

UNIVERSIDAD CENTROCCIDENTAL  
“LISANDRO ALVARADO”

**CONSTRUCCION COOPERATIVA DE ARQUITECTURAS 3D MEDIANTE  
LA OPTIMIZACIÓN POR ENJAMBRE DE PARTICULAS**

Por: ING. HENRY GIOVANNI ZAPATA ARAUJO

Barquisimeto, 2013

UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”  
DECANATO DE CIENCIA Y TECNOLOGÍA  
POSTGRADO DE CIENCIAS DE LA COMPUTACIÓN  
MENCION: INTELIGENCIA ARTIFICIAL

**CONSTRUCCION COOPERATIVA DE ARQUITECTURAS 3D MEDIANTE  
LA OPTIMIZACIÓN POR ENJAMBRE DE PARTICULAS**

Proyecto de Trabajo de Grado para Optar al grado de  
Magíster Scientiarum

Por: ING. HENRY GIOVANNI ZAPATA ARAUJO

Barquisimeto, 2013



UNIVERSIDAD CENTROCCIDENTAL  
"LISANDRO ALVARADO"  
DECANATO DE CIENCIAS Y TECNOLOGIA  
COORDINACION DE POSTGRADO

## ACTA VEREDICTO TRABAJO DE GRADO

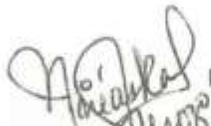
Nosotros, Miembros del Jurado Examinador del Trabajo de Grado titulado: "CONSTRUCCIÓN COOPERATIVA DE ARQUITECTURAS 3D MEDIANTE LA OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS", presentado por HENRY GIOVANNI ZAPATA ARAUJO, titular de la Cédula de Identidad N° 11.759.714, como requisito para optar al grado académico de **MAGÍSTER SCIENTIARUM EN CIENCIAS DE LA COMPUTACIÓN**, ofrecido por el programa de Maestría en Ciencias de la Computación del Decanato de Ciencias y Tecnología de la Universidad Centroccidental "Lisandro Alvarado", hacemos constar que hoy jueves doce de diciembre del año dos mil trece (12/12/2013) a las tres de la tarde (3:00 p.m.), se realizó el examen Público de Defensa de Trabajo de Grado, de acuerdo a lo establecido en la Normativa sobre Trabajos de Grado de la UCLA. Una vez rendido el examen, este Jurado emite el siguiente veredicto: El Trabajo de Grado fue:

\*\*\*\*\*

**APROBADO**

\*\*\*\*\*

Dando fe de ello, levantamos la presente acta en la ciudad de Barquisimeto a los doce días del mes de diciembre del año dos mil trece.

  
**Prof. Ninfaska Perozo**  
Tutora – Jurado Principal  
C.I. N° 11.593.374

  
**Prof. José Gregorio Sánchez**  
Presidente de Jurado  
C.I. N° 7.356.417



  
**Prof. Joyné Contreras**  
Jurado Principal  
C.I. N° 11.594.977

## **AGRADECIMIENTOS**

A Dios, por mantenerme saludable y permitirme la búsqueda del conocimiento.

A mi Madre, y mi familia por su apoyo incondicional.

A mi Tutora, por sus excelentes orientaciones que despertaron mi espíritu de investigador, y que gracias a sus exigencias fue posible consolidar este proyecto, además, su ejemplo de docente me compromete a estimular la investigación en todos mis ámbitos.

A los profesores Joyne y Wilfredo por su colaboración, con el ejercicio de sus profesiones, durante la búsqueda de la función objetivo, utilizada en esta investigación.

A la UCLA, sus Autoridades, Profesores, Empleados y Obreros, por permitirme un espacio para incrementar mi formación profesional.

## ÍNDICE

	<b>PÁG.</b>
ÍNDICE DE CUADROS.....	VII
ÍNDICE DE FIGURAS.....	VIII
INDICE DE GRÁFICAS.....	IX
RESUMEN.....	X
INTRODUCCIÓN.....	1
<b>CAPITULO</b>	
<b>I</b> <b>EL PROBLEMA.....</b>	<b>2</b>
Planteamiento del Problema.....	2
Objetivo General.....	4
Objetivos Específicos.....	4
Justificación.....	5
Alcances y Limitaciones.....	6
<b>II</b> <b>MARCO TEÓRICO.....</b>	<b>7</b>
Antecedentes.....	7
Bases teóricas.....	8
Inteligencia Colectiva.....	8
Modelo de Auto-ensamblaje.....	12
Micro-reglas.....	22
Requerimientos para el Auto-ensamblaje.....	27
Optimización de enjambre de partículas.....	29
Algoritmo de la PSO.....	31
<b>III</b> <b>MARCO METODOLOGICO .....</b>	<b>35</b>
Naturaleza de la Investigación.....	35
Fase I Diagnostico.....	35
Fase II Factibilidad.....	36

IV	PROPUESTA DEL ESTUDIO.....	37
	Descripción de la Propuesta.....	37
	Software Para Simulación.....	40
	Micro-reglas para Evaluación.....	43
V	EJECUCIÓN Y EVALUACIÓN DE LA PROPUESTA.....	45
	Descripción del Software.....	45
	Evaluación.....	47
	Primer Escenario.....	47
	Segundo Escenario.....	61
	Comparación de los Resultados.....	66
VI	CONCLUSIONES Y RECOMENDACIONES.....	67
	REFERENCIAS BIBLIOGRAFICAS.....	70

## ÍNDICE DE CUADROS

	<b>PÁG.</b>
Cuadro N° 1. Algunos ejemplos de enjambres en la naturaleza.....	11
Cuadro N° 2. Algoritmo de Construcción de Nidos de Avispas.....	19
Cuadro N° 3. Ejemplo 1. Extracción de la vecindad ( $V_1$ ) en el Cubo.....	24
Cuadro N° 4. Ejemplo 2. Extracción de la vecindad ( $V_2$ ) en el Cubo.....	24
Cuadro N° 5. Ejemplo de una tabla de consulta.....	25
Cuadro N° 6. Tabla de Consulta.....	26
Cuadro N° 7. Diferencia de las versiones del Nest.....	28
Cuadro N° 8. Requerimientos deseables para el Auto-ensamblaje.....	29
Cuadro N° 9. Algoritmo de PSO.....	34
Cuadro N° 10. Algoritmo de Auto-Ensamblaje Combinado con PSO.....	38
Cuadro N° 11. Estructura del archivo para las Micro-Reglas.....	41
Cuadro N° 12. Mecanismos de inercia dinámicos para PSO.....	42
Cuadro N° 13. Banco de Micro-reglas e imagen de la estructura 1.....	43
Cuadro N° 14. Banco de Micro-reglas e imagen de la estructura 2.....	44
Cuadro N° 15. Descripción de la herramienta de simulación.....	45
Cuadro N° 16. Parámetros para simular Auto-ensamblaje clásico Estructura 1.	47
Cuadro N° 17. Parámetros para simular Auto-ensamblaje clásico Estructura 2.	61
Cuadro N° 18. Parámetros de los experimentos de la estructura 2 con PSO.	62
Cuadro N° 19. Diferencia entre el modelo clásico y el propuesto.....	66

## ÍNDICE DE FIGURAS

	PÁG.
Figura N° 1. Celdas construidas y etiquetadas.....	17
Figura N° 2. Vecindad de una celda.....	23
Figura N° 3. Extracción de la vecindad en el cubo y el hexágono....	24
Figura N° 4. Estructura construida con Auto-ensamblaje.....	27
Figura N° 5. Interfaz grafica para Auto-ensamblaje.....	46
Figura N° 6. Interfaz grafica construyendo estructura 1.....	60
Figuras 7-23. Estructuras construidas con diferentes parámetros.....	62
Figura N° 24. Interfaz grafica construyendo estructura 2.....	65

## ÍNDICE DE GRÁFICAS

	PÁG.
Grafica N° 1. Actividad de construcción de la estructura uno con Auto-ensamblaje clásico.....	47
Grafica N° 2. Actividad de construcción de la estructura 1 variando el componente social.....	48
Grafica N° 3 y N° 4. Actividad de construcción de la estructura 1 variando el componente cognitivo.....	49
Grafica N° 5. Actividad de construcción de la estructura 1 variando el parámetro de inercia.....	50
Grafica N° 6, N°7 y N°8. Actividad de construcción de la estructura 1 variando el componente $C_2$ .....	51
Grafica N° 9, N°10 y N°11. Actividad de construcción de la estructura 1 variando el componente $C_1$ .....	52
Grafica N° 12, N°13 y N°14. Actividad de construcción de la estructura 1 variando el parámetro $w$ .....	54
Grafica N° 15 hasta N°25. Actividad de construcción de la estructura 1 utilizando inercia dinámica.....	56

UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”  
DECANATO DE CIENCIA Y TECNOLOGÍA  
POSTGRADO DE CIENCIAS DE LA COMPUTACIÓN  
MENCION: INTELIGENCIA ARTIFICIAL

**CONSTRUCCION COOPERATIVA DE ARQUITECTURAS 3D MEDIANTE  
LA OPTIMIZACIÓN POR ENJAMBRE DE PARTICULAS**

**Autor:** ING. HENRY G. ZAPATA ARAUJO

**Tutor:** DRA. NIRIASKA PEROZO GUÉDEZ

**RESUMEN**

Mediante el Auto-ensamblaje basado en la construcción de nidos de avispa, se pueden dibujar arquitecturas en un espacio 3D utilizando un conjunto de micro-reglas, sin embargo su tiempo de convergencia en algunas ocasiones puede no ser ideal, debido a la lenta acción de algunos agentes constructores al explorar y moverse aleatoriamente en lugar de construir. Para minimizar el tiempo total del algoritmo y favorecer el tiempo de construcción, se propone utilizar la optimización por enjambre de partículas, basada en el comportamiento colectivo de bandadas de aves o peces.

En este trabajo se pueden observar los resultados de combinar dos técnicas emergentes como: el Auto-ensamblaje y la optimización por enjambre de partículas a través de un entorno interactivo desarrollado para tal fin, donde según los resultados obtenidos se logra mejorar el tiempo de convergencia del algoritmo de Auto-ensamblaje sin perder su comportamiento emergente, para explorar a ciegas y coordinar dinámicamente a los agentes en la construcción de estructuras 3D.

**Palabras clave:** Construcción Cooperativa, Inteligencia Colectiva, Inteligencia de Enjambre, Optimización por Enjambre de Partículas, Auto-ensamblaje.

## INTRODUCCIÓN

El auto-ensamblaje, inspirado en la construcción de los nidos de avispa fue desarrollado por Bonabeau y otros [1]. Es un modelo para la construcción de estructuras complejas a partir de micro-reglas simples en un espacio 3D, a través de un sistema multi-agente que introduce la comunicación indirecta o estigmergia (“Stigmergy”, en inglés) como paradigma de comunicación. Según Theraulaz y Bonabeau [2], la estigmergia resuelve la paradoja de la coordinación; es decir los individuos no interactúan directamente sino que lo realizan de forma indirecta, y aunque no lo parezca cada uno de ellos se encuentra inmerso en un comportamiento colectivo y coordinado.

Los promotores de este modelo de auto-ensamblaje, consideran que para explotar su extraordinario potencial, es necesario explorar procedimientos para la generación de las micro-reglas que permiten la construcción de una determinada estructura, y además integrar algún mecanismo que permita mejorar el tiempo de convergencia del algoritmo.

El contenido de esta investigación permite conocer todo lo relacionado al Auto-ensamblaje, basado en la construcción de nidos de avispa y la optimización por enjambre de partículas, basado en bandadas de especies. Además se muestra una integración de éstas dos técnicas de inteligencia colectiva para mejorar el tiempo de convergencia del algoritmo de Auto-ensamblaje.

# CAPITULO I

## EL PROBLEMA

### Planteamiento del problema

Estudios biológicos resaltan las extraordinarias capacidades desarrolladas por varias especies en su evolución, como la auto-organización y la interacción indirecta entre los individuos, las cuales al aplicarlas computacionalmente permiten la posibilidad de encontrar soluciones inteligentes colectivamente a problemas complejos. Las interacciones indirectas ocurren cuando un individuo modifica su medio ambiente o entorno y otros individuos responden al cambio en un tiempo más tarde, según Camelia y otros [3], a este tipo de comportamiento se le llama estigmergia.

Las micro-reglas pueden ser generadas en forma determinista o estocástica según Theraulaz y Bonabeau [4]. En el primer caso, se definen puntualmente basado en pruebas y heurísticas, es decir: se encuentran asociadas al criterio de un observador y su experiencia. En el segundo caso, se obtiene configurando la activación de micro-reglas a partir de probabilidad predefinida entre cero y uno, durante la exploración del espacio de construcción.

Omitiendo la técnica o procedimiento utilizado para crear el conjunto de micro-reglas que define una determinada estructura, se hace relevante el tiempo utilizado por el algoritmo de auto-ensamblaje para lograr la convergencia, es decir: el tiempo requerido por los agente constructores para permitir que emerja una estructura o patrón en el espacio 3D como resultado de un trabajo cooperativo. Obviamente este tiempo se encuentra asociado a la aleatoriedad introducida para lograr el comportamiento emergente a través del algoritmo, específicamente en el movimiento aleatorio de los agentes constructores. Considerando que la nueva posición del agente

constructor es aleatoria, éste puede realizar una de dos acciones; (*estar inerte o colocar un bloque de construcción*), y dependen del banco de micro-reglas. Cuando el agente constructor ejecuta la acción de estar inerte, se desperdicia un tiempo de procesamiento, el cual al repetirse por varios agentes en cada iteración incrementa el tiempo de convergencia.

Tratar de reducir el tiempo de convergencia del modelo de auto-ensamblaje basado en la construcción de nidos de avispas, es el tema principal de ésta investigación, considerando el uso de una técnica emanada de la inteligencia de enjambre como lo es; la Optimización por Enjambre de Partículas (*“Particle Swarm Optimization PSO”, en inglés*), particularmente, para localizar la nueva posición del agente constructor, con el fin de minimizar las acciones inertes de los agentes en las iteraciones. El comportamiento emergente se mantiene debido a que se encuentra intrínseco en la PSO.

La PSO es un algoritmo bioinspirado, basado en una analogía al comportamiento social de ciertas especies como: bandadas de pájaros y cardúmenes de peces, según Cervantes [5]. En este algoritmo los agentes llamados partículas se lanzan al espacio de búsqueda aleatoriamente, guiadas por la partícula líder que es la que ha encontrado la mejor solución, cada una evoluciona teniendo en cuenta la mejor solución encontrada en su recorrido. En cada iteración las partículas actualizan su velocidad y ubicación hacia la mejor solución de su entorno, teniendo en cuenta la información del líder hasta lograr minimizar o maximizar una función objetivo, la cual en nuestro caso se encuentra asociada a la distancia entre el bloque de inicial y el último bloque construido, sin embargo inicialmente se creía que la función objetivo estaba asociada al banco de micro-reglas, ya que éste define la relación entre el espacio de construcción y la estructura o patrón que emerge en el espacio 3D.

Considerando que las estructuras generadas por el modelo de auto-ensamblaje basado en la construcción de nidos de avispa, pueden ser similitudes biológicas de

interés para los biólogos o patrones útiles para los ingenieros, surge las siguientes interrogantes, ¿Es posible mejorar el tiempo de convergencia del algoritmo de Auto-ensamblaje a través de la PSO?, ¿Se pueden reducir las acciones inertes de los agentes constructores?, ¿Cuáles pueden ser los parámetros a considerar en la función objetivo para minimizar el tiempo de construcción de los agentes?

### **Objetivo General**

Mejorar el proceso de construcción cooperativa de Arquitecturas 3D mediante la optimización por enjambre de partículas (PSO).

### **Objetivos Específicos**

- a) Investigar los aspectos teóricos relacionados al modelo de Auto-ensamblaje basado en la construcción de nidos de avispa.
- b) Investigar aspectos teóricos de la PSO.
- c) Determinar los requerimientos mínimos y adicionales para desarrollar el entorno requerido de acuerdo a los entornos existentes, para simular el proceso de Auto-ensamblaje.
- d) Seleccionar software con los requerimientos necesarios para desarrollar el entorno requerido considerando las necesidades del Auto-ensamblaje.
- e) Desarrollar entorno para la implementación del Auto-ensamblaje.
- f) Simular algunas arquitecturas en 3D utilizando el modelo clásico del Auto-ensamblaje basado en la construcción de nidos de avispa.
- g) Integrar la PSO en el modelo clásico de Auto-ensamblaje para optimizar el movimiento coordinado de los agentes constructores.
- h) Simular arquitecturas en 3D incorporando la PSO.

- i) Comparar los resultados obtenidos con el modelo clásico Vs. los resultados obtenidos mediante el modelo de Auto-ensamblaje con PSO.

### **Justificación**

La evolución de la inteligencia artificial a través de las ciencias de la computación, ha permitido el uso de modelos obtenidos mediante el estudio y la observación del comportamiento de los seres vivos y plantas, inmersas en el medio ambiente donde el hombre se desenvuelve. Entre estos modelos se encuentran los basados en inteligencia de enjambre (avispa, hormiga, peces, aves, entre otros), que durante su evolución han brindado los recursos algorítmicos suficientes, que permiten encontrar soluciones a problemas complejos implementando técnicas simples, basadas en la imitación del comportamiento de estos enjambres.

Sin embargo para explotar el potencial de estos algoritmos, se encuentran limitaciones en cuanto a software que permitan su implementación con fines de investigación académica. Actualmente es posible experimentar de dos formas con el Auto-ensamblaje basado en la construcción de nidos de avispa; primero utilizando un software propietario desarrollado para el estudio de procesos de construcción colectiva en sociedades de insectos llamado Nest. Este fue creado para los experimentos de Theraulaz y Otros, el cual fue actualizada por James Adam [6], a su versión más reciente (*Nest 3.0*). La segunda forma para implementar este modelo de Auto-ensamblaje es mediante los entornos de desarrollo con capacidad de sistemas multiagente y visualización en 3D, como la realizada por Marcin [7] en el entorno *BREVE*.

Esta investigación permite generar aportes a la inteligencia de enjambre mediante la implementación del modelo de Auto-ensamblaje integrado con la PSO para mejorar su rendimiento. Además se desarrolla un entorno de simulación específica para la investigación del Auto-ensamblaje basado en la construcción de

nidos de avispa, con la intención de eliminar las restricciones que envuelven el software propietario, y las limitaciones encontradas por Marcin [7], en cuanto a los requerimientos y alcances del algoritmo de Auto-ensamblaje, como la coordinación y velocidad de convergencia.

## **Alcances y Limitaciones**

### **Alcances**

Este trabajo permite implementar la inteligencia de enjambre para explorar la solución de problemas complejos, como lo es la construcción cooperativa de estructuras en 3D mediante auto-ensamblaje basado en la construcción de nidos de avispa, mejorando el tiempo de construcción con la optimización por enjambre de partículas, además se desarrolla un entorno que permita la investigación y simulación del auto-ensamblaje con fines académicos, es decir un entorno que permite al investigador la definición de micro-reglas, tener la capacidad de cargar las micro-reglas a partir de archivos, y la posibilidad de integrar el Auto-ensamblaje con otros tipos de Algoritmos como PSO.

### **Limitaciones**

Hasta el momento de formalizar la inscripción de esta investigación, no se han presentado situaciones que supongan el retardar o impedir la ejecución de este proyecto.

## **CAPITULO II**

### **MARCO TEÓRICO**

#### **Antecedentes**

La inteligencia colectiva ha incrementado sus seguidores desde diferentes áreas de investigación, convirtiéndose en punto de convergencia multidisciplinaria entre las ciencias y la tecnología, debido a sus aportes simples para la solución de problemas complejos. Las investigaciones realizadas en este campo se encuentran enfocadas hacia diferentes ciencias y sus aplicaciones son diversas, sin embargo considerando el espectro de esta investigación a continuación conoceremos los trabajos más significativos que ayudaran a consolidar los objetivos de este trabajo.

Zhongkui W. (2009) [8], estudió la construcción cooperativa de enjambres como una conducta basada en estigmergia mediante un sistema multi-agente, donde los agentes del enjambre no se comunican directamente entre ellos, no tienen la representación global de la estructura o modelo a construir y solo pueden percibir la información disponible, almacenada o modificada a través del ambiente, es decir el enjambre de agentes se coordina a través del entorno, comunicándose indirectamente para lograr el auto-ensamblaje de una estructura global. También logró modelar el comportamiento estigmergi en el espacio 3D como un sistema de eventos discretos (DES).

Quintero (2010) [9], realizó el Reconocimiento de Rostros con Visión por Computadora y Optimización Basada en Cúmulo de Partículas, donde expone con suficiente claridad la forma de utilizar la PSO en la solución de problemas de optimización, además concluyó que el algoritmo tiene la capacidad de responder con gran eficiencia, a la solución de problemas que requieren la exploración de un espacio de búsqueda grande.

Casto (2011) [10], desarrolló un conjunto de librerías utilizando la Arquitectura Unificada de dispositivos de Cómputo, CUDA (*“Compute Unified Device Architecture” en inglés*), para optimización con enjambre de partículas y un algoritmo genético. Además realizó implementación para evaluar las dos estrategias de optimización y poder establecer comparaciones entre los dos métodos. Este trabajo ayudó a identificar como definir la función objetivo que deberá minimizar el enjambre partículas durante el proceso de Auto-ensamblaje.

