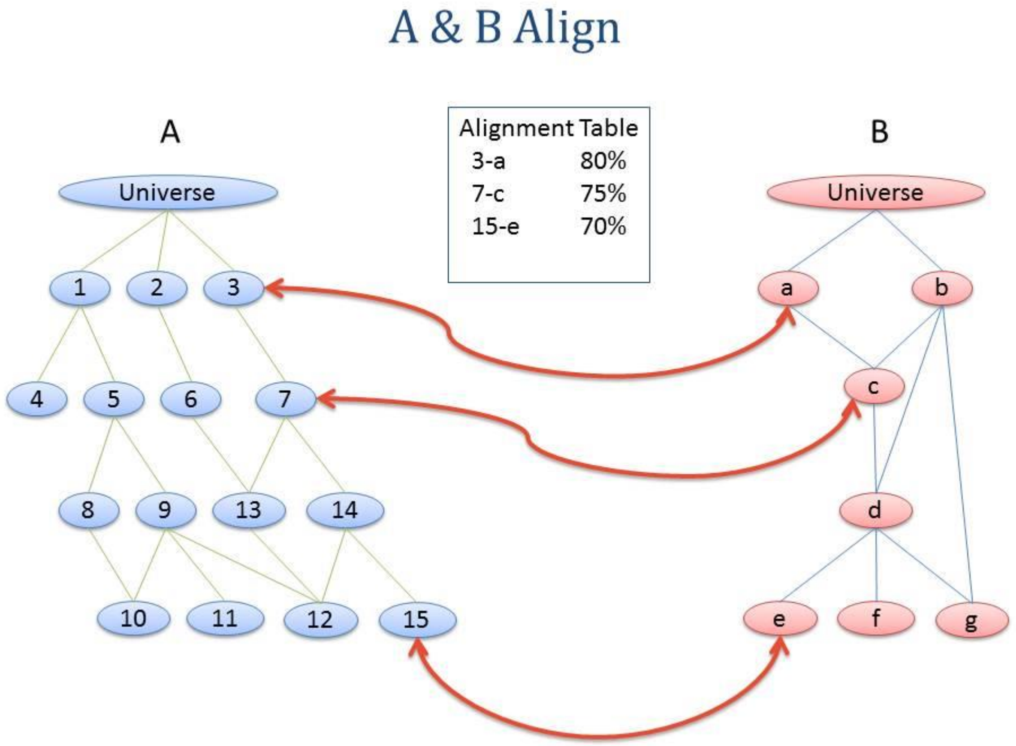
Computación Emergente: Aplicaciones

An Approach for the Emerging Ontology Alignment based on the Bees Colonies

Introduction

A first step to compare the ontologies that will be integrated, is apply **ontologies alignment techniques,** in order to find the correspondences between the concepts of the ontologies.



Alignment Table	
3-a	80%
7-c	75%
15-e	70%

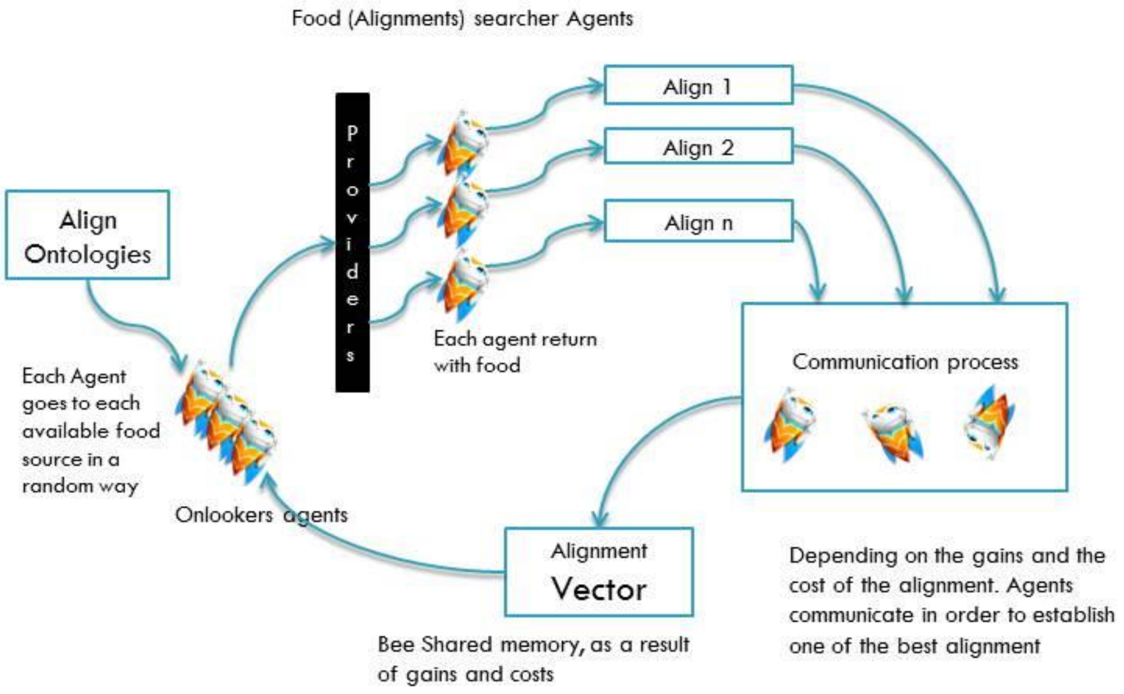
ABC Algorithm: the main steps

The algorithm based on the Colonies of Bees, called Artificial Bee Colony (**ABC**), motivated by the intelligent behavior of bees.

1. Send the scouts bees to find food sources
2. *REPEAT*
    a. *Send the employed bees to identified food sources and determine their amounts of nectar.*
    b. *Calculate the probability value of the sources (quality) with which the onlookers bees will prefer sources.*
    c. *Send onlookers bees to food sources using a stochastic selection process based on the amount of nectar in each source.*
    d. *Stop the process of exploitation of sources exhausted by bees.*
    e. *Send scouts to the search area to discover new food sources randomly.*
    f. *Save the best food source found so far.*
3. *UNTIL (the requirements are met)*

Emergent Alignment by using our ABC Approach

The **problem** of the ontology **alignment** is to be able to decide which of the techniques of semantic alignment must be used. For it, it is used the **ABC** algorithm in order to let it **choose automatically** the **technique** to perform the alignment.



Emergent Alignment by using our ABC Approach

The gain $G(S_i)$ is calculated as follows, in the equation

$$G(S_i) = \frac{S_a(S_i)}{CA(S_i)} \times P_c$$

- **S_i:** Service that can be utilizes to resolve a request (In our case, the alignment techniques). That is, **each alignment technique** is a source of nectar.
- **$G(S_i)$:**): Profit, that is obtained by the use of the service Sj (one alignment technique), defined by the equation, which determine the **quality of nectar** (alignment technique).
- **$Sa(S_i)$:** Satisfaction of the Bee, when the service Sj is performed. It is also relate with the quality of nectar; in our case is the **number of aligned nodes** of the ontologies
- **$CA(S_i)$:** Cost, it is represented in this work as **the execution time** of the service Sj to return a result (also affects the quality of nectar).
- **Pc:** Probability of preserving the food source. **Pseudo-random** value, with a **normal distribution** within the range of 0 and 1, which changes the value of $G(S_i)$

Used Alignment Techniques

a) **Class Structure:** based in finding similitude taking into consideration the **graph structure** from the classes.

b) **Distance edited name:** semantic distance required to make **the classes names** equals.

c) **Distance edited subclass name:** semantic distance required to make **the sub-classes names** equals, from the parent classes

d) **Name and properties:** **similitude** between classes names and properties.

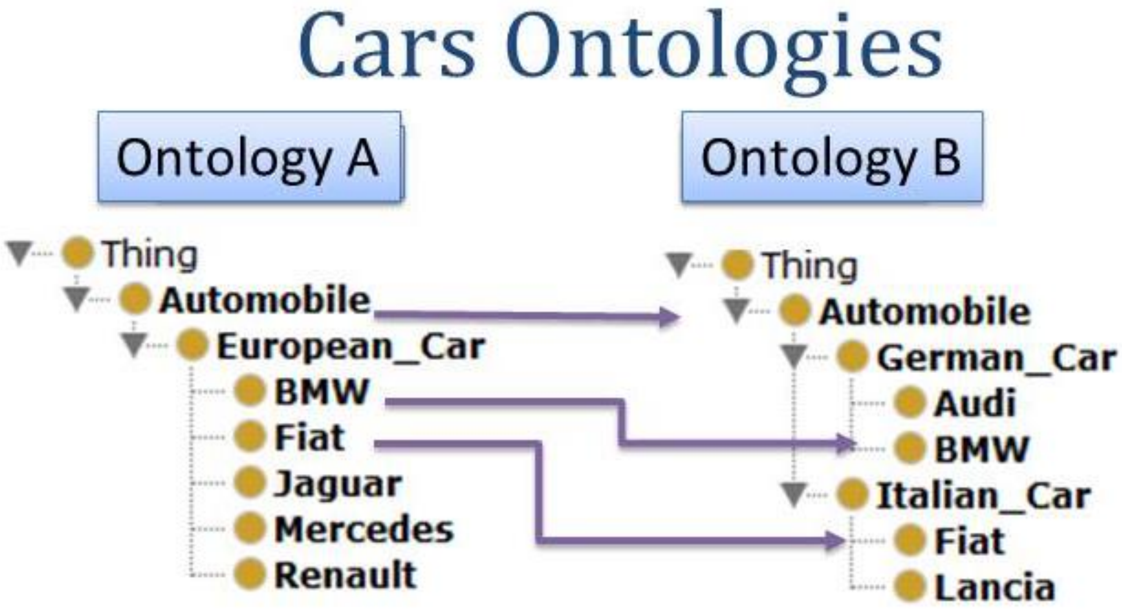
e) **Same names:** semantic similitude if **names from classes** are equals.

f) **Distance SMOA name** (A String Metric for Ontology Alignment): similitude between entities as **parts in common minus their differences**.

g) **String Distance:** takes classes, properties and instances **as simples strings** to compare them.

h) **Sub structures distance:** based in finding similitude taking into consideration the **graph structure** from the **sub-classes**.

