del Cromosoma: 0.5166666507720947
[15, 20, 18, 23, 12, 9, 17, 25, 19, 22, 3, 16, 21, 1, 13, 5, 2, 14, 4, 7, 24, 11, 6, 10, 8]
Posicion: 15
[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 1, 2, 2, 2, 1, 4, 2, 2, 0, 1, 3, 2, 2, 3, 2]
Tasa Clasificación del Clon: 0.5166666507720947
[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 1, 2, 2, 2, 2, 4, 2, 2, 0, 1, 3, 2, 2, 3, 2]
Tasa Clasificación del Clon: 0.4333333373069763
[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 1, 2, 2, 2, 0, 4, 2, 2, 0, 1, 3, 2, 2, 3, 2]
Tasa Clasificación del Clon: 0.4166666567325592
[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 1, 2, 2, 2, 2, 4, 2, 2, 0, 1, 3, 2, 2, 3, 2]
Tasa Clasificación del Clon: 0.4333333373069763
[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 1, 2, 2, 2, 2, 4, 2, 2, 0, 1, 3, 2, 2, 3, 2]
Tasa Clasificación del Clon: 0.4333333373069763
Cromosoma en la Población
[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 1, 2, 2, 2, 1, 4, 2, 2, 0, 1, 3, 2, 2, 3, 2]
-----
```


BEA - Ejemplo

Cromosoma 5\$\$\$\$\$\$

[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 4, 3, 4, 0, 2, 2, 3, 3, 2, 3, 2, 4, 3, 2]

Tasa Clasificación del Cromosoma: 0.6083333492279053

[17, 7, 23, 24, 25, 18, 9, 15, 6, 20, 8, 10, 1, 14, 13, 21, 4, 16, 11, 3, 12, 22, 5, 19, 2]

Posicion: 17

[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 4, 3, 4, 0, 2, 2, 3, 3, 2, 3, 2, 4, 3, 2]

Tasa Clasificación del Clon: 0.6083333492279053

[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 4, 3, 4, 0, 2, 4, 3, 3, 2, 3, 2, 4, 3, 2]

Tasa Clasificación del Clon: 0.4749999940395355

[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 4, 3, 4, 0, 2, 1, 3, 3, 2, 3, 2, 4, 3, 2]

Tasa Clasificación del Clon: 0.6083333492279053

[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 4, 3, 4, 0, 2, 1, 3, 3, 2, 3, 2, 4, 3, 2]

Tasa Clasificación del Clon: 0.6083333492279053

[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 4, 3, 4, 0, 2, 0, 3, 3, 2, 3, 2, 4, 3, 2]

Tasa Clasificación del Clon: 0.49166667461395264

Cromosoma en la Población

[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 4, 3, 4, 0, 2, 2, 3, 3, 2, 3, 2, 4, 3, 2]

BEA - Ejemplo

```
-----  
*****  
Cromosoma 9$$$$$$  
[2, 0, 2, 2, 1, 4, 1, 4, 3, 2, 0, 1, 2, 4, 1, 3, 0, 1, 4, 2, 3, 1, 3, 3, 2]  
Tasa Clasificación del Cromosoma: 0.550000011920929  
[15, 4, 3, 1, 14, 20, 22, 2, 17, 5, 12, 8, 13, 6, 7, 19, 18, 25, 23, 9, 10, 16, 24, 21, 11]  
Posicion: 15  
[2, 0, 2, 2, 1, 4, 1, 4, 3, 2, 0, 1, 2, 4, 1, 3, 0, 1, 4, 2, 3, 1, 3, 3, 2]  
Tasa Clasificación del Clon: 0.550000011920929  
[2, 0, 2, 2, 1, 4, 1, 4, 3, 2, 0, 1, 2, 4, 0, 3, 0, 1, 4, 2, 3, 1, 3, 3, 2]  
Tasa Clasificación del Clon: 0.550000011920929  
[2, 0, 2, 2, 1, 4, 1, 4, 3, 2, 0, 1, 2, 4, 0, 3, 0, 1, 4, 2, 3, 1, 3, 3, 2]  
Tasa Clasificación del Clon: 0.550000011920929  
[2, 0, 2, 2, 1, 4, 1, 4, 3, 2, 0, 1, 2, 4, 0, 3, 0, 1, 4, 2, 3, 1, 3, 3, 2]  
Tasa Clasificación del Clon: 0.550000011920929  
[2, 0, 2, 2, 1, 4, 1, 4, 3, 2, 0, 1, 2, 4, 0, 3, 0, 1, 4, 2, 3, 1, 3, 3, 2]  
Tasa Clasificación del Clon: 0.550000011920929  
Cromosoma en la Población  
[2, 0, 2, 2, 1, 4, 1, 4, 3, 2, 0, 1, 2, 4, 1, 3, 0, 1, 4, 2, 3, 1, 3, 3, 2]  
-----
```

BEA - Ejemplo

