

# Efectos de la comunicación directa entre agentes en los algoritmos de agrupación de clases basados en el comportamiento de insectos sociales

Marco A. Montes de Oca, Leonardo Garrido y José Luis Aguirre

Centro de Sistemas Inteligentes.  
Tecnológico de Monterrey, Campus Monterrey.  
Eugenio Garza Sada. No. 2501. Col. Tecnológico.  
Monterrey, N.L. México. C.P. 64849  
{A00788072, leonardo.garrido, jlguirre}@itesm.mx

## Resumen

Los algoritmos de agrupación de clases basados en el comportamiento colectivo de algunas especies de insectos sociales son herramientas para el descubrimiento de conocimiento que se basan en la comunicación estigmérgica, esto es, en la comunicación indirecta entre agentes a través de modificaciones locales al ambiente. Inspirados en algunos aspectos del fenómeno de la trofalaxis, o el intercambio de alimento líquido entre insectos, estudiamos dos estrategias de comunicación directa entre agentes en este tipo de algoritmos: (i) compartición de memoria y (ii) compartición de mapas ambientales. El objetivo del intercambio de información entre agentes es mejorar la calidad de la agrupación final. El efecto de la inclusión de estos mecanismos de comunicación se evalúa mediante la comparación de la calidad de la agrupación final obtenida con comunicación directa y sin comunicación (aparte de la estigmérgica). Se muestra además, que los resultados dependen de la densidad de agentes en el ambiente, así como de la utilidad de la información intercambiada.

**Palabras clave:** Agrupación de clases, Algoritmos, Insectos sociales, Comunicación directa, Agentes.

## 1. Introducción

Algunos comportamientos colectivos de los insectos sociales son resultado de un proceso de autoorganización [9]. Bonabeau et al. [2] definen autoorganización como “un conjunto de mecanismos dinámicos en los que aparecen estructuras a nivel global a partir de interacciones entre sus componentes de menor nivel” (p. 9). Normalmente, para que se pueda dar la autoorganización, es necesario contar con un mecanismo que permita múltiples interacciones entre los componentes de un sistema. En el caso de las colonias de insectos sociales, uno de los mecanismos que permiten tales interac-

ciones es la *estigmergia*.

La stigmergia puede ser definida como la influencia indirecta en el comportamiento de otros a través de modificaciones locales al ambiente. El término fue propuesto por Grassé [7], quien observó que cuando termitas trabajadoras de la especie *Macrotermes bellicosus* eran puestas en un contenedor con algunos granos de arena, los insectos cargaban y depositaban los granos aparentemente de forma aleatoria después de una fase de exploración en la cual los insectos se movían por el contenedor sin llevar a cabo ninguna acción. En esta etapa, un grano recién depositado por una termita es comúnmente recogido de nuevo y

puesto en otro lugar por otra termita. Cuando un grano es depositado sobre otro, la estructura resultante les es mucho más atractiva a las termitas para depositar otros granos cerca de ella. Esto hace a la zona aun más atractiva para seguir depositando granos [13].

La estigmergia no es la única forma de interacción entre insectos sociales. La *trofalaxis*, o el intercambio de alimento líquido entre miembros de la misma colonia, juega un papel importante en su organización social [13]. Considérese el caso de algunas especies de termitas que requieren de microorganismos intestinales para derivar beneficios de la celulosa. Sus larvas son alimentadas por alguna de las dos formas de trofalaxis existentes: oral y anal. La segunda permite la infección por bacterias contenidas en el líquido proctodeal. La asociación subsocial que resulta de esta codependencia ha evolucionado en una estructura social y morfológica bastante compleja [5].

La trofalaxis oral es la inspiradora del presente trabajo pues, en la mayoría de los casos, los individuos que participan en ella se comunican directamente para determinar quién será el proveedor y quién el consumidor del alimento a intercambiar [13]. Dado que la negociación previa al intercambio de alimento es parte inseparable del proceso, creemos que la trofalaxis es una buena metáfora para la interacción directa entre insectos artificiales.

En este trabajo investigamos los efectos que tiene en los algoritmos de agrupación de clases el intercambio directo entre agentes de información y conocimiento acerca de la distribución espacial de datos en el ambiente. La idea es que si dejamos a los agentes intercambiar información acerca de la distribución espacial de los datos en el ambiente, les debe ser posible enriquecer o actualizar sus propias representaciones del ambiente. El objetivo último del intercambio de información es permitir a los agentes elegir la mejor localidad en donde depositar un dato, y por tanto, crear mejores agrupaciones. Comparamos el desarrollo del proceso de agrupación del algoritmo de Lumer y Faieta [12] que utiliza memorias de corto plazo en los agentes con una extensión del mismo en donde los agentes comparten mutuamente sus memorias de corto plazo. Además, se evalúa un algoritmo en el que los agentes crean y comparten mapas ambientales. Mostramos evidencia experimental que muestra las ventajas de la comunicación directa en la calidad de la agrupación final.

## 2. Trabajos Previos

El primer algoritmo de agrupación de clases basado en el comportamiento de hormigas fue propuesto por Deneubourg et al. [4] usando un modelo para el ordenamiento espacial. En el modelo, un grupo de agentes con el mismo comportamiento se mueven aleatoriamente sobre una cuadrícula toroidal bidimensional en la que se encuentran esparcidos aleatoriamente objetos. Estos objetos pueden ser recogidos, transportados o depositados en cualquier localidad de la cuadrícula que esté libre. Un objeto es recogido con alta probabilidad si no está rodeado por otros objetos del mismo tipo, y es depositado por un agente cargado si su vecindario está poblado por otros objetos del mismo tipo que su carga y la localidad del agente se encuentra desocupada.