Experiments

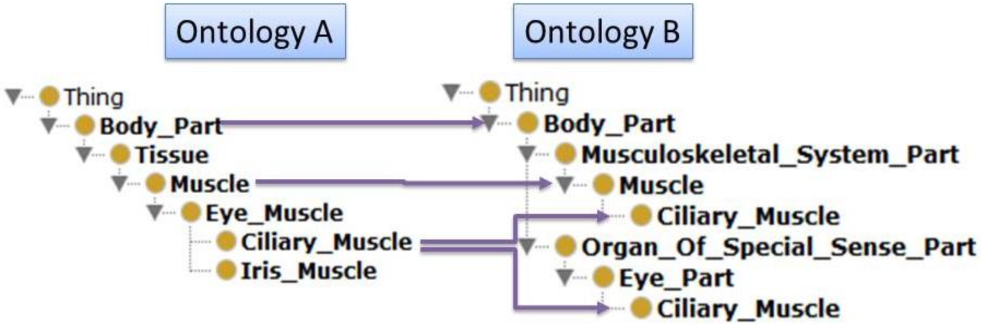


We use different pairs of ontologies to test our ABC algorithm:

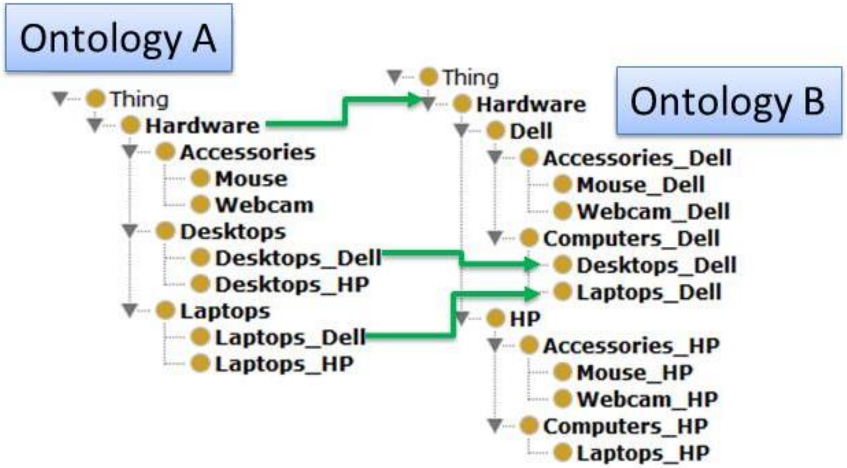
- **One pair about cars,**
- **Another of the anatomy of the eye, and finally**
- **One pair about computers**

Experiments

Anatomy Sub-set



Computers Ontologies



Results

Set	Execution Time (sec)	# Aligned nodes	#Times that each Align Technique is chosen
Cars	1:00	0	a) 0
	0:50	3	b) 6
	0:51	3	c) 3
	0:49	3	d) 2
	0:41	**3**	**e) 7**
	1:00	**3**	**f) 7**
	0:59	3	g) 3
	1:12	3	h) 2
Anatomy Sub set	0:50	0	a) 0
	0:59	3	b) 0
	0:53	3	c) 0
	0:47	5	**d) 12**
	0:45	3	e) 0
	0:35	5	f) 8
	0:59	3	g) 0
	0:53	5	h) 10
Computers	1:02	0	a) 0
	1:06	4	b) 0
	0:56	7	c) 0
	0:49	9	d) 12
	0:49	4	e) 0
	0:46	**9**	**f) 14**
	1:06	4	g) 0
	0:59	9	h) 4

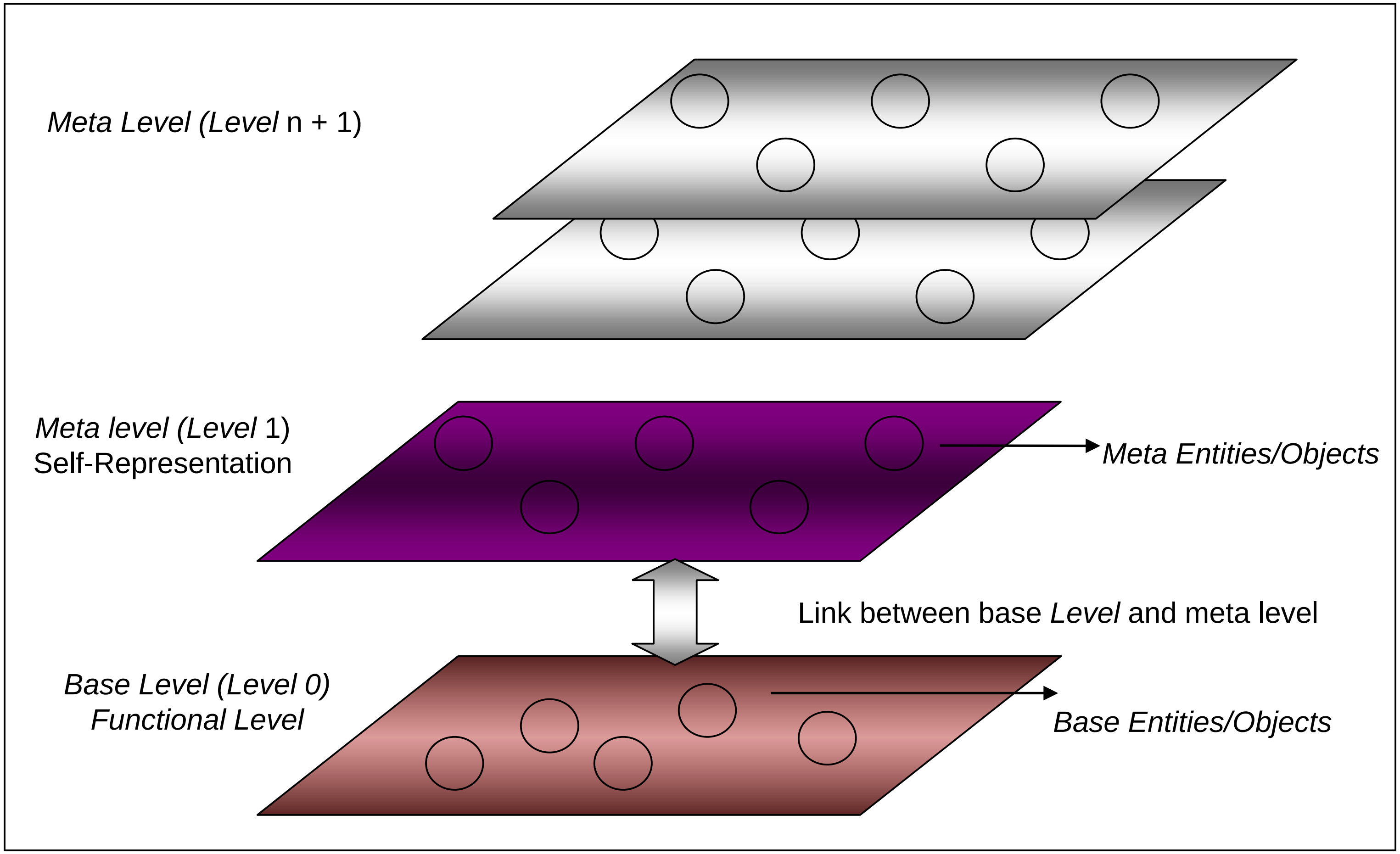
In this cases, all techniques from b) to h) align **the same numbers of nodes**, as the **times are closed**, it is chosen any of this techniques

In this two cases the number of aligned nodes are different, it is chosen any of the techniques with the bigger **quantity of aligned nodes**, **choosing the one with less time**.

Max cycle = 25 runs
Colony size = sources of nectar

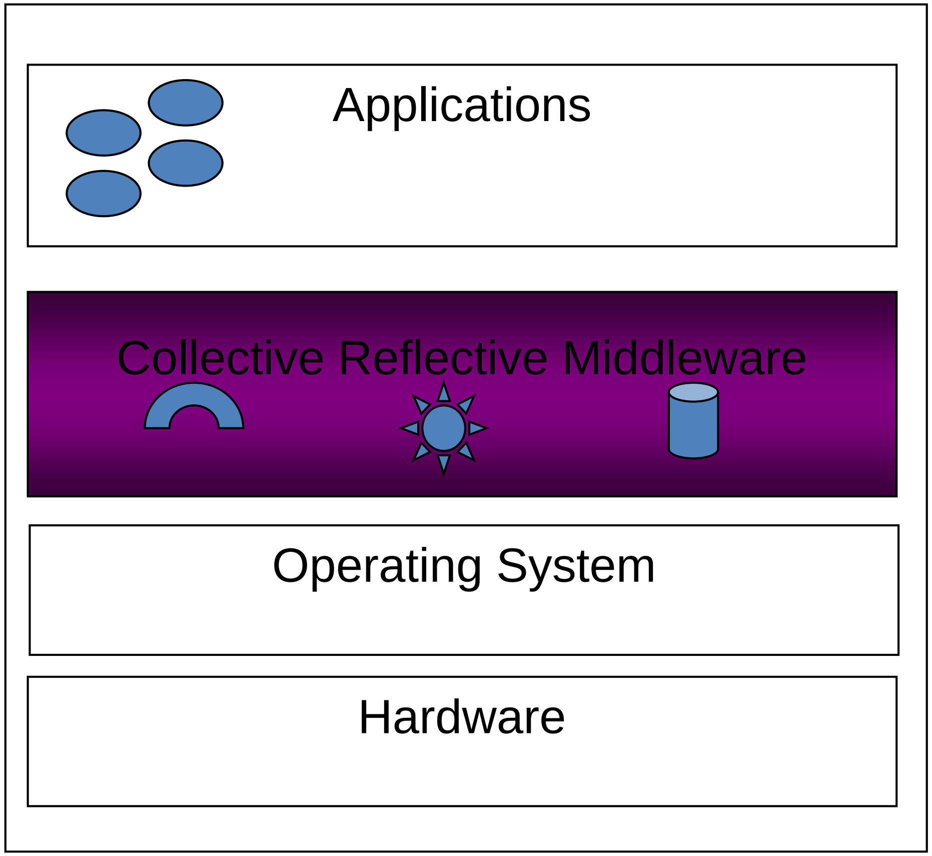
Aplicación 2

Reflection

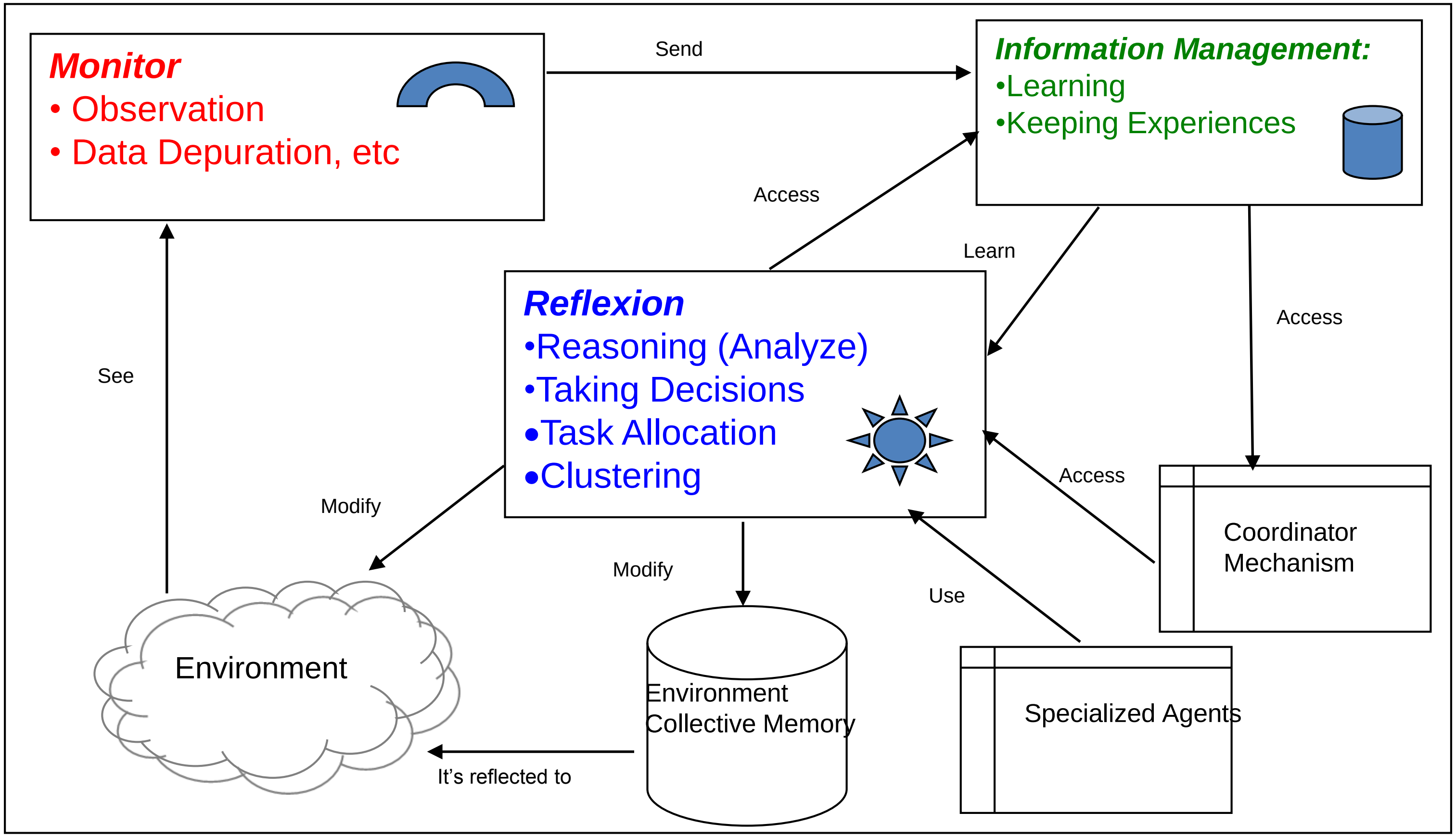

Meta Level (Level n + 1)
Meta level (Level 1)
Self-Representation
Meta Entities/Objects
Link between base *Level* and meta level
Base Level (Level 0)
Functional Level
Base Entities/Objects

"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”



Monitor

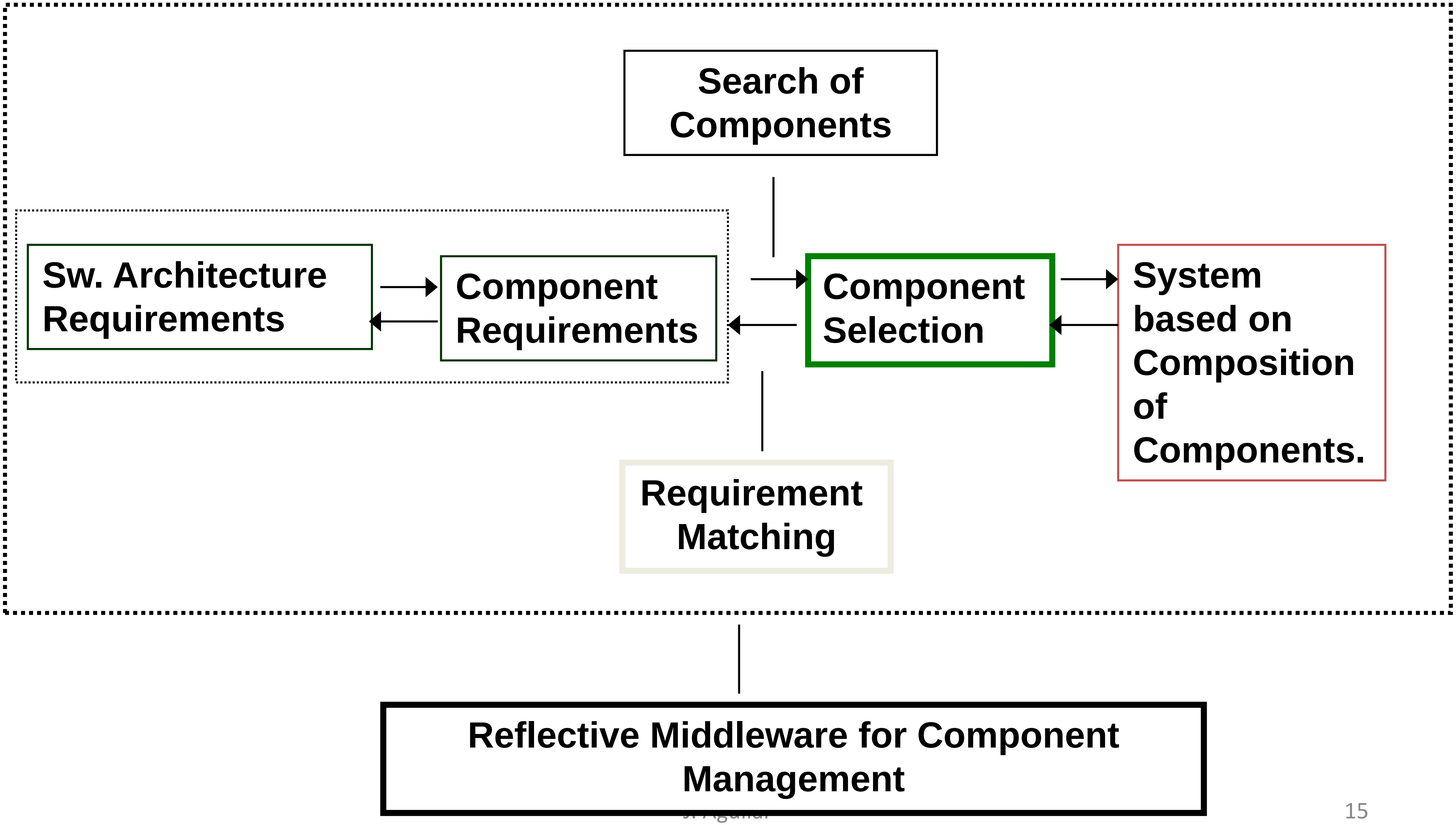
- Observation
- Data Depuration, etc

Information Management:

- Learning
- Keeping Experiences

Reflexion

- Reasoning (Analyze)
- Taking Decisions
- Task Allocation
- Clustering



Search of Components

Sw. Architecture Requirements

Component Requirements

Component Selection

System based on Composition of Components.

Requirement Matching

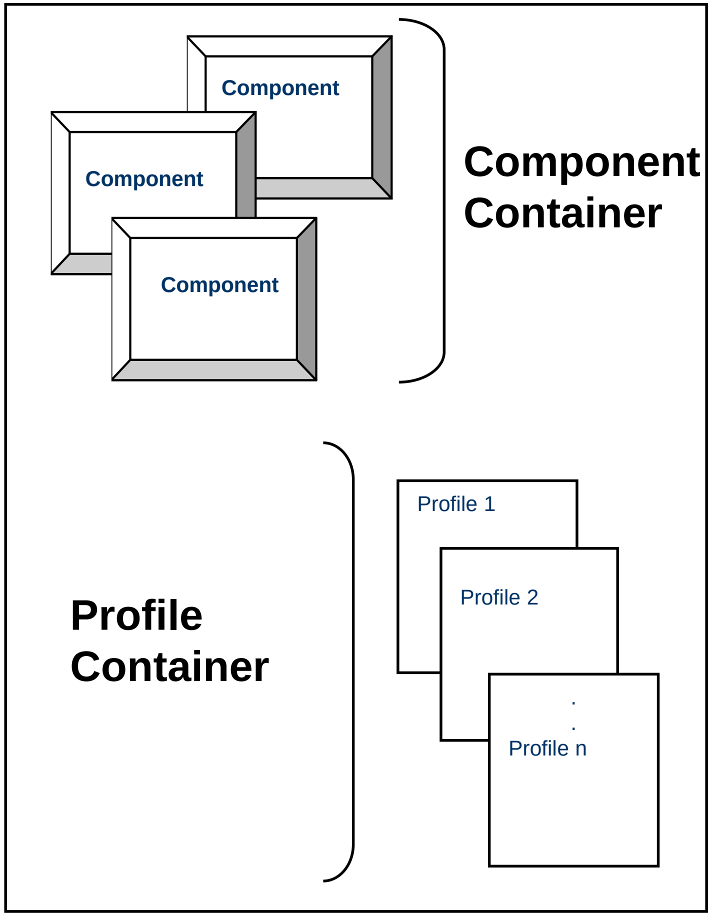
Reflective Middleware for Component Management

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.

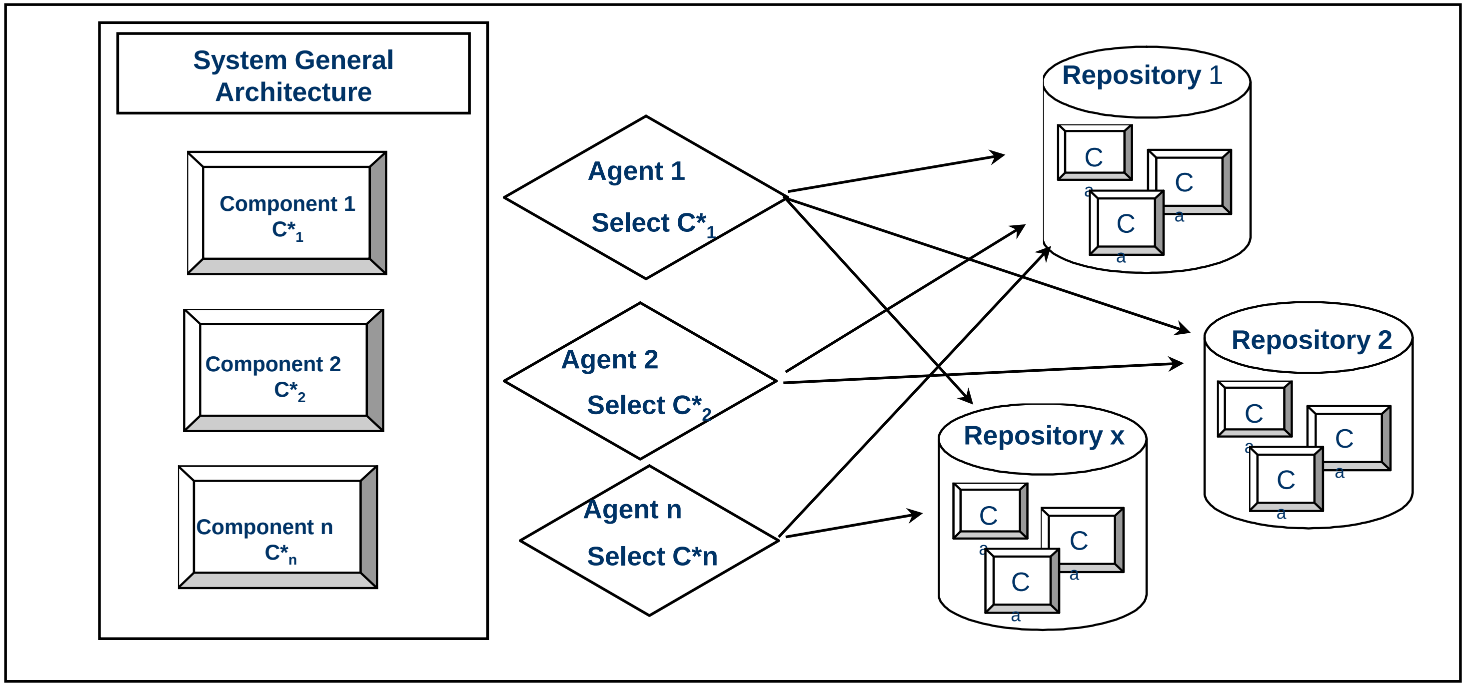
Component Selection Algorithm



```
<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):

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

X -> Component Matching
H -> Ideal Component Characteristics
N -> Real Characteristics
j: Component, l: Repository, u: Profile, k: Agent.

Selection Probability (Equation 2):

$$P_{lju}^{k}(t) = \left[Y_{lju}(t)\right]\left[X^{k}{}_{lju}\right]^{-1} \Big/ \sum_{rsn}\left[Y_{rsn}(t)\right]\left[X^{k}{}_{rsn}\right]^{-1}$$

Component Selection Algorithm

Pheromone Update (equation 5)

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

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

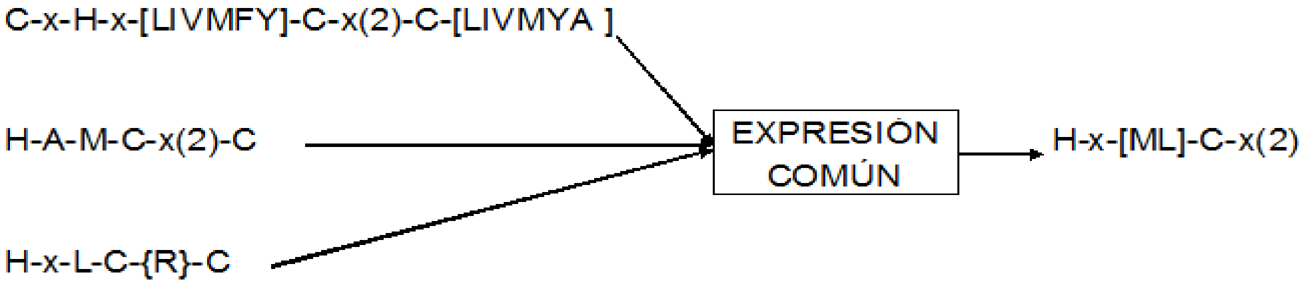
$$R = f(ET, M) \qquad (4)$$

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

Aplicación 2

PROBLEMA

Fusion de Motivos



C-x-H-x-[LIVMFY]-C-x(2)-C-[LIVMYA]

H-A-M-C-x(2)-C

H-x-L-C-{R}-C

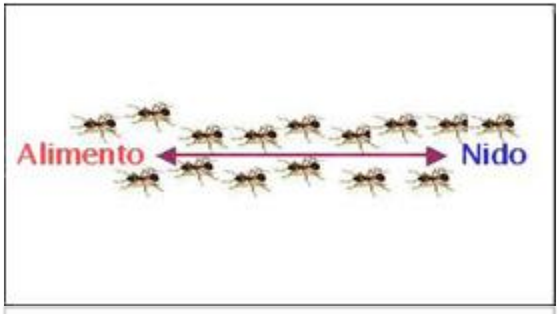
EXPRESIÓN COMÚN

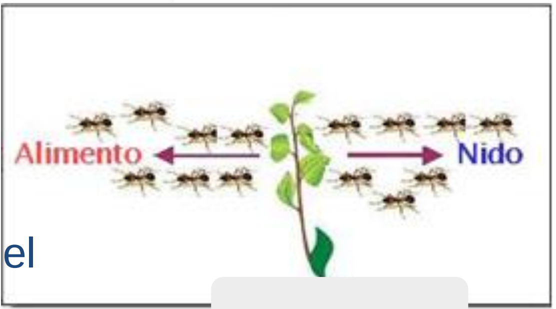
H-x-[ML]-C-x(2)

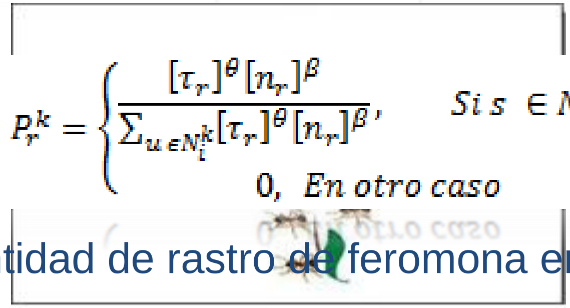
Construir un patrón común para las expresiones regulares que tienen un alto grado de semejanza

FUSION DE MOTIVOS

Colonia de Hormigas





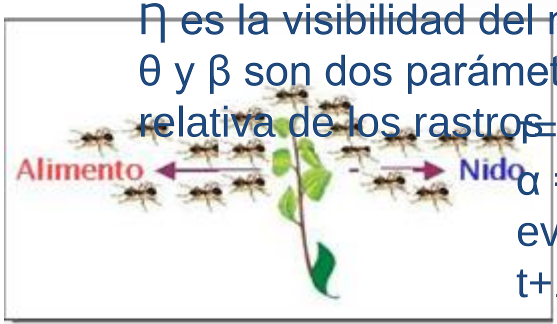


$$P_r^k = \begin{cases} \dfrac{[\tau_r]^\theta [n_r]^\beta}{\sum_{u \in N_i^k} [\tau_r]^\theta [n_r]^\beta}, & Si\ s\ \in N_k(i) \\ 0, & En\ otro\ caso \end{cases}$$

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

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

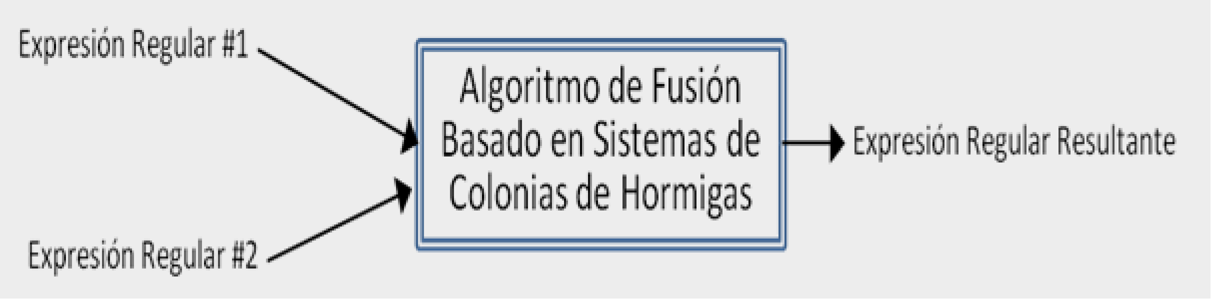
$$\Delta\tau_r = \sum^{m} \Delta\tau_r^k$$

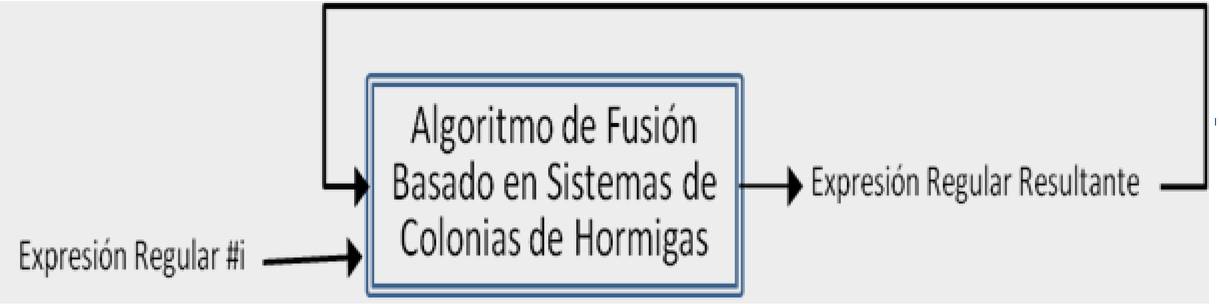


r= Intensidad de la feromona en el momento t
α = coeficiente tal que (1 – α) representa la tasa de evaporación de la feromona entre el intervalo de tiempo t y t+1 y Δτr = cantidad de feromona dejada en un nodo r

FUSION DE MOTIVOS

1. **Crea**
2. **Rec**
3. **Selección de los mejores nodos.**
4. **Con**





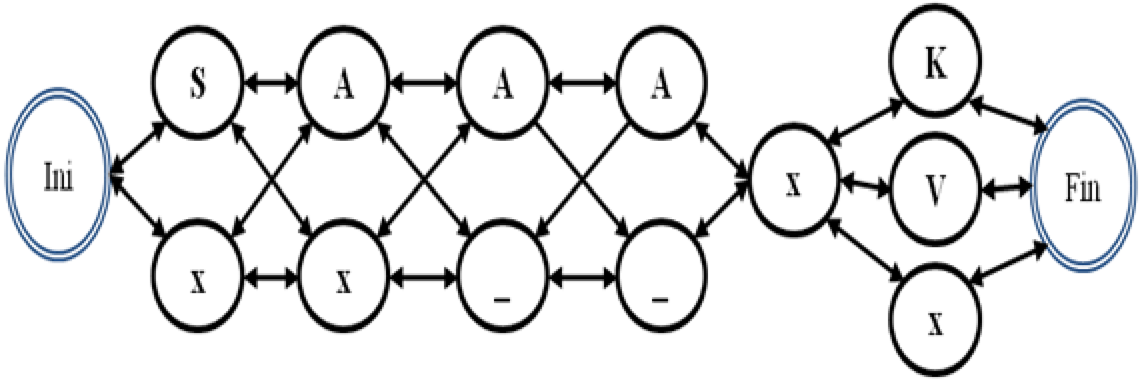
SUB-SISTEMA DE FUSION DE MOTIVOS

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]

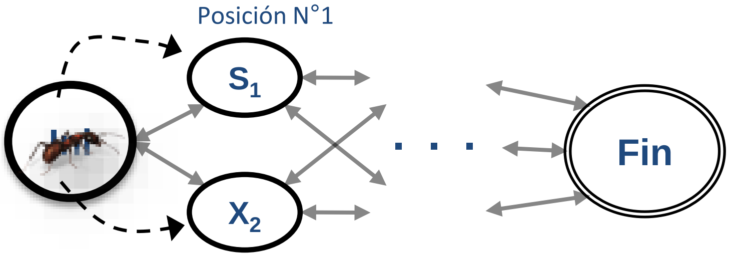


FUSION DE MOTIVOS

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.

ER2: L(2)-A(2)-Q

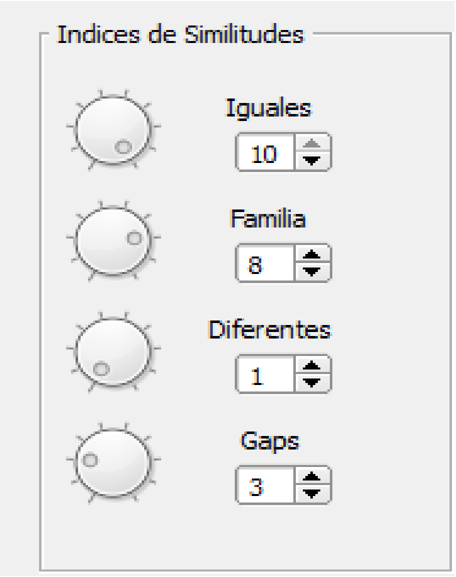
L(2)
A(2)
Q

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: 10

Familia: 8

Diferentes: 1

Gaps: 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.

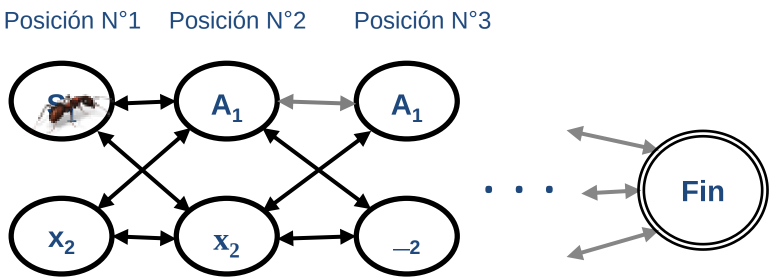
FUSION DE MOTIVOS

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

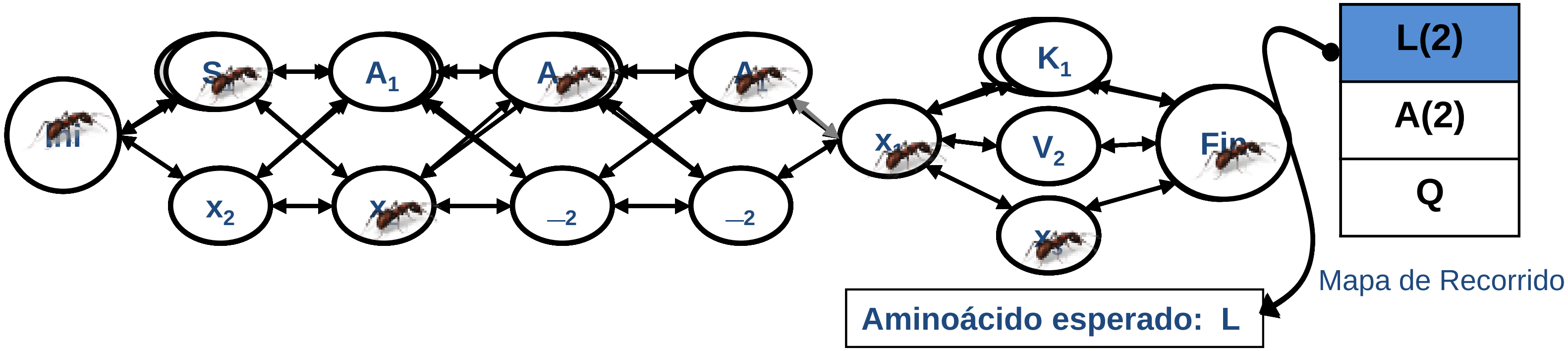
$$\tau_r^k = \tau_r^k + \sigma * \varphi_r^k$$



L(2)
A(2)
Q

FUSION DE MOTIVOS

Recorrido de la Colonia de Hormigas



Aminoácido esperado: L

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

Cuando una hormiga se desplaza a un nodo del grafo deposita un rastro de feromona

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

$$1 \quad \text{si } n = 1$$

P(r) = probabilidad de la función de similitud

τ = concentración actual de feromona

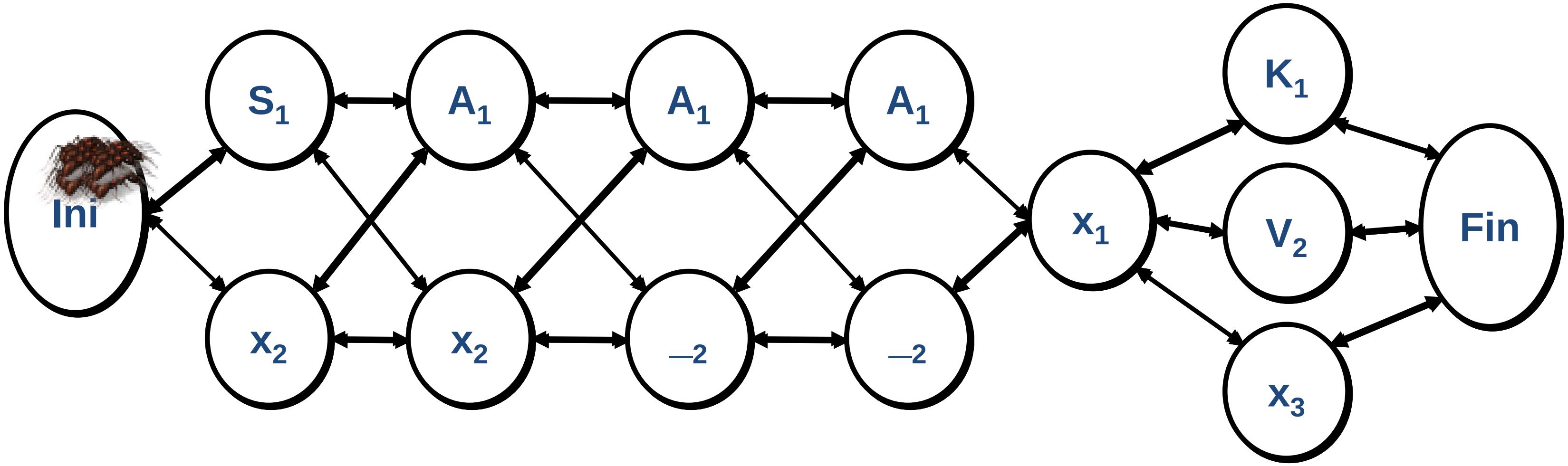
σ = coeficiente de incremento de feromona

“φr” = 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 los niveles de feromona en todos los nodos del grafo)