```
Nueva Población *****
[3, 0, 3, 3, 1, 3, 3, 4, 4, 2, 1, 1, 2, 2, 1, 4, 0, 0, 3, 0, 3, 2, 3, 3, 2]
Tasa Clasificación: 0.7583333253860474
[3, 1, 3, 3, 2, 0, 3, 0, 2, 0, 2, 1, 3, 3, 2, 3, 3, 0, 2, 2, 2, 1, 2, 2, 1]
Tasa Clasificación: 0.8166666626930237
[1, 3, 3, 2, 1, 1, 3, 1, 1, 0, 1, 4, 0, 1, 0, 2, 1, 3, 2, 2, 0, 4, 3, 0, 1]
Tasa Clasificación: 0.4333333373069763
[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 2, 1, 2, 2, 1, 4, 2, 2, 0, 1, 3, 2, 3, 3, 2]
Tasa Clasificación: 0.800000011920929
[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 4, 3, 4, 0, 2, 2, 3, 3, 2, 3, 2, 4, 3, 2]
Tasa Clasificación: 0.6083333492279053
[0, 1, 2, 0, 0, 3, 2, 3, 3, 2, 1, 1, 2, 2, 1, 2, 1, 3, 3, 2, 1, 4, 2, 1, 0]
Tasa Clasificación: 0.7749999761581421
[2, 1, 3, 2, 2, 2, 2, 3, 3, 2, 1, 3, 2, 4, 2, 1, 3, 1, 3, 0, 3, 4, 1, 1, 2]
Tasa Clasificación: 0.5833333134651184
[2, 0, 3, 0, 0, 2, 2, 2, 2, 1, 4, 2, 0, 0, 1, 2, 1, 2, 2, 1, 1, 0, 1, 2, 1]
Tasa Clasificación: 0.6583333611488342
[2, 1, 2, 2, 1, 3, 1, 4, 3, 2, 0, 1, 2, 2, 1, 3, 0, 1, 4, 2, 3, 2, 3, 3, 2]
Tasa Clasificación: 0.824999988079071
[3, 2, 3, 4, 2, 2, 2, 3, 3, 2, 3, 4, 4, 0, 1, 2, 1, 2, 2, 1, 3, 0, 3, 4, 0]
Tasa Clasificación: 0.8583333492279053|
```

BEA - Ejemplo

Orden de la Reglas:

10

9

2

4

6

1

8

5

7

3

Cantidad de Cambios: 10

Cambio - Regla Superior: 6, Regla Inferior: 3, Posicion: 7

Cambio - Regla Superior: 4, Regla Inferior: 8, Posicion: 20

Cambio - Regla Superior: 4, Regla Inferior: 8, Posicion: 2

Cambio - Regla Superior: 4, Regla Inferior: 5, Posicion: 12

Cambio - Regla Superior: 10, Regla Inferior: 7, Posicion: 5

Cambio - Regla Superior: 2, Regla Inferior: 1, Posicion: 22

Cambio - Regla Superior: 10, Regla Inferior: 1, Posicion: 3

Cambio - Regla Superior: 4, Regla Inferior: 8, Posicion: 20

Cambio - Regla Superior: 6, Regla Inferior: 8, Posicion: 8

Cambio - Regla Superior: 6, Regla Inferior: 1, Posicion: 12

BEA - Ejemplo

Nueva Población &&&&&&&&&&&&&

```
[3, 0, 3, 3, 1, 3, 3, 4, 4, 2, 1, 1, 2, 2, 1, 4, 0, 0, 3, 0, 3, 1, 3, 3, 2]
```

Tasa Clasificación: 0.708333134651184

```
[3, 1, 3, 3, 2, 0, 3, 0, 2, 0, 2, 1, 3, 3, 2, 3, 3, 0, 2, 2, 2, 1, 2, 2, 1]
```

Tasa Clasificación: 0.8166666626930237

[1, 3, 3, 2, 1, 1, 2, 1, 1, 0, 1, 4, 0, 1, 0, 2, 1, 3, 2, 2, 0, 4, 3, 0, 1]

Tasa Clasificación: 0.4333333373069763