## **Bases teóricas**

### **Inteligencia Colectiva**

También llamada inteligencia de enjambre, es un campo de investigación científica, multidisciplinaria, que se interesa en los procesos distribuidos (no supervisados) de una organización. Este campo ha propuesto un gran número de sistemas artificiales bio-inspirados para diferentes propósitos: *resolver problemas de optimización, coordinación de robots, auto-ensamblaje, organización de bases de datos, protección de virus*, entre otras. Este término aparece por primera vez a finales de la década de los ochenta, sin embargo para finales de los noventa, Bonabeau, Dorigo y Theraulaz presentan la concepción de algoritmos de resolución distribuida, de problemas inspirados por el comportamiento colectivo de las colonias de insectos sociales y de otras sociedades animales, que ha inspirado diversas aplicaciones según Perozo (2011) [11].

Los comportamientos colectivos mostrados por las colonias de insectos se pueden categorizar de acuerdo a cuatro funciones que emergen a nivel de la colonia y que organizan su comportamiento global. Estas funciones son: *coordinación, cooperación, deliberación y colaboración*. Así, estas funciones apoyan el procesamiento de información de la colonia de acuerdo a dos ejes principales:

- a) La coordinación y colaboración forman estructuras sociales, temporales y espaciales que resultan del trabajo de la colonia. La coordinación regula la densidad espacio-temporal de los individuos mientras que la colaboración regula la asignación de sus actividades.
- b) La cooperación y deliberación provee herramientas para que la colonia pueda enfrentar los retos ambientales. La deliberación permite los mecanismos que apoyan las decisiones de la colonia, mientras la cooperación representa los mecanismos que superan las limitaciones individuales.

Las estructuras complejas a nivel de colonia, y muchos aspectos de la llamada inteligencia de enjambre de los insectos sociales, pueden ser comprendidos en término de redes de interacción y ciclos de retroalimentación entre los individuos. Los sistemas realizados bajo el enfoque de la inteligencia colectiva se caracterizan por exhibir características que hacen a las sociedades de insectos exitosas en su entorno, tales como: flexibilidad, robustez, control descentralizado y auto-organización. Particularmente, cuando hablamos de inteligencia colectiva debemos tratar dos aspectos claves: *la auto-organización* y *la interacción indirecta*.

- a) ***La auto-organización*** en este tipo de sistemas; es un proceso donde el patrón (arreglo organizado particular de objetos en espacio o tiempo) a nivel global de un sistema, emerge solamente de numerosas interacciones entre los componentes del nivel inferior. Por otra parte, las reglas que especifican las interacciones entre los componentes del sistema se ejecutan usando solamente información local, sin referencia al patrón global. Por ejemplo, las estructuras emergentes en el caso de la búsqueda de comida (forrajeo) en las hormigas incluyen redes organizadas espacial y temporalmente de rastros de feromona (sustancia química excretada por algunos animales que influye en el comportamiento de los de su misma especie). La esencia de la auto-

organización es que un sistema adquiere una estructura espacial, temporal o funcional sin interferencia específica del exterior. La organización puede desarrollarse en el tiempo o en el espacio, puede mantenerse en una forma estable, o puede mostrar fenómenos transitorios. La característica principal de estos sistemas auto-organizativos es su capacidad de lograr tareas colectivas complejas con comportamientos individuales simples, sin un control central o estructura jerárquica. Esta capacidad para resolver problemas complejos se debe al *Comportamiento emergente* (un comportamiento nuevo y estable) que es producido por el sistema y, que es más complejo que el comportamiento de componentes individuales del sistema. La auto-organización requiere de cuatro Elementos:

- **Retroalimentación Positiva.** Está constituida por reglas de comportamiento simple que promueven la creación de estructuras. Por ejemplo, el reclutamiento y el reforzamiento realizado en las hormigas a través del rastro dejado por la feromona, o por la danza en el caso de las abejas.
- **Retroalimentación Negativa.** Permite el equilibrio con la retroalimentación positiva y ayuda a estabilizar el patrón colectivo: puede tomar la forma de saturación, agotamiento o competición. En el ejemplo de forrajeo, la retroalimentación negativa proviene del número limitado de hormigas disponibles, agotamiento de la fuente de alimento, multitud en la fuente de alimento, o competición entre las fuentes de alimento.
- **Amplificación de las Fluctuaciones (Aleatoriedad).** La aleatoriedad es frecuentemente crucial ya que permite el descubrimiento de nuevas soluciones que pueden actuar como semillas para que nuevas estructuras se desarrollen y se fortalezcan. Por ejemplo, una hormiga que se pierda siguiendo el rastro de sus compañeras. Aunque este fenómeno puede parecer ineficiente, esto favorece la exploración de nuevas áreas, y poder encontrar fuentes de alimento no explotadas.
- **Múltiples Interacciones.** Un simple individuo puede generar una

estructura auto-organizada, tal como un rastro estable provisto de un tiempo de vida feromonal suficiente. Sin embargo, la auto-organización generalmente requiere de una densidad mínima de individuos tolerantes mutuamente. Además, los individuos deben ser capaces de hacer uso de los resultados de sus propias actividades, como también de las actividades de los otros. Por ejemplo, redes de rastro pueden auto-organizarse y ser usadas colectivamente si los individuos usan la feromona de los otros. Un enjambre está definido como la auto-organización útil de múltiples entidades a través de interacciones locales. En la siguiente tabla se resumen algunos ejemplos de este comportamiento en la naturaleza.

Comportamiento Enjambre	Entidades
Generación de patrones	Bacterias
Formación de caminos	Moho Fangoso
Organización de nidos	Hormigas
Transporte cooperativo	Hormigas
Selección de la fuente de alimentos	Hormigas
Termo-regulación	Hormigas, Abejas
Asignación de tareas	Abejas
Construcción de colmenas	Avispas
Sincronización	Abejas, Avispas, Avispones, Termitas
Construcción de redes	Luciérnagas, bancos de peces, bandadas de aves
Cardumen “Schooling”	Arañas
Bandadas “Flocking”	Peces
Asedio de presas “Prey Surrounding”	Aves
	Lobos

**Cuadro N° 1. Algunos ejemplos de enjambres en la naturaleza. Perozo (2011) [11].**

- b) **Interacción Indirecta** (“*Stigmergy*”, en inglés). En el caso de “*Stigmergy*”, es un mecanismo de retroalimentación usado para reflejar las interacciones indirectas entre los agentes y el entorno. Permite alcanzar formas emergentes de comportamiento coordinado a nivel de la sociedad. Este concepto fue introducido por Grassé en 1959, según Theraulaz [2]. Grassé estudio el comportamiento de un tipo de termitas durante la construcción de sus nidos, y notó que el comportamiento de las obreras durante el proceso de construcción era influenciado por la estructura de las construcciones. Este mecanismo es un poderoso principio de cooperación en las sociedades de insectos. Ha sido observado dentro de muchas otras sociedades de insectos, como las avispas,

abejas y hormigas. Está basado en el uso del entorno como medio de inscripción de los efectos de comportamientos pasados, para influenciar el futuro. Se inspira en las investigaciones de algunas sociedades biológicas de insectos que han demostrado que se coordinan produciendo un campo disipante o de acción en su ambiente. Este mecanismo está basado en un proceso llamado autocatalítico, que significa que mientras más ocurre un proceso más posibilidad tiene de ocurrir en el futuro.

## **Modelo de Auto-ensamblaje**

### ***Origen***

Las avispas pueden socialmente construir estructuras simples y complejas, que dependen de la forma en que las partes del nido se organizan. En las especies de avispas han sido distinguidas varias arquitecturas principales, por el naturista francés Saussure (1583 – 1858), y posteriormente modificado por Richards y Richards (1951) según Bonabeau [1]. A continuación se describe brevemente la arquitectura de los nidos de avispas más observados.

- c) Los nidos de *Stelocytarus* tienen panales relacionados con el sustrato y, cuando hay varios panales mutuamente están relacionados por el tallo con una estructura llamada pedicelos. Se encuentran dos tipos de estos nidos. En un grupo donde los panales están encerrados con una envoltura externa, llamados calyptodomous (cubierto con una envoltura), Construidas por varias familias de avispas como la *Vespa*, *Vespula*, *Provespa*, *Angiopolybia*, *Chaterginus*, *Leipomeles* y *Parachartergus*. El segundo grupo no tiene una envoltura y los Nidos se denominan gymnodomous, Este tipo de arquitectura se ve en *Polistes*, *Mischocytarus*, *Belonogaster*, y algunas especies de *ropalidia* y *Stelopolybia*.

- d) El nido de las *Astelocytarus* son calyptodomous, pero el panal entero está construido directamente sobre el sustrato. La posición del orificio de entrada varía de una especie a otra y la estructura de la envoltura es más simple que la estructura de envoltura de los nidos *stelocytarus*.
- e) Los Nidos de *Phragmocytarus* tienen estructuras modulares. El primer módulo es un panal horizontal cubierta con una envoltura que tiene una entrada. El segundo módulo está construido directamente sobre la cara exterior del primer módulo y las celdas se agregan en la cara interior de la envoltura. Módulos siguientes se agregan en la misma forma y la estructura resultante parece ser una serie muy bien organizada de panales regularmente espaciadas dentro de un cilindro. Dependiendo de la posición del orificio de entrada en el módulo, una abertura de comunicación central o periférico pasa a través de panales sucesivos, lo que permite a las avispas para pasar de un piso a otro. Este es el tipo de arquitectura construida por *Polybia*, *Chartergus*, *Epipona*, *Brachygastra* y *Protonectarina*.

Bonabeau [1] encontró que Wenzel en el 1991 realizó una clasificación de las arquitecturas de nido de avispa en más de sesenta subtipos diferentes, con muchos intermediarios entre las formas extremas descritas anteriormente. Un nido maduro puede tener desde unas pocas celdas hasta un millón en panales apilados, esta última suele ser construido por las especies altamente sociales. Una revisión de los tipos de nidos realizado por *Vespidae* revela la existencia de tres grandes cambios arquitectónicos que dan lugar a nidos altamente estructurados en las especies eusociales más avanzados. Estos cambios incluyen la agrupación de celdas aisladas del panal organizado, el desprendimiento del panal del sustrato a través del pedúnculo, y la protección de los panales con una envoltura.

Las principales funciones biológicas del nido de avispa son reproductivas, para proteger a las crías de los depredadores y asegurar las condiciones adecuadas para el

desarrollo de las crías. El nido también posee una función social, facilitar la cohesión de la sociedad, una identidad colonial, y la comunicación entre los compañeros de nido.

La primera hipótesis para justificar la capacidad de construcción supone que las avispas individualmente tenían internamente (innatas) una representación global de la estructura que se produciría. De esta manera, todas las decisiones adoptadas por las avispas se podrían hacer sobre la base de esa representación. Pero pruebas experimentales no pudieron confirmar esta hipótesis, sin embargo experimentos realizados posteriormente en el barro con la avispa *Paralastor*, donde las estructuras de nido fueron modificados por las perturbaciones experimentales, reveló la insuficiencia de la hipótesis propuesta y destacó, que por el contrario, el comportamiento de la construcción se basaba en un programa de construcción innata, donde se debe identificar cuáles son las características principales de este programa que fortalece la coordinación y cómo surgen cuando varios insectos juntos construyen su nido.

La principal característica identificada consiste en que la construcción del nido se produce como una secuencia de estímulo-respuesta, en donde la culminación de una etapa proporciona el estímulo para el comienzo del siguiente, es decir; una avispa comienza con la excavación de un agujero estrecho de aproximadamente ocho centímetros de largo y ocho milímetros de ancho, cuando el agujero es completamente y rayado con lodo, la avispa inicia la construcción de un elaborado embudo de barro grande y por encima de su entrada, el embudo se basa en cinco etapas distintas. La primera etapa consiste en la construcción del embudo, en la segunda etapa la avispa deja de construir de manera uniforme hacia arriba y comienza la construcción de una curva uniforme en el vástago del embudo, luego que la curva se ha completado, en la tercera fase se inicia con la formación de una campana con la separación de los troncales para formar un reborde uniforme de aproximadamente dos centímetros de diámetro, con la fase cuatro el reborde se amplía más en el lado más cercano hacia tallo que en otros lugares, dando así una campana con característica de

asimetría en una dirección, finalmente en la quinta etapa los lados de la campana se forman mediante la construcción de manera uniforme hacia abajo desde el borde.

Al final de cada etapa de construcción, los estímulos para las respuestas que conducen a la realización de la próxima etapa son encontrados por la avispa como consecuencia de su comportamiento anterior. Una consecuencia de este comportamiento es que el orden en que los estímulos se presentan en el curso de la construcción debe seguir una secuencia precisa, pero si por alguna razón un estímulo desencadena la construcción de acciones que causan el sub-elemento anterior de la arquitectura que se produce en una fase posterior, el resultado es la construcción automática de una estructura redundante y una arquitectura anormal de nido.

Es fácil imaginar que si dos avispas distintas no distinguen el producto de su actividad de otros, ellas podrían completar la estructura del mismo nido. Una avispa podría continuar el trabajo realizado por otra avispa, cualquiera que sea la etapa de construcción del nido. Este mecanismo puede convertirse en un paso hacia la cooperación indirecta entre los individuos. Este es precisamente el tipo de mecanismo que Grasse tenía en mente cuando se introdujo el concepto de stigmergy.

### ***Estigmergia***

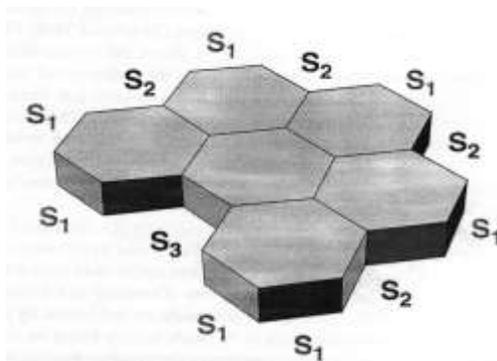
Es un paradigma de comunicación mediante el cual se pueden alcanzar la coordinación de las actividades a través del medio ambiente, es un mecanismo importante de la construcción del nido en los insectos sociales. Dos tipos de stigmergia se distinguen: cuantitativos o stigmergia continua, en donde los diferentes estímulos que desencadenan la conducta son cuantitativamente diferentes, y cualitativa o stigmergia discreta, en donde los estímulos se pueden clasificar en diferentes clases que difieren cualitativamente, para este trabajo se abordará el enfoque cualitativo.

La estigmergia cualitativa o discreta difiere de estigmergia cuantitativa en que los entes interactúan mediante estímulos cualitativos o discretos (*etiquetas*), mientras que por ejemplo; cuando las termitas construyen pilares responden a los estímulos cuantitativos determinados por los campos y gradientes de feromonas. La estigmergia cualitativa en los insectos se basa en un conjunto de tipos de estímulos discretos: por ejemplo, un insecto responde a un estímulo tipo-1 con una acción A y responde a un estímulo tipo-2 con una acción de B.

### ***Reglas individuales de construcción***

El número de sitios potenciales en los que una nueva celda se puede agregar aumenta a medida que avanza la construcción y también los permisos de construcción para continuar las actividades en paralelo. Debido a las acciones paralelas, la construcción puede llegar a ser desorganizado y conflictivas si las acciones se realizan simultáneamente. La cuestión de dónde colocar las nuevas celdas puede ser resuelta por las limitaciones de la propia arquitectura, que al parecer pueden canalizar adecuadamente la actividad de construcción. Pero ¿Cómo? Consideremos, por ejemplo, el panal se muestra en la figura N° 1, con seis celdas ya construidas.

Este panal tiene doce obras de construcción posibles: siete sitios con una de las paredes adyacentes (*con la etiqueta S1*), cuatro con dos paredes adyacentes (*S2*), y otro con tres paredes adyacentes (*S3*). Se ha verificado que el número medio de las obras de construcción potenciales varía según el número de celdas del panal ya construidas. Puesto que el crecimiento del panal se produce por igual en todas las direcciones. Surge la pregunta: ¿Cómo son agregadas las celdas para garantizar que el panal crece de manera uniforme en todas direcciones desde el pedicelo?



**Figura N° 1. Celdas construidas y etiquetadas.**

Estudios previos realizados por Bonabeau [1], mostraron que los algoritmos de construcción individual de Polistes consisten en una serie de decisiones "si-entonces" y "sí-no". Mediante un cuidadoso estudio de la dinámica de las configuraciones locales de las celdas durante el desarrollo de la construcción, se pudo observar que no hay el mismo número de sitios con una, dos o tres paredes adyacentes, la mayoría son sitios con dos paredes adyacentes. El número de sitios con una de las paredes, evidentemente, está fuertemente correlacionado con el número de sitios con tres paredes, es decir, cada vez que un sitio con tres paredes adyacentes es creado, dos sitios con una de las paredes se insertan automáticamente en la estructura. Otras investigaciones sobre *P. dominulus* y *P. fuscus* revelaron resultados similares.

Obviamente las avispa están influenciadas por la construcción anterior, las decisiones de construcción parecen estar hechas a nivel local basada en las configuraciones percibidas, de esta manera es posible que se limite la dinámica de construcción.

Una regla de construcción es que las avispa tienden a terminar una fila de celdas antes de iniciar una nueva fila, pero esta regla no es determinista, también se ha comprobado que la probabilidad para adicionar una celda en un sitio de tres paredes es aproximadamente diez veces superior a la de un sitio de dos paredes. ¿Cuáles son las consecuencias de esta estigmergia discreta probabilística en el desarrollo de la estructura de panal? Para estudiar este punto se introduce un modelo

de estigmergia cualitativo para ayudarnos a entender de una manera más general, los tipos de restricciones asociadas a los nidos construidos en un entorno distribuido muy discreto.

### *Modelo 3D para el Enjambre (Grid)*

Inspirado por las observaciones y datos empíricos descritos anteriormente, Bonabeau [1] introduce una clase de algoritmos para explorar la estigmergia cualitativa como un modelo de construcción del nido. Se considera una serie de agentes que se mueven de forma asincrónica en un espacio discreto en tres dimensiones y se comporta a nivel local mediante máquinas de estímulo-respuesta, los agentes eran constructores que respondieron a los estímulos específicos presentados por la estructura de panal en desarrollo. El espacio tridimensional en el que los agentes se mueven se compone de un grid cúbico o hexagonal, de ahí el nombre de grid de enjambres. El depósito de un bloque en la construcción por un agente depende de la configuración local de los ladrillos en las celda que rodean la celda ocupada por el agente (*denominaremos a esta configuración, micro-regla*), los ladrillos no podían ser eliminados después de ser depositados en la estructura de construcción. Aumentar el número de tipos de ladrillos incrementa la cantidad de las configuraciones locales que los agentes utilizan para responder.

Una micro-regla se define como la asociación entre una configuración particular de un estímulo, y el ladrillo que se va a depositar, y un algoritmo se define como conjunto de micro-reglas compatibles. Por lo tanto, el algoritmo se compone de un conjunto de reglas si-entonces donde la parte "si" es para chequear la configuración particular del bloque en la estructura, y la parte "entonces" tiene como resultado depositar en ese sitio uno de los tipos de bloques.

El algoritmo puede ser caracterizado por su tabla de micro-regla: la cual es una tabla de consulta que contiene todas las micro-reglas. Las configuraciones de estimulación y acciones asociadas, pueden ser determinista si todas las micro-reglas

tienen una probabilidad unitaria y si ésta probabilidad es asignada entre 0 y 1, para cada micro-regla entonces el algoritmo es probabilístico.

En este modelo la arquitectura puede ser completada por un solo autómatas, pero el comportamiento de construcción individual está determinado por las configuraciones locales que desencadenan acciones de desarrollo y tiene que ser organizado de tal manera que un grupo de autómatas pueden producir también la misma arquitectura. A continuación se describe el algoritmo de construcción presentado por Bonabeau [1].

```
/* Inicialización */
Construir tabla de consulta /* igual para todos los agentes */
Colocar el primer bloque en un sitio predefinido /* Seleccionar un sitio del Grid */
/* Distribuir los m agentes*/
Para k = 1 hasta m
  Colocar k agentes aleatoriamente en sitios desocupados del grid.
Fin /*del Para que distribuye los agentes */
/* Bucle Principal */
Para t = 1 hasta t_maximo /* Iteraciones que Realizaran los agentes */
Para k = 1 hasta m /* Acciones Ejecutadas por cada agente */
  Evaluar la configuración Local
  Si /* La configuración Local está en la tabla de consulta*/
    Entonces
      Construir el bloque especificado por la tabla de consulta
    Si-no /*La configuración Local no se encuentra definida en la tabla de consulta*/
      No construir
  Fin /*Fin de la evaluación "si-entonces" ó "Si-no"*/
  Moverse aleatoriamente a un sitio vecino y desocupado en el grid.
Fin /* Para t que controla las acción ejecutada por cada agente m */
Fin /* Para k que controla las iteraciones realizan los agentes */
/* Parámetros Utilizados en simulación */
m = 10
```

**Cuadro N° 2. Algoritmo de Construcción de Nidos de Avispas.**

Este algoritmo como lo expresan sus promotores es muy simple de implementar pero requiere de un entorno 3D que utilice sistemas multi-agentes. El potencial para construir estructuras complejas dependen de sus micro-reglas, para las cuales se deben considerar dos restricciones: primero, todos los agentes deben responder de manera uniforme al conjunto de configuraciones locales que se encuentran en la tabla de consulta, y segundo, que la arquitectura debe construirse a través de una sucesión de etapas constructivas cualitativamente distintas, cada una de estas etapas se caracteriza por una serie de estimulaciones configuradas localmente. Cuando cada agente del grupo, se une a la construcción de una arquitectura, actuando sobre el medio ambiente en el tiempo  $t$ , y su propia acción en el tiempo  $t + 1$  puede ser detectada por otro agente en el próximo ambiente.