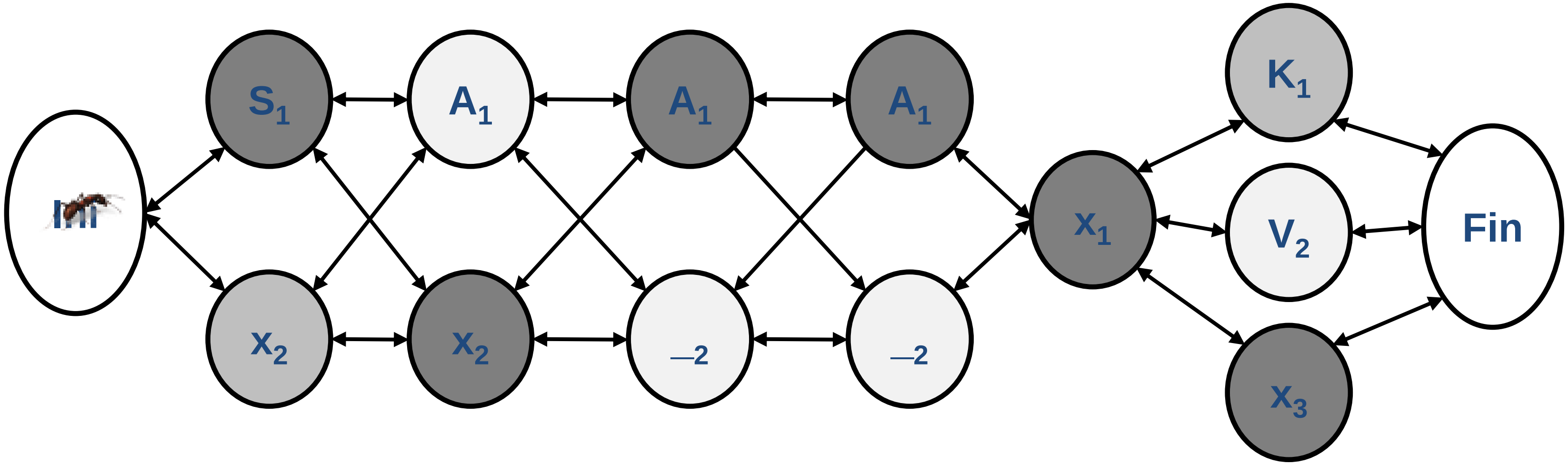
$$\tau_r^k = (1-\rho) * \tau_r^k$$

ρ = coeficiente de evaporación de feromona

FUSION DE MOTIVOS

Selección de los mejores nodos

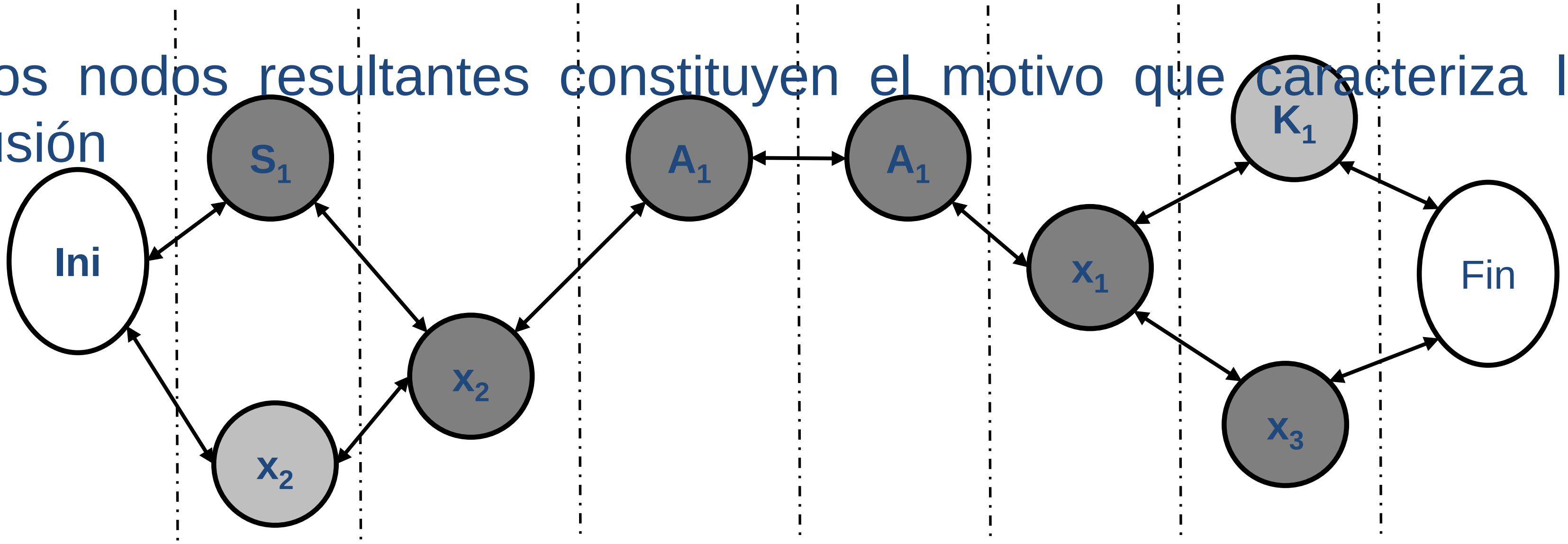
Se fija un valor de umbral para el nivel de feromona y se procede a escoger los mejores nodos



FUSION DE MOTIVOS

Construcción de la Expresión Regular de la Fusión

Los nodos resultantes constituyen el motivo que caracteriza la fusión



S	x	A	A	x	x

⇩

S	x	A(2)	x(2)

⇩

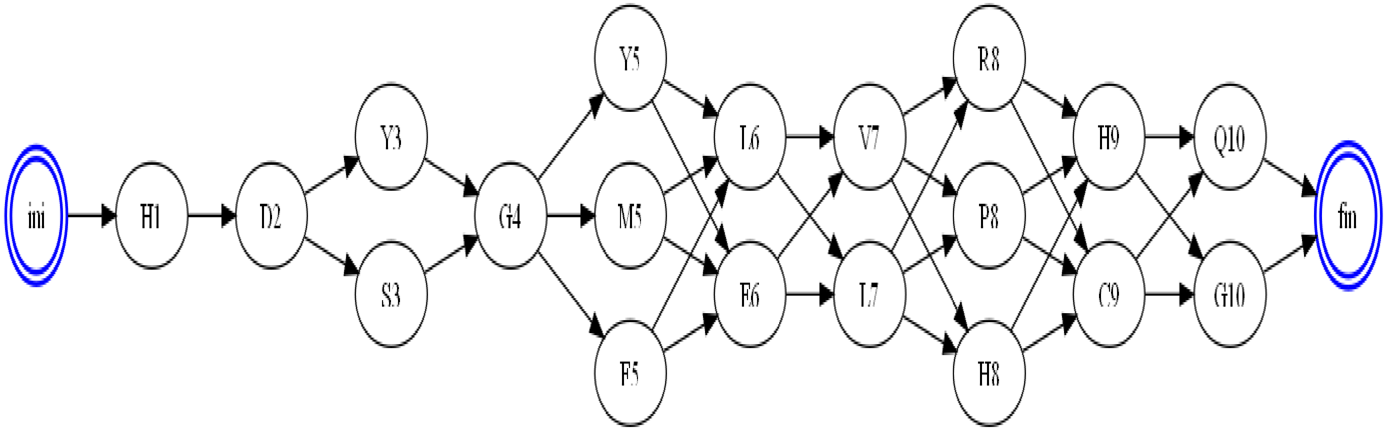
S-x-A(2)-x(2)

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]

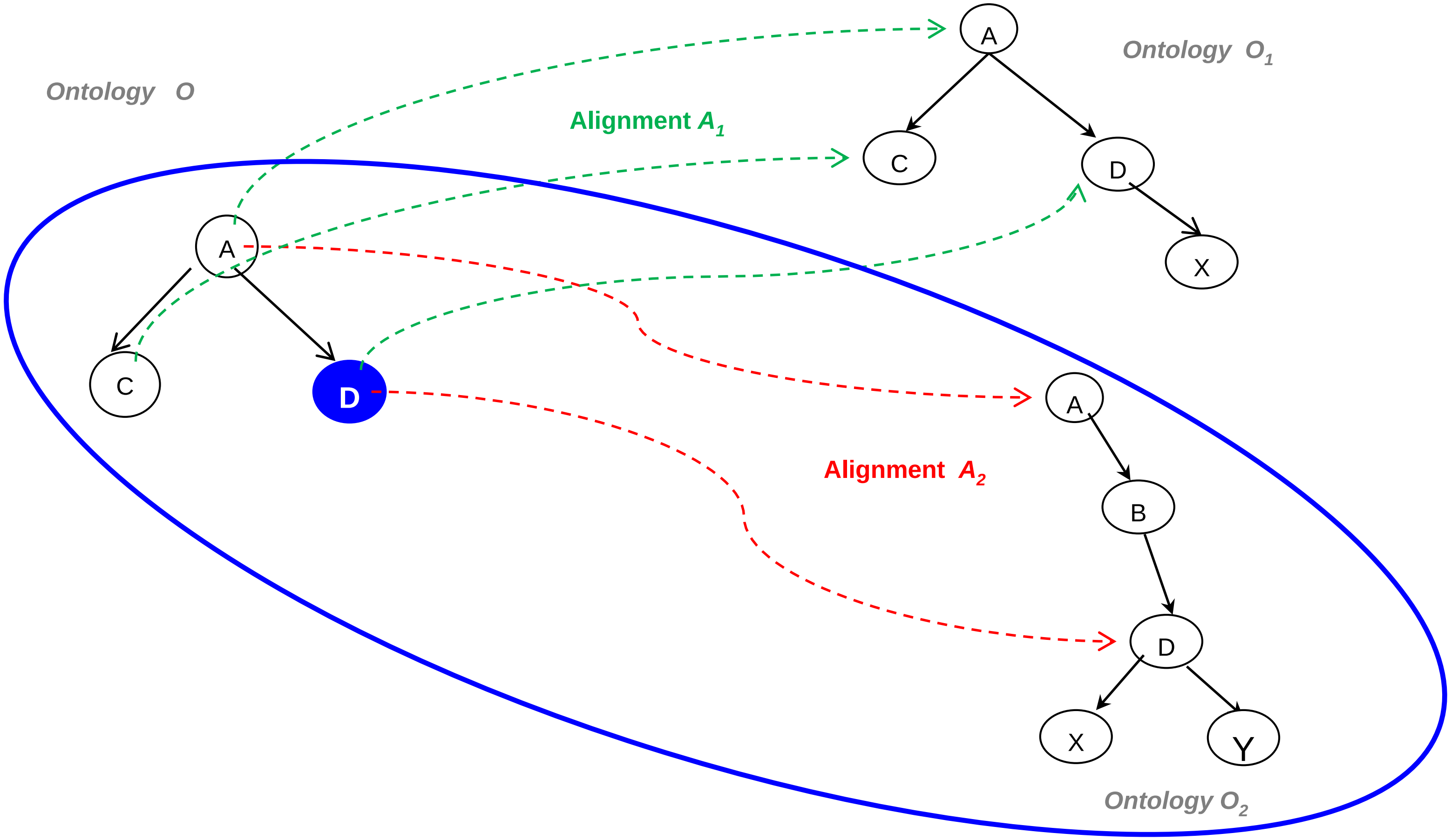
Familia	**Nombre de la Proteína**	**Organismo**
Proteína Precursora Amiloidea (APP)	A4_BOVIN (Q28053)	Bos tauros (Bovino)
Proteína Precursora Amiloidea (APP)	A4_CANFA(Q28280)	Canis familiaris (Perro)
Proteína Precursora Amiloidea (APP)	A4_CAVPO(Q60495)	Cavia porcellus (Cerdo de Guinea)
Proteína Precursora 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)

An approach for Multiple Combination of Ontologies based on the Ants Colony Optimization Algorithm

PROPOSAL

The enrichment of an objective ontology from multiple ontologies, can be defined as an optimization problem

The idea to apply ACO in the process of combining one ontology with other **N** ontologies, in which alignments already exists, is to determine the most appropriate alignment for each concept of the ontology, in relation to the criteria of: higher similarity and enrichment.



PROPOSAL

For each concept, the ants can select one of the alignments as part of the solution, through the **heuristic information** as a result of a similarity measure among the concepts, and also through the pheromone trails.

The ant **will explore among all possible solutions and will build the solution in a graph**, that will take account all the possible alignments that exists for each one of the concepts.

The alignments must be already defined, based on concepts of the objective ontology, in order to build the graph.

To apply ACO algorithm to the problem is necessary to define:

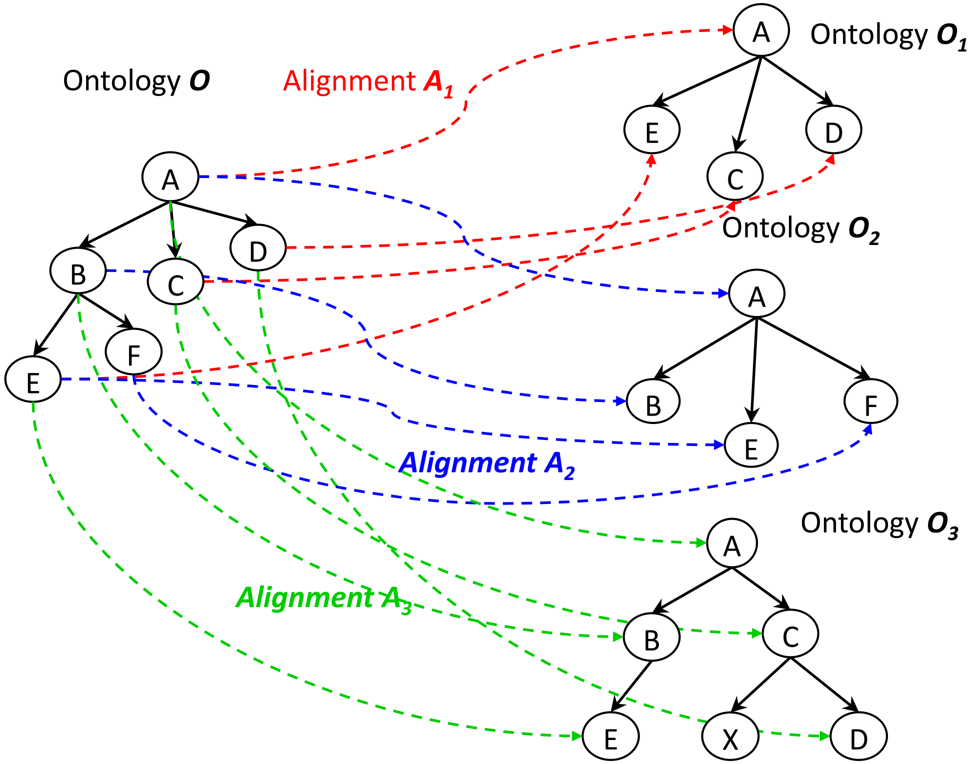
- The Space of Solutions
- The Similarity Measure
- Quality of the Solution

PROPOSAL

DETERMINATION OF THE SPACE OF SOLUTIONS

To the application of ACO in a problem is necessary to define the space of solutions that will visit the ants colony in search of the optimal solution.

For the enrichment of the ***O*** ontology using the ontologies ***O_1, O_2,…. O_N*** there are already defined a set of alignments ***A_1, A_2,… A_N*** and the idea is to select for each concept ***C*** belonging to ***O***, the best possible alignment from the ***N*** alignments already defined, **based on the highest similarity and enrichment** that may obtain the ***O*** ontology.



PROPOSAL

DETERMINATION OF THE SPACE OF SOLUTIONS

The alignments A_1, A_2 y A_3 already defined can be described as follows:

A_1 = {(A, A'_1), (C, C'_1), (D, D'_1), (E, E'_1) }
A_2 = {(A, A'_2), (B, B'_2), (E, E'_2), (F, F'_2) }
A_3 = {(A, A'_3), (B, B'_3), (C, C'_3), (D, D'_3), (E, E'_3) }

Among each pair of concepts must be defined, by the existing alignments, a similar measure:

Sim(C,C')

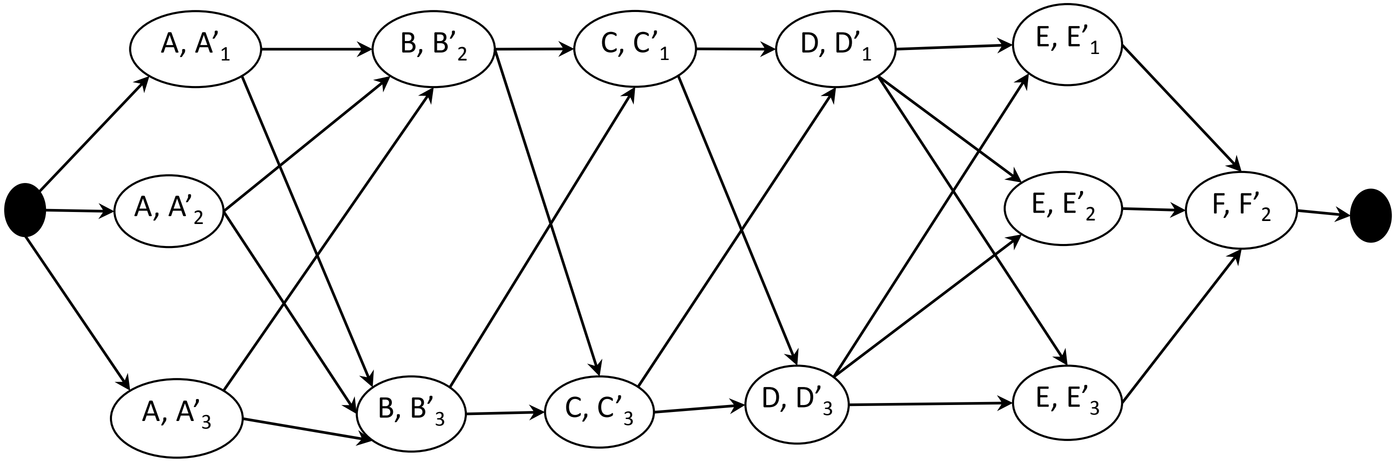
This similarity measure must be ranging from 0 to 1, which refers to the **grade of similarity with the aligned concepts, that can be given for a lexical similarity**.

PROPOSAL

DETERMINATION OF THE SPACE OF SOLUTIONS

For **M** concepts in the source ontology and **N** alignments, we have a maximum of $\mathbf{N}^{\mathbf{M}}$ possible solutions for this problem.

For this specific case of ACO, it must be defined a starting point where all ants start the route, in this case it will be the root concept of the source ontology that we want to enrich. Each ant starts selecting the concept that it considers the best alignment of the root concept, then with that selection as part of the solution, it selects the best alignment for the next concept,



PROPOSAL

DETERMINATION OF THE SIMILARITY MEASURE

The “Similarity Measure” between *C* and *C’* concepts for to select the best alignment for a concept *C*, a comparison is made based on three aspects:

$$MS(C,C') = \frac{SA(C,C') + SD(C,C') + SS(C,C')}{3}$$

SA : Similarity of Ancestor of C and C’

SD: Similarity of Siblings of C and C’

SS: Similarity of Descendants of C and C’

PROPOSAL

Similarity of Ancestor (SA)

The similarity of C and C’ concepts will be proportional to the similarity of ancestral concepts. In this case, **a relative percentage to each ancestor of concept C that is also aligned to an ancestor of concept C is given as**

$$SA(C,C') = PC_A \times Sim(C,C') + \frac{2(1-PC_A)}{n(n+1)} \sum_{j=1}^{m} \sum_{i=1}^{n} (n+1-i) Sim(Anc_i(C), Anc_j(C'))$$

Where:

- PC_A: It is a percentage of contribution of the similarity between C and C’, and determines the contribution of the similarity of the ancestors.
- Sim(C,C’) : similarity measure between C and C’ concepts.
- Anc_i (C) : ancestor i between concept C and the root.
- Anc_j (C’) : ancestor j between concept C’ and the root.
- n: It is the number of ancestors of concept C.
- m: It is the number of ancestors of concept C’

PROPOSAL

Similarity of Siblings (SS)

The similarity of C and C’ concepts will be proportional to the similarity with the siblings**.** In this case, **a percentage is given to each sibling of C concept that is also aligned with a sibling of C’ concept**, as follows:

$$SS(C,C') = PC_S \times Sim(C,C') + \frac{1-PC_H}{n} \sum_{i=1}^{n} \max\left(Sim(S_i,S'_1), \ldots, Sim(S_i,S'_m)\right)$$

<u>Where:</u>

- PC_S: It is the percentage of contribution of the similarity between C and C’ and determines the contribution of the similarity of siblings.
- Sim(C,C’) : similarity measure between C and C’ concepts.
- S_i: It corresponds to the i sibling of C concept.
- S'_m: It corresponds to the m sibling of C’ concept.
- n: It is the number of siblings of C concept.
- m: It is the number of siblings of C’ concept.

PROPOSAL

Similarity of Descendants (SD)

The similarity between the two concepts C and C' will also be proportional to the similarity of direct descendants. In this case, a percentage to each child of concept C is given, which is also aligned with a child of concept C' as follows:

$$SD(C,C') = PC_D \times Sim(C,C') + \frac{1-PC_D}{n} \sum_{i=1}^{n} \max\left(Sim\left(H_i H'_1\right), \ldots, Sim(H_i, H'_m)\right)$$

Where:

- PC_D: It is a percentage of contribution of the similarity between *C* and *C*' and determines the contribution of the similarity of children.
- Sim(C,C') : similarity measure between C and C' concepts.
- H_i: It corresponds to the *i* child of concept *C*.
- H'_m: It corresponds to the *m* child of concept *C'*.
- *n:* It is the number of children of concept *C*.
- *m:* It is the number of children of concept *C'*.

PROPOSAL

QUALITY OF THE SOLUTION

To determine the quality of the selected alignment, we will consider a criterion that takes account the **“Grade of Enrichment” (GE),** which will be an indicator of the amount of new concepts obtained by the source ontology after selecting an alignment for a concept. The **GE** of the ontology after selecting the alignment for a ***C*** concept with ***C’*** from new concepts that can be added to the ontology correspondents to:

- Children concepts of ***C’*** non-aligned (New **Hyponyms**) and their descendants.
- Siblings concepts of ***C’*** non-aligned with aligned immediate ancestor (father) (New **Cohyponyms**) and their descendants.
- Ancestors concepts of ***C’*** non-aligned (**New Hyperonyms**).

The non-aligned concepts are those that do not have any input in the table “Similarity Values among Concepts”.

$$GE(C,C') = CHildren\_Non\_Aligned\,(C') + Siblings\_Non\_Aligned(C') \\ + Ancestors\_Non\_Aligned(C')$$

PROPOSAL

THE ACO ALGORITHM

Probability of Transition

Determines the route that the ant selects. In order to build the solution, each ant must choose the next element of the solution from the *"r"* location. To do this, it uses a function of probability to select the element *"s"* that is calculated as follows

$$P^{k}_{(r,s)} = \frac{\gamma_{(r,s)}{}^{\alpha} \cdot \eta_{(r,s)}{}^{\beta}}{\sum_{u \in J^{k}_{r}} \gamma_{(r,u)}{}^{\alpha} \cdot \eta_{(r,u)}{}^{\beta}} \quad \text{Si } s \in J^{k}_{r}$$

Where:

γ_{rs} : is the amount of pheromone.

η_{rs} : is the heuristic information (Similarity Measure (MS)).

J^{k}_{r} : is the node not visited yet by the k ant from r.

α y β: define the importance of the memory information (pheromone) and heuristic information.

PROPOSAL

Pheromone Updating

While an ant is in the construction process of the solution, each selected edge must update the pheromone, delivering an amount of pheromone, according to the following function:

$$\gamma_{(r,s)} = \gamma_{(r,s)} + \Delta\gamma_{(r,s)}$$

Where:

$\Delta\gamma_{(r,s)}$: It is the increment of the pheromone, corresponding to sum of amounts of pheromone leaving by ants in the edge (r, s);

$$\Delta\gamma_{(r,s)} = \sum_{k=1}^{M} \Delta\gamma_{(r,s)}^{K}$$

Where:

$\Delta\gamma_{(r,s)}^{K}$: It is the amount of pheromone leaving by k ant in the edge (r,s) which is directly related to the “Quality of the Solution” found by k ant.

$$\Delta\gamma_{(r,s)}^{K} = f(GE(C, C_s))$$

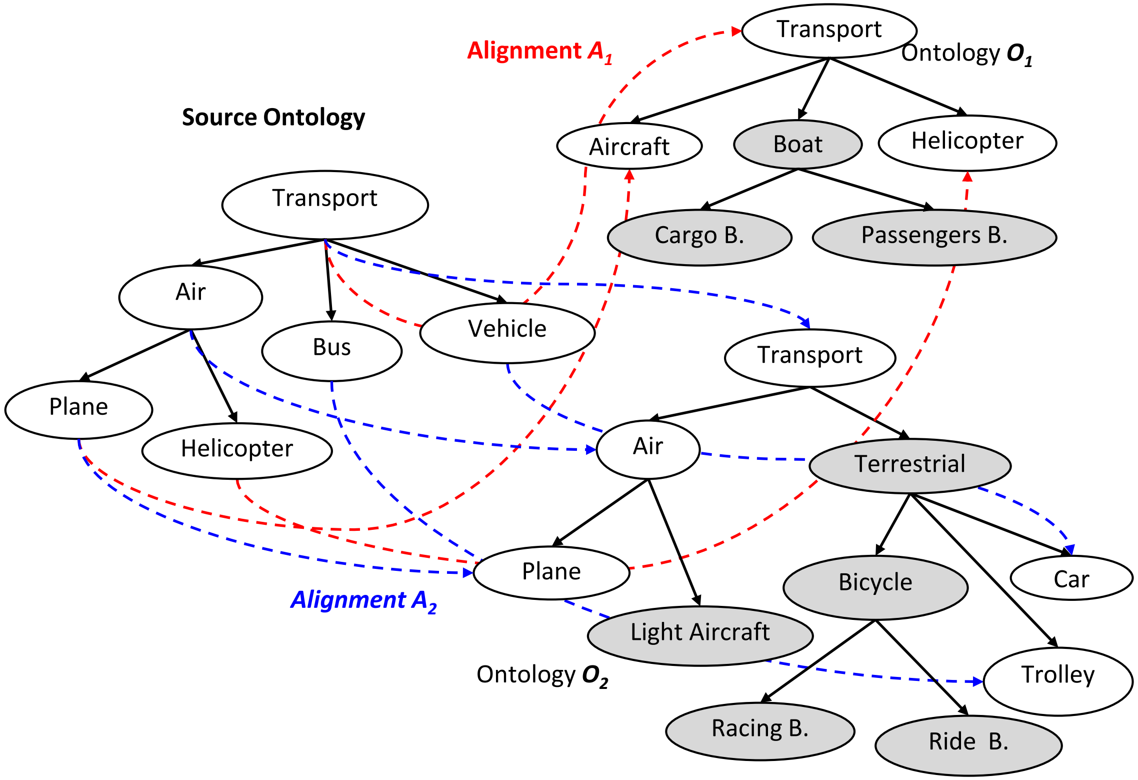
Where:

C_s: selected alignment by k ant as the best solution for C

$GE(C, C_s)$: Grade of enrichment of the alignment.

EXAMPLE

Alignments among Ontologies about Transportation