```
[0, 4, 3, 0, 2, 4, 3, 1, 3, 0, 2, 1, 2, 2, 1, 4, 2, 2, 0, 1, 3, 2, 3, 3, 2]
```

Tasa Clasificación: 0.800000011920929

```
[4, 4, 1, 2, 1, 0, 0, 4, 4, 1, 4, 1, 3, 4, 0, 2, 2, 3, 3, 2, 3, 2, 4, 3, 2]
```

Tasa Clasificación: 0.6000000238418579

```
[0, 1, 2, 0, 0, 3, 2, 3, 3, 2, 1, 1, 2, 2, 1, 2, 1, 3, 3, 2, 1, 4, 2, 1, 0]
```

Tasa Clasificación: 0.7749999761581421

```
[2, 1, 3, 2, 2, 2, 2, 3, 3, 2, 1, 3, 2, 4, 2, 1, 3, 1, 3, 0, 3, 4, 1, 1, 2]
```

Tasa Clasificación: 0.5833333134651184

```
[2, 4, 3, 0, 0, 2, 2, 3, 2, 1, 4, 2, 0, 0, 1, 2, 1, 2, 2, 1, 1, 0, 1, 2, 1]
```

Tasa Clasificación: 0.6583333611488342

```
[2, 1, 2, 2, 1, 3, 1, 4, 3, 2, 0, 1, 2, 2, 1, 3, 0, 1, 4, 2, 3, 2, 3, 3, 2]
```

Tasa Clasificación: 0.824999988079071

```
[3, 2, 3, 4, 2, 2, 2, 3, 3, 2, 3, 4, 4, 0, 1, 2, 1, 2, 2, 1, 3, 0, 3, 4, 0]
```

Tasa Clasificación: 0.8583333492279053

BEA - Ejemplo

Población Final &&&&&&&&&&

[2, 2, 3, 2, 1, 3, 2, 3, 4, 2, 2, 1, 2, 2, 1, 4, 0, 0, 3, 0, 2, 1, 3, 3, 2]

Tasa Clasificación: 0.925000011920929

[3, 2, 2, 2, 1, 3, 3, 0, 3, 2, 3, 2, 3, 3, 2, 3, 3, 0, 2, 2, 2, 1, 2, 2, 1]

Tasa Clasificación: 0.8416666388511658

[2, 1, 2, 2, 1, 1, 1, 3, 1, 0, 2, 2, 3, 2, 1, 2, 2, 2, 3, 2, 1, 4, 3, 0, 1]

Tasa Clasificación: 0.75

```
[0, 4, 3, 0, 1, 3, 1, 4, 3, 2, 2, 1, 2, 2, 1, 1, 1, 2, 2, 0, 2, 2, 3, 3, 2]
```

Tasa Clasificación: 0.7916666865348816

[2, 4, 1, 2, 0, 3, 3, 3, 4, 2, 4, 1, 2, 4, 0, 2, 1, 3, 3, 2, 3, 2, 3, 3, 2]

Tasa Clasificación: 0.6583333611488342

[0, 1, 2, 0, 0, 3, 2, 3, 4, 2, 2, 1, 2, 2, 1, 2, 1, 3, 3, 2, 1, 4, 2, 1, 0]

Tasa Clasificación: 0.8916666507720947

```
[2, 1, 2, 2, 1, 3, 2, 3, 4, 2, 2, 1, 3, 3, 2, 1, 3, 1, 2, 0, 3, 4, 3, 1, 2]
```

Tasa Clasificación: 0.8916666507720947

[2, 4, 2, 0, 1, 3, 2, 3, 3, 2, 4, 2, 0, 0, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 1]

Tasa Clasificación: 0.8416666388511658

[2, 1, 2, 2, 1, 2, 1, 3, 3, 2, 0, 1, 2, 2, 1, 3, 0, 1, 4, 2, 3, 2, 3, 4, 2]

Tasa Clasificación: 0.9083333611488342

```
[3, 2, 3, 4, 2, 2, 1, 3, 3, 2, 3, 4, 4, 0, 1, 2, 1, 2, 2, 1, 3, 0, 3, 4, 0]
```

Tasa Clasificación: 0.8916666507720947

GRACIAS

Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe