

Computación Emergente: Modelos Sociales y aplicaciones

Modelo de enjambre

Optimización por enjambre de partícula (PSO)

- La posición de la partícula i durante la iteración $t+1$, es definida por

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

donde, $x_i(t)$ es la posición actual de la partícula, y $v_i(t+1)$ es la velocidad que lleva la partícula.

$$v_{ij}(t+1) = v_{ji}(t) + c_1 r_{1j}(t) [y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t) [\hat{y}_j(t) - x_{ij}(t)]$$

c_1 y c_2 son constantes de aceleración, que definen la importancia del componente cognitivo y social. respectivamente, r_{1j} y r_{2j} son números aleatorios en el rango $[0, 1]$, y_{ij} es la dimensión j de la mejor posición alcanzada por la partícula i (componente cognitivo), y $\hat{y}_j(t)$ es la dimensión j de la mejor posición global alcanzada por el enjambre hasta la iteración

optimización por enjambre de partícula (PSO)

Crear e inicializar un enjambre n-dimensional;

Repetir hasta alcanzar condición de parada

por cada partícula $i = 1, n$

si $f(x_i) < f(y_i)$ entonces

$$y_i = x_i$$

si $f(y_i) < f()$ entonces

$$\hat{y} = y_i$$

por cada partícula $i = 1, n$

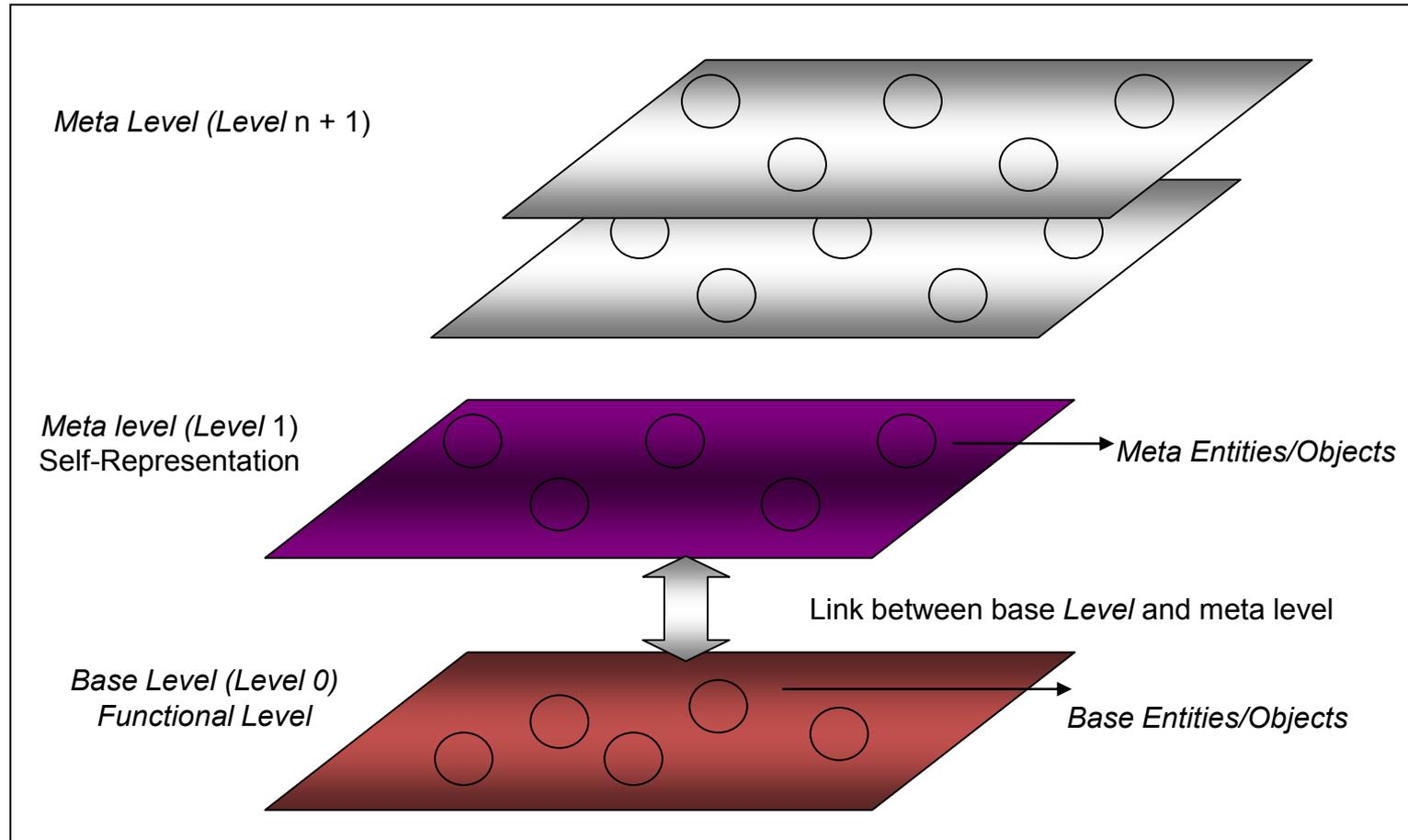
actualizar la velocidad

actualizar la posición

Algoritmo GBEST

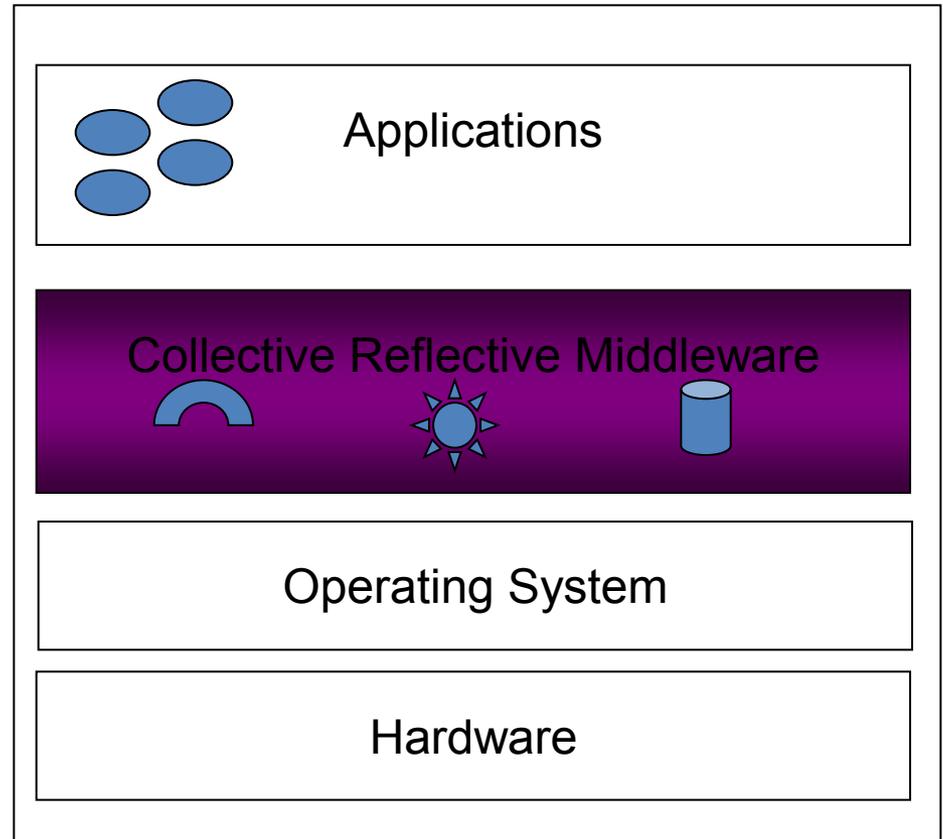
Aplicación 1

Reflection

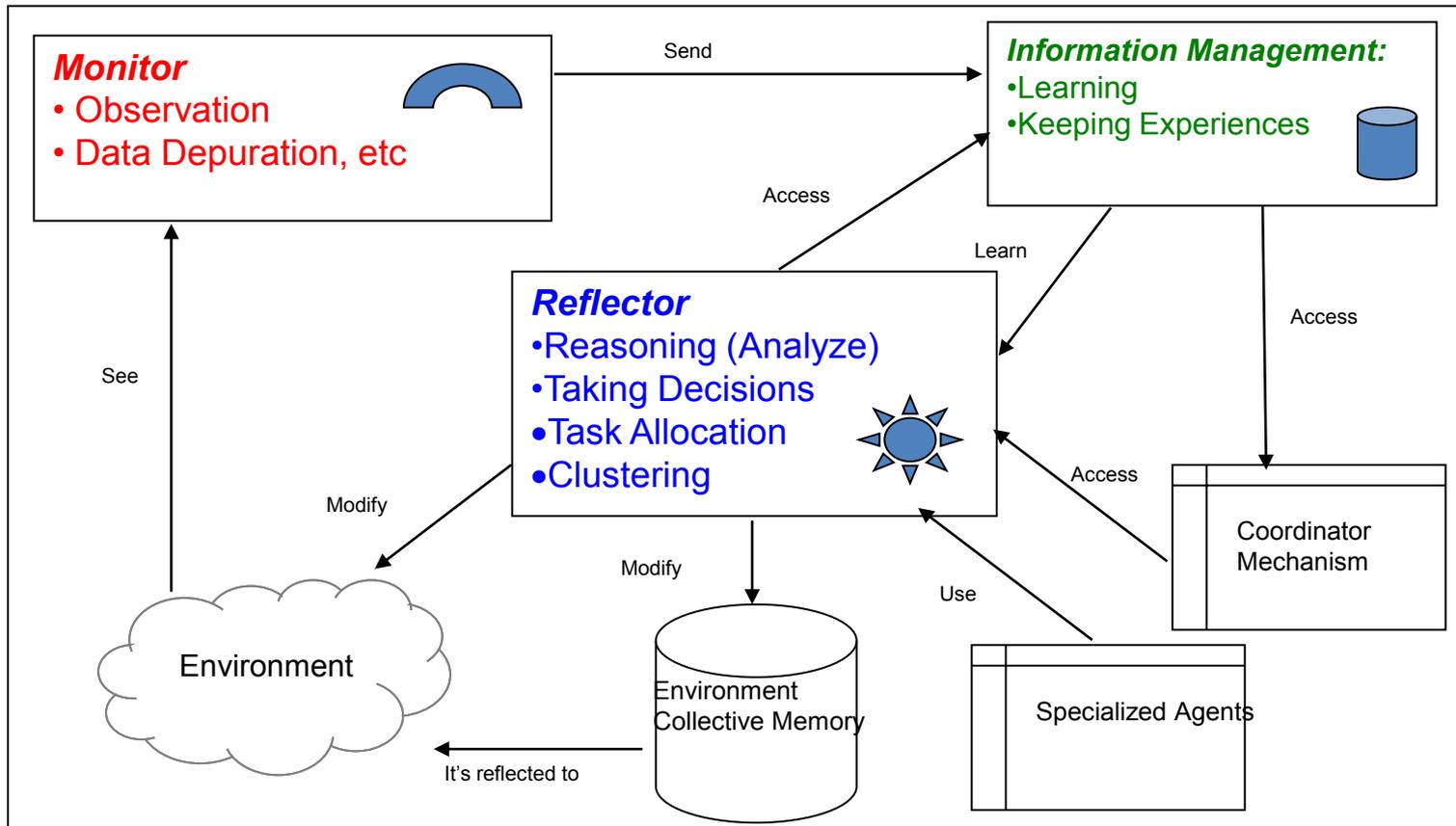


“Collective Reflective Middleware”

- Use of Artificial Collective Intelligence.
- They are inspired in component models.
- They have reflective layers oriented to generate collective introspection (state observation).
- Collective intercession (Model 1).
- Individual intercession (behavior and alteration of itself execution or meaning) is based on collective introspection (Model 2).
- Each software component exists inside a collective.
- Collective goals.
- There is a knowledge base that store present and historic information.

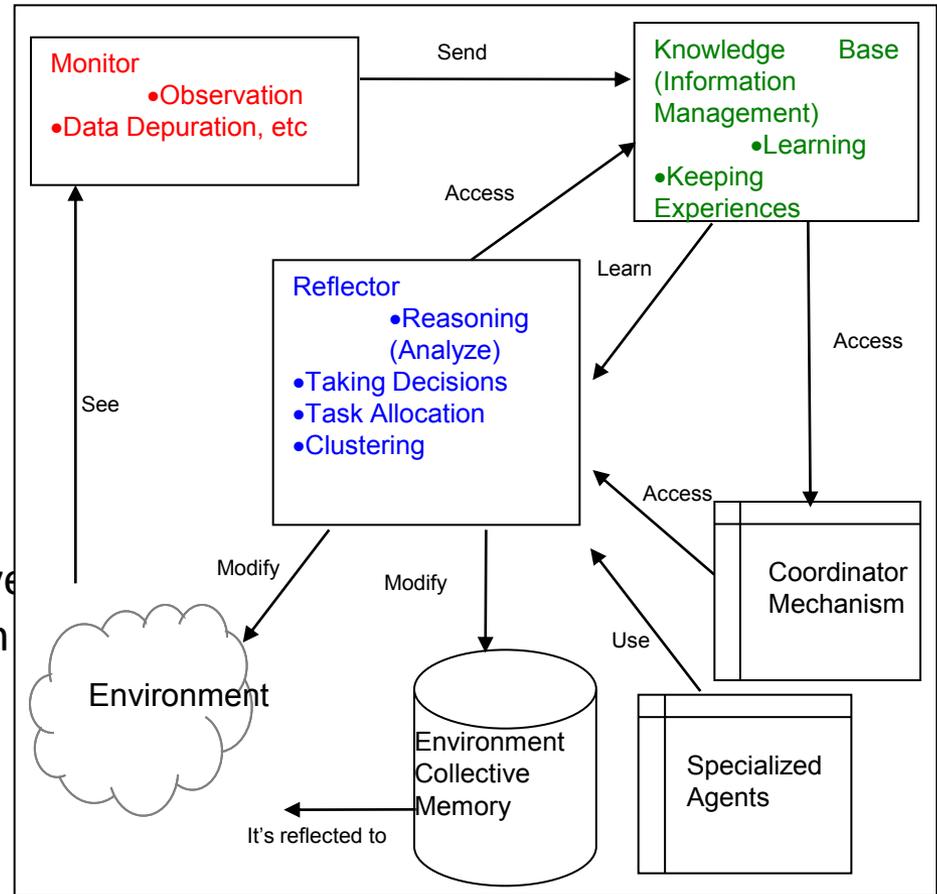


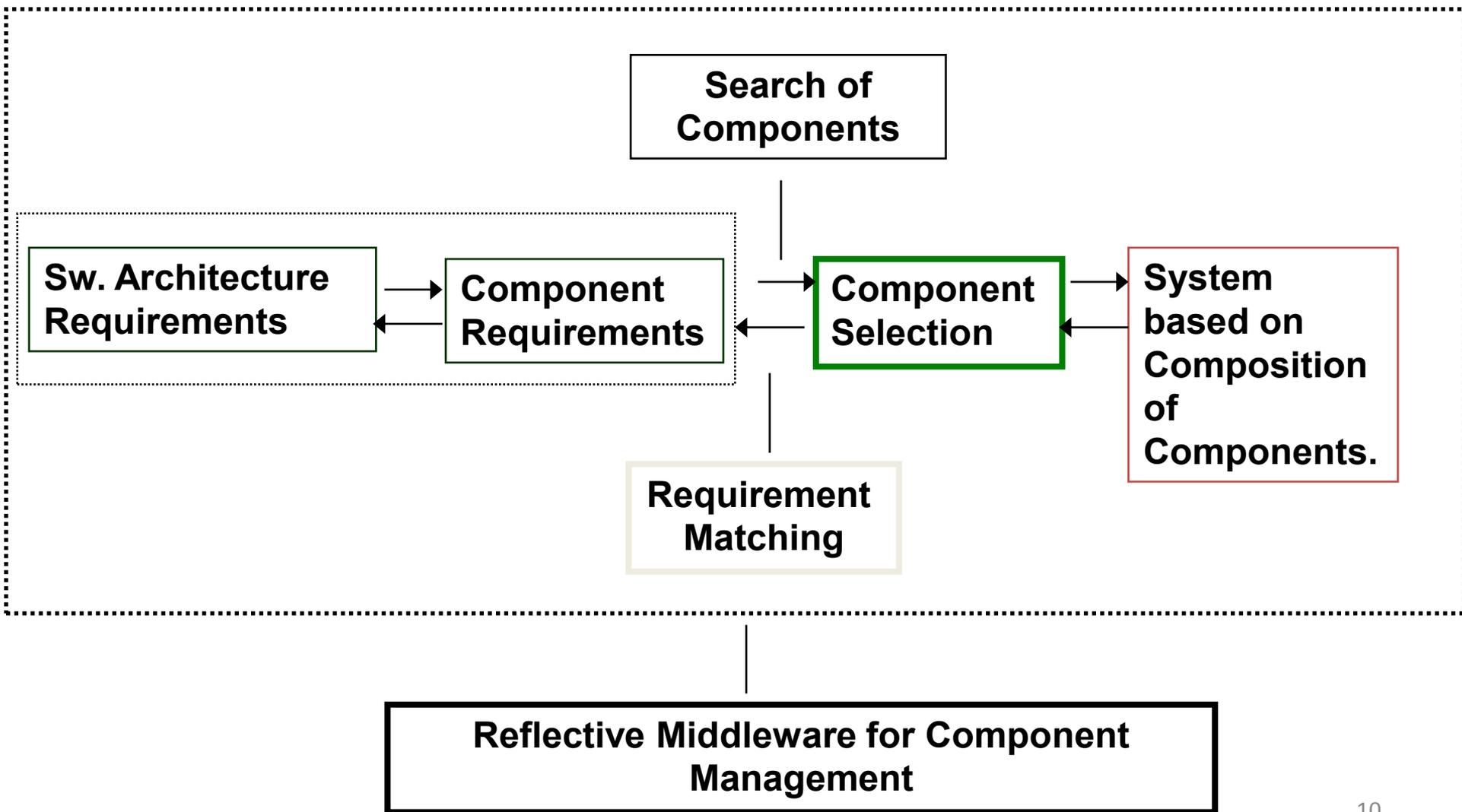
“Collective Reflective Middleware”