SIMILARITY VALUES AMONG CONCEPTS

Ontology O	Ontology O1	Sim (C,C´)
Transport	Transport	1
Air	-	-
Bus	-	-
Vehicle	-	-
Plane	Aircraft	0.8
Helicopter	Helicopter	1
Ontology O	**Ontology O2**	**Sim(C,C´)**
Transport	Transport	1
Air	Air	1
Bus	Trolley	0.5
Vehicle	Car	0.8
Plane	Plane	1
Helicopter	-	-

In this example there are 6 concepts in source ontology and 2 alignments, the space of solutions is 2^6 = 64 possible solutions.

EXAMPLE

POSIBLE SOLUTIONS AND ITS GRADES OF ENRICHMENT

Sol.	**Alignments Concepts**						**GE**	**New Concepts**							
	C_1	C_2	C_3	C_4	C_5	C_6		Boat	Car. B.	Pas. B.	Terr	Air L.	Bicy	Rac. B.	Rid.B.
1	1	1	1	1	1	1	3	x	x	x					
…															
24	**1**	**2**	**1**	**2**	**2**	**2**	**8**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
25	**1**	**2**	**2**	**1**	**1**	**1**	**8**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
…															
49	**2**	**2**	**1**	**1**	**1**	**1**	**8**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
50	**2**	**2**	**1**	**1**	**1**	**2**	**8**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
…															
62	**2**	**2**	**2**	**2**	**1**	**2**	**8**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
63	**2**	**2**	**2**	**2**	**2**	**1**	**8**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
64	2	2	2	2	2	2	5				x	x	x	x	x

EVALUATION RESULTS

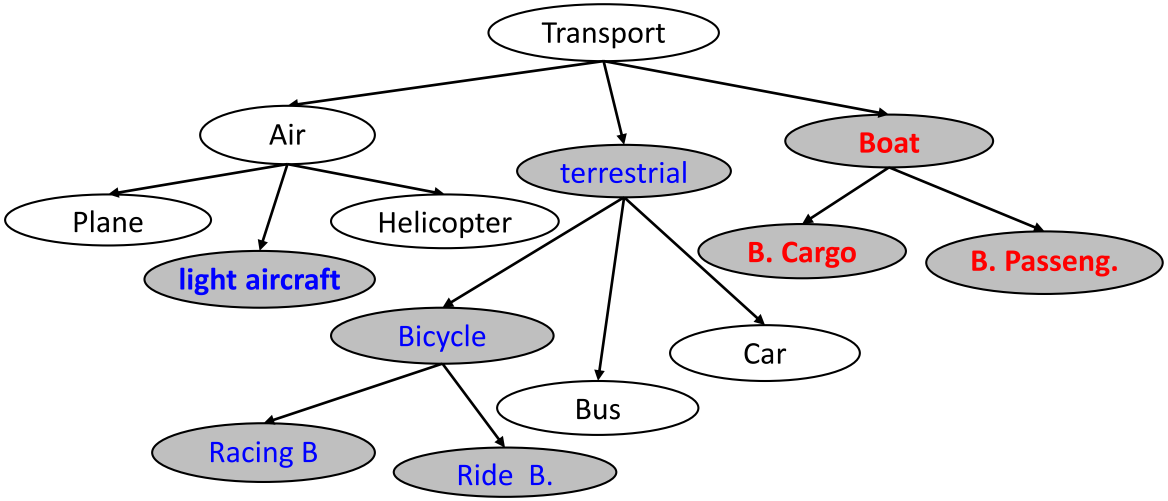
	Parameters						**GE = 8 (% converg)**
Sim.	**Alpha**	**Beta**	**evap rathe**	**Inic. pherom**	**N. Ant**	**N Iter.**	
1	0.2	0.2	0.1	1	50	100	40%
2	0.2	0.5	0.1	1	50	100	20%
3	0.2	0.8	0.1	1	50	100	30%
4	**0.5**	**0.2**	**0.1**	**1**	**50**	**100**	**90%**
5	0.5	0.5	0.1	1	50	100	30%
6	0.5	0.8	0.1	1	50	100	20%
7	**0.8**	**0.2**	**0.1**	**1**	**50**	**100**	**90%**
8	**0.8**	**0.5**	**0.1**	**1**	**50**	**100**	**80%**
9	0.8	0.8	0.1	1	50	100	40%
10	0.9	0.9	0.1	1	50	100	50%

Values of the last row tell us about proportion of ants that converge on the same solution.

- Percentages of higher convergences, is for values of α a little bit higher than β.
- When value of α is higher, means that a higher importance to pheromone is given, that is, to collective information.
- If the value of β is higher, more relevance is given to heuristic information, i.e, to similarity value.

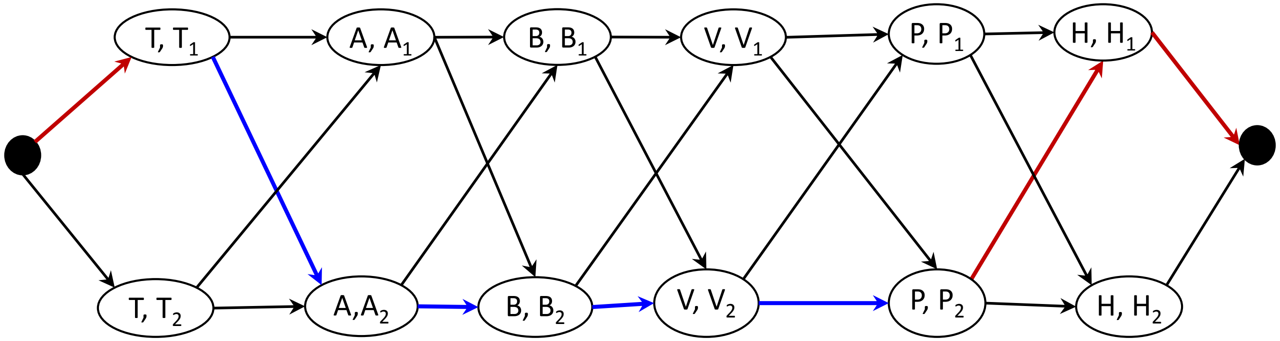
EXAMPLE

A possible resultant ontology after combination process, with a GE of 8, where new concepts acquired by ontology are highlighted.



This is a route graph for the solution obtained

Concepts
Transport (T)
Air(A)
Bus (B)
Vehicle (V)
Plane (P)
Helicopter (H)



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.

$$ind_i(j,t) = \{j \mid d(i,j,t) \leq g\_radio_i, 1 \leq j \leq n, j \neq i\}$$

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

$$dir_a(t) = \arctan \frac{\sum_{j=1}^{k} \sin\left(cabecera_j(t)\right)}{\sum_{j=1}^{k} \cos\left(cabecera_j(t)\right)} \qquad j \in ind_i(j,t)$$

Sincronización de las velocidades en el grupo

$$dir_c(t) = \arctan \frac{\sum_{j=1}^{k} \sin\left(cabecera_j(t) - cabecera_i(t)\right)}{\sum_{j=1}^{k} \cos\left(cabecera_j(t) - cabecera_i(t)\right)} \qquad j \in ind_i(j,t)$$

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