El proceso de construcción se puede seguir adelante de dos formas, mediante una sucesión de configuraciones individuales que permiten a los agentes construir un ladrillo y producir la siguiente configuración de estimulación, y a través de un proceso paralelo donde varias configuraciones permiten simultáneamente la construcción de ladrillos.

Para crear una determinada arquitectura mediante un enjambre de agentes, utilizando un conjunto de comportamientos estigmergicos cualitativos deben cumplirse dos condiciones: La construcción debe ser descompuesta en un número finito de etapas constructivas, y la configuración de la estimulación local creada en un momento dado deben ser diferentes de las creadas por una fase de construcción anterior o siguiente. Para analizar el funcionamiento del algoritmo se describe a continuación las consideraciones necesarias para realizar su implementación.

- f) El algoritmo de Auto-ensamblaje requiere un entorno de desarrollo que permita disponer de un espacio tridimensional (Grid) donde cada celda tenga capacidad

de modificar sus propiedades particulares como el color, forma y estado entre otros.

- g) El entorno debe tener la capacidad para implementar sistema multi-agente donde cada agente puede moverse aleatoriamente a través del Grid, y logre actualizar las propiedades de una celda, las cuales dependen de las Micro-reglas.
- h) Una micro-regla se puede considerar como un vector cuyo tamaño depende de la forma geométrica de la celda definida en el Grid (*Cubo: 26, Hexágono: 20*), y representan los estados de la vecindad que se desea tenga la celda que el agente está evaluando. Cuando se agrupa un conjunto de micro-reglas es lo que recibe el nombre de tabla de consulta, la cual debe estar disponible para todos los agentes ya que define la estructura a ser construida.
- i) Colocar un bloque inicial que puede ser aleatorio o específico en el Grid dependiendo de la complejidad y espacio requerido por la estructura.
- j) Liberar aleatoriamente cada uno de los agentes a utilizar en celdas diferentes desocupados del grid.
- k) Se puede observar en el algoritmo que el bloque principal tiene un bucle anidado donde el externo determina cuantas iteraciones se exige al grupo agentes, y en el interno como controla la ejecución de todos los agentes, es decir; en este caso una iteración es la ejecución de 10 agentes que necesariamente no representa acciones de construcción porque recordemos esto depende de las micro-reglas.

- l) La ejecución del agente en su primera etapa consiste en evaluar la configuración local donde se encuentra el agente, esto significa conocer los estados de la vecindad referido a la celda donde se encuentra el agente, y debe tener simetría con los registros de la tabla de consulta.
  
- m) Una segunda etapa es la acción que realizará el agente y depende de comparar su configuración local, con cada una de las micro-regla de la tabla de consulta, si existe alguna similitud entonces actualizar las propiedades de la celda donde se encuentra el agente según las especificaciones de la micro-regla (*hay construcción se coloca un ladrillo*), de lo contrario el agente no actualiza la celda (*no hay construcción*).
  
- n) Y la última etapa de la ejecución del agente indiferentemente si hubo o no construcción es que éste se mueve aleatoriamente a otra celda vecina y desocupada del grid.
  
- o) Finalmente solo esperamos las iteraciones necesaria para visualizar como va emergiendo la estructura en el Grid definida por las micro-reglas, y lograr apreciar cómo es posible esta técnica de construcción, sin la comunicación entre los agentes, pero coordinando sus acciones a través de su entorno, mediante una comunicación indirecta llamada estigmergia cualitativa.

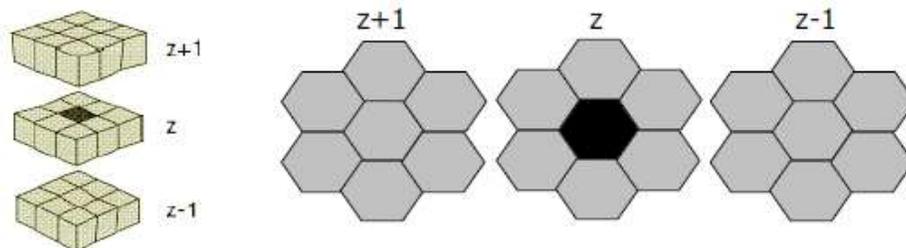
### **Las Micro-reglas**

Son las encargadas de definir la identidad de una determinada estructura o patrón, a través del modelo de Auto-ensamblaje basado en la construcción de nidos, es decir; una micro-regla aporta la información suficiente para responder las dos condiciones fundamentales del Auto-ensamblaje; "si-entonces", "sí-no". En el primer caso la micro-regla indica que están dadas las condiciones para realizar una acción de

construcción y entrega las especificaciones con la cual se debe fabricar en el determinado punto donde se consulta la micro-regla, y el segundo caso la micro-regla permite determinar que no están dadas las condiciones para realizar construcción y que es necesario moverse hacia otro punto del área de construcción.

La información de una estructura se encuentra distribuida en un conjunto de micro-reglas (*tabla de consulta*), el cual se encuentra disponible para ser consultado por los agentes constructores, para decidir cuál de sus dos tareas ejecutar: *construir con las especificaciones indicada por la micro-regla y moverse a otro punto o solo moverse aleatoriamente*. Según Marcin [7], el tamaño de una micro-regla se encuentra asociado a la vecindad, que simplemente es el número de celdas que tridimensionalmente envuelven un punto determinado donde hay que considerar la forma geométrica del Grid, clásicamente el cubo el tiene 26 y el hexágono 20 elementos en su vecindad respectivamente.

Los estados se pueden definir mediante símbolos numéricos o alfanuméricos, que representan los atributos de cada celda del espacio de construcción. Por ejemplo, considerando el utilizar números enteros para representar la definición de los estados tenemos; un 0: representa celda vacía, 1: un tipo de Bloque, 2: otro tipo de bloque y así sucesivamente, se pueden definir cualquier número de estados que represente una característica particular en la construcción. Veamos a continuación como se identificaría; una vecindad, los estados, las micro-reglas y una tabla de consulta en las siguientes figuras.



**Figura N° 2. Vecindad de una celda cubica y hexagonal.** (*La Celda Negra Representa la posición del Agente y todo lo demás es la vecindad.*)

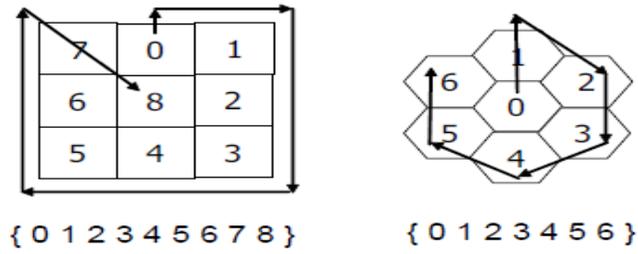


Figura N° 3. Secuencia de extracción de la vecindad en el cubo y el hexágono.

<b>Z-1</b>									
Posición	0	1	2	3	4	5	6	7	8
Estado	0	0	0	0	0	0	0	0	0
<b>Z</b>									
Posición	0	1	2	3	4	5	6	7	8
Estado	0	0	0	0	0	0	0	0	0
<b>Z+1</b>									
Posición	0	1	2	3	4	5	6	7	8
Estado	0	0	0	0	0	0	0	0	0
$V_1 = \{0,0,0,0,0,0,0,0,0,0\}$									

Cuadro N° 3. Ejemplo 1. Extracción de la vecindad ( $V_1$ ) en el Cubo.

<b>Z-1</b>									
Posición	0	1	2	3	4	5	6	7	8
Estado	1	0	0	0	0	0	0	0	0
<b>Z</b>									
Posición	0	1	2	3	4	5	6	7	8
Estado	0	0	0	2	0	0	0	0	0
<b>Z+1</b>									
Posición	0	1	2	3	4	5	6	7	8
Estado	0	0	0	0	0	0	0	0	2
$V_2 = \{1,0,0,0,0,0,0,0,0,2\}$									

Cuadro N° 4. Ejemplo 2. Extracción de la vecindad ( $V_2$ ) en el Cubo.

<b>Reglas</b>	<b>Z-1</b>	<b>Z</b>	<b>Z+1</b>	<b>Nuevo estado</b>
Regla_1	1,0,0,0,0,0,0,0,0	0,0,0,2,0,0,0,0	0,0,0,0,0,0,0,0,2	1
Regla_2	2,2,2,2,2,2,2,2	0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0	2
Regla_n	2,0,0,0,0,0,0,0,2	0,1,0,0,0,0,0,0	0,0,0,0,2,0,0,2,0	2

**Cuadro N° 5. Ejemplo de una tabla de consulta.**

Para lograr que se construya una estructura con el algoritmo de auto-ensamblaje, es necesario que los agentes constructores en cada iteración realicen una validación entre la vecindad de la celda que se evalúa y cada una de las micro-reglas disponibles en la tabla de consulta, para determinar el “si-entonces”, o “si-no” se requiere un cambio de estado en la celda evaluada, lo cual significa la colocación de un bloque que contribuye a la construcción de una estructura global. Por ejemplo:

*Suponemos el ejemplo N° 1*, sea extraer la vecindad y tenemos:

Vecindad = { 0,0 }

*Validamos:* Se compara la vecindad con cada micro-reglas disponible en banco de micro-reglas presentado en el cuadro N° 4.

Si Vecindad = regla\_1 entonces cambiar estado a (1);

Si Vecindad = regla\_2 entonces cambiar estado a (2),

Sino Dejar igual

No se cumple ninguna micro-reglas no hay construcción, (condición si-no)

*Ahora el ejemplo N° 2*, cuya vecindad es:

Vecindad = { 1,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,2 }

*Validamos:*

Si Vecindad = regla\_1 entonces cambiar estado (1);

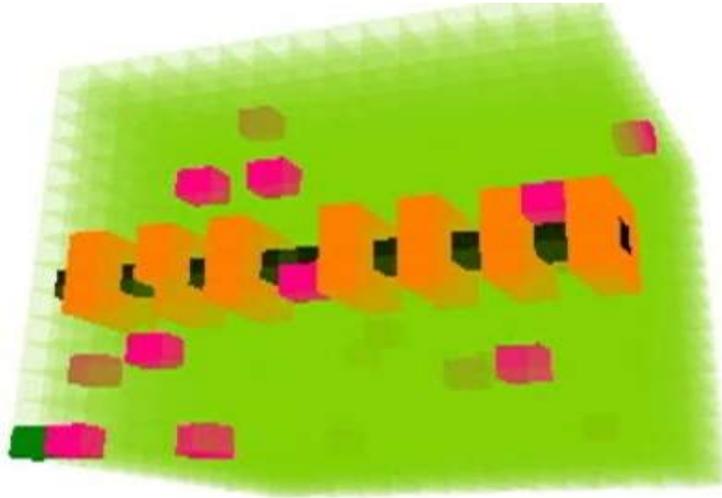
Si Vecindad = regla\_2 entonces cambiar estado (2),

Sino Dejar igual

Se puede ver que la vecindad es igual a la regla\_1 entonces la celda donde se encuentra el individuo de construcción cambiara su estado a 1, lo que significa un cambio de las características de la celda, según la definición del estado. A continuación se presenta una pequeña implementación de auto-ensamblaje, se definen tres estados; 0: Celda Libre, 1: Bloque de Color Negro, 2: Bloque de color Naranja, 3: Los Agentes. El siguiente cuadro Muestra el conjunto de micro-reglas predefinidas para construir mediante auto-ensamblaje la estructura que se puede apreciar en la figura N° 4.

<b>Micro-Reglas</b>	<b>Estados Requeridos en la Vecindad que habilitan el cambio a un Nuevo Estado</b>	<b>Nuevo estado</b>
R0	0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	1
R1	0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0	1
R2	0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0	2
R3	0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0	2
R4	0, 0, 0, 1, 0, 0, 0, 0, 0, 2, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0	2
R5	1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0	2
R6	0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0	2
R7	0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0	2
R8	0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0, 1, 0, 2, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0	2
R9	0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1	2

**Cuadro N° 6. Tabla de consulta.**



**Figura N° 4. Estructura construida con Auto-ensamblaje utilizando la tabla de consulta predefinidas en el cuadro N° 5.**

### **Requerimientos para el Auto-ensamblaje**

Para simular un proceso de Auto-ensamblaje como se pudo apreciar en el segmento anterior es necesario un ambiente 3D con capacidad para una distribución geoméricamente uniforme del espacio de construcción, y un sistema multi-agente que tenga para ejecutar las tareas de construcción con acceso a la tabla de consulta. Sin embargo en los inicio de esta investigación fue posible identificar una herramienta computacional llamada *Nest*, dedicada exclusivamente al Auto-ensamblaje basado en la construcción de nidos de avispa, por supuesto netamente propietaria.

La herramienta *Nest* en su primera versión disponía de ciertas limitaciones en cuanto a las geometrías disponibles para cada celda del Grid, las cuales fueron superadas en la versión *Nest 3.0*. A continuación se presenta una comparación de la evolución de esta aplicación según James [6].

<b>Característica</b>	<b>Nest-2.11.1</b>	<b>Nest-3.0</b>
Lenguajes Implementados	C++, X11	Ruby, C++, OpenGL
Plataforma	Unix	Unix, Windows, Mac OS X
Extensibilidad	Requiere recompilar el código fuente.	Basado en comandos de alto nivel mediante Ruby.
Arquitectura Geométrica	Cubico, Hexagonal; codificado en el sistema	Puede soportar múltiples Arquitecturas
Tipos de Bloques	Limitado a 4 Colores	Virtualmente Ilimitado el numero de colores.
Reglas	Tamaño fijo con rotación alrededor del eje Z	Tamaño ilimitado con rotación en los tres ejes
Comportamiento de la construcción	Los agentes solo pueden colocar ladrillos en la posición Vacía donde se encuentra.	Los agente pueden colocar en cualquier espacio vacío alrededor, y también es posibles la eliminación (excavación)

**Cuadro N° 7. Diferencias de la Versiones del Nest, James [6].**

En este punto ya es obvio tener identificado los requerimientos ideales para experimentar ampliamente con Auto-ensamblaje, sin embargo es significativo resaltar que existen entornos de desarrollo con capacidad 3D y con sistemas multi-agentes, pero presentan dificultad para introducir la implementación de algoritmos adicionales como AG, PSO u otros. Por esta razón, se considera desarrollar para esta investigación una herramienta computacional con las especificaciones mínimas y con una estructura que permita alcanzar los requerimientos considerados ideales identificados en el desarrollo de esta investigación, tal como se muestra en el cuadro 8.

<b>Especificaciones Mínimas</b>
<ul style="list-style-type: none"> <li>• Sistemas Multi-Agente.</li> <li>• Representación en 3D del Grid.</li> <li>• Capacidad para que los agentes puedan moverse aleatoriamente en las celdas del Grid</li> <li>• Mecanismo para colocar a disposición de los agentes la tabla de consulta (<i>las micro-reglas</i>).</li> <li>• Actualización o modificación de las propiedades (<i>geometría, color, estado, posición</i>) de cada celda desde el agente.</li> <li>• Geometría de la Celda o bloque: Cubico, Hexagonal</li> <li>• El agente ubicado en una celda debe poder validar los estados de su vecindad para compararlos con la tabla de consulta, esto le permitirá decidir si la celda requiere alguna actualización.</li> <li>• Exportación de estructura construida</li> </ul>
<b>Especificaciones Adicionales</b>
<ul style="list-style-type: none"> <li>• Geometría de la Celda o bloque: Múltiples</li> <li>• El agente ubicado en una celda pueda validar su vecindad, ampliándola en diferentes niveles, rangos o criterios de referencia.</li> <li>• Generador interactivo de micro-reglas.</li> <li>• Configuración interactiva del Grid, Agente</li> <li>• Vista desde diferentes ejes incluyendo inmersas.</li> <li>• Mecanismo para cargar micro-reglas a partir de archivos.</li> <li>• Extensibilidad para integrar otros algoritmos.</li> </ul>

**Cuadro N° 8. Requerimientos Ideales Para la Implementación a través de Auto-ensamblaje.**

**Optimización por Enjambre de Partículas (PSO)**

Para Cruz [12], la PSO es una metaheurística evolutiva inspirada en el comportamiento social de la bandada de pájaros o bancos de peces, desarrollada por Eberhart y Kennedy en 1995. Este comportamiento consiste en la transmisión de sucesos de cada individuo a todos los demás individuos de la bandada o enjambre, logrando una sinergia que según Juaquin [13], permite a los individuos satisfacer sus necesidades inmediatas.

González Y Otros [14], considera que la PSO trabaja con un conjunto de soluciones candidatas denominado enjambre. Cada miembro del enjambre es una partícula, y esta posee un vector de solución denominado posición. Cada partícula conoce la mejor posición encontrada por el enjambre o mejor global. Si se definen subconjuntos de partículas del enjambre, cada uno de estos se denomina vecindad. Para este caso, cada partícula conoce la mejor posición de su propio vecindario o mejor local. Una vecindad define como las partículas interactúan entre ellas, y determina como la información se propaga en el enjambre afectando la convergencia del algoritmo según Lin [15].

Como introducción se presentan algunas características que permiten conocer el origen y evolución de la optimización por enjambre de partículas según Fernández [16] y Lazinica [17].

- p) Surge en los 60 de la simulación y teorías, de adaptación social y de la generación de gráficos tridimensionales por computadora. El sistema está compuesto por partículas que se moverían en el espacio con características como: posición inicial, velocidad inicial, longevidad, tamaño inicial, color, etc. y su movimiento se rige por leyes físicas y su estética por otras leyes.
- q) Luego Reynolds empleó una bandada de pájaros, utilizando letras para indicar el movimiento, comportamiento de las partículas, y las características del entorno. Presenta tres reglas: evitar colisiones, relación de velocidades (se siguen a los mejores vecinos), y mantener la bandada centrada es decir juntar las partículas sin que haya divergencia.
- r) Seguidamente Kennedy le introduce al modelo un aporte relevante, como lo es el intercambio de información entre las partículas, argumentando que así como los humanos imitan a sus vecinos para mejorar y evolucionar, si las partículas toman como referencia sus mejores compañeras pueden avanzar hacia una solución óptima.

- s) Finalmente se han realizado diversas variaciones del modelo PSO, donde diferentes investigadores han explorado otras alternativas con intenciones de mejorar la eficiencia del algoritmo.

El Algoritmo de PSO tiene tres versiones relevantes las cuales permiten aplicaciones en espacio continuo, discreto y binario. Sin embargo para utilizar el PSO discreto y binario, se toma como punto de partida el continuo, y mediante algunas expresiones matemáticas se convierte en discreto o binario respectivamente [18]. Para ésta investigación se implementará el modelo PSO para aplicaciones en espacios discretos.

### **Algoritmo de la PSO**

Son diversas las versiones presentadas sobre el algoritmo de PSO original, desde un punto de vista comprimido o expandido, que en algunas casos es difícil reconocer algunas secuencias, sin embargo para este trabajo iniciaremos, presentando a continuación la descripción y modelo del algoritmo de PSO presentado por Juakin y Otros [13].

“La PSO consiste en un algoritmo iterativo basado en una población de individuos denominada enjambre, en la que cada individuo denominada partícula, se dice sobrevuela el espacio de decisión en búsqueda de soluciones óptimas.”

Cada partícula conoce su posición actual y la velocidad con la cual llega, estas son determinadas con las siguientes expresiones.

$$v_{ij}^{t+1} = w \times v_{ij}^t + C_1 \times rand() \times (P_{ij}^t - x_{ij}^t) + rand() \times (g_j^t - x_{ij}^t), \quad Ecu. 1$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1}, \quad Ecu. 2$$

A demás todas las partículas deben conocer su mejor posición individual, y la partícula que posee la mejor posición global en el enjambre. A continuación se define los términos utilizados en las ecuaciones y el algoritmo de PSO.

$v_{ij}^{t+1}$  : Velocidad de la partícula.

$x_{ij}^{t+1}$  : Posición de la partícula en espacio continuo.

**Inercia (w):** Es un parámetro utilizado para controlar el impacto de las velocidades anteriores en el cálculo de la nueva velocidad de la partícula, controla el balance entre la exploración global y local del espacio de búsqueda, un valor grande permite que se exploren nuevas áreas del espacio de búsqueda (*exploración*), y uno pequeño estimula la exploración del área cercana a la posición actual de la partícula (*explotación*). En este trabajo se considera este parámetro en el rango [0, 1], lo cual evita que las velocidades de las partículas crezcan sin control durante las iteraciones, a demás se introducirán diversos mecanismos donde la inercia pueden variar en cada iteración como los presentados por Ahmad y Otros [17].

**Parámetro Cognitivo (C<sub>1</sub>):** Es una constante de aceleración hacia la mejor posición local encontrada por la partícula.

**Parámetro Social (C<sub>2</sub>):** Es una constante de aceleración hacia la mejor posición global encontrada por el enjambre.

**Rand ( ):** Es una función que retorna un numero aleatorio en el intervalo [0,1].

**Mejor Posición:** Es posible percibir un poco de ambigüedad en este término, sin embargo en PSO su característica de mejor, se refiere a la distancia que se reduce durante las iteraciones, entre una solución optima deseada y la solución dada por las posiciones de la partículas cuando es evaluada la función objetivo.

$(P_{ij}^t)$  : Mejor Posición Individual lograda por la partícula,

$(g_j^t)$  : Mejor Posición global lograda por el enjambre.

**Función Objetivo:** Como su nombre lo indica, la PSO tiene la tarea de alcanzar una meta, que puede ser minimizar o maximizar una función como las encontrados en la implementación de PSO realizada por Guerra y Crawford [19].

**PSO Discreto:** La Optimización por Enjambre de Partículas nativamente se encuentra definido para el espacio continuo, y para utilizarlo en espacio discreto es necesario realizar una conversión mediante las siguientes ecuaciones [18]:

$$S_{id} = \frac{M}{1 + e^{-V_{id}}} , \quad Ecu. 3$$

$$X_{id} = round(S_{id} + (M - 1) \times \sigma \times rand ()) , \quad Ecu. 4$$

*Condiciones:*

$$Si X_{id} > M - 1 \quad Entoces \quad X_{id} = M - 1$$

$$Si X_{id} < 0 \quad Entoces \quad X_{id} = 0$$

$S_{id}$  : Velocidad para espacio Discreto.

$V_{id} = v_{ij}^{t+1}$  : Velocidad en espacio Continuo.

$M$  : Cantidad de espacio Discreto.

$X_{id}$  : Posicion en espacio Discreto.

$round()$  : Función para redondear.

$\sigma$  : Varianza.

$rand()$  : Funcion para Numero aleatorio entre el rango [0, 1].

```

/* Parámetros Iniciales (Inercia, Cognitivo, Social, Mejor global, Partículas, componente*/
w=0.7; C1=0.2; C2=0.4; g=1; cant_partículas=20; N=3, Itera=500;
Para i=1 hasta cant_partículas /* Inicializando el Enjambre */
    Para j=1 hasta N /* Para Cada Componente (x, y, z) */
        Partícula[i].posición[j] = rand ( );
        Partícula[i].velocidad[j] = rand ( );
    Fin /* Componentes */
    /* Evaluar Función Objetivo */
    Partícula[i].evaluación = evaluar (partícula[i].posición);
    Para j=1 hasta N /* Inicializar Mejor Posición individual */
        Mejor[i].posición[j] =Partícula[i].posición[j];
    Fin /* Mejor Posición Individual */
    /* Determinar Mejor posición Global */
    Mejor[i].evaluación = Partícula[i].evaluación;
    SI (Mejor[i].evaluación ES_MEJOR_QUE Mejor[g].evaluación) ENTONCES
        g=i; /* Actualiza el índice del mejor global */
    Fin /* De la mejor Posición Global */
Fin /* De la inicialización del enjambre */

Para k=1 hasta Itera /* Se realizan las iteraciones */
    Para i=1 hasta cant_partículas /* Actualizar la Velocidad y Posición de las partículas */
        Para j=1 hasta N /* Para Cada Componente (x, y, z) */
            Partícula[i].velocidad[j] = evaluar_ecu_1 ( );
            Partícula[i].posición[j] = evaluar_ecu_3_y_4 ( );
        Fin
    Fin /* De la Actualización de Las Partículas */
    /* Evaluar La posición actual de las partículas, Según Función Objetivo */
    Para i=1 hasta cant_partículas
        Partícula[i].evaluación = evaluar (partícula[i].posición);
        /* Guardar la mejor Posición individual de la partícula */
        Si Partícula[i].evaluación ES_MEJOR_QUE Mejor[i].evaluación Entonces
            Para j=1 hasta N /* Para Cada Componente */
                Mejor[i].posición[j] = Partícula[i].posición[j];
            Fin /* para Guardar la mejor posición Individual */
            Mejor[i].evaluación = Partícula[i].evaluación;
            /* Determinar si es una mejor posición global */
            Si Mejor[i].evaluación ES_MEJOR_QUE Mejor[g] Entonces
                g = i;
            Fin /* Para Guardar el Id, de la mejor Posición global */
            Fin /* para determinar el mejor Individual y Global */
        Fin /* De la Evaluación de las Partículas */
    Fin /* de la Iteración */
Retornar (Mejor[g].posición [ ]; /* Retorna Mejor solución Encontrada */

```

**Cuadro N° 9. Algoritmo de PSO.**

## **CAPITULO III**

### **MARCO METODOLÓGICO**

#### **Naturaleza de la Investigación**

Considerando que este trabajo explora técnicas computacionales de la inteligencia artificial, con la finalidad de buscar, implementar y validar estrategias que permitan optimizar la construcción de estructuras complejas o patrones en 3D, que pueden generar aporte a la solución de problemas en las ciencias y la ingeniería, permite la posibilidad de enmarcarlo en la modalidad de proyecto factible.

La investigación se basa en la implementación combinada de dos algoritmos de la inteligencia de enjambre; Auto-ensamblaje y PSO. El primero, para construir en 3D estructuras basadas en micro-reglas predefinidas o encontradas con la ayuda de otras técnicas de la inteligencia colectiva, y el segundo, la optimización por enjambre de partículas para optimizar el proceso de construcción del modelo de Auto-ensamblaje, por lo tanto el desarrollo de este trabajo se realizará en las siguientes fases:

- Fase I Diagnostico.
- Fase II Factibilidad.

#### **Fase I Diagnostico**

Durante esta fase se una realizó una recolección de información y análisis para alcanzar los objetivos planteados en esta investigación, estructurada de la siguiente manera:

- t) Se investigó y analizó el modelo de Auto-ensamblaje basado en la construcción de nido de avispa, fue posible realizar una implementación del algoritmo utilizando reglas-predefinidas para construir una estructura en 3D, lo cual permitió identificar su potencial y la necesidad de combinarlo con otra técnica computacional como la PSO, para intentar mejorar su velocidad de convergencia.
- u) Se investigó la optimización por enjambre de partículas, con el objetivo de establecer un mecanismo que permitiera integrarlo al algoritmo de Auto-ensamblaje, con la intención de reducir el tiempo de convergencia durante el proceso de construcción.
- v) Se realizó una revisión de los entornos de desarrollo disponibles para simular la implementación de los dos algoritmos con los requerimientos identificados, y los resultados obtenidos fueron poco satisfactorios, por esta razón, se desarrolló en esta investigación una herramienta computacional que permita simular la propuesta de este trabajo.

## **Fase II Factibilidad**

Luego de realizar la fase de diagnóstico, fue posible determinar que este proyecto es factible técnica y económicamente. Es factible técnicamente porque existe la documentación bibliográfica adecuada, y se dispone de expertos en el área de la inteligencia colectiva, además existen diversos entornos que pueden permitir el desarrollo en una aplicación computacional para satisfacer las necesidades de esta investigación. Y tiene factibilidad económica porque la inversión requerida para ejecutar este trabajo se encuentra dentro del rango admisible para el autor del proyecto.

## **CAPITULO IV**

### **PROPUESTA DEL ESTUDIO**

El modelo de Auto-ensamblaje basado en la construcción de nidos de avispa, permite la construcción de estructuras en 3D, previos experimentos realizados en [7], permitieron observar que el tiempo de construcción no es bueno en algunos casos (tiempo de convergencia muy alto), debido a que los agentes constructores navegan muy alejados del área de construcción, por esta razón, se propone evaluar la posibilidad de la capacidad construcción y reducir el tiempo ocioso de los agentes, mediante la combinación de la PSO con el modelo de Auto-ensamblaje. Para la ejecución de esta propuesta es necesario concretar las siguientes actividades: formalizar un algoritmo que permita combinar la PSO en el algoritmo original de Auto-ensamblaje, seleccionar una función objetivo, desarrollar una herramienta computacional que permita la construcción de estructuras en 3D mediante Auto-ensamblaje sin y con PSO, y finalmente, seleccionar dos conjuntos de micro-reglas que definan estructuras diferentes para la comparación del algoritmo propuesto y el original.

#### **Combinación de Auto-Ensamblaje y PSO**

Para establecer esta combinación se implementan los siguientes criterios:

1. En Auto-Ensamblaje el movimiento de un punto a otro, en el espacio 3D de los agentes se realiza de manera aleatoria, al incorporar la PSO, será ésta quien determina los puntos hacia donde se mueven los agentes.
2. La PSO en cada iteración realiza la búsqueda de un punto, donde se espera retorne con una mejor solución a la función objetivo. En Auto-ensamblaje, en cada iteración se intenta construir un bloque de la estructura según el estado de la vecindad. Al incorporar la PSO, cada agente realiza ambas acciones en cada iteración.

Considerando los criterios para la integración deseada, se presenta a continuación en cuadro 10, el algoritmo de Auto-Ensamblaje original con las modificaciones necesarias para incorporar la PSO.

```

/* Inicialización */
Construir tabla de consulta /* igual para todos los agentes */
Colocar el primer bloque en un sitio predefinido /* Seleccionar un sitio del Grid */
/* Distribuir los m agentes*/
Para k = 1 hasta m
Colocar k agentes aleatoriamente en sitios desocupados del grid.
Si Utiliza_PSO Entonces sitio= PSOParaAUTO (parámetros);
/* PSOParaAUTO(parámetros): Es la función que ejecuta la PSO y retorna un punto */
/* (x,y,z) del Grid según los criterios propuestos.*/
SINO sitio = rand ( );
Fin
Colocar agente k en sitio.
Fin /*del Para que distribuye los agentes */
/* Bucle Principal */
Para t = 1 hasta t_maximo /* Iteraciones que Realizaran los agentes */
Para k = 1 hasta m /* Acciones Ejecutadas por cada agente */
    Evaluar la configuración Local
    Si /* La configuración Local está en la tabla de consulta*/
        Entonces
            Construir el bloque especificado por la tabla de consulta
        Si-no /*La configuración Local no se encuentra definida en la tabla de consulta*/
            No construir
    Fin /*Fin de la evaluación “si-entonces” ó “Si-no”*/
Moverse aleatoriamente a un sitio vecino y desocupado en el grid.
Si Utiliza_PSO Entonces sitio= PSOParaAUTO (parámetros);
SINO sitio = rand ( );
Fin
Moverse hasta sitio.
Fin /* Para t que controla las acción ejecutada por cada agente m */
Fin /* Para k que controla las iteraciones realizan los agentes */
/* Parámetros Utilizados en simulación */
m = 10

```

**Cuadro N° 10. Algoritmo de Auto-Ensamblaje Combinado con PSO.**

La función *PSOparaAUTO* (*parámetros*), ejecuta el algoritmo de PSO pero realizando una iteración, y retorna un sitio del grid (*punto*  $x, y, z$ ) desocupado. Los parámetros son los requeridos por la PSO. Mediante este criterio se pretende lograr que los sitios suministrados por la PSOparaAUTO, se acerquen más en cada iteración al área de construcción, esperando una mejor cooperación de los agentes durante el proceso de construcción, mediante la coordinación de sus movimientos a través de una función objetivo.

### **Función Objetivo**

Para definir la función objetivo se considera descartar una expresión matemática que pueda representar una determinada estructura, porque estaría en contra de los fundamentos del modelo de Auto-Ensamblaje basado en la construcción de nidos de avispa, donde quedó establecido que las avispas no tienen una representación global de la estructura, sino que ejecutan un programa de construcción en función de los cambios percibidos en el ambiente donde se realiza la construcción. Considerando esto, se propone para formular la función objetivo las siguientes heurísticas, evaluadas en diversos experimentos:

- Se implementará la distancia euclidiana para determinar la distancia entre dos puntos  $d_E(P_1, P_3)$ .
- Se utilizará como punto de referencia fijo, la posición del bloque inicial ( $P_1$ ).
- La posición inicial de los agentes constructores es aleatoria, y varía en cada iteración ( $P_3$ ).
- Tomaremos como punto de referencia móvil, la posición del último bloque construido ( $P_2$ ), cuando se minimice la distancia entre  $d_E(P_1, P_3)$ .
- Se espera lograr el movimiento de los agentes entre el bloque inicial, y el último bloque colocado respectivamente, reduciendo la distancia entre los puntos.
- La lógica propuesta para el movimiento del enjambre consiste en un ciclo, que

se ejecuta tratando de moverse desde  $P_3$ , hasta  $P_1$ , en cada iteración, luego de que ésta distancia es minimizada con la función objetivo  $d_E(P_3, P_1)$ , el enjambre cambiará de dirección y se intentará mover hacia  $P_2$ , y la función objetivo pasa a ser la minimización de  $d_E(P_2, P_1)$ , y luego de minimizar esta distancia se repite el ciclo.

Tomando en cuenta las heurísticas propuestas tenemos que nuestra función objetivo está conformada por la distancia euclidiana implementada bajo cierta condición, como se puede apreciar a continuación:

$$d_E(P_3, P_1) = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2} \quad \text{Ecu. 5}$$

Condición:

$$\text{Si } d_E(P_3, P_1) < 1 \quad \text{Entonces } P_1 = P_2$$

**Esta condición le permite al enjambre moverse hacia lo que llamaremos *la referencia*, (que puede ser el bloque fijo  $P_1$ , ó bloque móvil  $P_2$ ) durante la minimización de la función objetivo, lo que permite a los agentes mejorar su posibilidad de construcción durante el recorrido, porque los mantiene aleatoriamente cerca del área de construcción. Esto se verá gráficamente en los experimentos realizados en el capítulo V.**

### **Software para Simulación**

Considerando la necesidad de apreciar visualmente, cómo emerge una estructura que se construye a partir de un comportamiento cooperativo, utilizando Auto-ensamblaje con PSO en un espacio 3D, se propone desarrollar una herramienta computacional en MATLAB, implementando el modelo de Auto-ensamblaje basado en la construcción de nidos de avispa, e incorporando la optimización por enjambre de partículas, tal como se muestra en el algoritmo presentado en el cuadro N° 9.



3. tener la opción de utilizar el Auto-Ensamblaje basado en la construcción de nidos de avispa, sin o con, la optimización por enjambre de partículas.
4. Brindar otras opciones que permitan utilizar mecanismos de inercia ( $w$ ) dinámicos para PSO, como los propuestos por Ahmad y otros [17], que pueden ser apreciados en el cuadro 12, con el fin de aprovechar el efecto que proporciona la explotación y exploración al variar la inercia en cada iteración.

$$w(iter) = 0.5 + \frac{rand}{2} \quad (WD1)$$

$$w(iter) = \frac{itermax - iter}{itermax} * (Wmax - Wmin) + Wmin \quad (WD2)$$

$$w(iter) = \frac{(itermax - iter)^2}{itermax^2} * (Wmax - Wmin) + Wmin \quad (WD3)$$

$$w(iter) = (Wmax - Wmin) * \frac{itermax - iter}{itermax} + Wmin * z \quad (WD4)$$

$$z = 4 * z' * (z' - 1),$$

$$z' = rand$$

$$w(iter) = \frac{1 - \beta}{1 - s\beta} \quad (WD5)$$

$$\beta = \frac{iter}{itermax} \quad s > -1$$

$$w(iter) = \left(\frac{2}{iter}\right)^{0.3} \quad (WD6)$$

$$w(iter) = Winicial * u^{iter} \quad \text{rango de } u \text{ [1.0001,10005]} \quad (WD7)$$

**iter:** Iteración.  
**w(iter):** Inercia por Iteración.  
**rand:** Numero Aleatorio [0,1].  
**itermax:** Cantidad de iteraciones.  
**Wmax:** Máxima Inercia.  
**Wmin:** Mínima Inercia.  
**Winicial:** Inercia de Inicio.  
**WD1, WD2, WD3, WD4, WD5, WD6 y WD7:** Inercias Dinámicas.

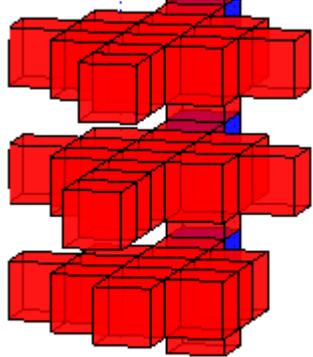
**Cuadro N° 12. Mecanismos de inercia dinámicos para PSO.**

5. Disponer de un espacio 3D, para visualizar la estructura que emerge mediante Auto-ensamblaje con y sin PSO.
6. Mostrar gráficamente en cada experimento, la tendencia de la distancia entre los agentes y el bloque de referencia, la cantidad de bloques construidos por cada iteración, además graficar la inercia dinámica cuando se utilice.



*Segundo Escenario – Estructura 2.*

Para este caso se propone una estructura más compleja que evoluciona en los tres ejes ( $x, y, z$ ), dependiendo de las condiciones establecidas en la micro-reglas, como se puede apreciar en el cuadro 14.

0,0,0,0,2,0,1	
0,0,0,0,1,0,2	
0,2	
0,0,0,0,0,2,0,2	
0,2	
0,2	
0,2	
0,2	
0,2	
0,2	

**Cuadro N° 14. Banco de Micro-reglas e imagen de la estructura que representa el escenario 2.**

La propuesta de evaluación en cada escenario considera acciones como:

- Construir la estructura utilizando de modelo de Auto-ensamblaje basado en la construcción de nidos de avispa clásico, y determinar el número de iteraciones que fijarán el umbral para establecer las comparaciones.
- Realizar múltiples experimentos para construir la estructura mediante Auto-ensamblaje con PSO, variando los parámetros de inercia, componente cognitivo y social, hasta calibrar los valores ideales para cada estructura.
- Finalmente establecer la comparación de los resultados obtenidos con el Auto-ensamblaje clásico y el propuesto, dentro del rango de iteraciones establecidas (umbral).

## CAPITULO V

### EJECUCION Y EVALUACIÓN DE LA PROPUESTA

Para implementar el modelo de Auto-ensamblaje basado en la construcción de nidos de avispa con o sin PSO, se desarrolló una herramienta computacional en MATLAB como se presenta en la figura N° 5, que se encuentra dividida en seis secciones que se describen a continuación en el cuadro 15:

<i>Secciones</i>	<i>Descripción</i>
<i>Menú e Iconos</i>	Esta sección está conformada por el menú y los iconos utilizados por el editor de graficas de Matlab.
<i>Parámetros de Auto-Ensamblaje</i>	Contiene las especificaciones necesarias para simular el Auto-ensamblaje, tales como: cantidad de iteraciones, número de agentes a utilizar, cantidad de celdas por eje distribuidas en el grid, botón para <b>“Cargar Micro-reglas”</b> desde un archivo, y un botón para <b>“Agregar bloque Inicial”</b> en una celda específica del grid.
<i>Parámetros de PSO.</i>	En esta sección se dispone de una caja de verificación <b>“incluye PSO”</b> , que habilita y deshabilita utilizar PSO en el modelo de Auto-ensamblaje, también se define los parámetros de inercia, componente cognitivo y social del enjambre.
<i>Otros Mecanismos de Inercia (w)</i>	Se dispone en este de espacio de siete mecanismos de inercia dinámica ( <i>ecuaciones matemáticas presentadas en el cuadro 12</i> ), donde en cada caso, la inercia varía por iteración según la expresión ecuación utilizada. Algunos de estos mecanismos utilizan un valor máximo y mínimo definido entre [0..1].
<i>Grid</i>	Representa el espacio 3D donde bloque a bloque, los agentes realizan la construcción de la estructura.
<i>Experimentos</i>	En este espacio se especifican las opciones relacionadas con los experimentos: <b>“Construir Estructura”</b> : Simula Auto-ensamblaje hasta alcanzar la cantidad de iteraciones especificadas. <b>“Gráfica”</b> : Muestra la distancia euclidiana entre el agente y la referencia, y la cantidad de bloques construidos en cada iteración, además si habilita la casilla <b>“Incluir Grafica de Inercia”</b> se podrá visualizar el valor de la inercia en cada iteración. <b>“Leer”</b> : Permite leer el archivo de experimentos almacenados. <b>“Guardar”</b> : Almacena un experimento realizado. <b>“Definir”</b> : Prepara un archivo para almacenar experimentos. <i>Lista de experimentos</i> : Muestra identificadores (ID) de los experimentos almacenados. <b>Para Graficar “...”</b> : Permite graficar hasta cinco experimentos almacenados, seleccionando los ID y el parámetro de comparación $w$ , $c_1$ , $c_2$ .

**Cuadro N° 15. Descripción de la herramienta de simulación implementada**

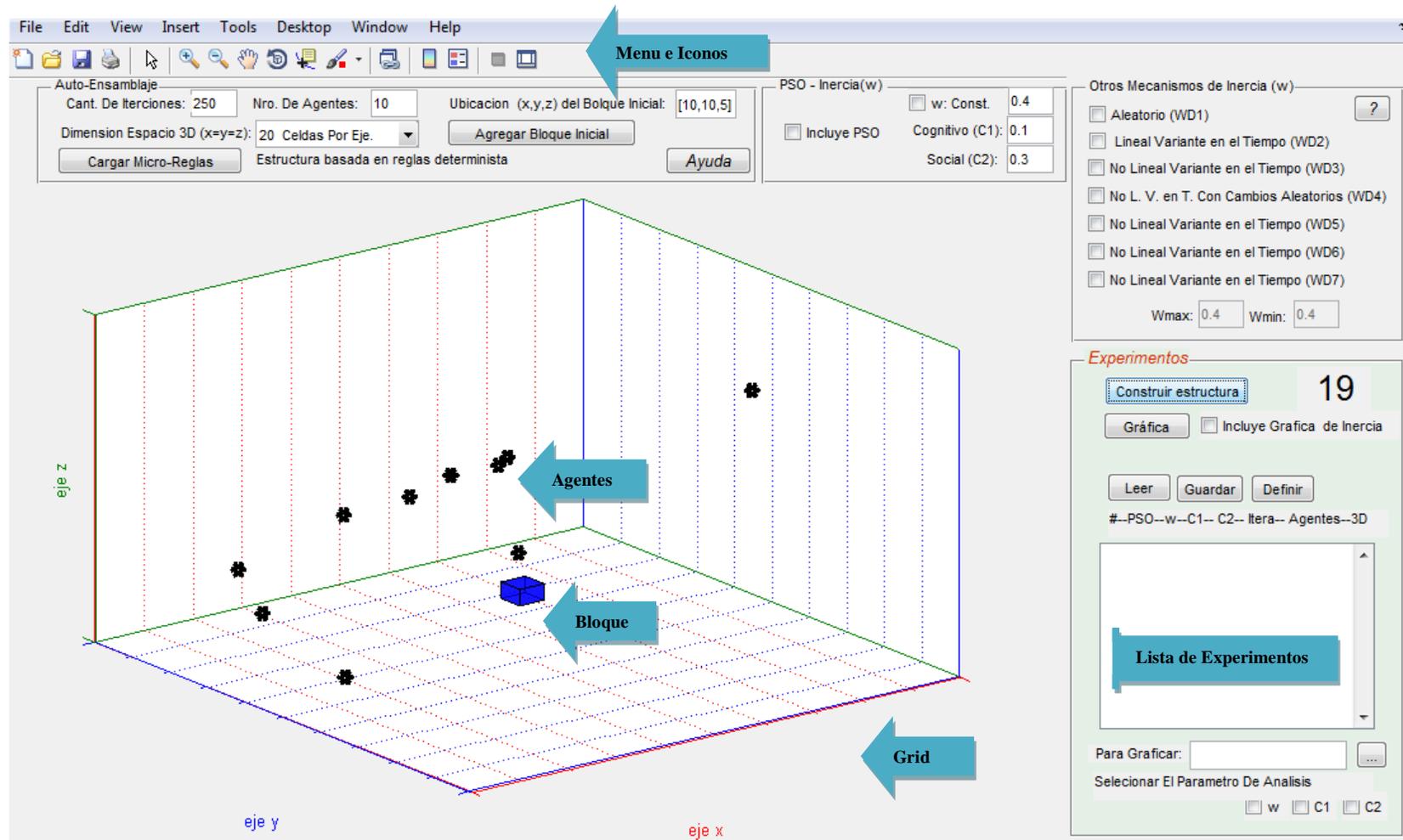


Figura N° 5. Interfaz Gráfica de la herramienta computacional desarrollada en Matlab para Auto-Ensamblaje.

## Evaluación

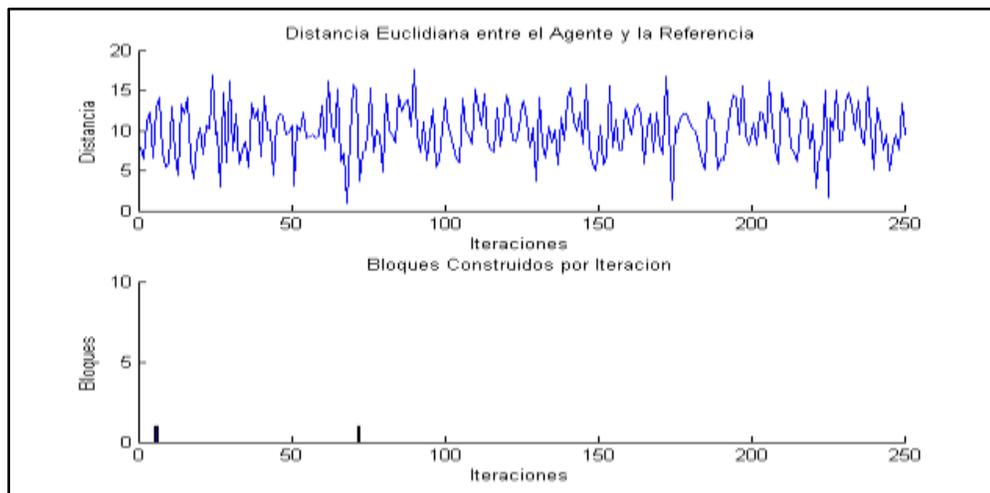
Utilizando la herramienta computacional desarrollada para simular el modelo de Auto-ensamblaje basado en la construcción de nidos de avispa con y sin PSO, se realizaron experimentos que permitieron identificar un estimado de las iteraciones necesarias, donde se pudiera apreciar visualmente la ejecución de ambos algoritmos en la construcción de la estructura propuesta. Es posible apreciar en cada escenario propuesto, que a partir de doscientos cincuenta iteraciones se puede observar la capacidad de construcción por iteración de cada modelo (*clásico y propuesto*). La evaluación se realiza en dos escenarios (*dos estructuras diferentes*), y en cada uno se ejecutan tres pasos, en el primero se trata de construir la estructura con el modelo de auto-ensamblaje clásico, en el segundo paso, se explora mediante experimentos para determinar los parámetros del PSO, y por último, se construye la estructura con el modelo de Auto-ensamblaje propuesto en este trabajo.

### Primer Escenario – Estructura 1.

*Paso 1:* Construir la estructura 1 con el modelo de Auto-ensamblaje clásico.



**Cuadro N° 16. Parámetros para simular el Auto-ensamblaje clásico con la Estructura 1.**

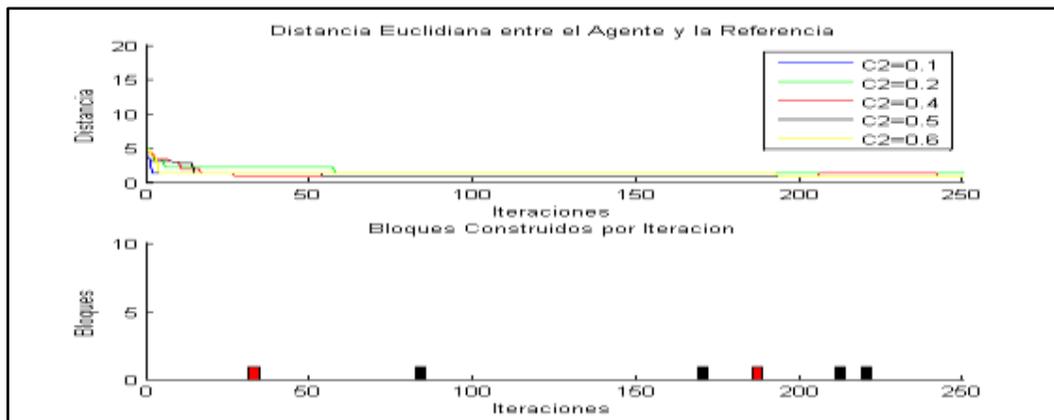


**Gráfica N° 1. Actividad de construcción de los agentes para la estructura 1 con Auto-ensamblaje clásico.**

Las gráficas que se muestran para cada evaluación, representan por cada iteración, en la parte superior, **la distancia euclidiana entre el agente y la referencia** (*definida en el planteamiento de la función objetivo*), y en la parte inferior, **la cantidad de bloques construidos**. Tomando como ejemplo la gráfica 1, se puede decir de la parte superior, que los agentes se mantuvieron explorando alejados del área de construcción, ya que la distancia euclidiana entre los agentes y la referencia (*en este caso el bloque inicial*), pocas veces logró ser menor que 1, por lo tanto hay pocos bloques construidos, como se puede apreciar la parte inferior de la gráfica 1. Cuando la distancia tiende 0, habrá muchos bloques construidos, porque el enjambre navega hacia el área de construcción.

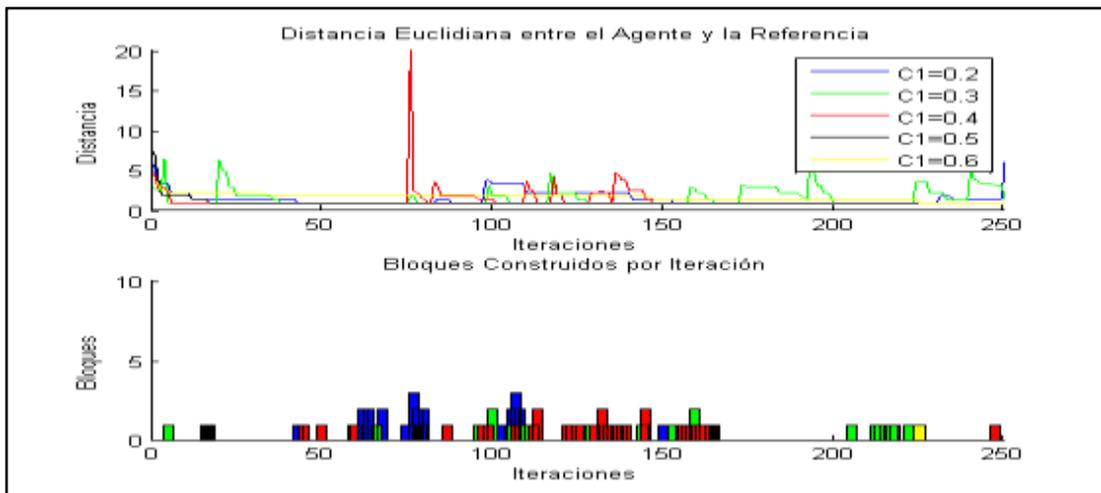
*Paso2:* Determinar mediante diferentes experimentos los parámetros ( $w$ ,  $c_1$ ,  $c_2$ ) que permitan una mejor actividad de construcción de los agentes con el modelo de Auto-ensamblaje propuesto con PSO.

- a) Se fijaron valores mínimos (0.1), para la inercia y el componente cognitivo ( $w, c_1$ ), y se varió en magnitud en un rango [0.1..0.9]; el componente social para cada experimento, logrando observar que el enjambre logra construir algunos ladrillos como se puede apreciar en la grafica N° 2, donde el componente social con valor de 0,5 es un buen punto de partida.

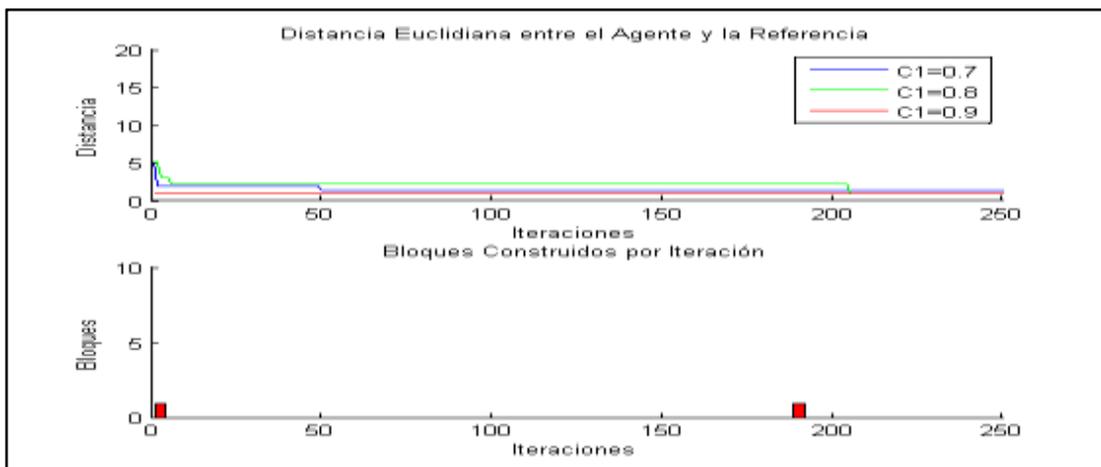


**Gráfica N° 2. Actividad de construcción de los agentes para la estructura 1 variando el componente social.**

b) Aplicando la estrategia anterior pero fijando a 0.1, la inercia ( $w$ ), y el componente social ( $c_2$ ), y variando el componente cognitivo, se logra observar que el mejor valor para  $c_1$  es 0.2, como se puede observar en las gráficas 3 y 4. Además a partir de estas gráficas se comienzan apreciar unos picos, en la distancia euclidiana, que representan la posición de los agentes constructores, cuando el enjambre cambia de dirección, después de minimiza la función objetivo.

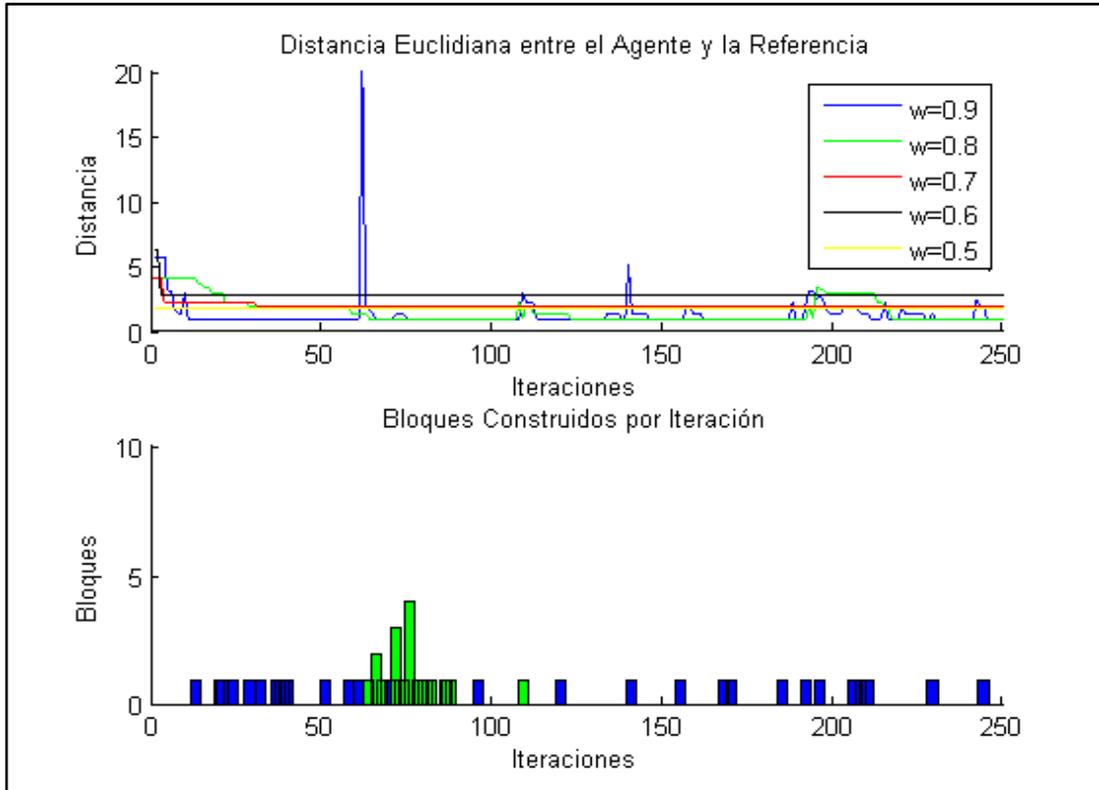


**Gráfica N° 3. Actividad de construcción de los agentes para la estructura 1 variando el componente cognitivo (Parte 1).**



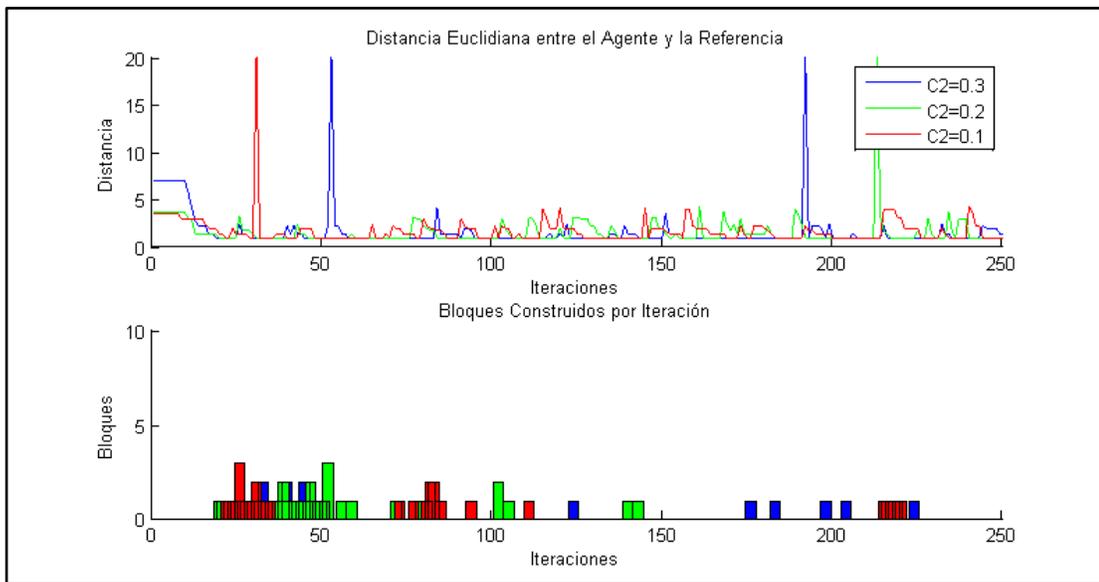
**Gráfica N° 4. Actividad de construcción de los agentes para la estructura 1 variando el componente cognitivo (Parte 2).**

- c) Utilizando la misma estrategia para obtener el mejor valor para el parámetro de inercia, se obtuvo que el valor de 0.9, para  $w$  puede ser un buen punto de partida.

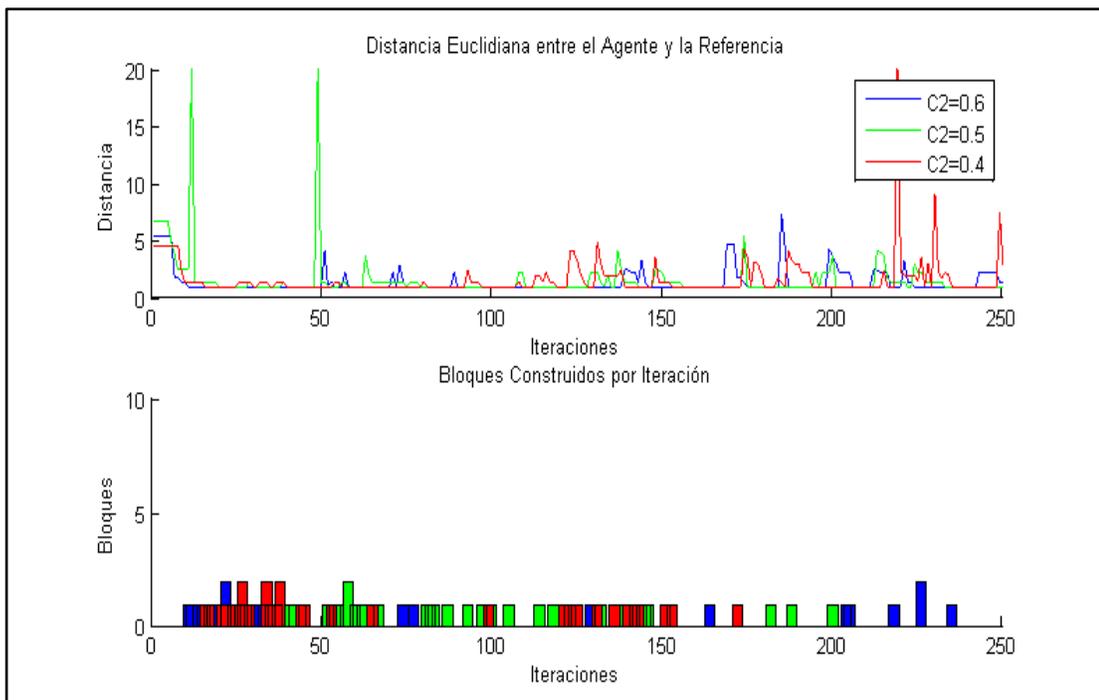


**Gráfica N° 5. Actividad de construcción de los agentes para la estructura 1 variando el parámetro de inercia.**

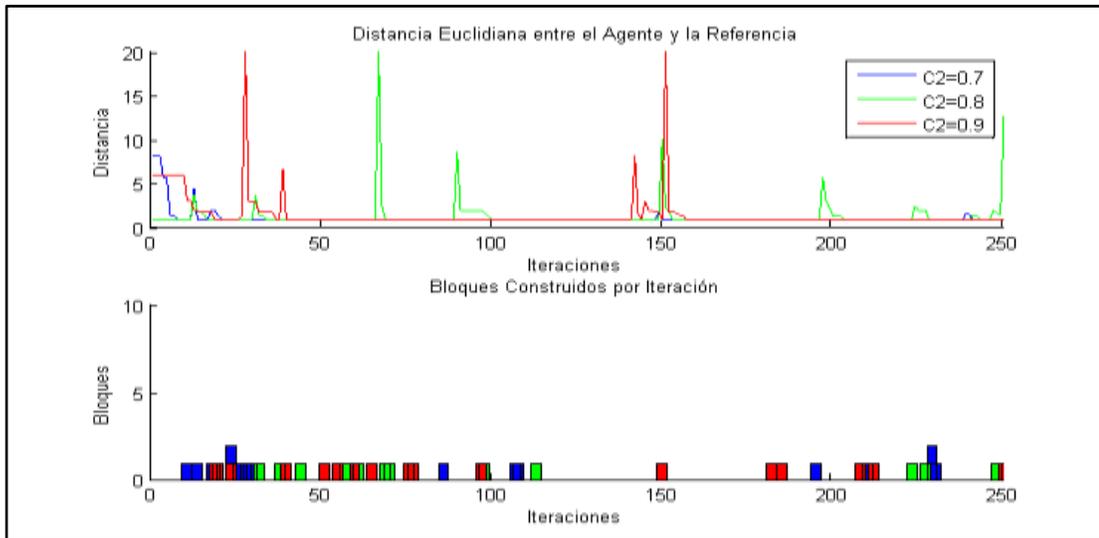
- d) Considerando los valores obtenidos para los parámetros utilizados por el PSO en los pasos anteriores, se repite la estrategia de exploración con  $w=0.9$ ,  $C_1=0.2$ , y variando  $C_2$  en magnitud en el rango  $[0.1..0.9]$ , fue posible encontrar un  $C_2 = 0.4$ , que puede ser más conveniente para la estructura 1, como se puede apreciar en las siguientes gráficas 6,7 y 8.



**Gráfica N° 6. Actividad de construcción de los agentes para la estructura 1 variando el componente  $C_2$  (Parte 1).**

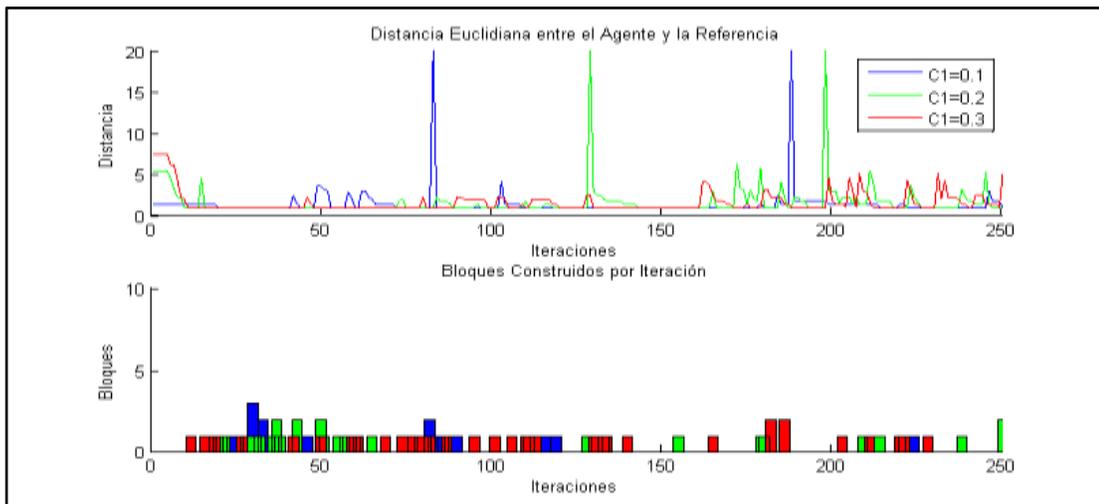


**Gráfica N° 7. Actividad de construcción de los agentes para la estructura 1 variando el componente  $C_2$  (Parte 2).**

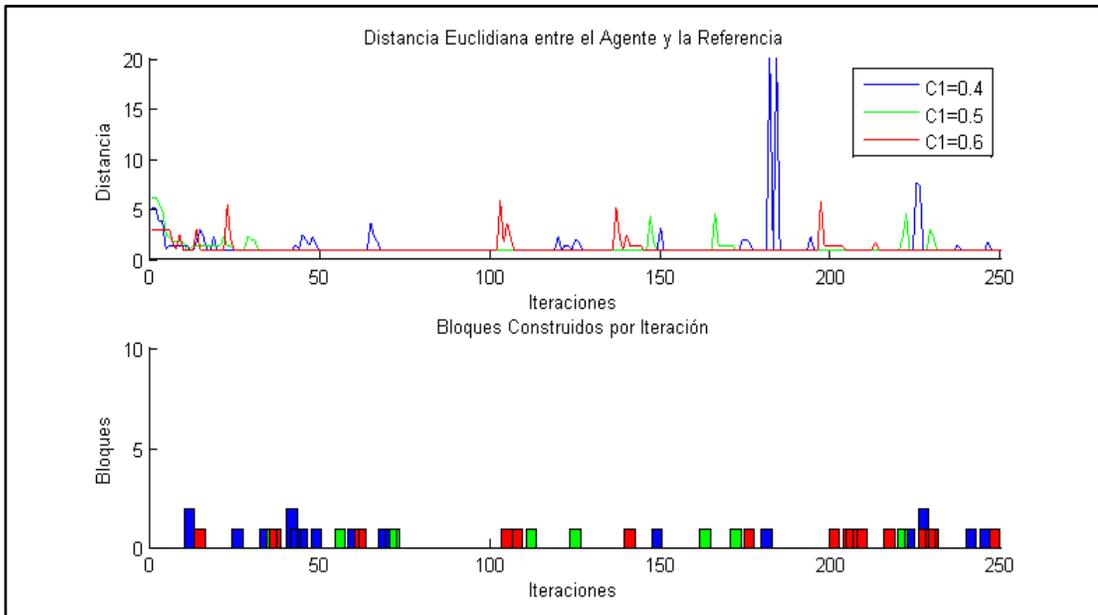


**Gráfica N° 8. Actividad de construcción de los agentes para la estructura 1 variando el componente  $C_2$  (Parte 3).**

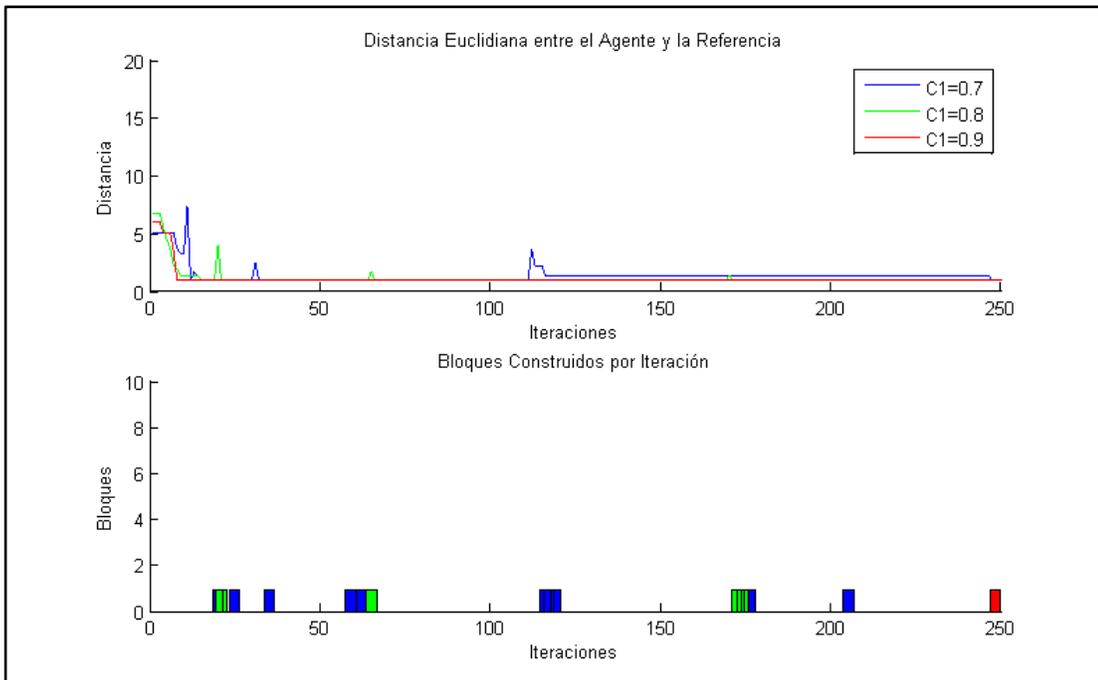
e) Aplicando el procedimiento anterior pero fijando  $w=0.9$ ,  $C_2 = 0.4$  y variando  $C_1$ , es posible considerar un mejor valor para este parámetro  $C_1 = 0.3$ , como se puede apreciar en las gráficas 9,10 y 11.



**Gráfica N° 9. Actividad de construcción de los agentes para la estructura 1 variando el componente  $C_1$  (Parte 1).**

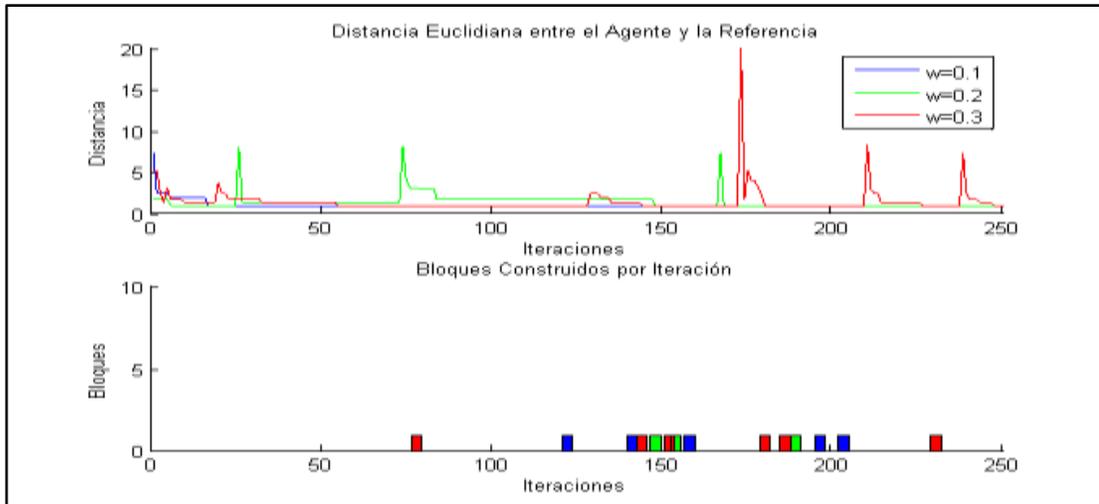


**Gráfica N° 10. Actividad de construcción de los agentes para la estructura 1 variando el componente  $C_1$  (Parte 2).**

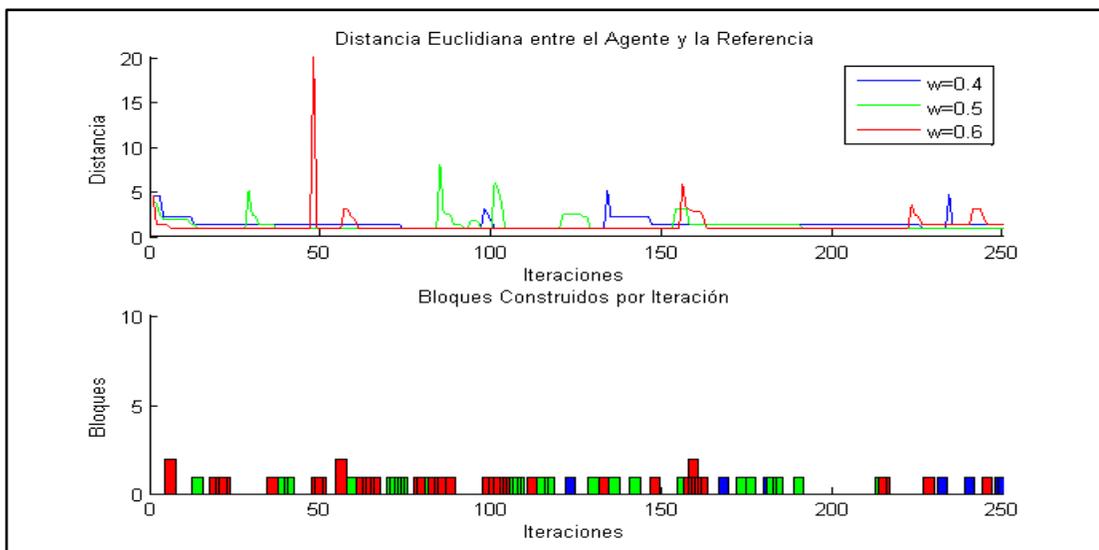


**Gráfica N° 11. Actividad de construcción de los agentes para la estructura 1 variando el componente  $C_1$  (Parte 3).**

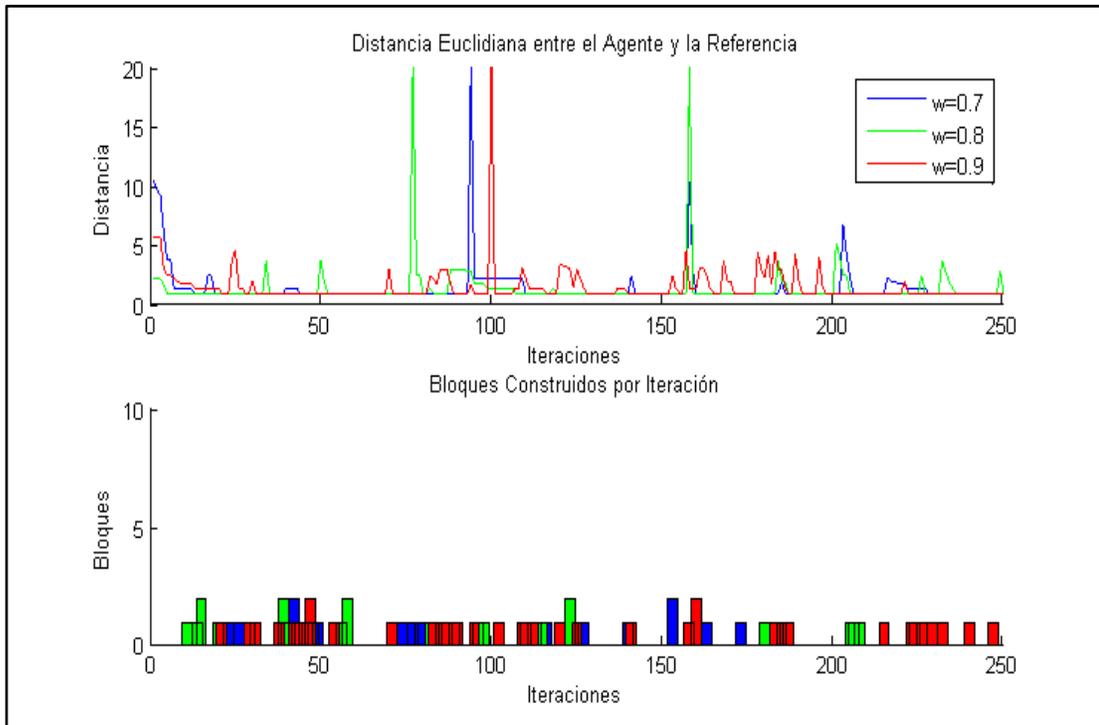
- f) Para buscar una inercia que pueda ser más conveniente para los nuevos valores encontrados para  $C_1=0.3$ , y  $C_2=0.4$ , se repite el procedimiento encontrando que  $w=0.9$  es adecuado para la estructura, como se puede apreciar en las gráficas 12, 13 y 14.



**Gráfica N° 12. Actividad de construcción de los agentes para la estructura 1 variando el parámetro  $w$  (Parte 1).**

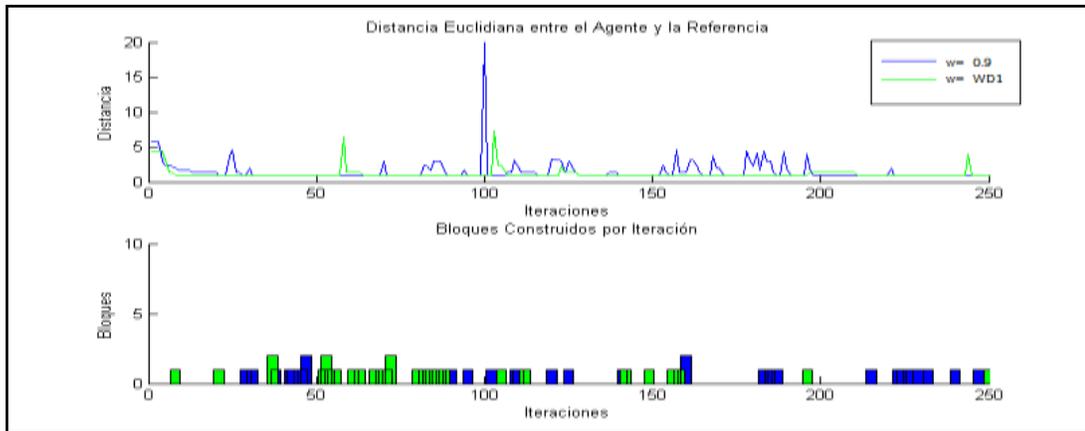


**Gráfica N° 13. Actividad de construcción de los agentes para la estructura 1 variando el parámetro  $w$  (Parte 2).**

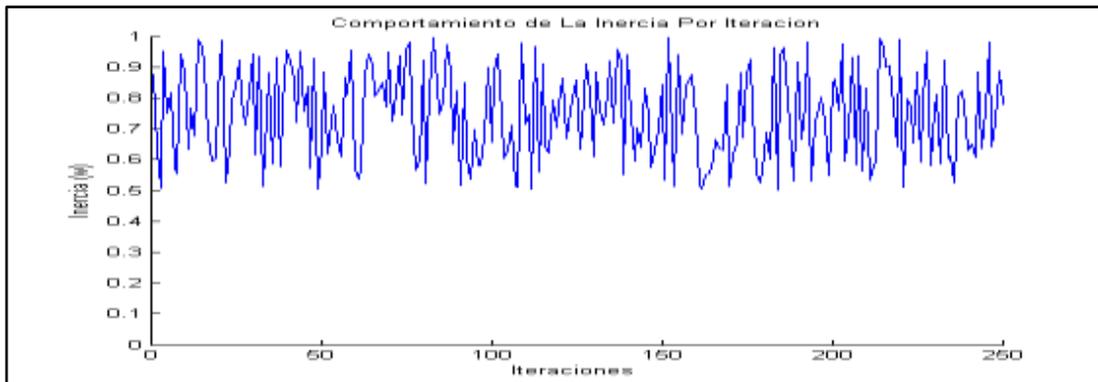


**Gráfica N° 14. Actividad de construcción de los agentes para la estructura 1 variando el parámetro  $w$  (Parte 3).**

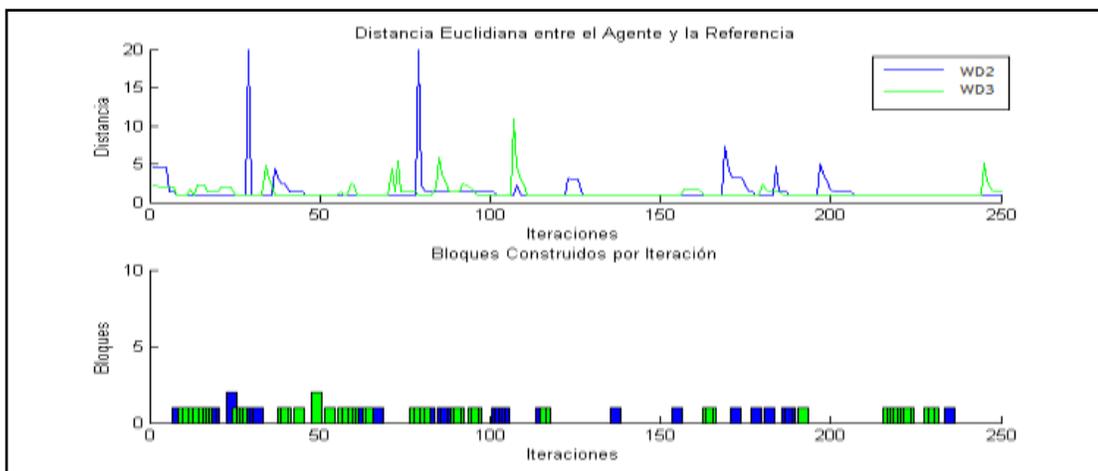
g) Finalmente se explora la posibilidad de encontrar un mejor valor para la inercia, de tipo dinámica, fijando los parámetros  $c_1=0.3$ , y  $c_2 = 0.4$ , y variando los mecanismos de inercia planteados en el cuadro 12, disponibles en la herramienta computacional. Se incorpora un nuevo gráfico que muestra el comportamiento de la inercia por iteración (*valor de  $w$  en cada iteración*), durante el proceso de construcción.



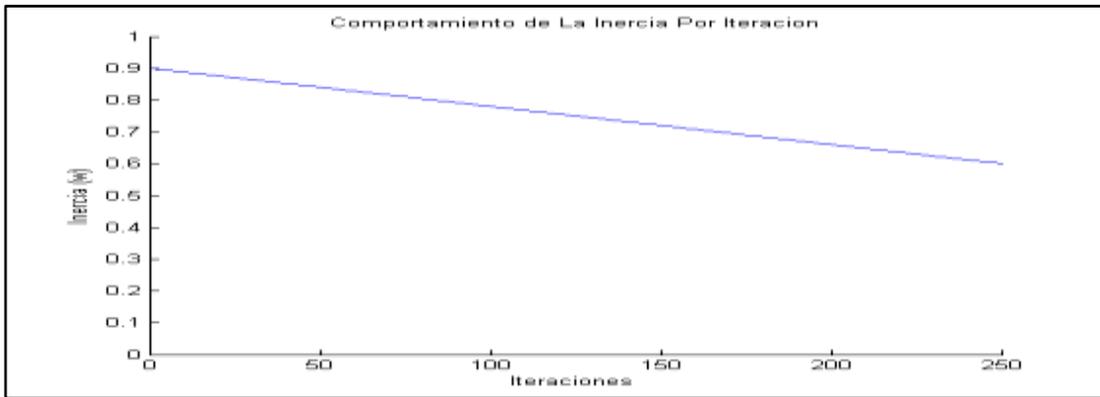
**Gráfica N° 15. Actividad de construcción de los agentes para la estructura 1 utilizando inercia dinámica *WDI*.**



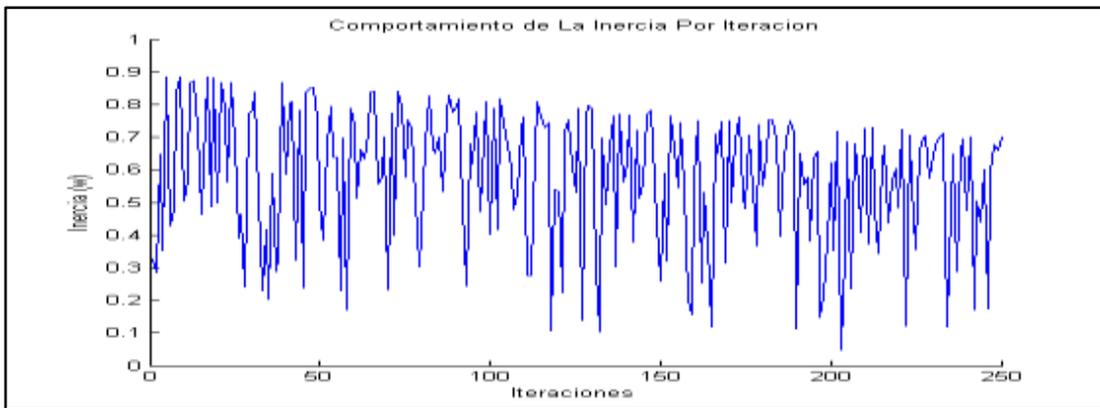
**Gráfica N° 16. Comportamiento de la inercia dinámica *WDI*.**



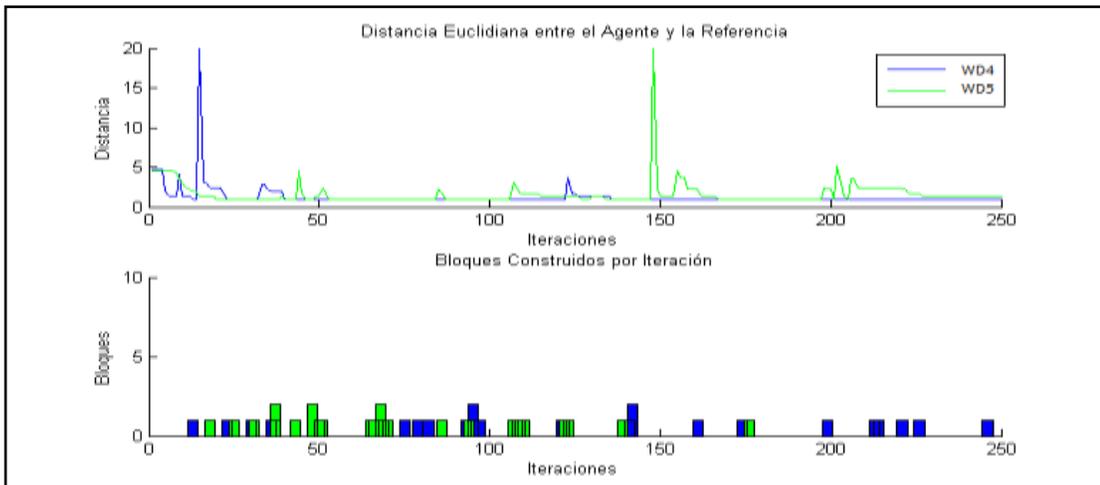
**Gráfica N° 17. Actividad de construcción de los agentes para la estructura 1 utilizando inercia dinámica *WD2* y *WD3*.**



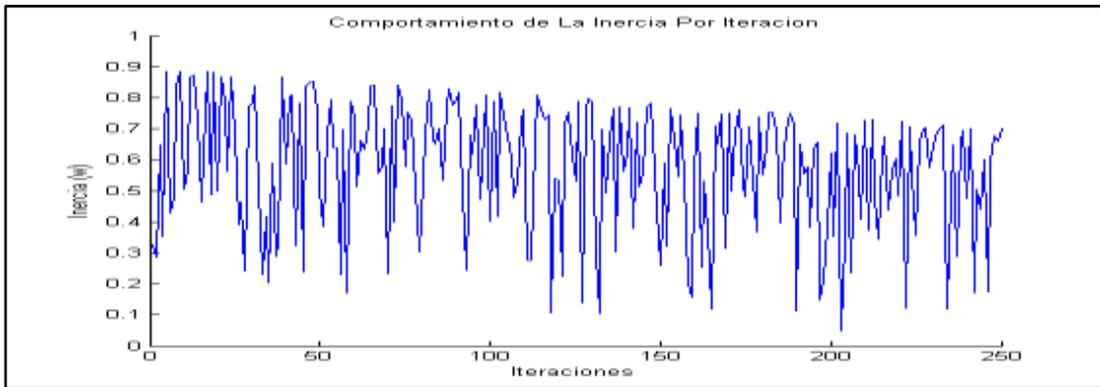
**Gráfica N° 18. Comportamiento de la inercia dinámica WD2.**



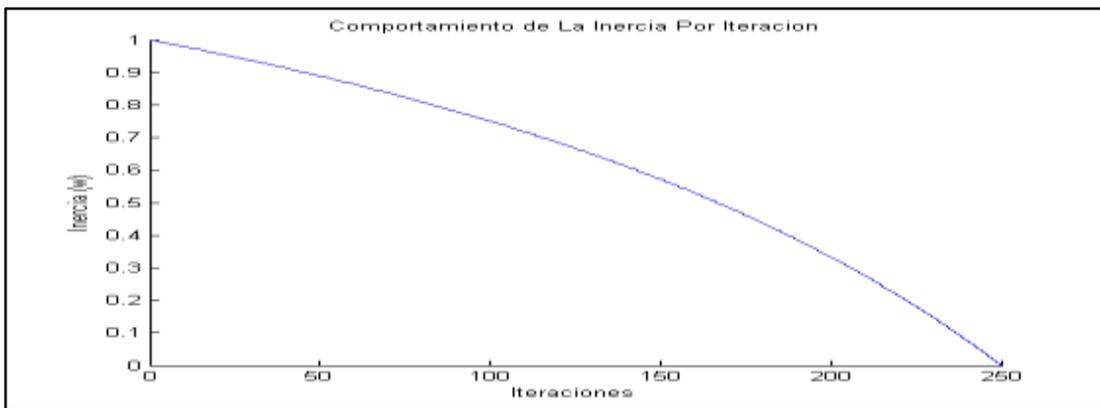
**Gráfica N° 19. Comportamiento de la inercia dinámica WD3.**



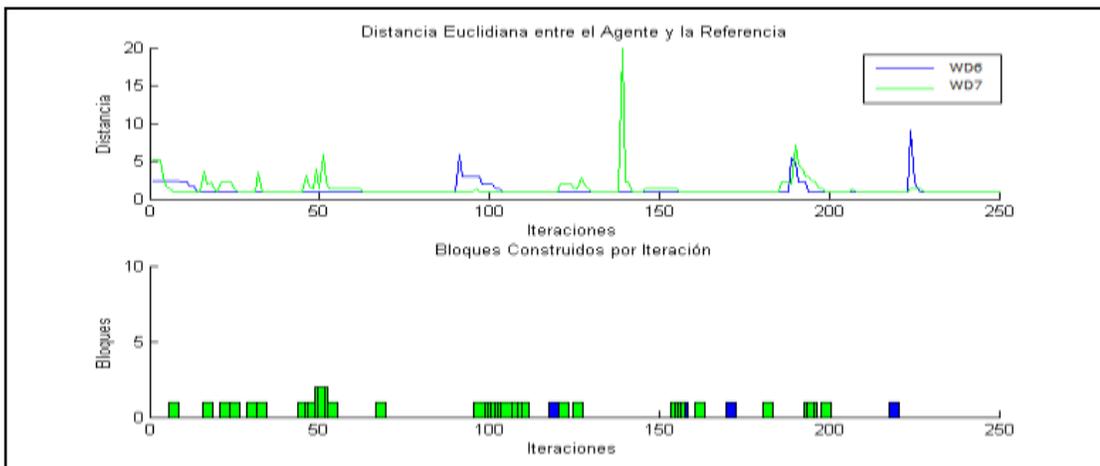
**Gráfica N° 20. Actividad de construcción de los agentes para la estructura 1 utilizando inercia dinámica WD4 y WD5.**



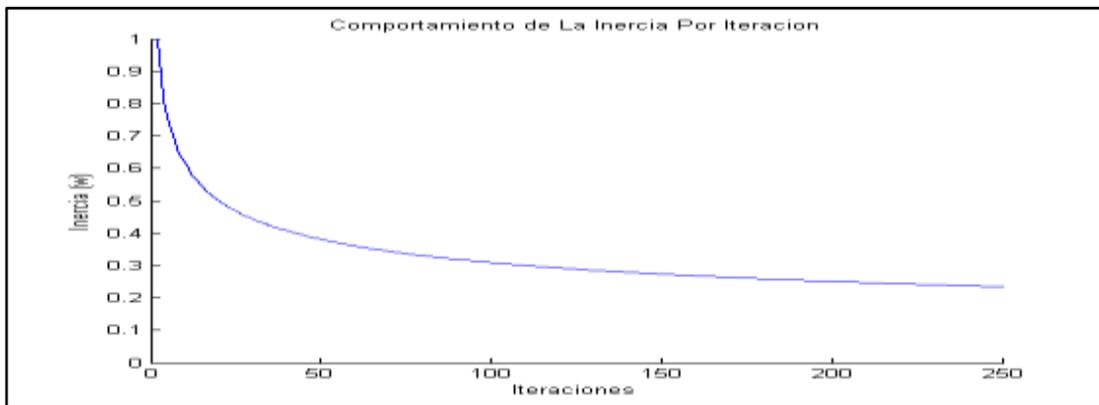
**Gráfica N° 21. Comportamiento de la inercia dinámica WD4.**



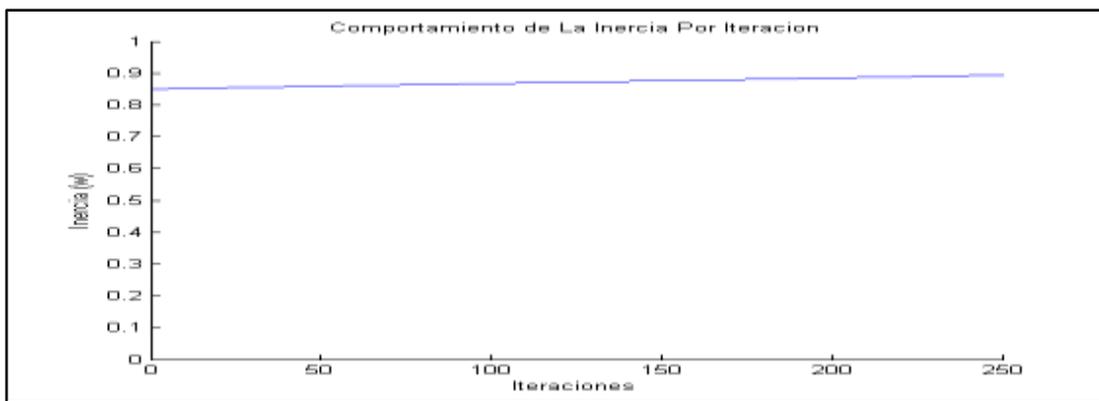
**Gráfica N° 22. Comportamiento de la inercia dinámica WD5.**



**Gráfica N° 23. Actividad de construcción de los agentes para la estructura 1 utilizando inercia dinámica WD6 y WD7.**



**Gráfica N° 24. Comportamiento de la inercia dinámica WD6.**



**Gráfica N° 25. Comportamiento de la inercia dinámica WD7.**

Analizando las gráficas es posible apreciar que utilizando inercia dinámica, el enjambre logra construir la estructura en la mayoría de los casos, con más rapidez que cuando se utiliza la inercia fija, porque se aprovechan las capacidades de exploración y explotación de los agentes, a través de variar los valores de la inercia. Sin embargo durante los experimentos, fue posible observar que los parámetros que permitieron al enjambre construir de manera más eficiente, y visiblemente coordinada mediante la estigmergia cualitativa son: Ecuación de Inercia Dinámica WD7 (ver cuadro 12), con una inercia inicial de 0.9,  $C_1 = 0.3$  y  $C_2 = 0.4$ .

Paso 3: Construir la estructura 1, utilizando Auto-ensamblaje con PSO con los valores obtenidos.

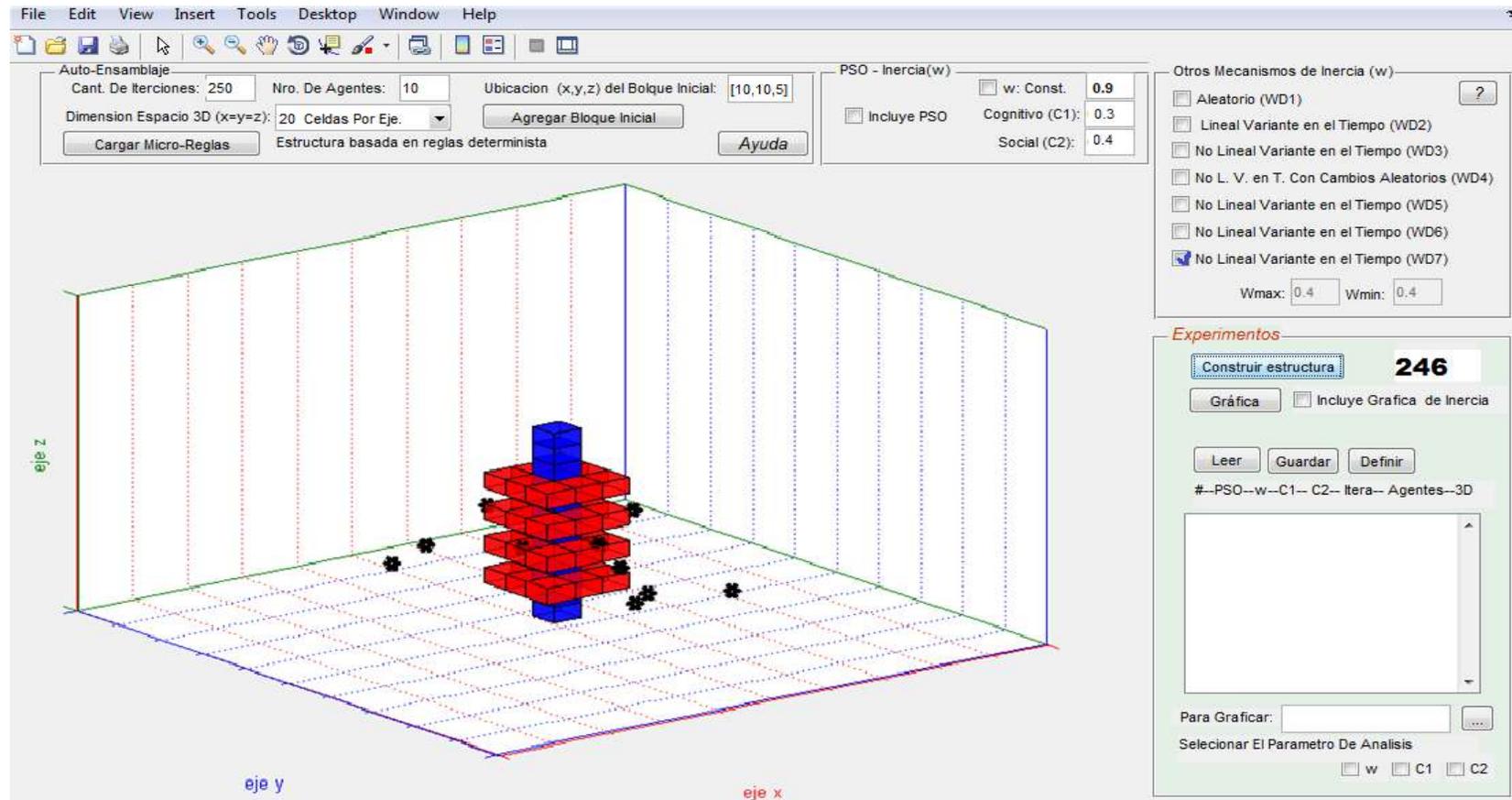


Figura N° 6. Resultados obtenidos con los valores  $W$ = inercia dinámica WD7, Inercia inicial=0.9,  $C_1$ =0.3 y  $C_2$ =0.4 para Auto-ensamblaje con PSO, de la construcción de la estructura 1.

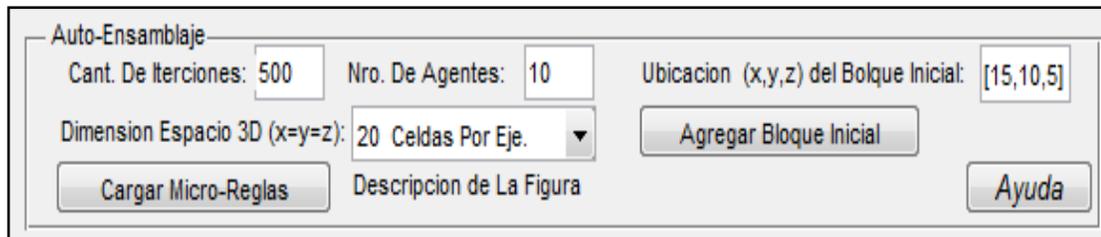
Analizando el primer escenario, en la gráfica 1, se muestra que en 250 iteraciones el Auto-ensamblaje clásico solo construyó 2 ladrillos de la estructura propuesta, mientras que el Auto-ensamblaje con PSO, con las mismas condiciones experimentales y sus parámetros calibrados, construyó la estructura deseada de 32 ladrillos, lo que permite concluir, que el Auto-ensamblaje con PSO, logra superar al Auto-ensamblaje clásico, en cuanto a la capacidad de construcción, es decir; se reduce el tiempo construcción utilizando Auto-ensamblaje con PSO.

### Segundo Escenario – Estructura 2

Para el escenario uno, se presentaron gráficas que permitían observar la distancia entre los agentes y la referencia, además la cantidad de ladrillos construidos por iteración. En el segundo escenario, se considera una estructura más compleja que la presentada anteriormente, es conveniente observar la estructura que se puede construir en cada experimento, para ver la influencia de los parámetros clave del PSO, en la construcción que se realiza en cada experimento.

*Paso 1:* Construir la estructura 2 con el modelo de Auto-ensamblaje clásico.

Al igual que en el escenario uno, el auto-ensamblaje clásico, en diez experimentos no logró construir más de cinco ladrillos en la mayoría de los experimentos, donde se utilizaron los siguientes parámetros:



The image shows a software control panel for 'Auto-Ensamblaje'. It contains several input fields and buttons. The parameters are: 'Cant. De Iteraciones' set to 500, 'Nro. De Agentes' set to 10, and 'Ubicacion (x,y,z) del Bolque Inicial' set to [15,10,5]. The 'Dimension Espacio 3D (x=y=z)' is set to 20 Celdas Por Eje. There are buttons for 'Cargar Micro-Reglas', 'Agregar Bloque Inicial', 'Ayuda', and 'Descripcion de La Figura'.

Auto-Ensamblaje		
Cant. De Iteraciones: 500	Nro. De Agentes: 10	Ubicacion (x,y,z) del Bolque Inicial: [15,10,5]
Dimension Espacio 3D (x=y=z): 20 Celdas Por Eje.	Agregar Bloque Inicial	
Cargar Micro-Reglas	Descripcion de La Figura	Ayuda

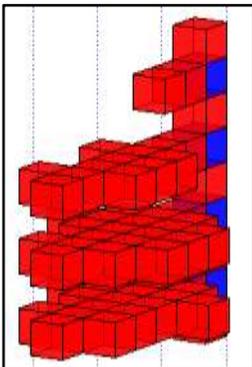
**Cuadro N° 17. Valores de los parámetros para simular el Auto-ensamblaje clásico con la Estructura 2.**

*Paso2:* Seleccionar mediante experimentos los valores para los parámetros clave del PSO.

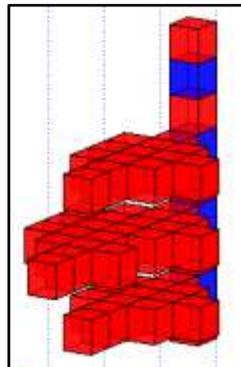
En este caso se tomarán como punto de partida los valores encontrados en el escenario 1, y variando los parámetros como en los experimentos anteriores según el cuadro 18, donde se relacionan las figuras construidas con los parámetros utilizados, además es importante ver que en el parámetro  $w$ , también se utiliza WD1, WD2, WD3, WD4, WD5, WD6 y WD7 para la inercia dinámica, como se presentó en cuadro 12 para tal fin.

Parámetros			Figura	Parámetros			Figura	Parámetros			Figura
w	C <sub>1</sub>	C <sub>2</sub>	N°	W	C <sub>1</sub>	C <sub>2</sub>	N°	W	C <sub>1</sub>	C <sub>2</sub>	N°
0.9	0.3	0.4	7	0.9	0.7	0.4	13	<b>WD3</b>	0.3	0.4	19
0.7	0.3	0.4	8	0.9	0.3	0.2	14	<b>WD4</b>	0.3	0.4	20
0.5	0.3	0.4	9	0.9	0.3	0.6	15	<b>WD5</b>	0.3	0.4	21
0.3	0.3	0.4	10	0.9	0.3	0.8	16	<b>WD6</b>	0.3	0.4	22
0.9	0.1	0.4	11	<b>WD1</b>	0.3	0.4	17	<b>WD7</b>	0.3	0.4	23
0.9	0.5	0.4	12	<b>WD2</b>	0.3	0.4	18				

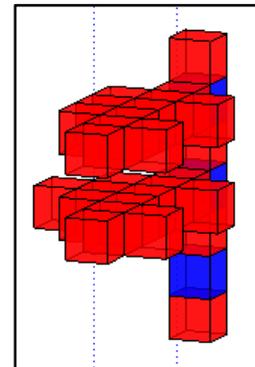
**Cuadro N° 18. Valores de los parámetros clave en diversos experimentos para construir la estructura 2 con PSO.**



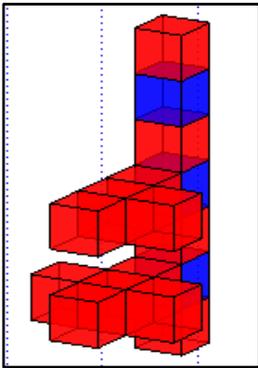
**Figura N°7.**



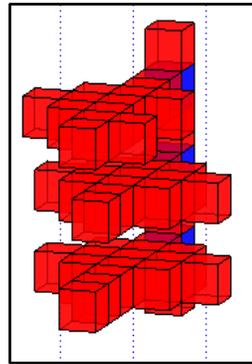
**Figura N°8.**



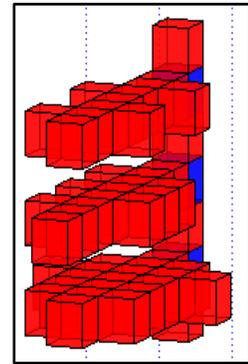
**Figura N°9.**



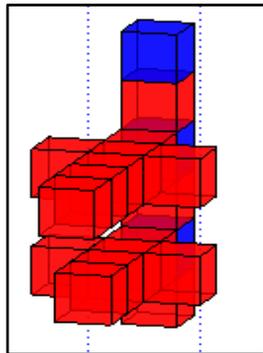
**Figura N°10.**



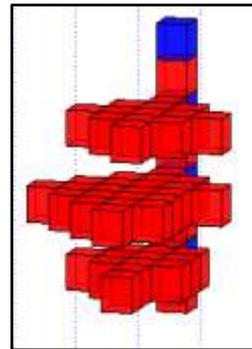
**Figura N°11.**



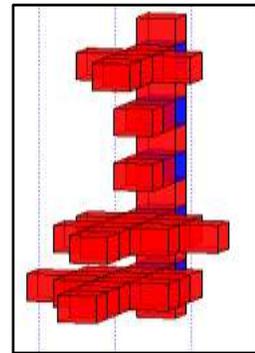
**Figura N°12.**



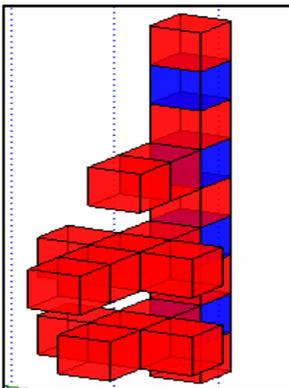
**Figura N°13.**



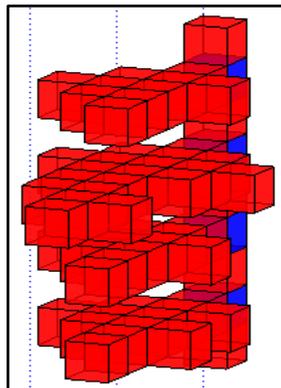
**Figura N°14.**



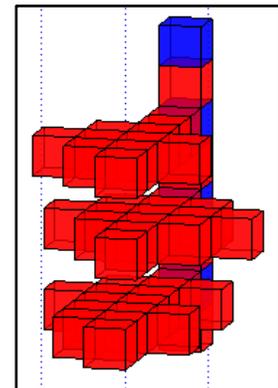
**Figura N°15.**



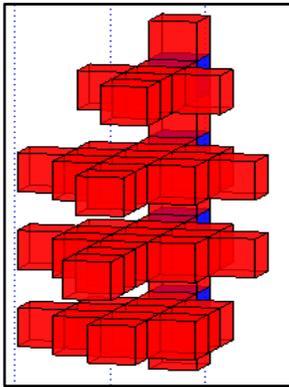
**Figura N°16.**



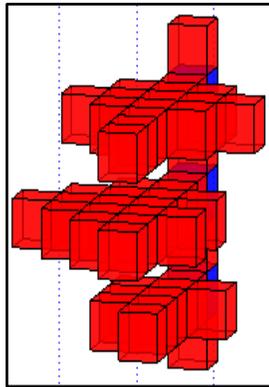
**Figura N°17.**



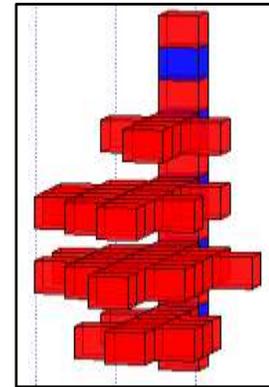
**Figura N°18.**



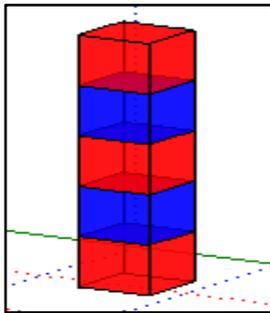
**Figura N°19.**



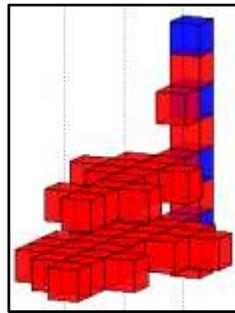
**Figura N°20.**



**Figura N°21.**



**Figura N°22.**



**Figura N°23.**

Analizando los resultados obtenidos se puede apreciar que hay varias selecciones de parámetros que permiten construir la estructura 2, ya que en los experimentos se observó que el componente social y cognitivo encontrados en el escenario 1, aportan un comportamiento similar al construir la estructura 2. Sin embargo para este escenario seleccionaremos los parámetros utilizados en la figura 17, debido a que los agentes además de mantenerse cerca del área de construcción logran construir de manera eficiente (mayor cantidad de bloques construidos), permitiendo observar como emerge la estructura 2 a través de la coordinación mediante estigmergia cualitativa empleada.

Paso 3: Construir la estructura 2, utilizando Auto-ensamblaje con PSO con los valores obtenidos.

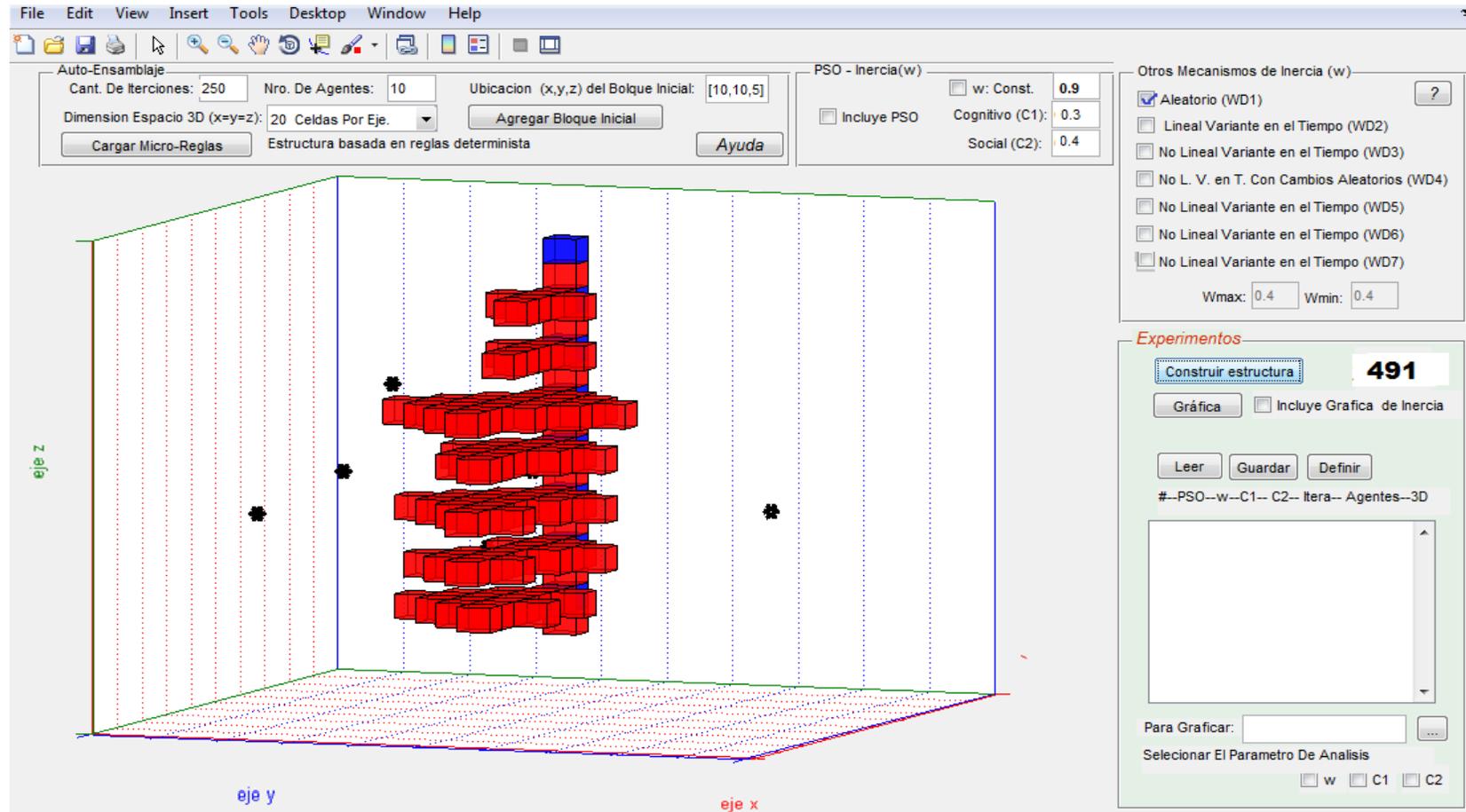


Figura N° 24. Resultados obtenidos con la inercia dinámica WD1,  $C_1=0.3$  y  $C_2=0.4$  para Auto-ensamblaje con PSO, en la construcción de la estructura 2.

**Modelo de Auto-Ensamblaje clásico Vs. El Modelo Auto-Ensamblaje con PSO Propuesto.**

Las observaciones realizadas mediante diferentes simulaciones con diversos valores de los parámetros clave en los dos escenarios propuestos, permiten verificar la eficiencia de los dos algoritmos (*clásico y con PSO*) en la construcción de las estructuras propuestas.

<b>Auto-Ensamblaje Clásico</b>	<b>Auto-Ensamblaje Con PSO</b>
1) En los experimentos realizados para los dos escenarios con 500 iteraciones no llegó a construir más de cinco ladrillos. 2) Las posiciones de los agentes durante sus movimientos para realizar la edificación, se mantuvo alejada del área de construcción. 3) Construir cada una las estructuras propuestas requiere más de 6000 iteraciones.	1) En la mayoría de los experimentos realizados con 500 iteraciones logró construir la estructura propuesta. 2) El movimiento de los agentes, mantenía una cercanía en dirección al área de construcción. 3) Fue posible construir cada una de las estructuras propuestas, con solo el 10% de las iteraciones requeridas por el auto-ensamblaje clásico.

**Cuadro N° 19. Diferencia entre el Modelo de Auto-ensamblaje Clásico y el Propuesto.**

## CAPITULO VI

### CONCLUSIONES Y RECOMENDACIONES

En el presente trabajo se construyeron estructuras en 3D a través de dos técnicas emergentes, utilizando un modelo propuesto de *Auto-Ensamblaje (basado en la construcción de nidos de avispa)*, que incorpora la optimización por enjambre de partículas (*Basado en el comportamiento colectivo de bandada de aves y peces*), con el fin de mejorar el proceso de construcción de estructuras en 3D para reducir el tiempo de convergencia.

Para apreciar visualmente las mejoras del modelo propuesto y lograr establecer comparaciones con el modelo de Auto-ensamblaje clásico, en el proceso de construcción de estructuras en 3D, se desarrolló una herramienta computacional en MATLAB, para simular el proceso de construcción con los dos modelos de Auto-ensamblaje. Con esta herramienta se logra aportar un software para experimentar con Auto-ensamblaje con y sin PSO, y además se superan algunas restricciones encontradas en las pocas herramientas disponibles para tal fin como por ejemplo:

- 1) Cargar el banco de consulta o micro-reglas desde un archivo (*.AVP*), que puede ser definido desde un editor de texto, simplemente utilizando el formato presentado en cuadro 11.
- 2) Configurar interactivamente los parámetros clave para el Auto-ensamblaje con y sin PSO, además se incluyeron mecanismos dinámicos para la inercia, que permiten aprovechar el efecto de la exploración y la explotación realizada por el enjambre, al variar este parámetro dinámicamente en cada iteración.
- 3) Graficar los resultados obtenidos en cada uno de los experimentos, para visualizar la distancia euclidiana entre los agentes y la referencia (*posición del bloque inicial o el último bloque construido*), la cantidad de bloques

construidos por iteración. Además, cuando se utiliza la inercia dinámica opcionalmente, se pueden graficar los valores de la inercia en cada iteración para medir su impacto en el proceso de construcción.

- 4) Gestionar los experimentos, es decir, proporciona la posibilidad de almacenamiento de los resultados obtenidos, para luego leer y realizar comparación gráficamente los resultados de hasta cinco experimentos simultáneamente.
- 5) Exportar la estructura construida hacia un formato de imagen como .jpg, .bmp y .png.

Gracias a la incorporación de todas estas características, fue posible apreciar el proceso de construcción en 3D utilizando la herramienta computacional desarrollada, donde utilizando el modelo propuesto se lograron construir las dos estructuras propuestas en menos de quinientas iteraciones, mientras que para el modelo de Auto-Ensamblaje clásico se requieren más de seis mil iteraciones, lo que nos permite suponer que si es posible mejorar el proceso de construcción cooperativa de Arquitecturas 3D mediante la optimización por enjambre de partículas.

La definición de la función objetivo requerida por la PSO fue complicada, ya que inicialmente se intentó definir basada en las micro-reglas, que fue descartado porque estaba en contra del fundamento principal del algoritmo de Auto-ensamblaje clásico, ya que las avispas no tienen una representación global de la estructura sino que ejecutan un programa de construcción en función de los cambios percibidos en el ambiente, donde realizan la construcción. Sin embargo, observando la construcción de las estructuras propuestas con el Auto-ensamblaje clásico, se fueron definiendo heurísticas que permitieron al enjambre navegar cerca del área de construcción, lo que permitió seleccionar la distancia euclidiana entre los agentes constructores y una referencia, con ciertas restricciones como función objetivo.

El utilizar dos estructuras, una simple (*escenario 1*), y otra más compleja (*escenario 2*), para implementar Auto-ensamblaje con PSO, permitió observar en los experimentos, que la eficiencia del algoritmo propuesto, depende de la calibración de los parámetros clave: inercia, componentes cognitivo y social, y no de la complejidad o tipo de estructura, para mejorar el tiempo de construcción de los agentes.

Finalmente, podría ser importante considerar para trabajos futuros:

- 1) Utilizar el movimiento browniano para la navegación del enjambre de agentes, por ser un modelo matemático que describe el movimiento aleatorio, que se observa en algunas partículas microscópicas en medio de un flujo, lo que permite especular que al ser un movimiento natural que se observa en estructura complejas como la materia, se puede suponer que puede brindar una buena navegación a los agentes constructores, o que se pueda utilizar como un mecanismo para definir el banco de micro-reglas de una determinada estructura o patrón.
- 2) Considerar la implementación del Auto-ensamblaje para la construcción de nano estructuras, porque los sistemas a nano escala, virtualmente se pueden construir con Auto-ensamblaje, solo requiere, primero, ajustar la forma geométrica de la celda del grid, para que acepte las formas geométricas utilizadas para representar, los diferentes elementos geométricos empleados en nano estructuras, y segundo, definir mediante las micro-reglas, qué forma geométrica podría construirse según las condiciones de la vecindad, y podemos especular que las micro-reglas, se pueden definir a partir de las leyes que rigen el comportamiento de las partículas, como las encontradas en la mecánica cuántica.
- 3) Llevar el Auto-ensamblaje a sus máximas exigencias, demanda un alto requerimiento computacional, que se incrementa al aumentar el tamaño del grid, por esta razón, se considera mejorar el requerimiento computacional, utilizando una arquitectura como CUDA [10].

## REFERENCIAS BIBLIOGRAFICAS

- [1] Bonabeau E., Dorigo M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems, Capitulo 6: Nest Building and Self-Assembling*, pp. 205-252. Oxford University Press, USA.
- [2] Theraulaz G, Bonabeau E. “A Brief History of Stigmergy”. *Massachusetts Institute of Technology: Artificial Life* Vol. 5, pp. 97–116, USA, 1999.
- [3] Camelia Chira, Camelia M. Pinteá, D. Dumitrescu. *Stigmergy and Sensitivity in Multi-Agent Models*, Computational Intelligence Reports No. 10, ISSN 1841-995X, Babes-Bolyai University, Rumanía, 2008.
- [4] Theraulaz G. Y Bonabeau E. *Modelling the Collective Building of Architectures in Social Insects with Lattice Swarms*. Academic Press Limited pp. 381-400, USA, 1995.
- [5] Cervantes R. Alejandro. *Clasificación Mediante Enjambre de Prototipos*. Tesis Doctoral, Universidad Carlos III De Madrid, 175 p., España, 2009.
- [6] James G. Adam. *Designing Emergence: Automatic Extraction of Stigmergic Algorithms From Lattice Structures*. Tesis Doctoral, University of Essex, 294 p. Reino Unido, 2005.
- [7] Marcin L. Pilat, *Wasp-Inspired Construction Algorithms*. Department of Computer Science, University of Calgary, 14 p., Canada, 2004.
- [8] Zhongkui Wang, B.S. *Cooperative Construction*. Trabajo de Grado, The Ohio State University, 53 p., USA, 2009.
- [9] Quintero M. *Reconocimiento de Rostros con visión por computadora y optimización basada en cúmulos de partículas*. Universidad Autónoma de Ciudad Juárez, 11 p., México, 2010.
- [10] Castro Iliana Liera. *Paralelización de algoritmos de optimización Basado en Poblaciones por medio de GPGPU*. Trabajo de Grado, Instituto Tecnológico de la Paz, 119 p., México, 2011.

- [11] Perozo Niriaska. *Modelado Multiagente para sistemas Emergentes y Auto-organizados*. Tesis Doctoral, Universidad de los Andes & Universidad Paul Sabatier, 145 p., Venezuela, 2011.
- [12] Cruz C. Carlos A. *Estrategias coordinadas paralelas basadas en soft-computing para la solución de problemas de optimización*. Tesis Doctoral, Universidad de Granada, 159 p. España, 2005
- [13] Juaquin Q. Lima M., Benjamín Barán C. *Optimización de enjambre de partículas aplicada al Problema del Cajero Viajante Bi-Objetivo*. Revista iberoamericana de inteligencia artificial, Vol. 10, pp. 67-76, Paraguay, 2006.
- [14] González V., Villagra M., Barán B. *Optimización por Enjambre de Partículas para Satisfacción de Fórmulas Booleanas*. Universidad Católica de Asunción, Paraguay, 2006.
- [15] Lin I-ling. *Particle swarm Optimization for solving constraint satisfaction problems*. Trabajo de grado, Simon Fraser University, 202 p., Canadá 2002.
- [16] Fernández J. *Optimización del Diseño de la Electrificación Ferroviaria*. Trabajo de grado, Universidad Pontificia Comillas, 92p., España 2010.
- [17] Ahmad N, y Otros. *A novel particle swarm optimization algorithm with adaptive inertia weight*. Applied Soft Computing, Amirkabir University of technology, Vol. 11, pp. 3658-3670, Iran, 2011.
- [18] Lazinica A. *Particle Swarm Optimization*. Publicado por In-tech, www.intechweb.org, Cap.28, pp.451- 460, Austria, 2009.
- [19] Guerra N., Crawford B. *Optimización de Funciones a través de Optimización por enjambre de partículas y algoritmo genéticos*. Pontificia Universidad Católica de Valparaíso, 10p., Chile, 2006.