Macro Algorithm (Middleware Behavior):

- *Constraints: A set of applications or software components (collective) is supposed to be loaded.*
- Monitor gets information from the collective environment.
- Monitor pre-process information.
- Information is sent to Reflector and to the Information Manager.
- Reflector takes a decision based on previous phase and Information Manager.
- Reflector makes changes in the collective
- Information Manager keeps information about collective performance permanently.
- Information Manager learns from the result of each decision taken



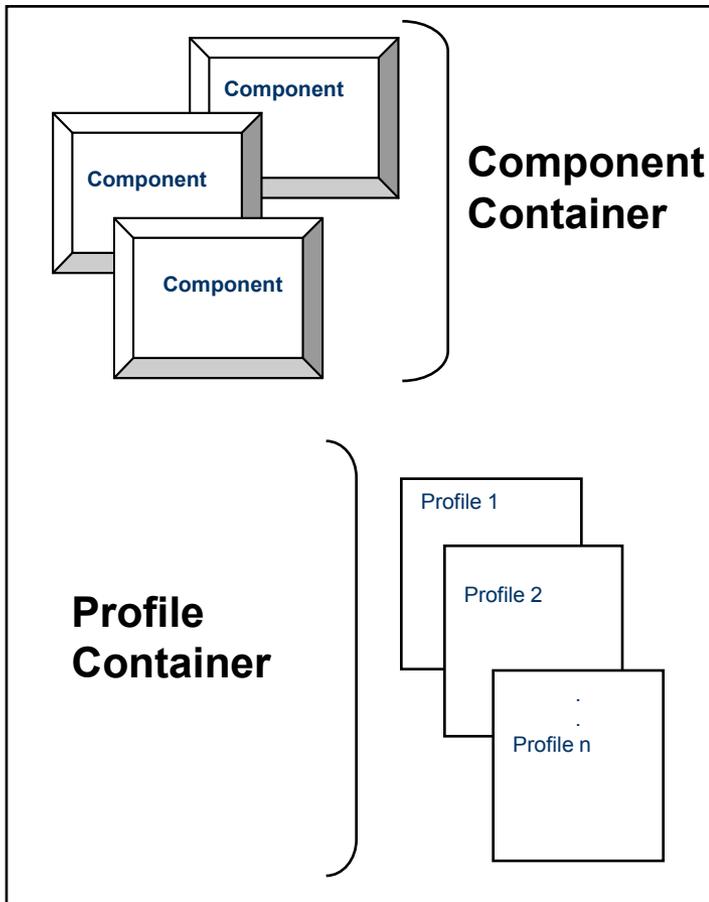


Component Selection Algorithm

This Selection Algorithm is inspired in a collective behavior, the following premises are needed:

- Each component must have a profile associated with its own characteristics. (.xml)
- The end user must give the initial requirements.
- The selection program should be able to update the component profile (its pheromone).
- Selection is based on a probabilistic method.

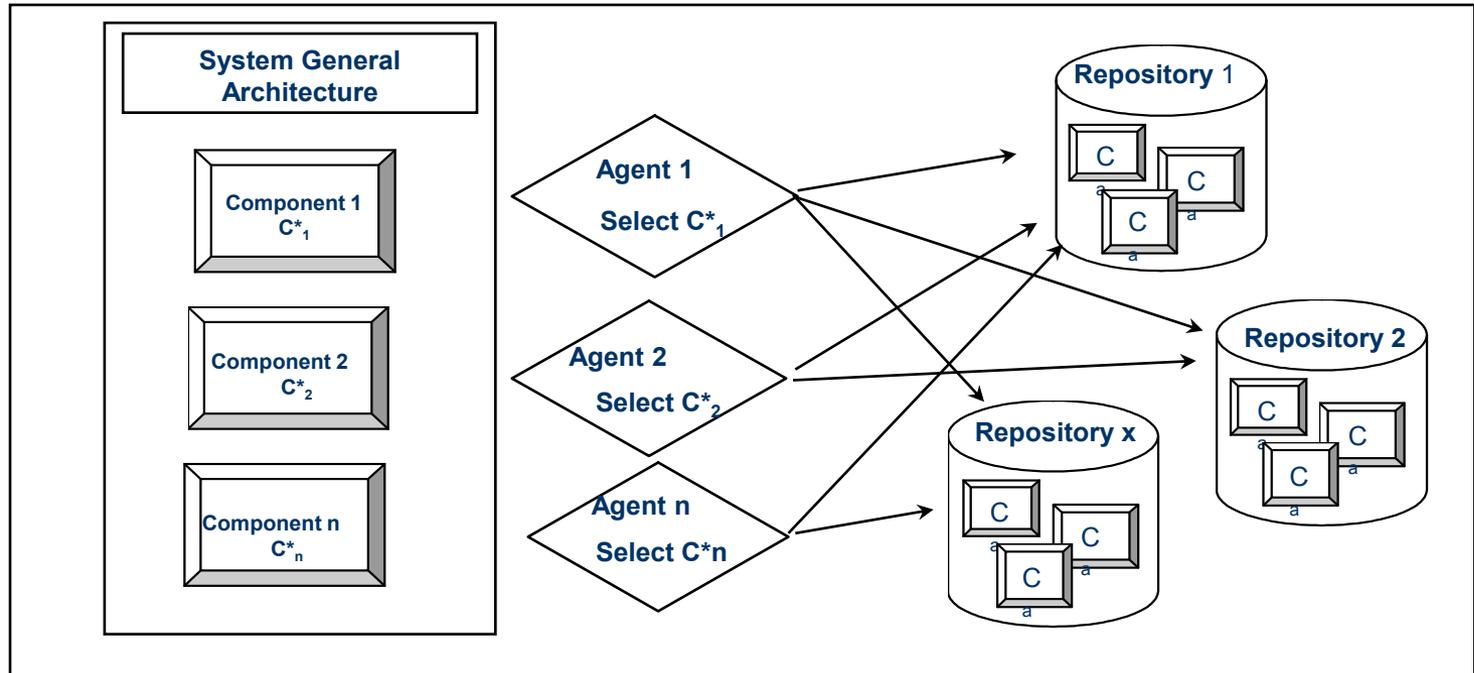
Component Selection Algorithm



```
<componentStaticInformation>
  <componentInformation>
    <uniqueID>SCAM2001</uniqueID>
    <name>Scriptable Application Manager</name>
  </componentStaticInformation>
  <profile1>
    <componentDynamicInformation>
      <pheromone>0.1</pheromone>
      <os>windows</os>
      <ms>3k</ms>
      <et>0.01</et>
    </componentDynamicInformation>
  </profile1>
  <profile2>
    <componentDynamicInformation>
      <pheromone>0.5</pheromone>
      <os>fedora</os>
      <ms>1k</ms>
      <et>0.05</et>
    </componentDynamicInformation>
  </profile2>
```

Component Selection Algorithm

Model A



Algorithm A

1. Create K agents, one for each component to select.
2. For $i = 1$ to K
 - 2.1. Identify possible group of C components to select using (1)
 - 2.2. Selection of component i using (2)
3. Analyze Performance (R) using (4)
4. Update Pheromone using (5) for each component profile of group C.

Component Selection Algorithm

Matching (Equation 1):

$$X_{ljk} = 1 + H_k - N_{ljk}$$

X -> Component Matching

H -> Ideal Component Characteristics

N -> Real Characteristics

j: Component, l: Repository, u: Profile, k: Agent.

Selection Probability (Equation 2):

$$P_{lju}^k(t) = \frac{[Y_{lju}(t)] [X_{lju}^k]^{-1}}{\sum_{rsn} [Y_{rsn}(t)] [X_{rsn}^k]^{-1}}$$

Component Selection Algorithm

Pheromone Update (equation 5)

$$Y_{lju}(t) \begin{cases} (1 - \alpha) * Y_{lju}(t) + \Delta Y_{lju}^k(t) & \text{If the component } lju \text{ is chosen} \\ (1 - \alpha) * Y_{lju}(t) & \text{Otherwise} \end{cases}$$

Where R and $\Delta Y_{lju}^k(t)$ are:

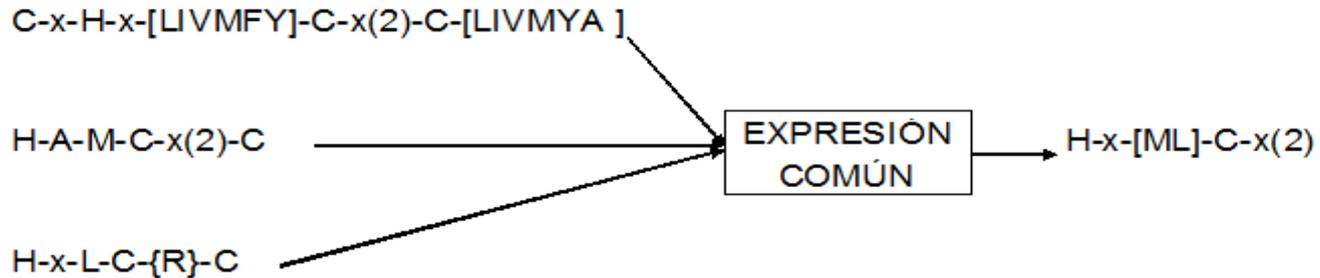
$$R = f(ET, M) \tag{4}$$

$$\Delta Y_{lju}^k(t) = (X_{lju}^k * R)^{-1}$$

Aplicación 2

PROBLEMA

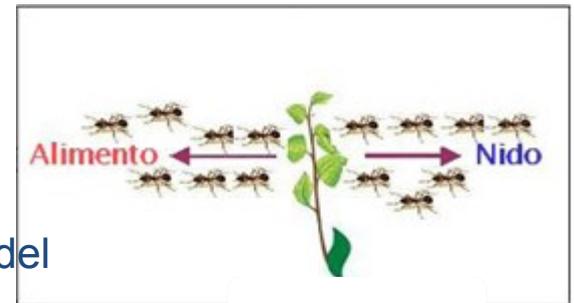
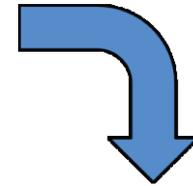
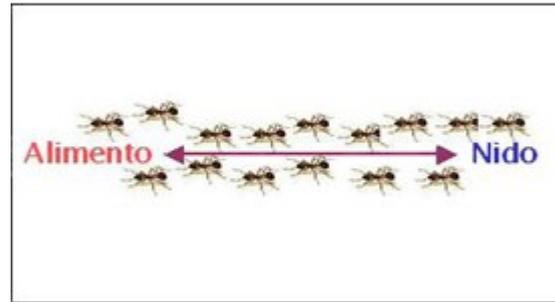
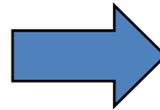
Fusion de Motivos



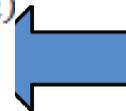
Construir un patrón común para las expresiones regulares que tienen un alto grado de semejanza

FUSION DE MOTIVOS

Colonia de Hormigas



$$P_r^k = \begin{cases} \frac{[\tau_r]^\theta [n_r]^\beta}{\sum_{u \in N_i^k} [\tau_r]^\theta [n_r]^\beta}, & \text{Si } s \in N_k(i) \\ 0, & \text{En otro caso} \end{cases}$$



τ es la cantidad de rastro de feromona en el nodo r del grafo.

η es la visibilidad del nodo r .

θ y β son dos parámetros que ponderan la importancia relativa de los rastros de feromona y la visibilidad.

$$\tau_r(t+1) = \alpha \tau_r(t) + \Delta \tau_r$$

$$\Delta \tau_r = \sum^m \Delta \tau_r^k$$

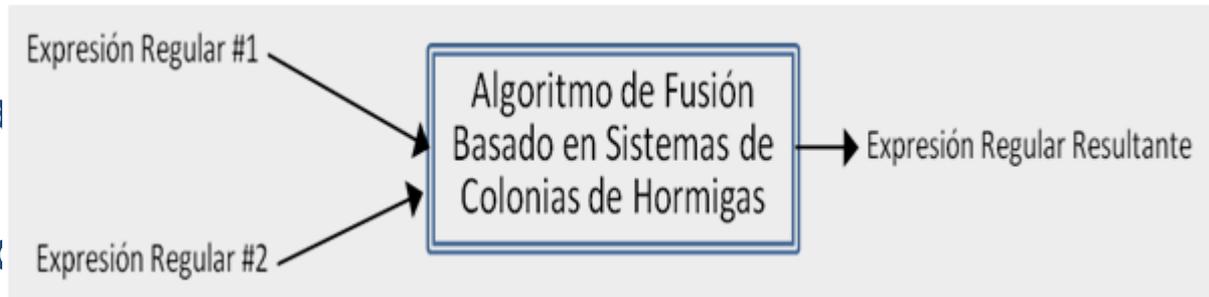
α = coeficiente tal que $(1 - \alpha)$ representa la tasa de evaporación de la feromona entre el intervalo de tiempo t y $t+1$ y $\Delta \tau_r$ = cantidad de feromona dejada en un nodo r



FUSION DE MOTIVOS

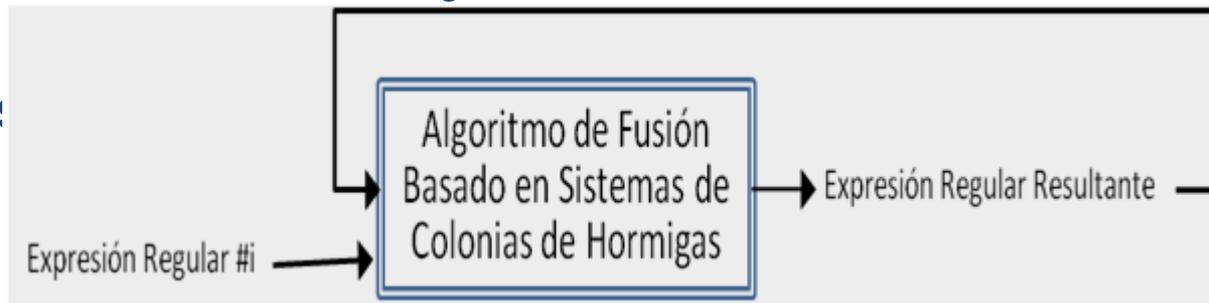
1. Crear

2. Reco



3. Selección de los mejores nodos.

4. Con



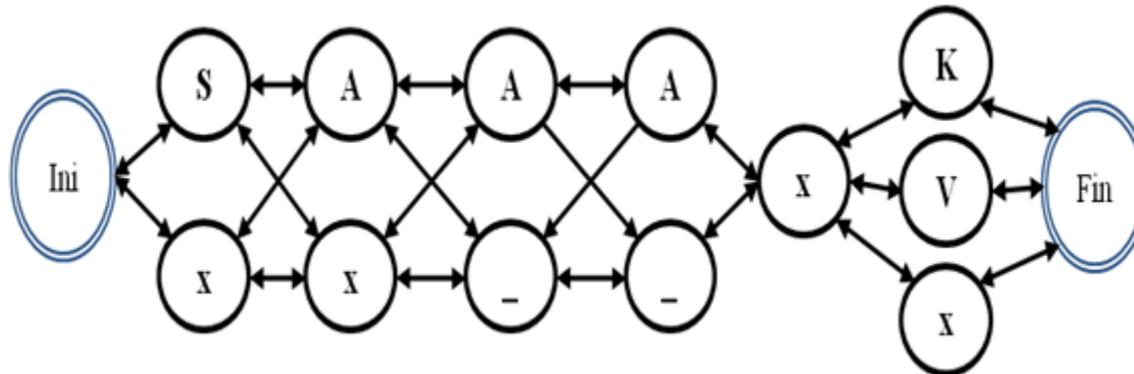
SUB-SISTEMA DE FUSION DE MOTIVOS

Creación del Grafo de Recorrido

Representa el mapa donde los nodos caracterizan cada uno de los aminoácidos que contiene un motivo dado.

Información que se quiere representar	Identificador especial seleccionado	Clasificador para las nuevas Familias
Gap	x	0
Vacio	-	-1
Inicio	Ini	-2
Fin	Fin	-2

S-A(1,3)-x-[KV]

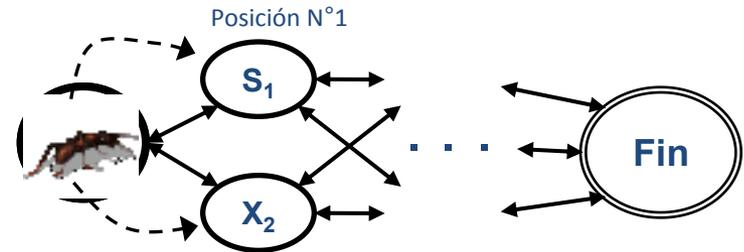


FUSION DE MOTIVOS

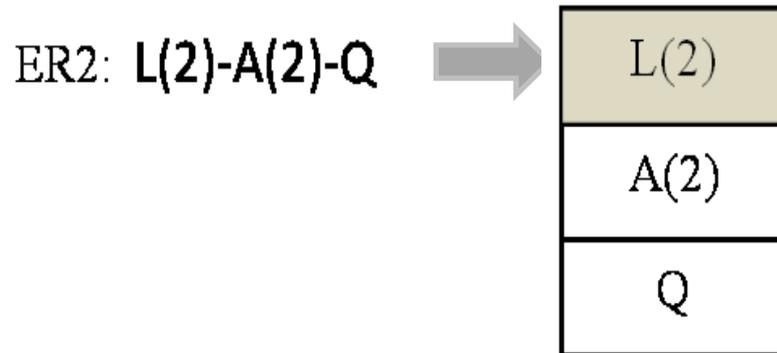
Recorrido de la Colonia de Hormigas

Estructura del TDA Agente Hormiga

- ✓ Nodo Inicial
- ✓ Mapa de Recorrido



Las hormigas realizarán el recorrido del grafo utilizando los aminoácidos del otro motivo.



FUSION DE MOTIVOS

Recorrido de la Colonia de Hormigas

Estructura del TDA Agente Hormiga

- ✓ Nodo Inicial
- ✓ Mapa de Recorrido
- ✓ Índices de Similitud

Indices de Similitudes

	Iguales	<input type="text" value="10"/>
	Familia	<input type="text" value="8"/>
	Diferentes	<input type="text" value="1"/>
	Gaps	<input type="text" value="3"/>

Tipos de Aminoácidos	Aminoácidos pertenecientes a la misma familia (notación de 1 Letra)	Clasificador para las Familias de Aminoácidos
Alifáticos	GALVIM	1
Aromáticos	FYW	2
Básicos	K R H	3
Neutros	S T N Q	4
Ácidos	D E	5
Con Azufre	C	6
Iminoácido	P	7

FUSION DE MOTIVOS

Recorrido de la Colonia de Hormigas

Estructura del TDA Agente Hormiga

- ✓ Nodo Inicial
- ✓ Mapa de Recorrido
- ✓ Índices de Similitud
- ✓ Similitud Aprobatoria

Indica a la hormiga el nivel de similitud mínimo para considerar que el nodo visitado es un hallazgo exitoso.

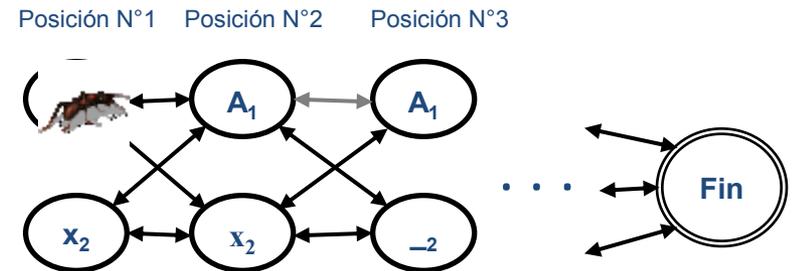
FUSION DE MOTIVOS

Recorrido de la Colonia de Hormigas

Estructura del TDA Agente Hormiga

- ✓ Nodo Inicial
- ✓ Mapa de Recorrido
- ✓ Índices de Similitud
- ✓ Similitud Aprobatoria
- ✓ Coeficiente del Incremento de Feromona

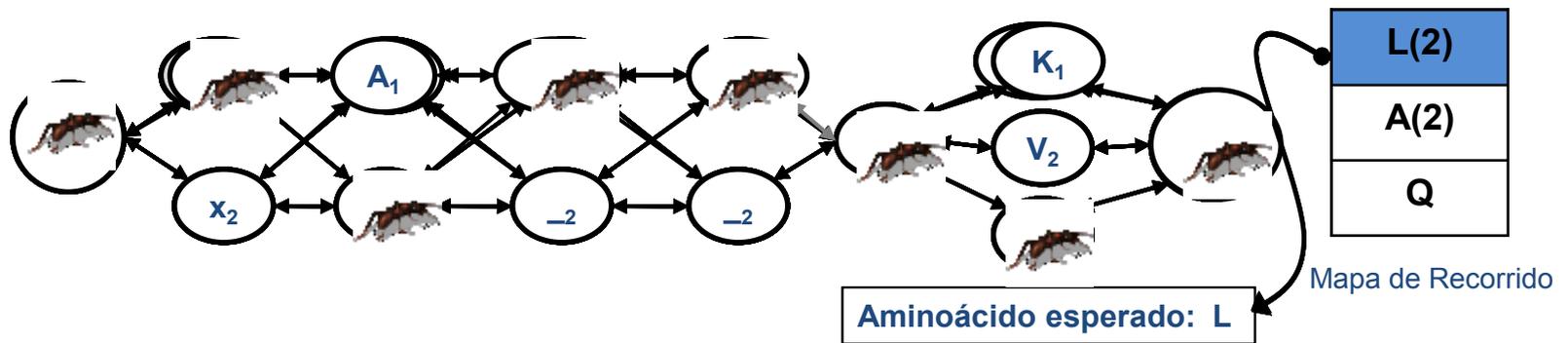
$$\tau_r^k = \tau_r^k + \sigma * \varphi_r^k$$



L(2)
A(2)
Q

FUSION DE MOTIVOS

Recorrido de la Colonia de Hormigas



Para desplazarse desde el punto inicial y el nodo final del gráfico, la hormiga utiliza una función de transición para conocer a que nodo dirigirse en un momento dado.

$$P_r^k = \tau_r^k + \sigma * \varphi_r^k \quad n > 1$$

$$\left(\begin{array}{l} 1 \\ 1 \end{array} \right) \quad \text{si } n = 1$$

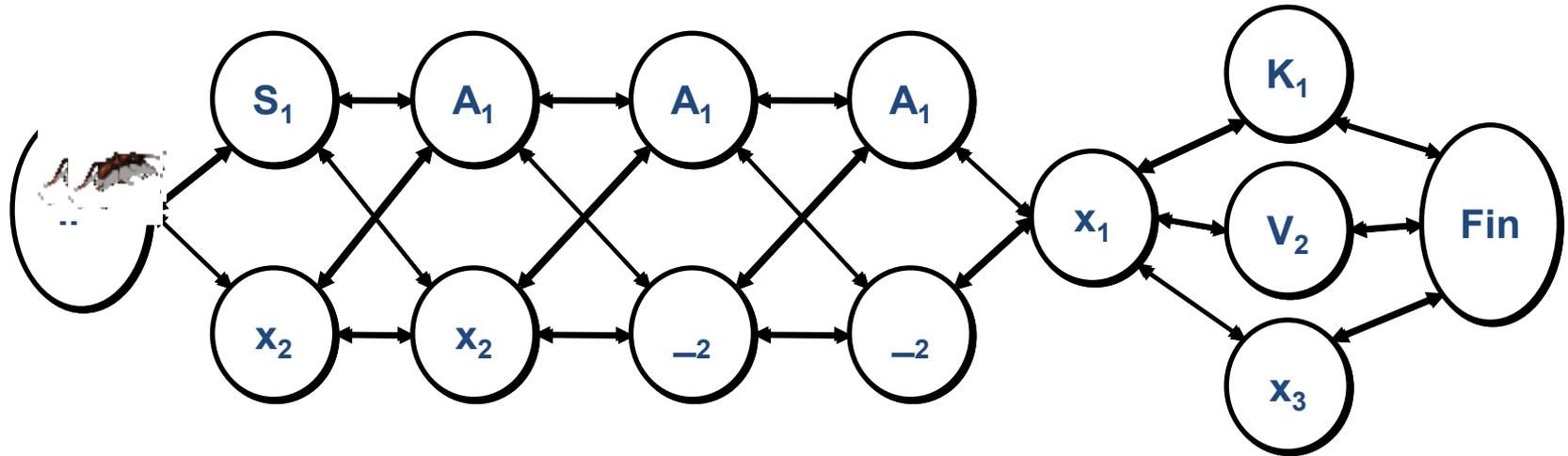
P_r^k = probabilidad actual de feromona
 τ_r^k = coeficiente de feromona
 φ_r^k = índice de similitud



Distribución de Monte Carlo

FUSION DE MOTIVOS

Recorrido de la Colonia de Hormigas



Se realiza la evaporación de la traza (disminución de los niveles de feromona en todos los nodos del grafo)

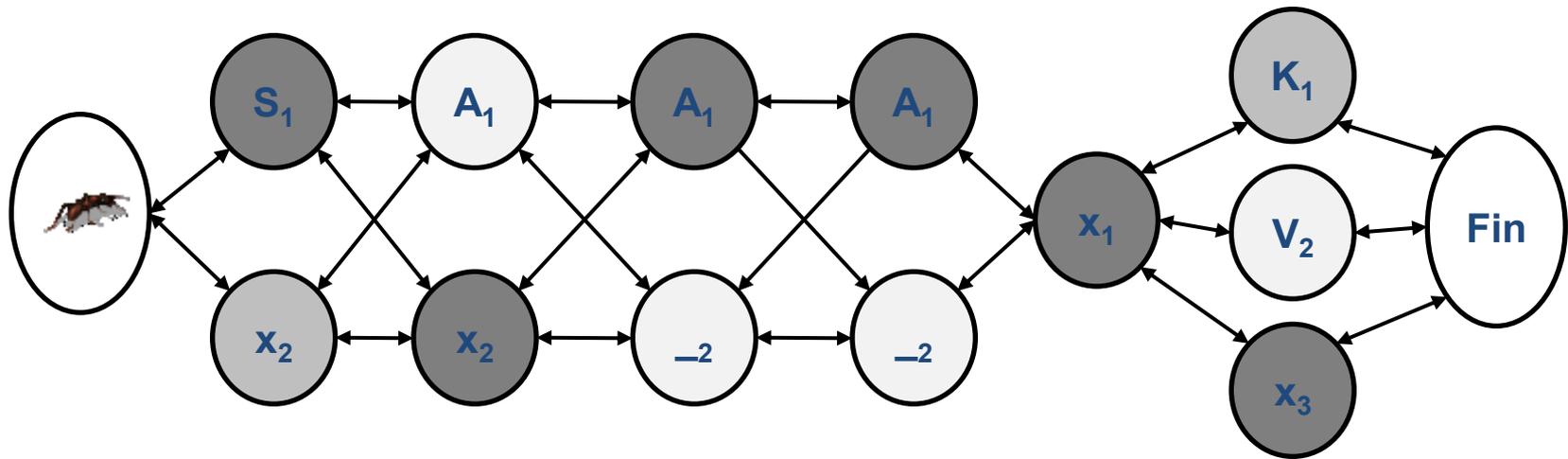
$$\tau_r^k = (1 - \rho) * \tau_r^k$$

ρ = coeficiente de evaporación de feromona

FUSION DE MOTIVOS

Selección de los mejores nodos

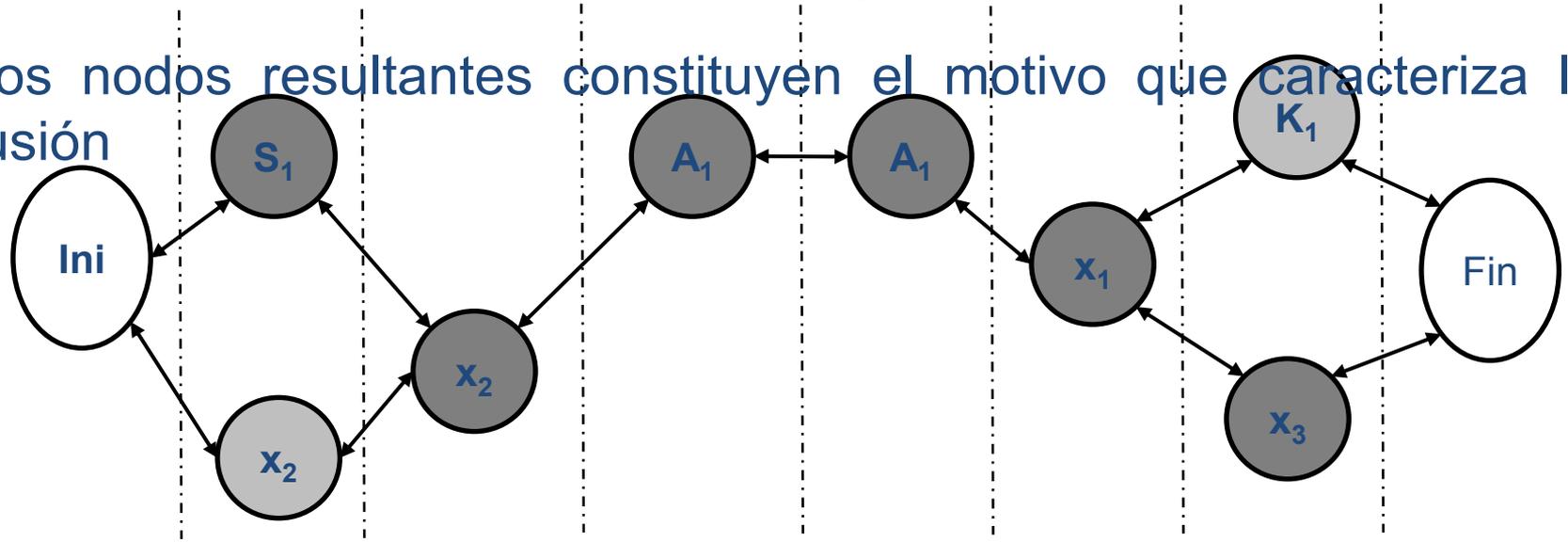
Se fija un valor de umbral para el nivel de feromona y se procede a escoger los mejores nodos



FUSION DE MOTIVOS

Construcción de la Expresión Regular de la Fusión

Los nodos resultantes constituyen el motivo que caracteriza la fusión



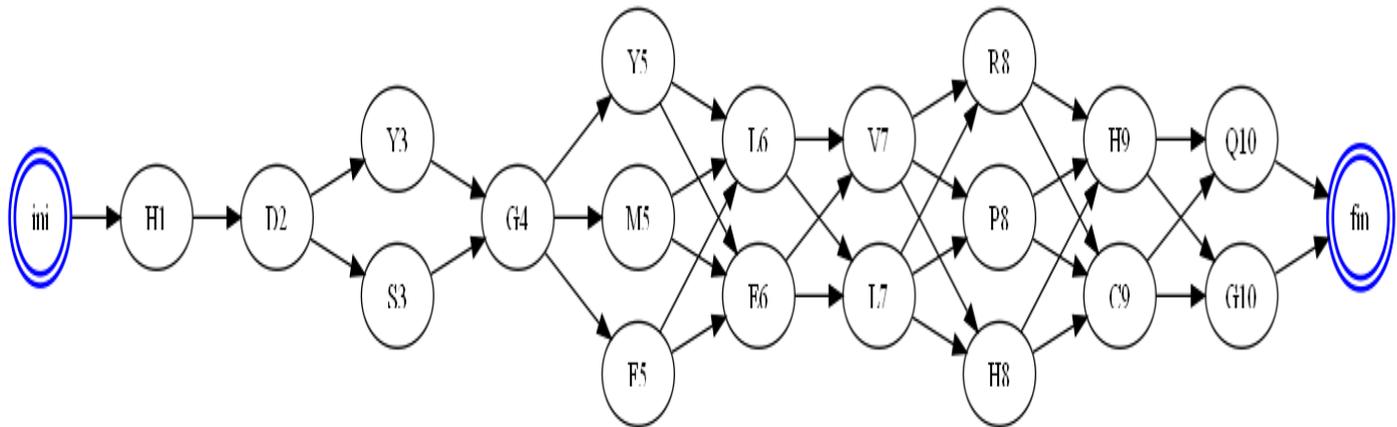
S-x-A(2)-x(2)

PRUEBAS

Fusionar un motivo con un conjunto de motivos

Motivo 1: H-D-[SY]-G-[FMY]-[EL]-[LV]-[HPR]-[CH]-[GQ]

Grafo de Recorrido



Motivo 2: H-D-[GPSTWY]-G-[FILMVY]-[EHIKLMQV]-[AGILV]-
[DEGHKNPQRSTY]-[ACHMTV]-[ACGHNPQST]

Mapa de ruta de las hormigas

PRUEBAS

Fusionar un motivo con un conjunto de motivos

El motivo de fusión resultante es:

H-D-[SY]-G-[FMY]-[EL]-[LV]-[HPR]-[CH]-[GQ]

Familia	Nombre de la Proteína	Organismo
Proteína Precursora Amiloidea (APP)	A4_BOVIN (Q28053)	Bos tauros (Bovino)
Proteína Precursora Amiloidea (APP)	A4_CANFA(Q28280)	Canis familiaris (Perro)
Proteína Precursora Amiloidea (APP)	A4_CAVPO(Q60495)	Cavia porcellus (Cerdo de Guinea)
Proteína Precursora Amiloidea (APP)	A4_FUGRU (O93279)	Fugu rubripes (pez globo japonês)
Proteína Precursora Amiloidea (APP)	A4_HUMAN (P05067)	Homo sapiens (Humano)
Proteína Precursora Amiloidea (APP)	A4_MACFA (P53601)	Macaca fascicularis (Macaco Cangrejero)

Modelos Sociales

Modelos de Preferencias

Filtrado colaborativo

- Se **registra un gran grupo de preferencias** de la gente
- Se crea un subgrupo de gente, cuyas preferencias son similares a las de la persona que busca asesoramiento, utilizando una **métrica de similitud**,
- Se **calcula un promedio** (posiblemente ponderada) de las preferencias para ese subgrupo
- Se usa la función de preferencia resultante, para **recomendar opciones** en las que quien busca ayuda no ha expresado ninguna opinión personal hasta el momento.

Modelos de Preferencias

Ejemplo modelo de preferencia

- Se basa en un protocolo binario (series de votos, con 2 opciones cada una) .
- Se escogen de dos en dos opciones, hasta ir eliminando todas las opciones y quedarse con una.

Protocolo de Borda

- Los agentes escogen todas sus preferencias,
- A partir de allí se desarrolla una función de bienestar social:
Asigna N puntos a la preferencia que ha sido elegida por todos los individuos, y así sucesivamente, siendo N el número de candidatos.

Modelos de Preferencias

Filtrado colaborativo y documentos relacionados

- Generar una **función de preferencia colectiva**,
- Dos documentos son más similares, entre **más usuarios los hayan consultado a los dos**.
- Probabilidad condicional de que un usuario puede consultar x , dado que ese usuario consulto y . $P(x|y)$, determina una matriz M_{xy} , que representa las fuerzas de las conexiones entre los documentos (**co-ocurrencia**).

$$M_{xy} = P(x|y) = \frac{\#(x\&y)}{\#(y)}$$

Donde, $\#(x)$ representa el número total de usuarios que consultaron x , y $\#(x\&y)$ el número total de usuarios que consultaron tanto x como y .

Modelos de Preferencias

El modelo de Confianza Limitada

- Es uno de los **modelos de formación de opinión** más famosos.
- Se basa en que el **agente ajusta su opinión**, cuando el acumulado de una opinión está por debajo de un umbral determinado.
- El modelo de Confianza Limitada es una **forma de comportamiento colectivo** de un gran número de agentes que interactúan, con un objetivo común del grupo
- **Modelo:**
 - población de N agentes, donde cada agente i tiene x_i opiniones, un radio de vista r_i , una función de confianza $w_i(j, t)$ con respecto al agente j , y una función de distancia $d(i, j, t)$, Y un umbral de confianza ε .
 - Cuando cada agente se mueve en el espacio, sólo puede influir en los agentes en su área de vista.

Modelos de Preferencias

El modelo de Confianza Limitada

- Conjunto de los vecinos del agente i ,

$$Vecino_i(j, t) = \{j \mid d(i, j, t) \leq r_j, 1 \leq j \leq n, j \neq i\}$$

- Los vecinos que pueden influir en él

$$Influyen_i(j, t) = \{j \mid |x_i - x_j| < \varepsilon_i, 1 \leq j \leq m, j \neq i\}$$

- función de confianza del agente i cumple con

$$\sum_{j=1}^k w_i(j, t) + w_i(i, t) = 1$$

Modelos de Preferencias

El modelo de Confianza Limitada

- La opinión x_i del agente i cambia en el tiempo t

$$x_i(t+1) = w_i(i,t)x_i(t) + \sum_{j=1}^k w_i(j,t)x_j(t)$$

- Cuando el proceso de ajuste de opinión termina, los agentes eligen sus nuevas direcciones de movimiento usando un valor repulsivo r_radio_i y un valor de gravitación g_radio_i .
- Cuando los agentes se mueven en el espacio, pueden ser influidos por los otros agentes en su radio de gravitación.

$$ind_i(j,t) = \{j | d(i,j,t) \leq g_radio_i, 1 \leq j \leq n, j \neq i\}$$

Modelos de Preferencias

El modelo de Confianza Limitada

- Un agente i tiene la dirección de movimiento $cabecera_i(t)$ en el tiempo t , la cual no cambian si $ind_i(j, t) = 0$
- De lo contrario, cambia eligiendo el vecino h cuyo $d(i, h, t)$ es mínimo. Si $d(i, h, t)$ se encuentra entre r_radio_i y g_radio_i , el agente i va a cambiar la dirección según:

$$cabecera_i(t + 1) = cabecera_i(t) + dir_a(t) + dir_c(t)$$

- donde

$$dir_a(t) = \arctan \frac{\sum_{j=1}^k \sin (cabecera_j(t))}{\sum_{j=1}^k \cos (cabecera_j(t))} \quad j \in ind_i(j, t) \quad \text{Sincronización de las velocidades en el grupo}$$

$$dir_c(t) = \arctan \frac{\sum_{j=1}^k \sin (cabecera_j(t) - cabecera_i(t))}{\sum_{j=1}^k \cos (cabecera_j(t) - cabecera_i(t))} \quad j \in ind_i(j, t) \quad \text{regla de cohesión}$$

Modelos de Preferencias

El modelo de Confianza Limitada

- Pero si $d(i,h,t)$ es menor a r_radio_i , entonces i y h están demasiado cerca y se deben repeler

$$cabecera_i(t + 1) = cabecera_i(t) - cabecera_h(t)$$

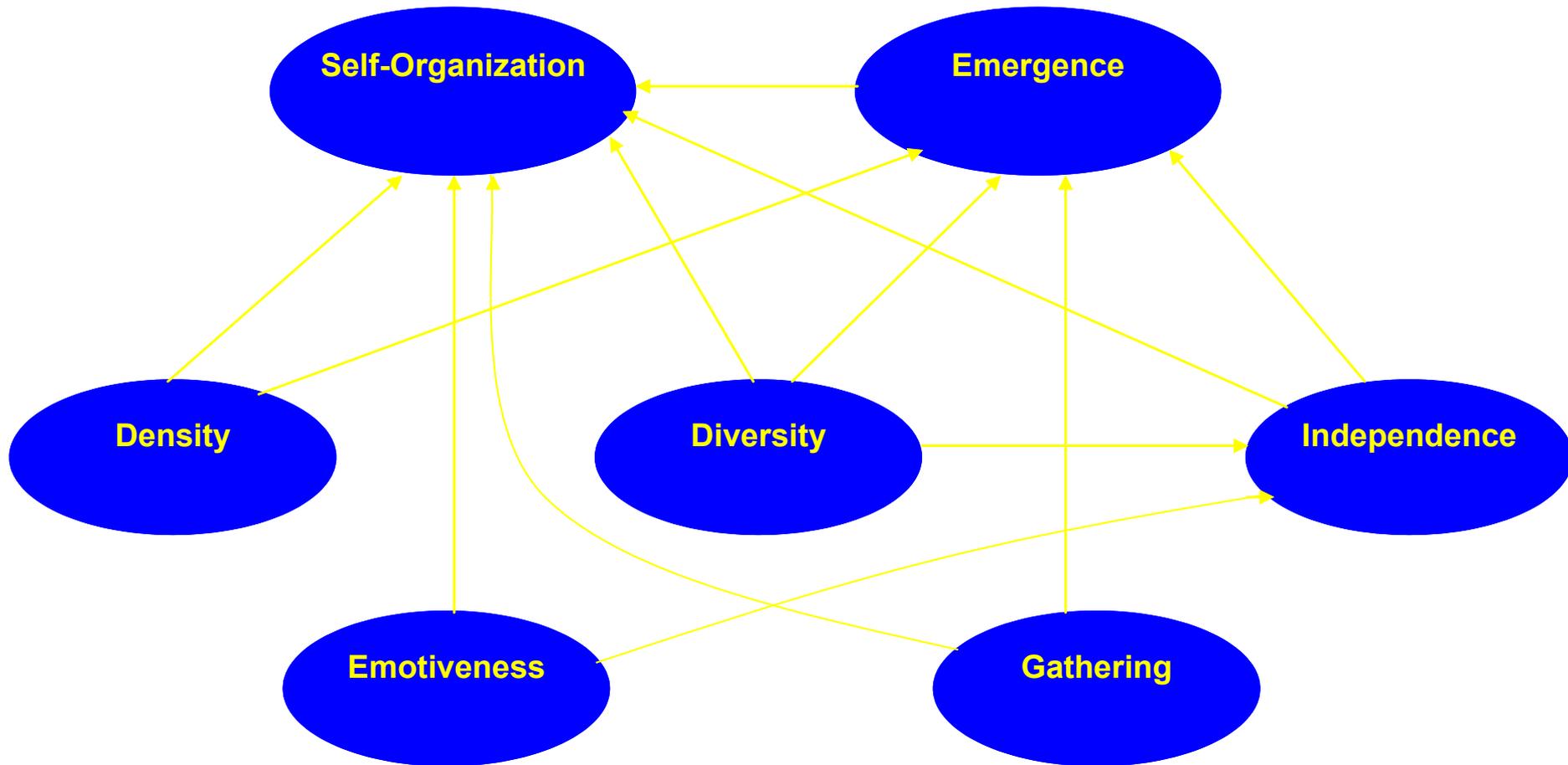
- Cada agente ajusta su dirección al mismo tiempo, se mueve un paso y espera para ajustar de nuevo las opiniones.

Este modelo puede utilizarse para simular el proceso de opinión pública en Internet, etc. Es un modelo de emergencia de soluciones, resolviendo los problemas de integración de opiniones

MASOES

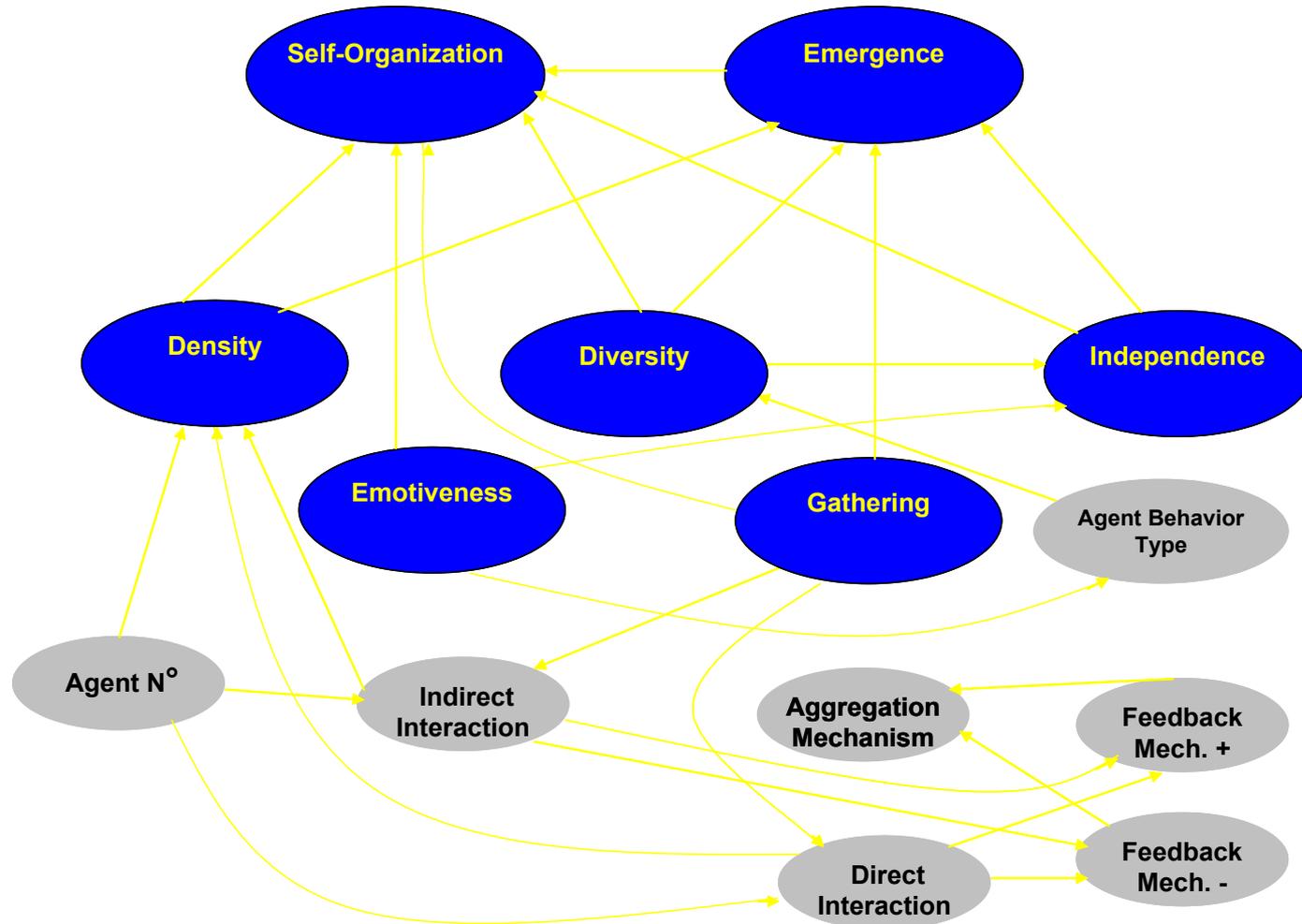
VERIFICATION METHOD

Level I : concepts linked to the emergent and self-organizing properties)



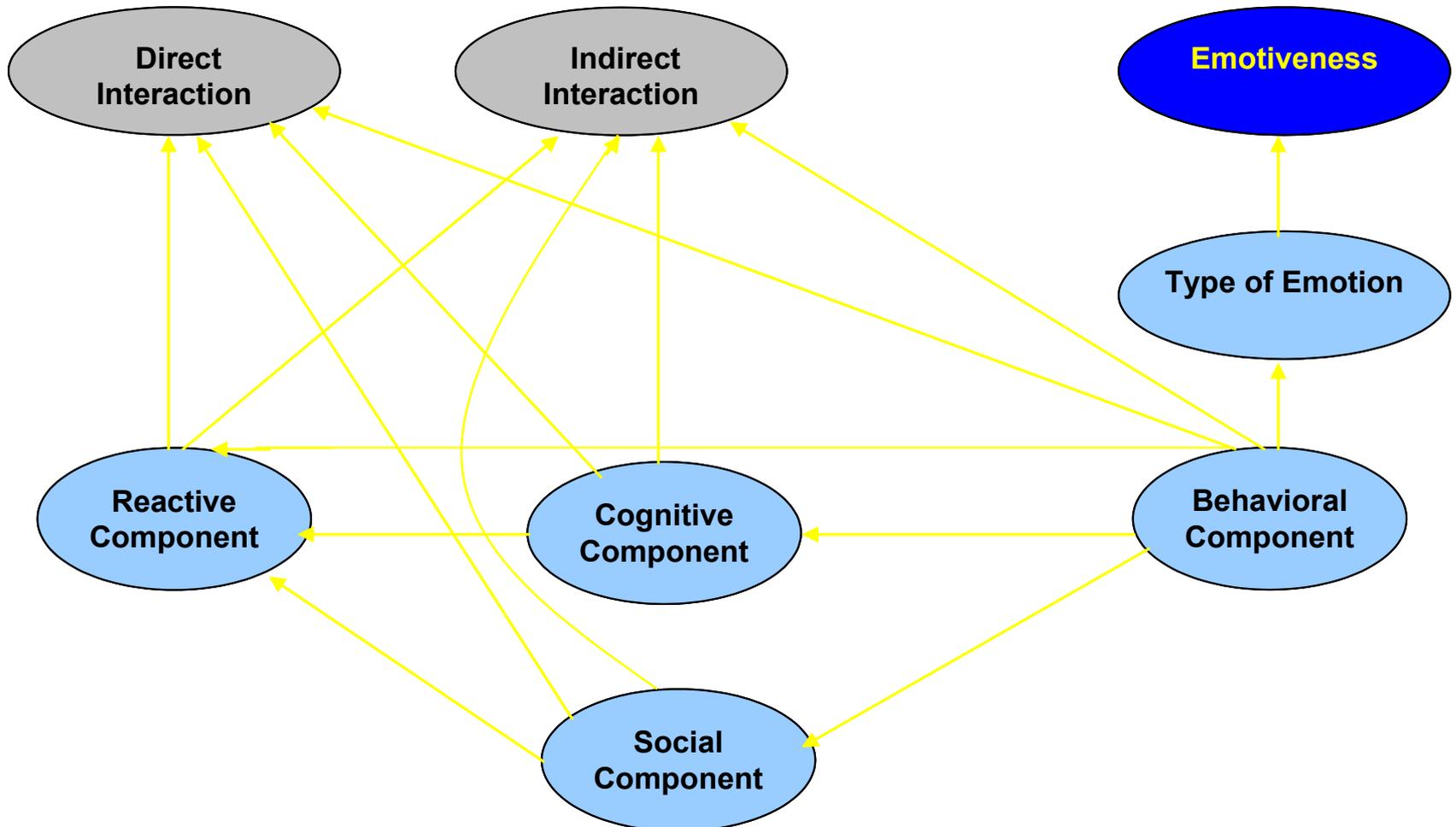
VERIFICATION METHOD

Level II : Architectonic concepts



VERIFICATION METHOD

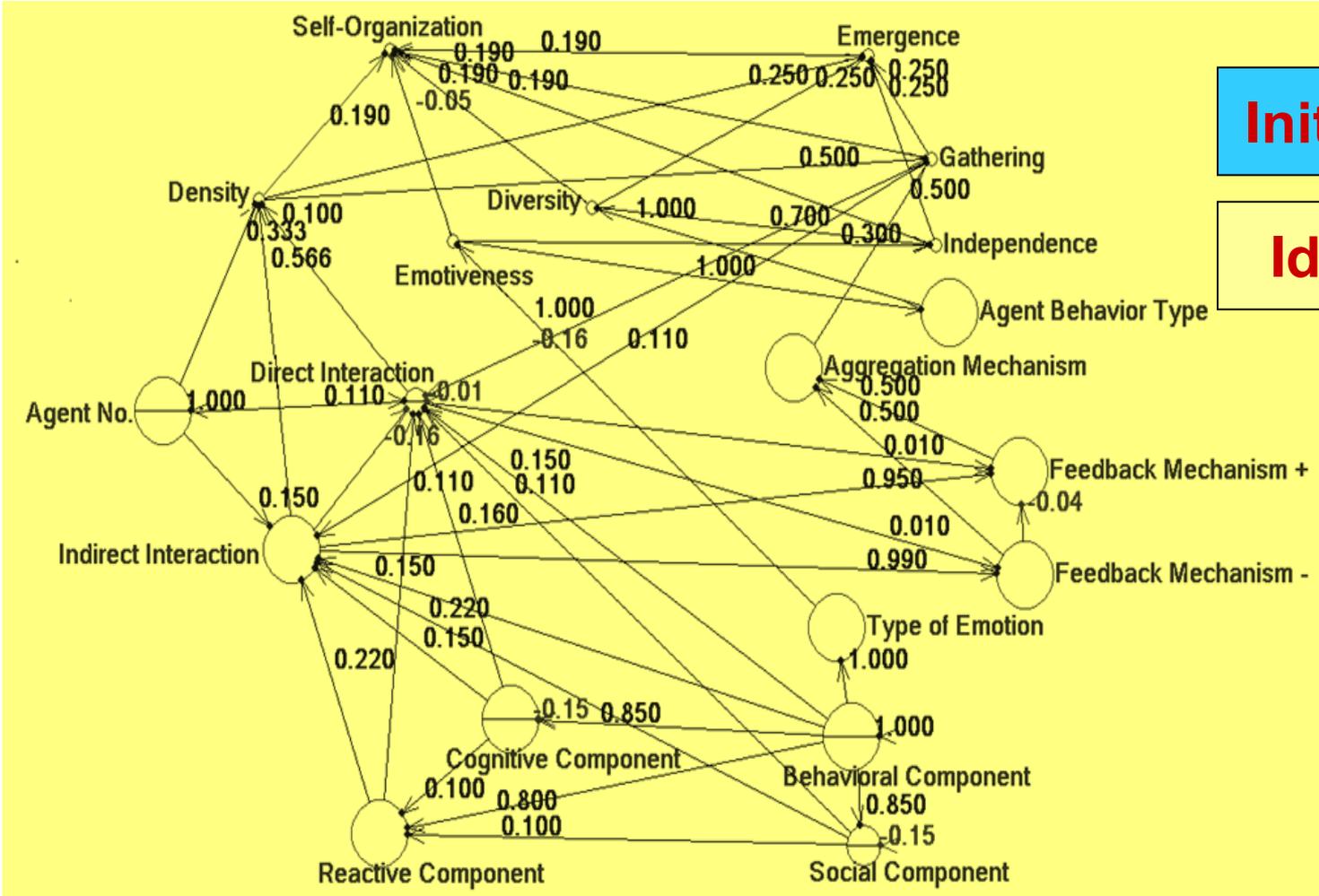
Level III : Causal Relationships



CASE STUDY: WIKIPEDIA



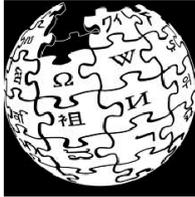
Scenario 1: Wikipedia in English Language



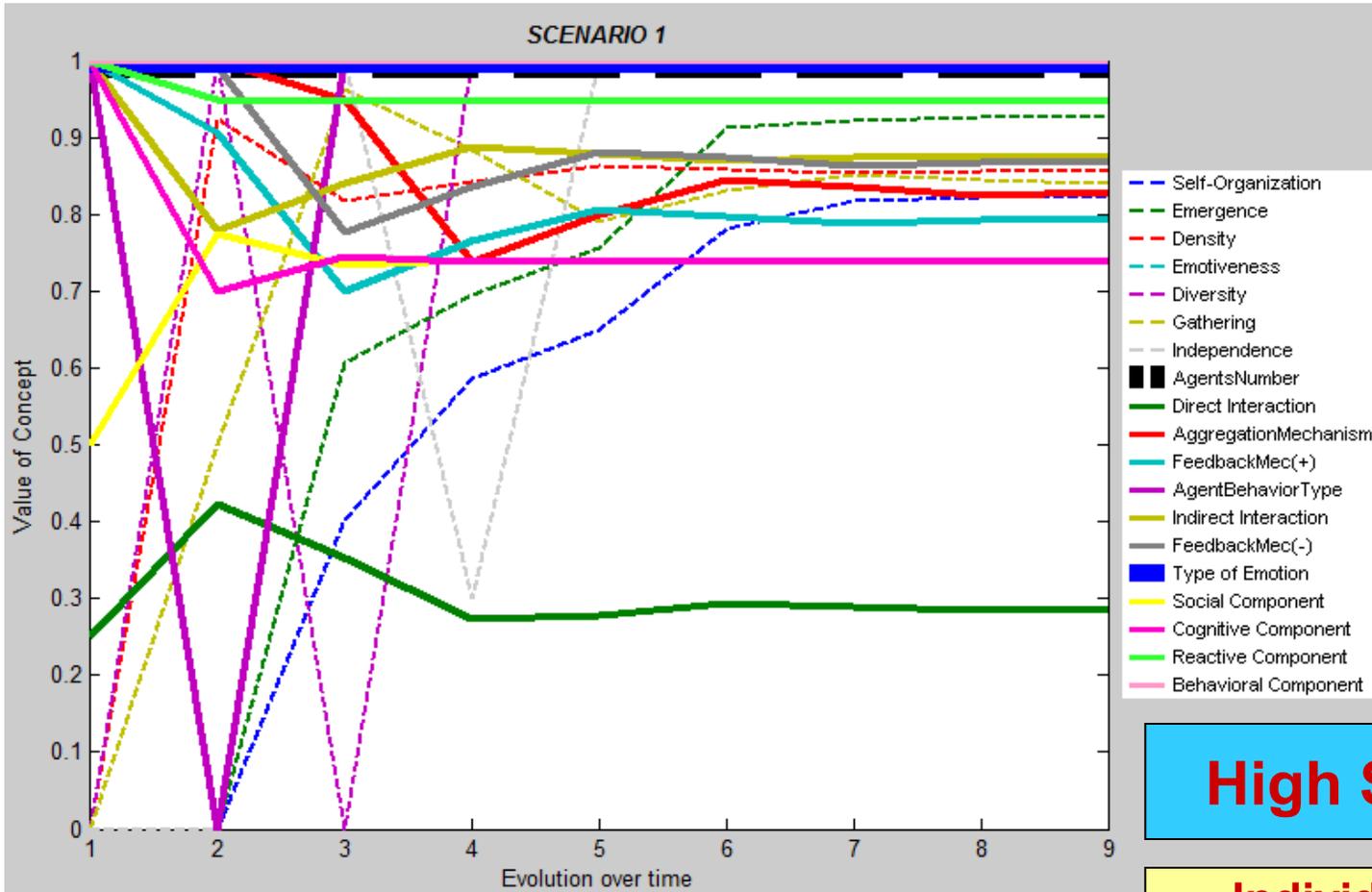
Initialization

Ideal Case

CASE STUDY: WIKIPEDIA



Obtained Results for Scenario 1

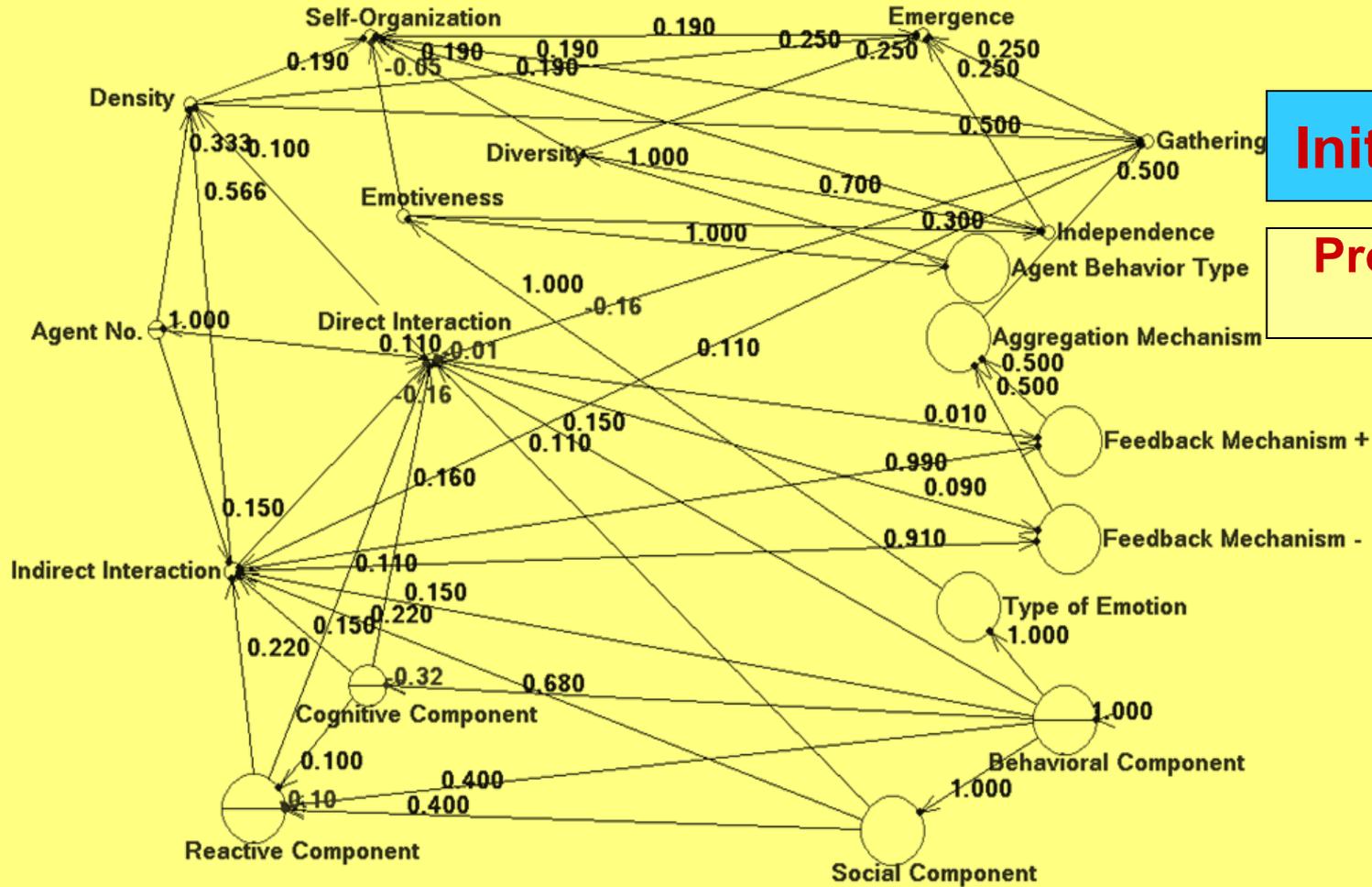
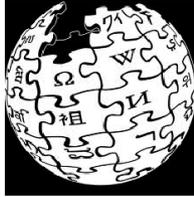


High S.O. and E.

**Individual Behavior
More than Social Behavior**

CASE STUDY: WIKIPEDIA

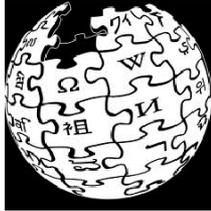
Scenario 2: Wikipedia in Spanish Language



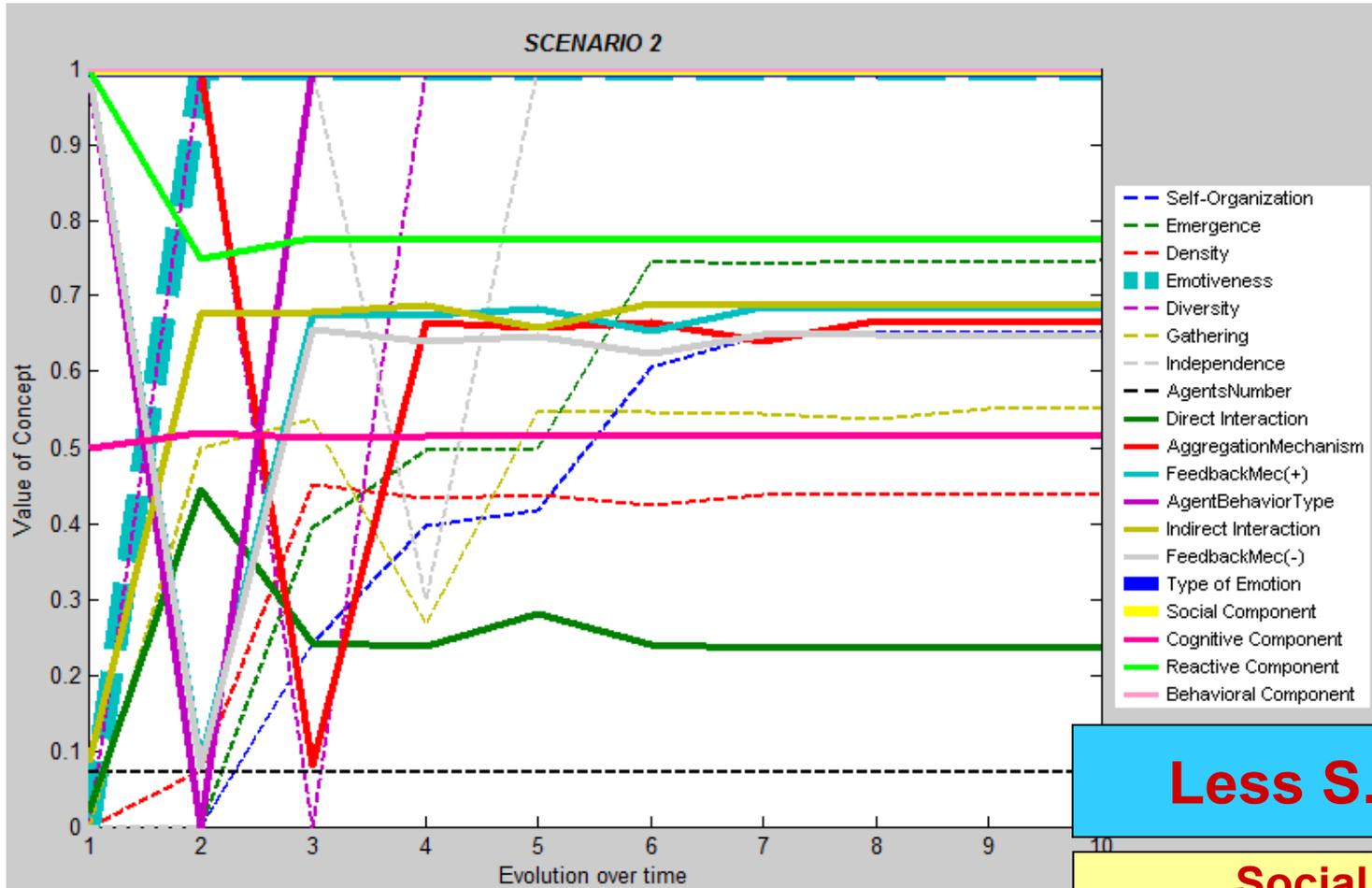
Initialization

Proportional Values

CASE STUDY: WIKIPEDIA



Obtained Results for Scenario 2



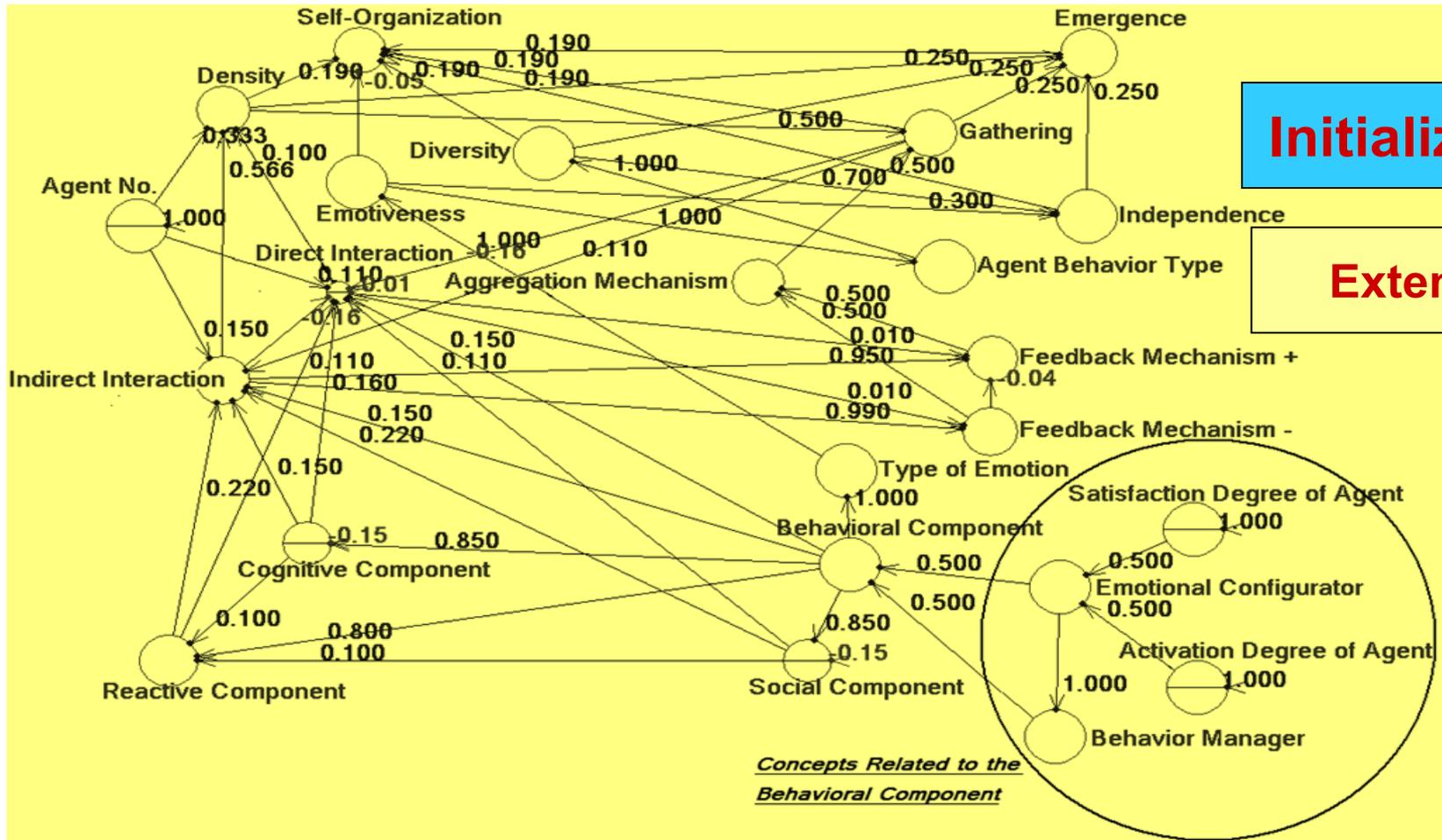
Less S.O. and E.

**Social Behavior
More than Social Behavior**

CASE STUDY: WIKIPEDIA



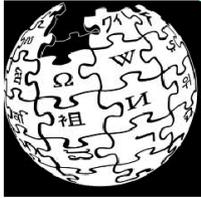
Scenario 3: Social and Emotional Regulation on Wikipedia



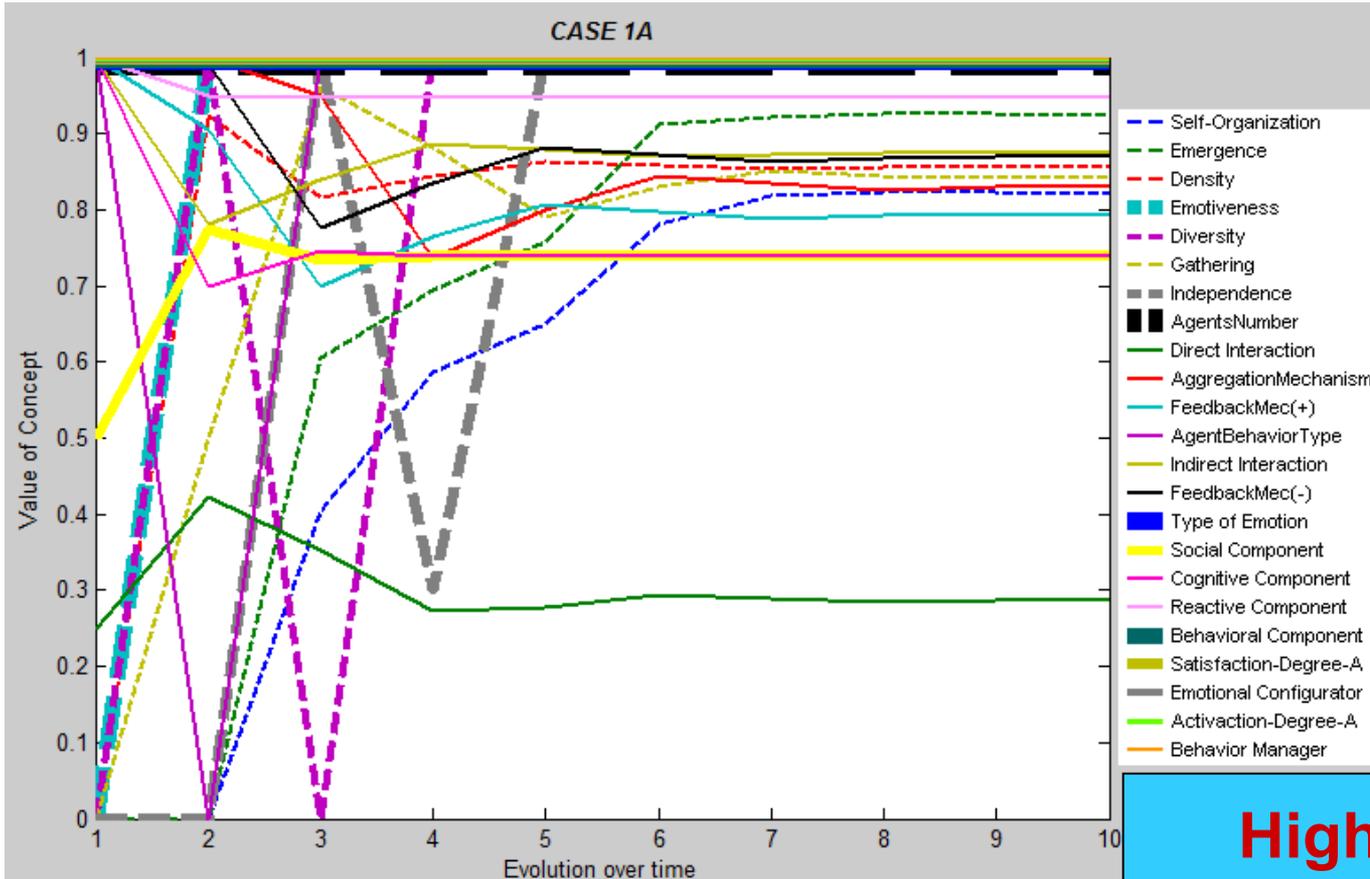
Initialization

Extension

CASE STUDY: WIKIPEDIA



Obtained Results for Scenario 3



High S.O. and E.

**High Satisfaction Degree
Social and Individual Emotions**

METROPOLIS

“un **juego emergente** aparece cuando un conjunto relativamente simple de reglas conducen a juego complejo ... un juego emergente significa que tiene vida propia, según lo que haga el jugador, tal que el entorno puede responder a sus acciones construyendo una historia que será siempre en esencia la misma, pero adaptada a las preferencias del jugador”

- El juego emergente puede verse como un concepto en el que **emergen dinámicas sociales** a partir de las decisiones que van tomando los jugadores

Las **acciones del jugador influyen** en el mundo que le rodea, conllevando a que se den en él comportamientos emergentes,

Tipos de emergencia de interés:

- Cuando en un juego, debido a las decisiones de los jugadores, surge un nuevo comportamiento.
- Cuando las reglas simples sobre las propiedades de los objetos del juego interactúan para crear un juego completamente nuevo

Premisas de Metropolis

- Es del tipo de videojuegos de construcción y gestión, subgénero de los videojuegos de simulación en el que los jugadores construyen, expanden o gestionan comunidades ficticias o proyectos con recursos limitados.
 - . Ejemplos son SimCity, LinCity, City Life y OpenCity,
- Contiene reglas que regulan las interacciones entre los agentes (jugadores) que desempeñan roles sociales en la sociedad.
- Las ciudades se auto-gestionan a partir de las decisiones que se van tomando colectivamente, en un entorno en el que todos los agentes (jugadores) ejercen un rol de igual importancia, .
- En el juego se persigue planificar colectivamente el crecimiento de comunidades, para después observar las ciudades desarrolladas.

Bases del Juego

- Elemento de base del juego : el consejo.
En el consejo cada agente perteneciente a un colectivo tiene la oportunidad de votar a favor o en contra de construir cierto tipo de construcción,
- Agentes de dos tipos, unos que representan a los jugadores, otros generados aleatoriamente y controlados por el juego.
- Se asignan personalidades a los agentes que indican sobre qué tipo de construcción tienen más interés, tanto de construir como para votar en el consejo.
- La ciudad evoluciona según lo que los agentes decidan en cuanto a que se construirá o destruirá en la misma.

Bases del Juego

- Cada jugador no sólo se representa a sí mismo, sino a un porcentaje de la sociedad de la ciudad (a un colectivo).
- A la hora de construir algo en la ciudad, cada jugador somete su proyecto a una votación en el consejo.
- El juego lleva un sistema de puntuación, el cual es calculado cada año que pasa en la ciudad.
- La puntuación está basada en:
 - *Los índices de avance de la ciudad:* calculados según el balance que haya entre las diferentes áreas de desarrollo de la ciudad (ambiental, educativo, industrial, etc.).
 - *Los índices de felicidad de los habitantes* se establece según la relación entre el perfil de cada habitante y el desarrollo de la ciudad

Reglas del Juego

- Pueden jugar uno o más jugadores.
- Se establece un tiempo límite, el cual es el número de años tras el cual el juego termina.
- Cada agente podrá votar en el consejo a favor o en contra de la construcción de una edificación en una determinada zona.
- Cada ciudad deberá tener 10 agentes, de los cuales al menos 5 son agentes artificiales.
- Los agentes pueden ser creados con personalidades al azar o definidos por el usuario al inicio del juego.
- El jugador podrá construir la edificación que desee, en la zona que prefiera.
- Cualidades de la ciudad: educación, ambiente, industria, ambiente, tecnologías. comercio

Caracterización de las Edificaciones

<i>Num</i>	<i>Tipo</i>	<i>Grupo</i>	<i>Radio</i>	<i>Bono</i>	<i>Penalización</i>	<i>Costo \$</i>
0	Vacio	--	--	--	--	0
1	Árbol	--	--	+1 Ambiente	--	20
2	Residencia	--	--	--	--	500
3	Calle	--	--	--	--	40
4	Institución de Salud	Primaria	7	+250 Salud	--	7000
5	Institución Educativa	Primaria	4	+150 Educacion	--	4000
6	Institución Ambiental	Primaria	7	+100 Ambiente	--	4000
7	Comercio	Primaria	2	+50 Comercio	--	5000
8	Institución Industrial	Primaria	7	+200 Industria	-75 Ambiente	7000
9	Institución Tecnológica	Primaria	3	+170 Tecnologia	--	3000
10	Institución de Salud	Secundaria	2	+15% Salud	--	1000
11	Institución Educativa	Secundaria	2	+10% Educacion	--	800
12	Institución Ambiental	Secundaria	2	+12% Ambiente	--	800
13	Comercio	Secundaria	1	+7% Comercio	--	600
14	Institución Industrial	Secundaria	3	+ 18% Industria	-8% Ambiente	1000
15	Institución Tecnológica	Secundaria	2	+10%Tecnologia	--	800

Personalidades de los Agentes e Índices de Felicidad

Trato	Salud	Educación	Ambiente	Comercio	Industria	Tecnología
Hipocondríaco	1250	1000	1000	1000	1000	1000
Saludable	750	1000	1000	1000	1000	1000
Con Dificultades	1000	1250	1000	1000	1000	1000
Autodidacta	1000	750	1000	1000	1000	1000
Indiferente	1000	1000	1250	1000	1000	1000
Ambientalista	1000	1000	750	1000	1000	1000
Ahorrador	1000	1000	1000	1250	1000	1000
Comprador	1000	1000	1000	750	1000	1000
Anti- Industrias	1000	1000	1000	1000	1250	1000
Industrialista	1000	1000	1000	1000	750	1000
Anticuado	1000	1000	1000	1000	1000	1250
Geek	1000	1000	1000	1000	1000	750

Penalizaciones o bonificaciones por proximidad de construcciones

	Árbol	Residencia	S1	E1	A1	C1	I1	T1	S2	E2	A2	C2	I2	T2
Árbol														
Residencia					X		X						X	
CL														
S1			-4			-2	-6	-2	+3				-4	
E1							-4	+3	+3	+3			-3	+3
A1		X					-3	+2			+3		-2	
C1			-2				-3					+3	-2	
I1		X	-6	-4	-3	-3		-3	-5	-3	-3	-3	+3	-2
T1			-2	+3	+2		-3			+3	+2		-2	+3
S2			+3	+3			-5			+3			-3	
E2				+3			-3	+3	+3		+2		-3	+3
A2					+3		-3	+2		+2			-2	+2
C2						+3	-3						-2	
I2		X	-4	-3	-2	-2	+3	-2	-3	-3	-2	-2		
T2				+3			-2	+3		+3	+2			

**Emergencia
de Patrones**

Formulas Matemáticas

Penalización del *Solapamiento*

$$PO_i = \frac{VC}{NO_i * NCO}$$

Donde: PO_i : Penalización solapamiento de edificaciones de tipo i ; VC : Valor de la Casilla; NO_i : Número de solapamiento (cantidad de construcciones i que tocan dicho espacio); NCO : Número de Casillas con dicho solapamiento.

Puntuación total de una construcción primaria i

$$PC_{ij} = \sum_r (P_{ij} - PO_i)$$

Donde: P_{ij} es el bono o penalización de la construcción i al índice j según tabla anterior, y r es la zona que abarca la edificación i

Formulas Matemáticas

Puntuación total de cada índice de felicidad j

$$PT_j = \sum_l \sum_i (PC_{ij}^l + \sum_s CER_{si}) + \sum_m \sum_k (PC_{kj}^m + \sum_s CER_{sk})$$

Donde: l recorre todas las edificaciones primarias y m las secundarias en la ciudad, respectivamente, y CER_{sk} son las bonificaciones o penalizaciones según las cercanías de las edificaciones s a i según la tabla anterior

Puntuación para el índice de felicidad del Jugador p

$$VI_p = \sum_j \frac{1000 * PT_j}{TRJ_{pj}}$$

Donde: TRJ_{pj} es el total Requerido por el Jugador p para el índice j .

Eventos naturales

Un evento se define como algo que ocurre al azar.

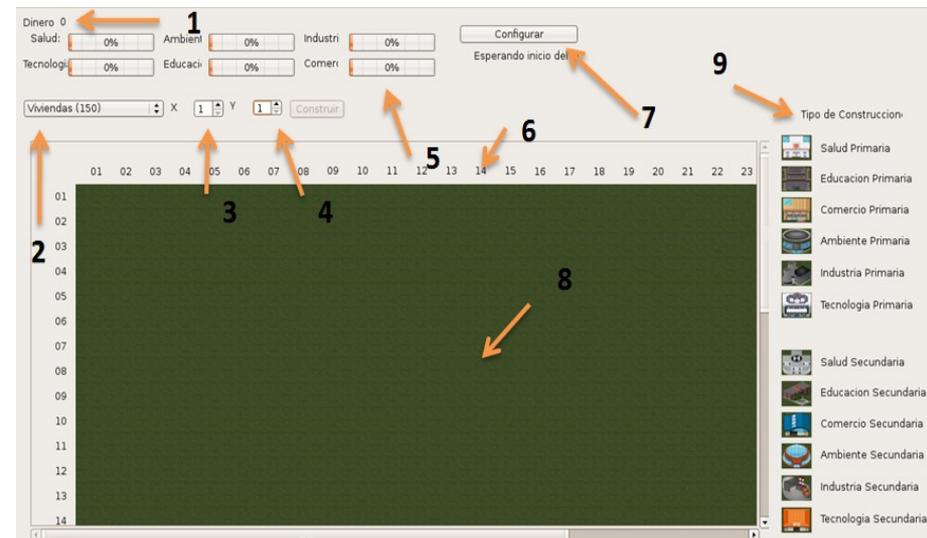
Se han definido 3 tipos de eventos: *incendios, terremotos y tornados.*

Estos están diseñados para destruir la ciudad, existiendo tres escalas de intensidad :1, 2 y 3; siendo tres la más devastadora.

La probabilidad de que ocurra un incendio es más alta de que ocurra un tornado o terremoto, siendo la probabilidad del tornado la menor

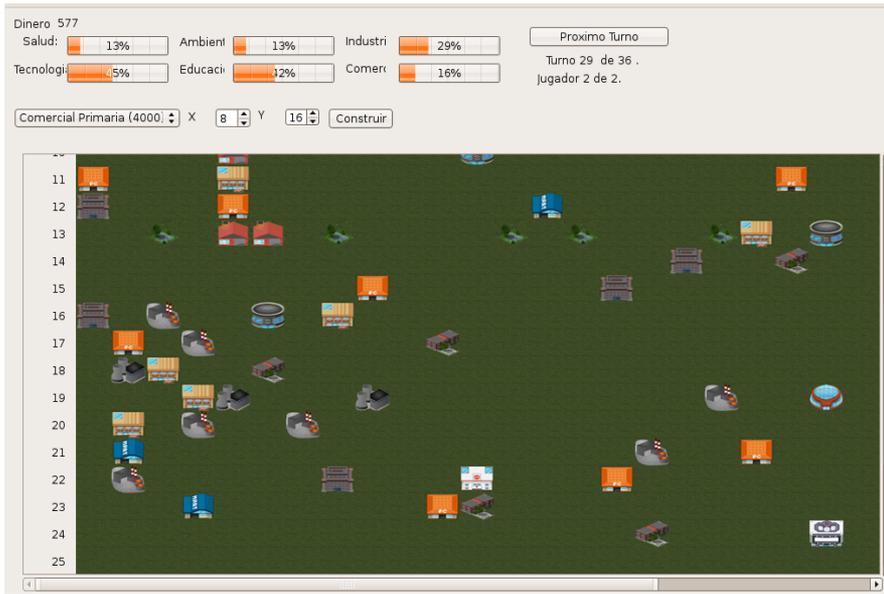
Un tornado de intensidad 3 destruye todo lo que está a su alrededor, mientras que un incendio de intensidad 1 destruye pocas cosas escogidas al azar (los dos extremos de efectos)

Pantalla Principal de Metropolis



1. Indica la cantidad de dinero que se tiene en el presupuesto actualmente.
2. Permite seleccionar el tipo de construcción e indica su costo.
3. Coordenada X del mapa.
4. Coordenada Y del mapa.
5. Indicadores de avance en salud, educación, ambiente, comercio, etc.
6. Muestra el número de la casilla para que el jugador pueda guiarse.
7. Botón de configuración de juego, es el primer paso para configurar los agentes.
8. Mapa donde se construyen las edificaciones.
9. Leyenda que muestra el icono asignado a cada tipo de construcción.

Caso 1: emergencia de patrones urbanísticos



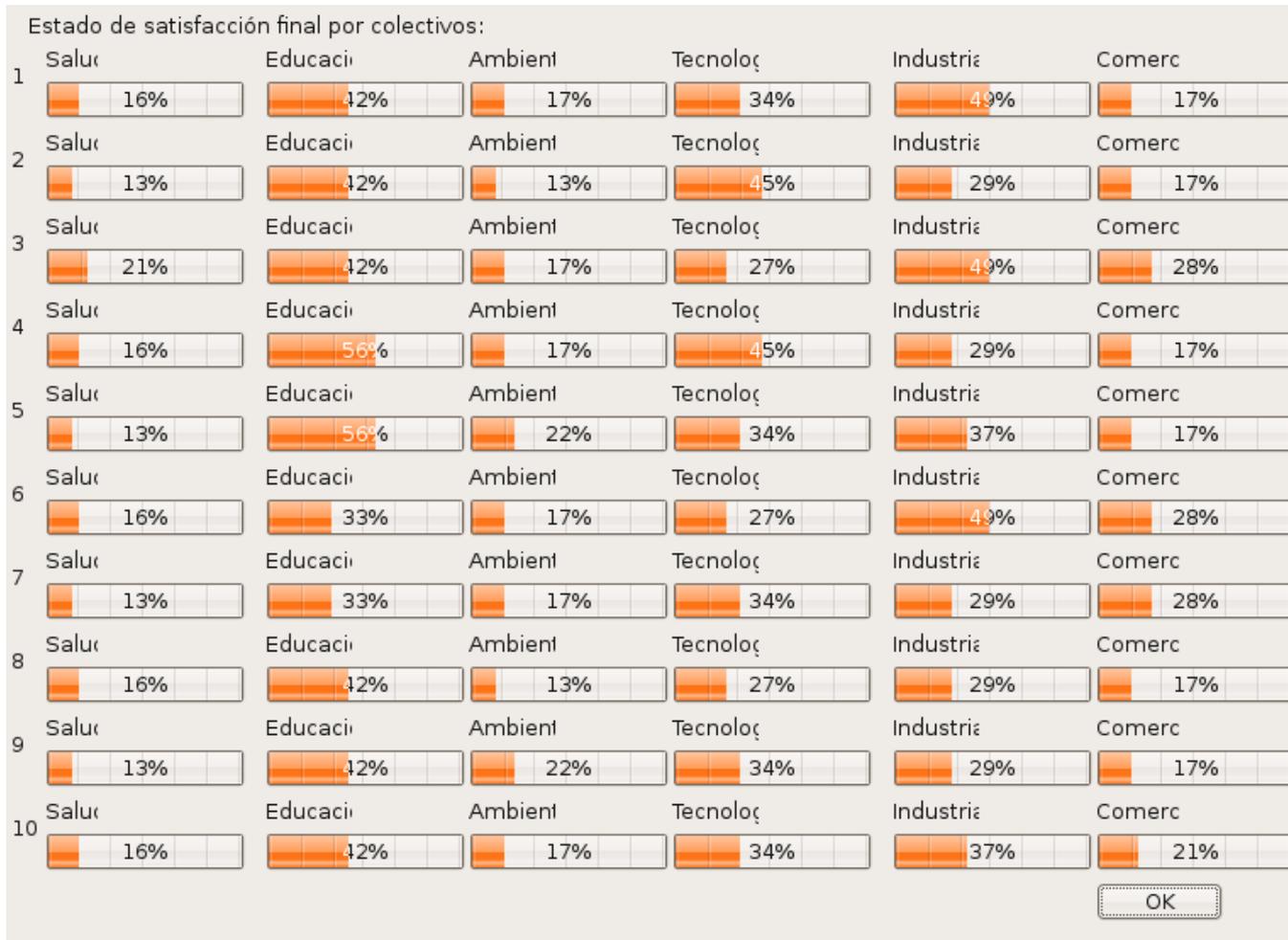
Turno 18



Índices de
Avance de
La Ciudad.
Año 2

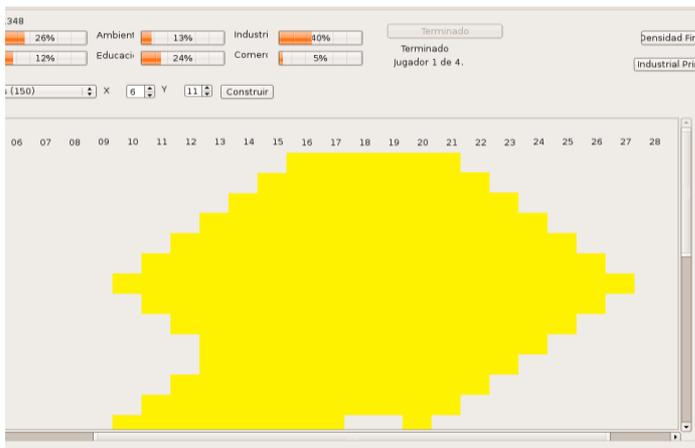
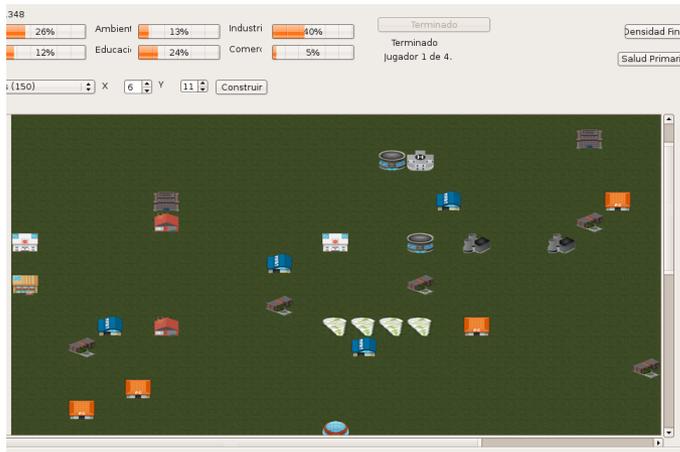
- Se seleccionan 2 jugadores y 3 años para el tiempo de juego.
- Se dejan las personalidades de los agentes al azar.
- Cada turno de juego representa un mes para cada uno de los jugadores.
- Para este caso se seleccionaron 3 años de juego,

Estudio de Comportamientos Emergentes



Índices de Felicidad
 para los Agentes
 Al 3r año

Estudio de patrones de ciudad particulares debido a necesidades específicas de los agentes



Para el segundo caso el tiempo de juego es 2 años y el número de jugadores es 4.

Se configuran los agentes 1 y 2 como:

Agente 1: con dificultades en la educación, industrialista e hipocondríaco.

Agente 2: hipocondríaco, ambientalista, ahorrador y autodidacta.
y el resto se dejan al azar

Los resultados finales de la ciudad son congruentes con las personalidades de los agentes

Emerge un patrón de conducta en la ciudad.

Metrópolis

Conceptos:

- Juegos Emergentes
- Radio de alcance
- Overlapping

Metrópolis

MainWindow

Diner: 63903

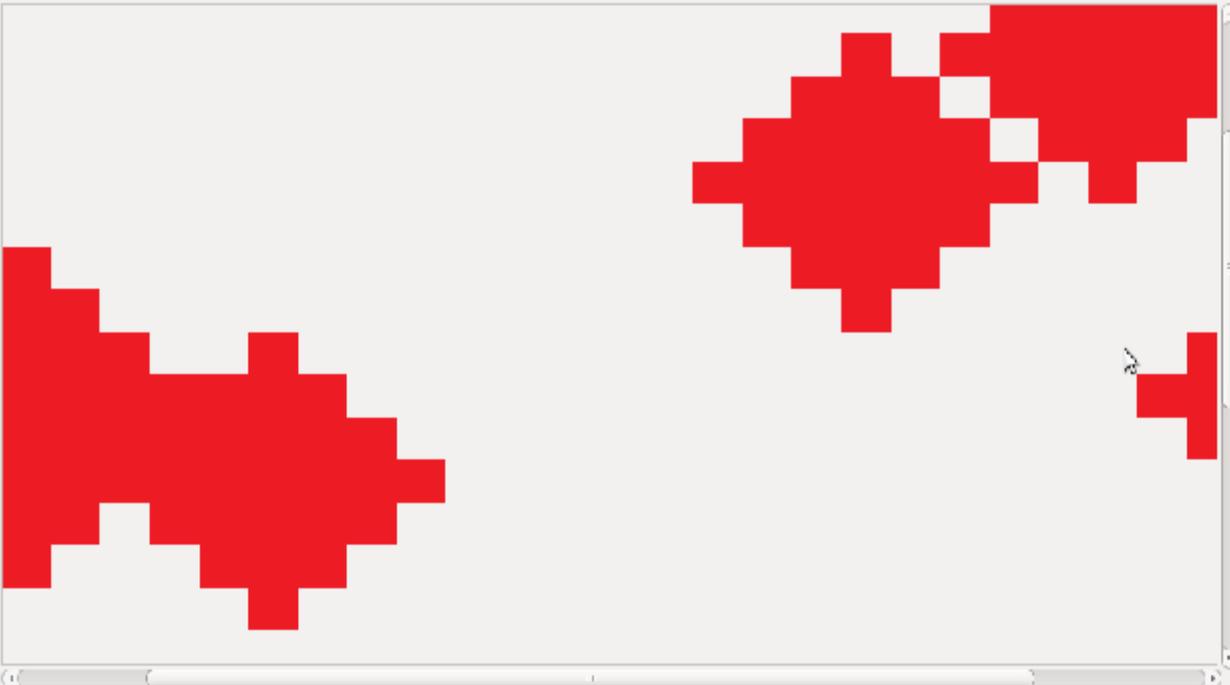
Salud: 85% Ambien: 42% Industr: 72%

Tecnolog: 53% Educaci: 78% Comer: 36%

Terminado
Terminado
Jugador 1 de 1.

Densidad Final de la Ciudad
Salud Secundaria

Viviendas (150) X 1 Y 1 Construir



Tipo de Construcción

- Salud Primaria
- Educacion Primaria
- Comercio Primaria
- Ambiente Primaria
- Industria Primaria
- Tecnologia Primaria
- Salud Secundaria
- Educacion Secundaria
- Comercio Secundaria
- Ambiente Secundaria
- Industria Secundaria
- Tecnologia Secundaria

Metrópolis

Estrategia de auto-modificación

Las construcciones tienen un valor de radio establecido por defecto en el simulador tanto para construcciones primarias como secundarias que se pueden observar en la Tabla 1, este valor determina el número de casillas que alcanza dicha construcción.

Se asigna como máximo radio el valor 7, y como mínimo radio el valor 1. Además hay construcciones que no se les toma en cuenta en la asignación de el valor del radio tales como las zonas residenciales y calles por ejemplo, ya que estas solo afectan a la

Metrópolis

Tipo	Radio Primaria	Radio Secundaria
Salud	7	2
Ambiente	7	2
Industria	7	3
Educación	4	2
Comercio	2	1
Tecnología	3	2

Valores de radio por defecto

Metrópolis

Algoritmo de modificación:

Variables adicionales de radio: rS , rE , rA , rC , rI , rT

Indices de Felicidad: $fsalud$, $feduccion$, $fambiente$, $fcomercio$, $findustria$, $ftecnologia$

Variable promedio: $fprom = (fsal + fedu + famb + fcom + find + ftec) / 6$

if(salud >= fprom){

*if(salud-fprom < fprom*0.33)*

rS=rS;

*else if(salud-fprom >= fprom*0.33 && salud-fprom < fprom*0.66)*

rS--;

*else if(salud-fprom >= fprom*0.66)*

rS-=2;

}

Metrópolis

```
else if(salud < fprom){  
    if(fprom-salud < fprom*0.33)  
        rS=rS;  
    else if(fprom-salud >= fprom*0.33 && fprom-salud < fprom*0.66)  
        rS++;  
    else if(fprom-salud >= fprom*0.66)  
        rS+=2;  
}
```

Nota: Este ejemplo es para el caso salud y se aplica igual para cada uno de los casos.

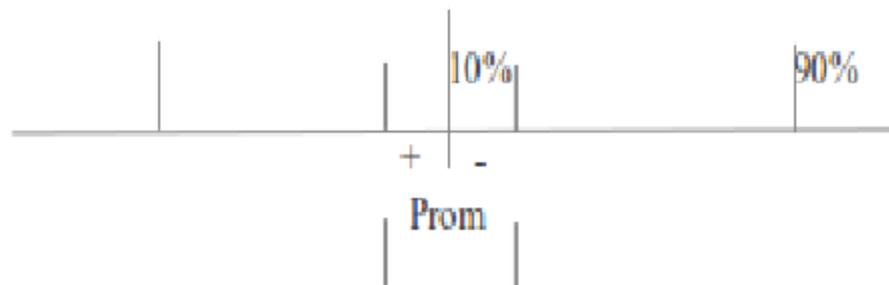
Metrópolis

Formulas Matemáticas

$$\begin{aligned} R_x &= R_x & F_{xi} < F_{Prom} * 0,10 \\ R_x - 1 & & F_{Prom} * 0,10 \leq F_{xi} < F_{Prom} * 0,90 \\ R_x - 2 & & F_{xi} \geq F_{Prom} * 0,90 \end{aligned}$$

$$\begin{aligned} R_x &= R_x & F_{xi} > F_{Prom} * 0,10 \\ R_x + 1 & & F_{Prom} * 0,10 \geq F_{xi} > F_{Prom} * 0,90 \\ R_x + 2 & & F_{xi} \leq F_{Prom} * 0,90 \end{aligned}$$

Promedio y Rangos de cercanía



Prueba 1

Con Mod R

Iteraciones	3	4	5	Prom
Salud	24	100	85	69,67
Educación	62	99	100	87,00
Ambiente	49	57	100	68,67
Tecnología	21	99	37	52,33
Industria	56	54	65	58,33
Comercio	46	19	77	47,33
Prom	43,00	71,33	77,33	

Sin Mod R

Iteraciones	3	4	5	Prom
Salud	13	34	78	41,67
Educación	35	36	83	51,33
Ambiente	31	37	95	54,33
Tecnología	37	39	52	42,67
Industria	37	55	37	43,00
Comercio	19	30	40	29,67
Prom	28,67	38,50	64,17	

Prueba 2

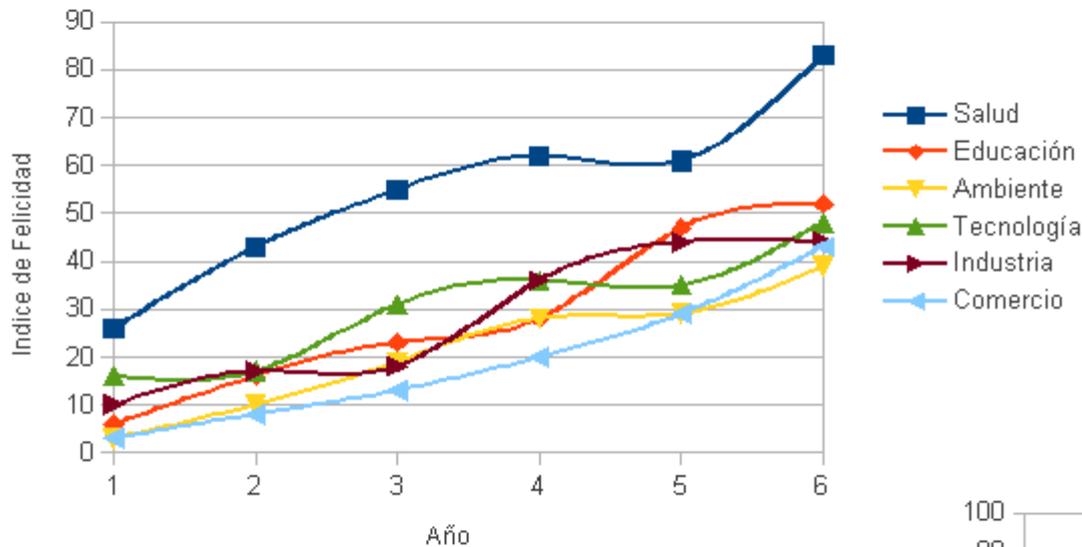
Con Mod R Sin Mod R

Iteraciones	4	4	Diferencia
Salud	73	69	4
Educación	60	34	26
Ambiente	38	23	15
Tecnología	56	48	8
Industria	53	47	6
Comercio	61	27	34
Prom	56,83	41,33	

Prueba 4

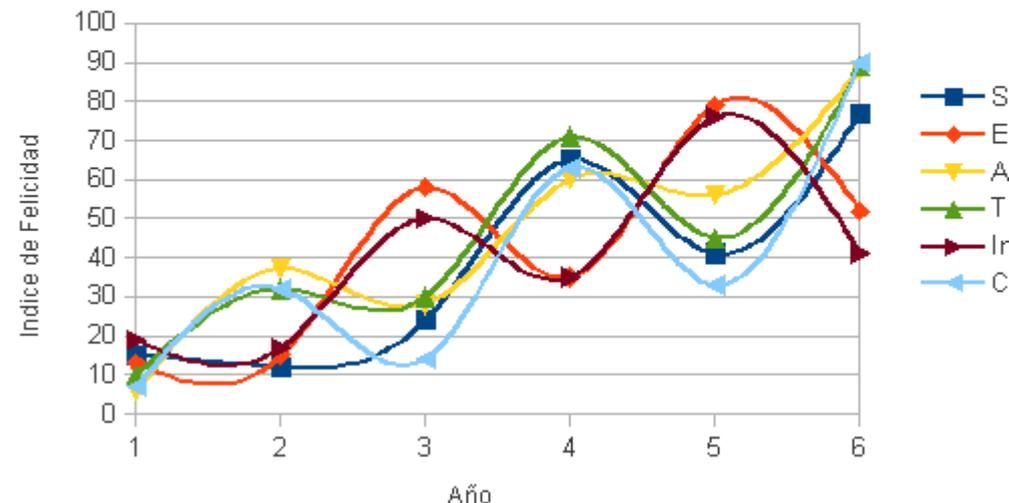
Simulación de 6 años

Sin modificación de radio

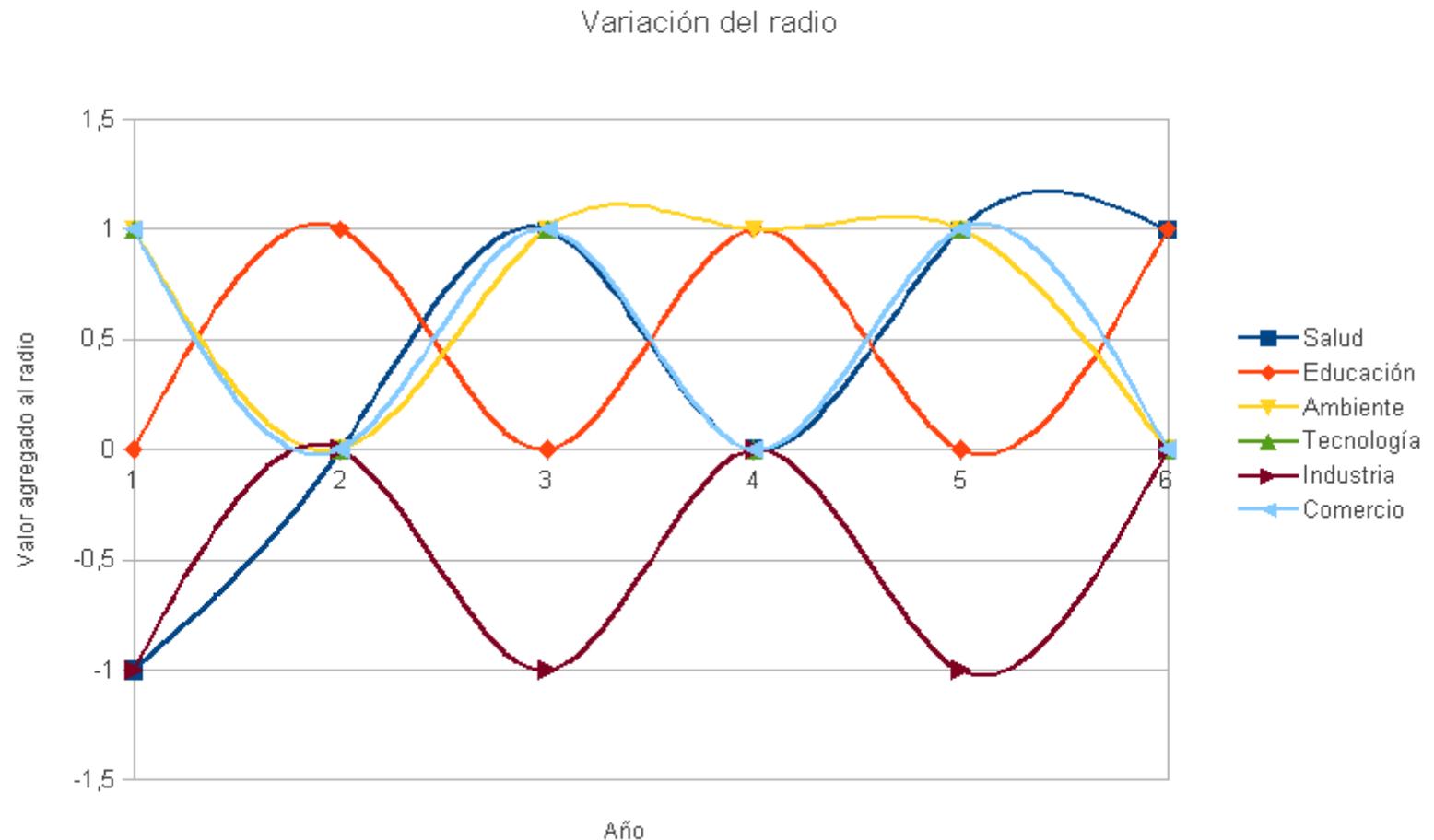


Simulación de 6 años

Con modificación de radio



Variación del radio agregado



Computación Distribuida

Aplicaciones distribuidas

- Reunir a una serie de partes interesadas mutuamente con diversos grados de relación previa (tal vez ninguno) que deseen compartir sus recursos con el fin de alcanzar los objetivos individuales o globales.
- Tal vez el mejor ejemplo de una aplicación distribuida es el conjunto de protocolos de red que se ejecutan y de control del tráfico de red en Internet.

**algoritmos muy robustos son altamente distribuidas ,
y muy resistente a los fallos .**

Aplicaciones distribuidas

lo difícil que es escribir aplicaciones distribuidas.

- son difíciles de modificar y mantener.
- Sin embargo, un problema más difícil es que el comportamiento de estas aplicaciones no se expresa por el cálculo local, sino por el comportamiento que emerge de las interacciones de todos los nodos de cálculo que interactúan entre sí

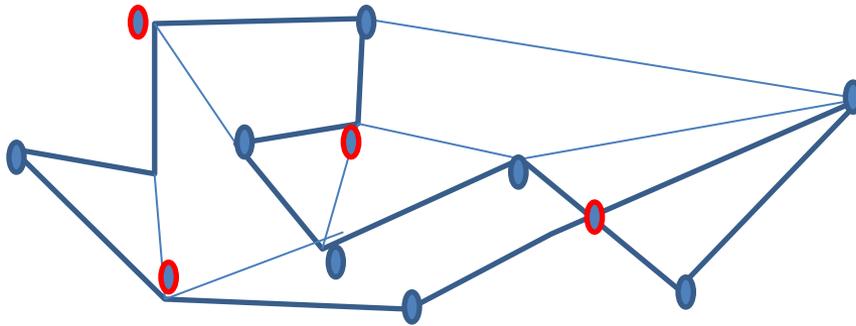
esto es muy difícil para un programador para visualizar estas interacciones .

Aplicaciones distribuidas

Ejemplos

- Algoritmos de Votos
- Algoritmos de enrutamiento
- Algoritmos de alineación den enjambre de robots

Algoritmo de Voto



- Cada nodo tiene una primera opinión de cualquiera bueno (azul) o mala (rojo)
 - Los nodos se comunican sus opiniones a sus vecinos
 - Nodo cambia de opinión si la mayoría de los nodos en el barrio son de la opinión contraria
 - informa del cambio a los vecinos
- Repita

Algoritmo de alineación den enjambre de robots

- Tenemos n robots en línea



- Queremos que se distribuyan uniformemente