El modelo de Deneubourg et al. [4] fue generalizado por Lumer y Faieta [12] para aplicarlo al análisis exploratorio de datos. Todos los algoritmos clásicos de agrupación de clases dependen de una medida de similitud o disimilitud para determinar si dos objetos se parecen o no. Este algoritmo no es ninguna excepción. Sin embargo, en lugar de tomar como entrada una matriz de similitudes, este algoritmo inicia con algo muy similar a lo que se describió en el párrafo anterior. La probabilidad de recoger un dato  $i$  se define como

$$p_p(i) = \left( \frac{k_p}{k_p + f(i)} \right)^2 \quad (1)$$

donde  $k_p$  es una constante y  $f(i)$  es una medida de densidad de similitud con respecto al dato  $i$ . De manera similar, la probabilidad de depositar un dato está dada por

$$p_d(i) = \begin{cases} 2f(i) & \text{si } f(i) < k_d \\ 1 & \text{caso contrario} \end{cases} \quad (2)$$

donde  $k_d$  es una constante. La medida de densidad de similitud  $f(i)$  para un dato  $i$ , en una localidad en particular  $\tau$ , se define como

$$f(i) = \begin{cases} \frac{1}{s^2} \sum_{j \in \text{Neigh}(\tau)} \left( 1 - \frac{d(i,j)}{\alpha} \right) & \text{si } f > 0 \\ 0 & \text{si } f \leq 0 \end{cases} \quad (3)$$

donde  $s^2$  es el área de la región de percepción de un agente con centro en su localidad y  $\alpha$  es un factor de escalamiento de la medida de disimilitud  $d(i, j)$  entre los datos  $i$  y  $j$ . Si se asume que los datos pueden ser representados como puntos

en un espacio  $n$ -dimensional, entonces la distancia Euclidiana puede utilizarse como medida de disimilitud.

En algunos casos, sin embargo, tal vez sea necesario usar una medida de similitud como lo es la métrica cosenoidal. En tales casos, la expresión 3 no es directamente aplicable. La siguiente expresión puede ser usada en su lugar

$$f(i) = \frac{1}{s^2} \sum_{j \in \text{Neigh}(\tau)} \left( \frac{1}{1 + e^{-S \frac{d(i,j)}{\alpha} + D}} \right) \quad (4)$$

donde  $S$  es la inclinación de la curva de respuesta y  $D$  sirve como un factor de desplazamiento. En nuestros experimentos fijamos  $S$  a 5 debido a que provee un valor de similitud cercano a 0 cuando la métrica cosenoidal es mínima, o sea, cuando la métrica toma un valor de  $-1$ , y  $D$  a 1 porque permite distinguir mejor entre vectores con ángulos de separación entre 0 y  $\pi/2$ . Esta expresión tiene la ventaja de limitar el rango de valores que  $\alpha$  puede tomar a  $(0, 1]$ , dado  $-1 \leq d(i, j) \leq 1$ , como es el caso para la métrica cosenoidal.

En experimentos, Lumer y Faieta mostraron que el algoritmo era capaz de clasificar correctamente datos sintetizados a partir de cuatro distribuciones normales bivariadas en el plano; sin embargo, el número de grupos encontrados fue, en general, mayor al esperado. Para corregir ese problema, ellos mismos propusieron tres modificaciones al algoritmo básico. Una de ellas fue la introducción de una memoria de corto plazo en cada agente, de manera que pudieran recordar las últimas  $m$  localidades en donde hayan depositado datos. La memoria de corto plazo trata de representar el estado del ambiente. Esta modificación tiene el efecto de reducir la probabilidad de que un dato recién recogido pueda crear un grupo independiente por sí mismo, reduciendo así el número final de grupos. Lo anterior sucede gracias a que cada vez que un agente recoge un dato, éste busca en su memoria al dato más cercano en el espacio de atributos. El agente, en lugar de moverse aleatoriamente, se dirige directamente a la localidad del dato más cercano. Esta representación de conocimiento acerca del estado del ambiente tiene problemas cuando el ambiente es dinámico. Memorizar las localidades de los datos previamente manipulados no reflejará el estado *real* del ambiente, particularmente durante las primeras etapas del proceso. Para eliminar ese problema, proponemos utilizar comunicación directa entre agentes y cambiar la representación de conocimiento empleada.

### 3. Estrategias de Comunicación y de Representación de Conocimiento

En todos los intentos previos por llevar a cabo tareas de agrupación de clases mediante una colonia de insectos simulada no hay interacción directa entre los agentes. Esto se debe, tal vez, al hecho de que los primeros trabajos en agrupación basada en el comportamiento de insectos se enfocaron en implementaciones robóticas [4, 9, 1, 11] en donde la comunicación directa en tiempo real es mucho más complicada que en simulaciones computacionales.

En nuestros experimentos, siempre que un grupo de agentes coincida en un punto de la cuadrícula, de entre los cuales al menos uno de ellos esté en búsqueda de un lugar donde depositar un dato, éste intercambiará conocimiento con los demás agentes acerca del estado del ambiente. El propósito de este intercambio es aprovechar el conocimiento acumulado de otros agentes acerca de la distribución de los datos en el ambiente para dirigir la caminata aleatoria de un agente en búsqueda de localidades apropiadas donde depositar datos.

La estrategia de comunicación arriba descrita invita a reflexionar acerca del tipo de conocimiento a intercambiar. Para tratar de abordar este tema, usamos primero una extensión natural del modelo con agentes con memoria de corto plazo de Lumer y Faieta [12]. Siempre que un agente cargado coincida en la cuadrícula con otro u otros agentes, éste explorará la memoria de los otros agentes en búsqueda de un dato más cercano a su carga que el dato que lo iba guiando en su búsqueda por localidades apropiadas para depositarla. Si lo encuentra, el agente se redirige hacia ese dato. Esta es la primera estrategia de representación de conocimiento que se explora en este artículo.

La segunda estrategia de representación de conocimiento explorada en este trabajo es la de mapas ambientales. En un mapa ambiental, un agente representa la distribución espacial de los datos. Una representación sucinta de una distribución espacial es la que ofrecen los mapas autoorganizativos [10]. Sin embargo, debido a sus limitaciones tales como el número fijo de neuronas establecido *a priori* y el problema de neuronas "muertas" o neuronas que no actualizan sus vectores de referencia debido a un mal posicionamiento inicial en

el espacio de atributos de los datos, cada agente debe crear un mapa autoorganizativo dinámico. Con el propósito de representar al ambiente lo más precisamente posible, el agente debe actualizar su mapa conforme explora el ambiente sin importar si está en búsqueda de alguna localidad para depositar algún dato o sólo desplazándose por el mismo.

Con la representación del mapa ambiental, tanto el comportamiento del agente como el intercambio de conocimiento entre agentes es diferente al caso anterior. Si un agente no cargado recoge un dato de la cuadrícula, éste lo clasifica usando su propia red neuronal y se mueve directamente hacia la localidad de la neurona ganadora. Cuando un agente cargado coincide con otros agentes en alguna localidad sobre la cuadrícula, clasifica el dato que lleva cargando usando las redes de esos agentes para determinar la neurona ganadora más cercana. Si esta neurona se encuentra más cerca en el espacio de atributos que su propia neurona ganadora, el agente se redirige hacia la localidad de esa neurona. La implementación particular del mapa autoorganizativo no debería alterar los efectos finales siempre y cuando represente de manera precisa la distribución espacial de los datos en el ambiente.

## 4. Experimentación

Para evaluar el efecto del intercambio de conocimiento entre agentes se utilizaron dos bases de datos reales del Repositorio para el Aprendizaje Automático de la UCI [3]. Estas fueron:

- Base de datos de plantas Iris. Esta base de datos se compone de 150 instancias con 4 atributos numéricos cada uno. Tiene 3 clases compuestas de 50 instancias cada una. De éstas, una es linealmente separable de las otras dos; estas últimas no son linealmente separables entre sí.
- Base de datos de reconocimiento de vinos. Esta base de datos se compone de 178 instancias con 17 atributos numéricos cada uno. Hay 3 clases en la base de datos, la clase 1 tiene 59 instancias, la clase 2 tiene 71 instancias y la clase 3 tiene 48 instancias.

Para eliminar el sesgo en las medidas de similitud provocado por la existencia de diferentes escalas en los atributos de los datos, se estandarizaron ambas bases de datos. La métrica de similitud utilizada en todos nuestros experimentos fue la métrica cosenoidal<sup>1</sup>.

### 4.1. Modelos de Agente

Las probabilidades con las que los agentes recogen y depositan objetos se obtuvieron usando las expresiones 1 y 2 respectivamente. Sin embargo, debido a que la métrica cosenoidal es una medida de similitud, en nuestros experimentos se aplicó la expresión 4.

La tabla que se presenta a continuación resume todos los demás parámetros de configuración de los agentes usados en nuestros experimentos<sup>2</sup>.

Parámetro	Valor
$k_p$	0,1
$k_d$	0,15
$\alpha$	0,7
Tamaño de vecindario	$5 \times 5$
Tamaño de memoria	8

Para el algoritmo con agentes creadores de mapas, se utilizaron redes neuronales crecientes del tipo “gas” para representar la distribución espacial de los datos. Estas fueron propuestas por Fritzke [6] como una manera de eliminar algunas de las limitaciones de los mapas autoorganizativos convencionales.

Una red neuronal del tipo “gas” se compone de:

- un conjunto  $A$  de unidades (o nodos). Cada unidad  $c \in A$  tiene asociado a un *vector de referencia*  $w_c \in \mathbf{R}^n$ . Los vectores de referencia se pueden considerar como las posiciones de las unidades en el espacio de entrada.
- un conjunto  $N$  de conexiones (o arcos) entre pares de unidades. Estas conexiones no están ponderadas. Su único propósito es la definición de una estructura topológica.

La idea detrás del algoritmo de entrenamiento es agregar sucesivamente nuevas unidades a una red

<sup>1</sup>En experimentos preliminares, demostró dar buenos resultados.

<sup>2</sup>Los parámetros de configuración de las tablas que aparecen en este trabajo fueron derivados experimentalmente, y demostraron dar buenos resultados con las bases de datos utilizadas.

inicialmente pequeña a través de la evaluación local de medidas estadísticas reunidas durante pasos previos en el proceso de adaptación. La topología de la red es generada incrementalmente mediante una regla Hebbiana de aprendizaje competitivo. La dimensionalidad depende de los datos de entrada y puede variar localmente.

El algoritmo de entrenamiento empieza con dos unidades localizadas aleatoriamente en el espacio de entrada y una señal de entrada que debe ser aprendida. La regla de aprendizaje utilizada para adaptar los vectores de referencia de la unidad más cercana en el espacio de entrada a la señal de entrada actual y de sus vecinas topológicas es

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \epsilon(\xi - \mathbf{w}^{(t)}) \quad (5)$$

donde  $\mathbf{w}$  es el vector de referencia de la unidad adaptativa,  $\epsilon$  es una constante llamada *tasa de aprendizaje*, y  $\xi$  es la señal de entrada que es aprendida.

El error cuadrático de la unidad más cercana en cada turno es acumulado, de manera que después de que se han aprendido  $\lambda$  señales de entrada, se crea una nueva unidad entre las dos neuronas vecinas con los errores acumulados más altos. La eliminación de unidades sucede cuando una unidad no se encuentra topológicamente conectada con ninguna otra unidad. Esto ocurre después de que un cierto número de señales de entrada no hayan disparado alguna unidad. Este proceso continúa hasta que algún criterio de paro es satisfecho. El algoritmo completo puede ser consultado en [6].

La red neuronal que acabamos de describir tiene un conjunto de parámetros de configuración por sí misma, referimos al lector a [6] para una explicación detallada del significado y efectos de todos estos parámetros. La tabla que se presenta a continuación resume el conjunto de parámetros de las redes utilizadas en nuestros experimentos.

Parámetro	Valor
Tasa de aprendizaje de la neurona ganadora $\epsilon_b$	1
Tasa de aprendizaje de las neuronas vecinas $\epsilon_n$	0.005
Edad máxima de arco	50
Umbral de crecimiento $\lambda$	150
Tasa de decrecimiento de error local $\alpha$	0.5
Tasa de decrecimiento de error global $d$	0.995

Una tasa de aprendizaje de 1 es normalmente con-

siderada demasiado alta, sin embargo, su uso es justificable en este caso debido al hecho de que el ambiente de los agentes es sumamente dinámico. Una tasa de aprendizaje con estas características es necesaria para permitir a la red aprender la distribución real de los datos sobre la cuadrícula, condición de suma importancia para el buen funcionamiento de este modelo.

## 4.2. Diseño de Experimentos

Para observar los efectos del intercambio de información entre agentes en el proceso de agrupación de clases, se aplicaron las cuatro medidas de desempeño utilizadas por Handl et al. [8] cada 10,000 ciclos de simulación al agrupamiento parcial hasta ese momento hecho. Estas medidas son:

**Medida  $F$ .** La media armónica del recuerdo y la precisión, también conocida como la *Medida  $F$* . El recuerdo y la precisión son medidas que nos dan una idea acerca de qué tan bien un algoritmo de agrupación de clases está identificando las clases presentes en la base de datos. La Medida  $F$  está limitada al intervalo  $[0, 1]$  con un valor de 1 para una clasificación perfecta.

En el contexto de la clasificación, el recuerdo se define como  $r(i, j) = \frac{n_{ij}}{n_i}$  donde  $n_{ij}$  es el número de elementos de la clase  $i$  en la agrupación  $j$  y  $n_i$  es el número de elementos de la clase  $i$ . La precisión se define como  $p(i, j) = \frac{n_{ij}}{n_j}$  donde  $n_j$  es el número de elementos en la agrupación  $j$ . Para la clase  $i$  y una agrupación  $j$  la *Medida  $F$*  se define como

$$F(i, j) = \frac{2p(i, j)r(i, j)}{p(i, j) + r(i, j)}$$

La *Medida  $F$*  general para la clasificación generada por el algoritmo de agrupación de clases está dada por

$$F = \sum_i \frac{n_i}{n} \max_j \{F(i, j)\} \quad (6)$$

donde  $n$  es el tamaño de la base de datos.  $F$  está limitada al intervalo  $[0, 1]$  con un valor de 1 para una clasificación perfecta.

**Índice Rand.** Determina una medida de similitud entre la clasificación correcta  $C$  y la

partición generada por el algoritmo de agrupación de clases  $P$ . El índice Rand está limitado al intervalo  $[0, 1]$  con un valor de 1 cuando la partición  $P$  es igual a la clasificación  $C$ .

Se define como

$$R = \frac{a + d}{a + b + c + d} \quad (7)$$

donde

- $a$  es el número de pares en los que ambos elementos pertenecen a la misma clase en  $C$  y al mismo grupo en  $P$ .
- $b$  es el número de pares en los que ambos elementos pertenecen a la misma clase en  $C$  y a distintos grupos en  $P$ .
- $c$  es el número de pares en los que ambos elementos pertenecen a distintas clases en  $C$  y al mismo grupo en  $P$ .
- $d$  es el número de pares en los que ambos elementos pertenecen a distintas clases en  $C$  y a distintos grupos en  $P$ .

Nótese que  $a + b + c + d = N(N-1)/2$  donde  $N$  es el tamaño de la base de datos.

**Varianza interna.** Esta medida trata de capturar la idea de que los elementos que pertenecen al mismo grupo deben parecerse entre sí lo más posible. El objetivo es minimizarla.

Está dada por

$$I = \sum_{i=1}^n \sum_{p \in C_i} \|p - \mu_i\|^2 \quad (8)$$

donde  $n$  es el número de grupos en la partición generada por el algoritmo de agrupación de clases y  $\mu_i$  es el centroide del grupo  $i$ .

**Índice Dunn.** Esta medida tomará un valor alto siempre que la partición describa grupos compactos y bien separados.

Esta medida está dada por

$$D = \min_{c,d \in C} \left\{ \frac{\|\mu_c - \mu_d\|}{\max_{e \in C} \{diam(e)\}} \right\} \quad (9)$$

donde  $\mu_i$  es el centroide del grupo  $i$  y  $diam(i)$  es el diámetro del grupo  $i$  que se considera una medida de dispersión. Se define como

$$diam(c) = \max_{x,y \in c} \|x - y\|$$

Cada ciclo de simulación se compone por  $N$  acciones individuales, donde  $N$  es el número de agentes en la simulación. Todos los algoritmos fueron puestos a prueba 30 veces con cada una de las bases de datos durante 1,000,000 ciclos.

En nuestros experimentos, el intercambio de información ocurre solamente cuando dos o más agentes se encuentran en un mismo punto de la cuadrícula al mismo tiempo. Podemos esperar, por tanto, que la probabilidad de un encuentro se incremente conforme la densidad de agentes en el ambiente se vea incrementada. Al tener más encuentros entre agentes, los efectos del intercambio de información deben tener un mayor impacto. Se experimentó, por tanto, con poblaciones de 10, 20 y 30 agentes en un ambiente de  $100 \times 100$  localidades en todos nuestros experimentos. No se experimentó con poblaciones mayores de agentes debido a que el número máximo de agentes debe ser menor al tamaño de la clase menor que se quiere encontrar<sup>3</sup>.

Para recolectar información de la partición espacial formada por la acción de los agentes en la cuadrícula, se aplicó un algoritmo jerárquico de agrupación de clases aglomerativo sobre los datos usando la distancia Euclidiana y la estrategia de enlace por miembros más cercanos. La distancia máxima para unir agrupamientos se fijó igual al valor del tamaño del vecindario de los agentes. La distancia utilizada para calcular la varianza y los diámetros de los agrupamientos también fue la distancia Euclidiana.

## 5. Resultados

Los resultados obtenidos por los algoritmos evaluados en las bases de datos de prueba se resumen y presentan agrupados por medida de validación. Los resultados obtenidos con 20 agentes no se muestran debido a que, en todos los casos, mostraban resultados que se encontraban entre los obtenidos con 10 y 30 agentes. En todas las gráficas que se presentan, las versiones de los algoritmos en los que no hay intercambio de información están representados por la línea cero, es decir, sólo se muestra la diferencia en el desempeño de los algoritmos en donde los agentes intercambian información. Es por esto que, a pesar de que el rango de las medidas de desempeño empleadas está en  $\mathbf{R}^+$ , las gráficas muestran valores negativos.

<sup>3</sup>Un dato que, por cierto, en situaciones reales normalmente no se conoce

### 5.1. Medida $F$

Las figuras 1 y 2 muestran los valores de la Medida  $F$  con respecto al tiempo para los algoritmos basados en el modelo de memoria de corto plazo y de mapas ambientales respectivamente, ambos con la base de datos Iris. Después de un cierto número de ciclos, se puede observar que los algoritmos con 10 agentes que pueden intercambiar conocimiento tienen una evaluación más alta que sus contrapartes en donde no se comunican. Las figuras 3 y 4 muestran los valores de la Medida  $F$  con respecto al tiempo para los algoritmos basados en el modelo de memoria de corto plazo y de mapas ambientales respectivamente, ambos con la base de datos de reconocimiento de vinos. Los resultados son los mismos cualitativamente hablando que los obtenidos en la base de datos Iris.

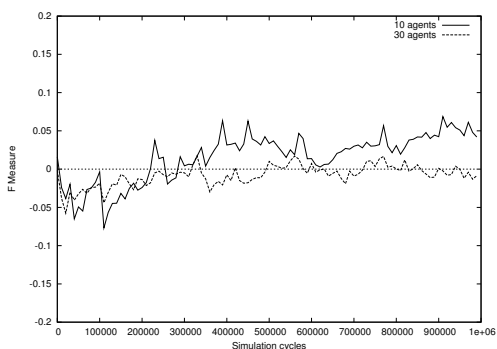


Figura 1. Medida  $F$  para el modelo de memoria de corto plazo para la base de datos Iris.

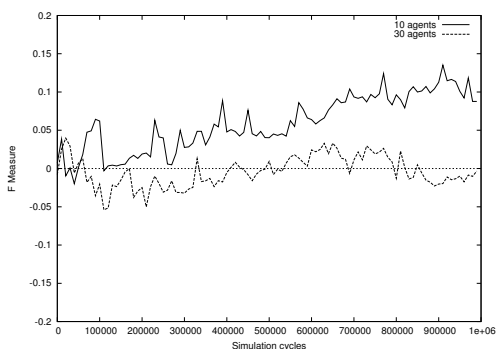


Figura 2. Medida  $F$  para el modelo de mapas ambientales para la base de datos Iris.

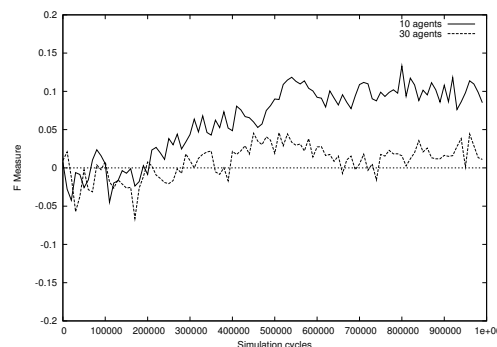


Figura 3. Medida  $F$  para el modelo de memoria de corto plazo para la base de datos de vinos.

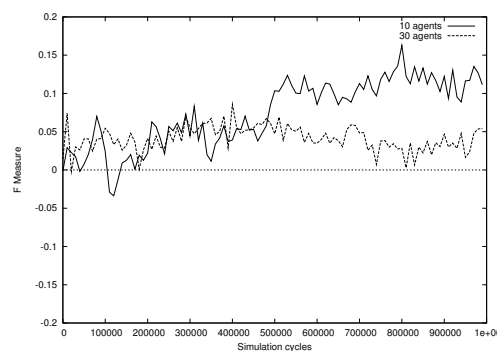


Figura 4. Medida  $F$  para el modelo de mapas ambientales para la base de datos de vinos.

### 5.2. Índice Rand

Las figuras 5 y 6 muestran los valores del índice Rand con respecto al tiempo para los algoritmos basados en el modelo de memoria de corto plazo y de mapas ambientales respectivamente, ambos con la base de datos Iris. Las figuras 7 y 8 muestran las mismas evaluaciones pero ahora con la base de datos de reconocimiento de vinos. Al igual que con la Medida  $F$ , las evaluaciones son más altas cuando se utiliza intercambio de información, aunque sólo cuando se usan 10 agentes.

### 5.3. Varianza Interna

Las figuras 9 y 10 corresponden a los valores de la varianza interna de los grupos encontrados por los algoritmos con 10 y 30 agentes en la base de

datos Iris. Como en el caso de la Medida  $F$  y el índice Rand, los algoritmos en los que los agentes pueden intercambiar información tienen mejor comportamiento que en los que no lo hacen. Recuerdese que el objetivo es minimizar esta medida, por tanto entre menor sea el valor de la varianza mejor será el agrupamiento. Las figuras 11 y 12 muestran la varianza interna con la base de datos de reconocimiento de vinos.

#### 5.4. Índice Dunn

Las figuras 13 y 14 muestran los valores del índice Dunn con respecto al tiempo para todos los algoritmos en la base de datos Iris.

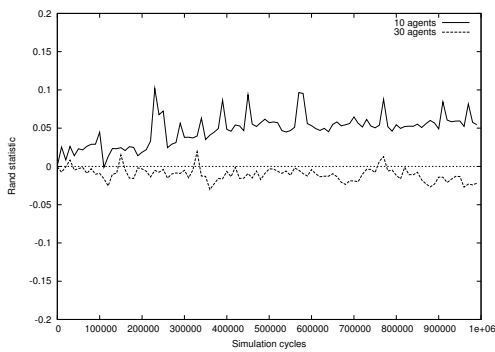


Figura 5. Índice Rand para el modelo de memoria de corto plazo para la base de datos Iris.

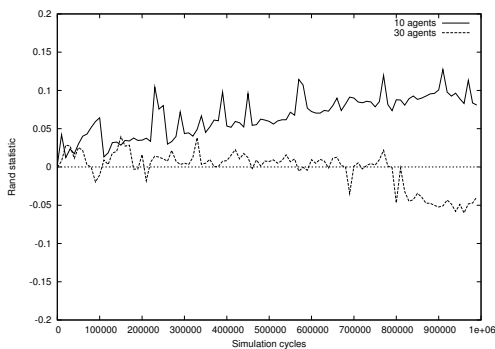


Figura 6. Índice Rand para el modelo de mapas ambientales para la base de datos Iris.

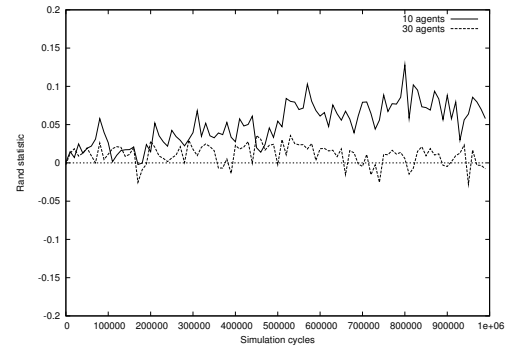


Figura 7. Índice Rand para el modelo de memoria de corto plazo para la base de datos de vinos.

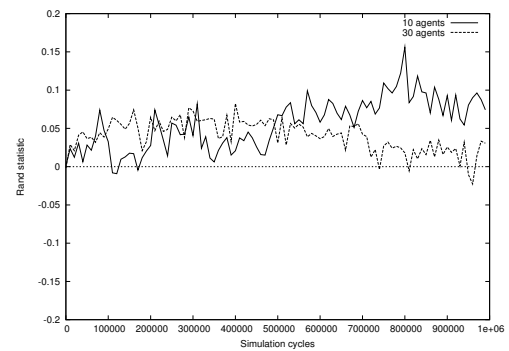


Figura 8. Índice Rand para el modelo de mapas ambientales para la base de datos de vinos.

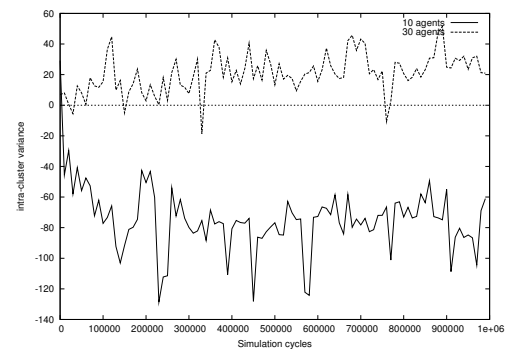


Figura 9. Varianza interna para el modelo de memoria de corto plazo para la base de datos Iris.



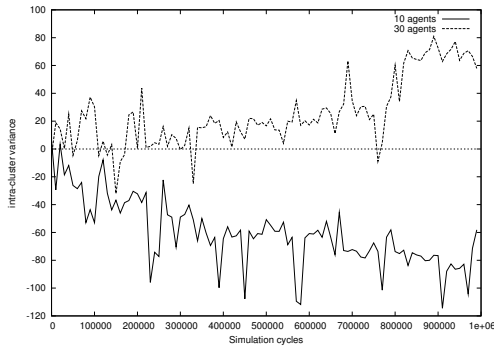


Figura 10. Varianza interna para el modelo de mapas ambientales para la base de datos Iris.

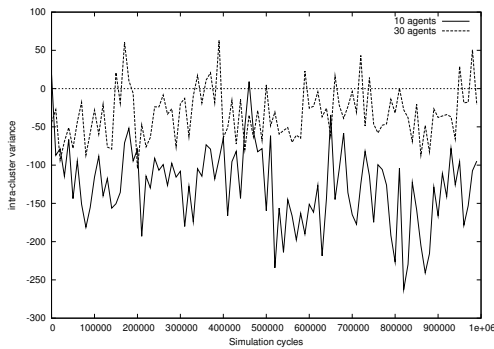


Figura 11. Varianza interna para el modelo de memoria de corto plazo para la base de datos de vinos.

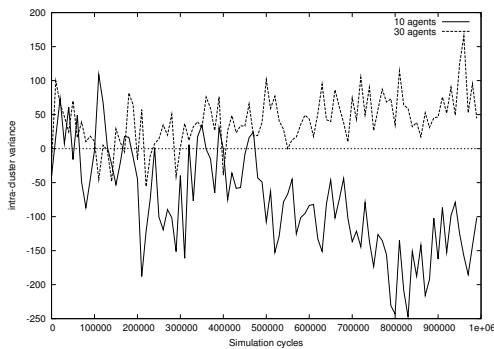


Figura 12. Varianza interna para el modelo de mapas ambientales para la base de datos de vinos.

Se puede observar que el desempeño del algoritmo con intercambio de información con 30 agentes tiene una evaluación muy alta. Este fenómeno

se puede explicar si se recuerda que en la base de datos Iris dos grupos no son linealmente separables, o sea, el algoritmo está confundiéndolos como un sólo grupo. Las figuras 15 y 16 muestran los valores del índice Dunn con respecto al tiempo para todos los algoritmos en la base de datos de reconocimiento de vinos.

## 6. Discusión

La probabilidad de un encuentro entre dos agentes en movimiento aleatorio se incrementa conforme el número de agentes aumenta. Esto es confirmado en los experimentos robóticos [9], donde ha sido reportado que el número de colisiones entre robots se incrementa cuando el número de robots aumenta.

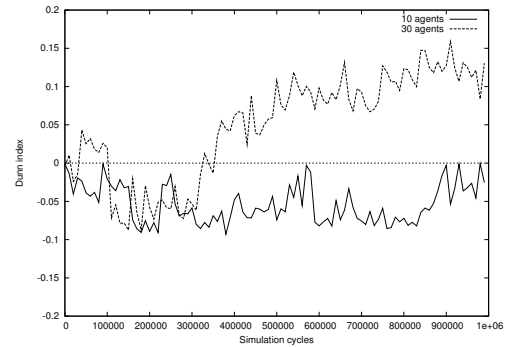


Figura 13. Índice Dunn para el modelo de memoria de corto plazo para la base de datos Iris.

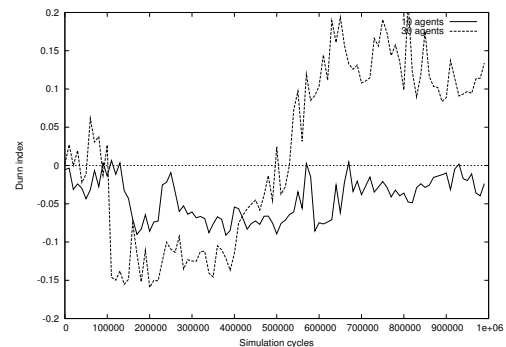
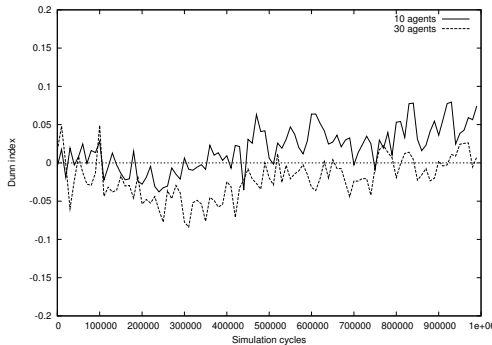
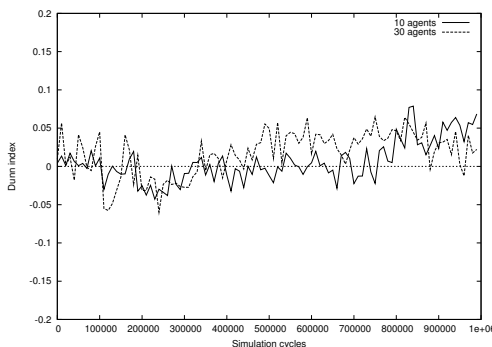


Figura 14. Índice Dunn para el modelo de mapas ambientales para la base de datos Iris.



**Figura 15. Índice Dunn para el modelo de memoria de corto plazo para la base de datos de vinos.**



**Figura 16. Índice Dunn para el modelo de mapas ambientales para la base de datos de vinos.**

En nuestros experimentos, tratamos de tomar ventaja de este hecho y usarlo para estudiar el efecto de incrementar la frecuencia de intercambios de conocimiento entre los agentes. Sin embargo, el efecto del intercambio de conocimiento depende no solamente de la frecuencia de los encuentros, sino también de su efectividad. Es decir, el intercambio de conocimiento será tan benéfico como lo sea el conocimiento intercambiado y la forma en que se utilice éste. En el contexto de agrupación de clases, la utilidad del conocimiento depende mucho de cómo el conocimiento de los agentes representa el estado *real* del ambiente. En las primeras etapas del proceso de agrupamiento de clases, esto no se logra debido al dinamismo del ambiente. Cuando esta etapa termina, una vez que algunos grupos existen, el intercambio de información es más efectivo debido a que el ambiente es menos dinámico. Esto en parte explica el rendimiento relativamente pobre del modelo con

intercambio de información durante los primeros ciclos de la simulación.

Nuestros resultados sugieren que tan sólo con compartir información acerca de las localidades de deposición memorizadas, el intercambio de información ofrece algunas ventajas sobre el modelo que usa solamente a la estigmergia como medio de interacción entre agentes. En particular, la calidad de la agrupación de clases generada es mejorada y esto es reflejado por la Medida  $F$ , el índice Rand y la varianza interna. Esto es verdadero, sin embargo, solamente cuando el número de agentes es pequeño. Este hecho sugiere que hay un número crítico de agentes que pueden intercambiar información, y cuando éste es sobrepasado, el resultado puede ser perjudicial. A pesar de esto, el intercambio de información puede ser útil cuando se trata con bases de datos grandes, donde ambos, el número de agentes y las dimensiones del ambiente deben ser incrementadas; por ejemplo, cuando uno no puede pagar el costo computacional asociado con el incremento de estas variables en las tareas de agrupación de clases.

## 7. Conclusiones y Trabajo Futuro

En este documento se exploran dos esquemas de intercambio de información. Primero, un esquema simple de intercambio de memorias de corto plazo donde los agentes usan las experiencias pasadas de otros agentes para encontrar localidades de deposición favorables sobre la cuadrícula. Luego, la explotación de mapas de distribución espacial que los agentes van construyendo conforme exploran el ambiente. El resultado es que cuando un agente está en búsqueda de localidades en donde depositar datos, éste puede cambiar su trayectoria de acuerdo con el conocimiento de otros agentes.

Hay muchas posibilidades para explorar la comunicación directa entre los agentes en algoritmos de agrupación de clases basados en el comportamiento de hormigas. En todas ellas, se tiene que determinar *qué* información intercambiar, *cómo* hacer este intercambio, *cuándo* es el momento más apropiado para hacerlo, y *qué hacer* con esta información. Algunos de estos aspectos han sido investigados en el campo de robótica, y a pesar de que los resultados son importantes, no son directamente aplicables a las implementaciones de *software* donde las restricciones físicas no necesitan ser consideradas.

Como parte de los trabajos futuros que se pueden hacer en esta área, están el explorar diferentes estrategias de intercambio de información que, manteniendo a los agentes simples, pudieran proporcionar mejoras en la calidad de los resultados de la agrupación de clases. Entre otras, el intercambio de información de manera indirecta a través de la deposición de paquetes de información sobre el ambiente. Otra línea de trabajo es aquella en la que se pudieran establecer criterios que permitan a los agentes decidir qué hacer con la información que reciben de otros agentes.

## Agradecimientos

Esta investigación fue posible gracias a la Cátedra de Investigación en Tecnologías de Conocimiento y Agentes Inteligentes del Tecnológico de Monterrey, Campus Monterrey. CAT-011.

## Referencias

- [1] Martinoli A. and F. Mondada. Collective and cooperative group behaviours: Biologically inspired experiments in robotics. In Khatib. O. and Salisbury. J.K., editors, *Proceedings of the Fourth International Symposium on Experimental Robotics ISER-95*, pages 3–10. Springer Verlag, 1995.
- [2] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence. From Natural to Artificial Systems*. Oxford University Press, 1999.
- [3] Blake C.L. and Merz C.J. UCI repository of machine learning databases. [<http://www.ics.uci.edu/~mllearn/mlrepository.html>]. University of California, Irvine, Dept. of Information and Computer Sciences, 1998.
- [4] J.-L. Deneubourg, S. Goss, N. Franks, Sendova-Franks A., Detrain C., and L. Chretien. The dynamics of collective sorting: Robot-like ants and ant-like robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 356–365. MIT Press, 1991.
- [5] Richard J. Elzinga. *Fundamentals of entomology*. Prentice Hall, 2000.
- [6] Bernd Fritzke. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7*. MIT Press, 1995.
- [7] Pierre-Paul Grassé. La reconstruction du nid et les coordinations inter-individuelles chez *bellicositermes natalensis* et *cubitermes sp.* la theorie de la stigmergie: Essai d'interpretation des termites constructeurs. *Insectes Sociaux*, 6(1):41–83, 1959.
- [8] Julia Handl, Joshua Knowles, and Marco Dorigo. On the performance of ant-based clustering. Design and application of hybrid intelligent systems. *Frontiers in Artificial Intelligence and Applications 104*. IOS Press, Amsterdam, The Netherlands, 2003.
- [9] Owen Holland and Chris Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5:173–202, 1999.
- [10] T. Kohonen, T.S. Huang, and M.R. Schroeder. *Self-organizing maps*. Springer Verlag, third edition, 2000.
- [11] C.R. Kube and H. Zhang. Collective robotic intelligence. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 460–468, 1992.
- [12] Erick D. Lumer and Baldo Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, pages 501–508. MIT Press, 1994.
- [13] Edward O. Wilson. *The Insect Societies*. The Belkap Press of Harvard University Press, 1971.