



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA**  
**INGENIERÍA DE SISTEMAS-INGENIERÍA INDUSTRIAL**

“ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN  
EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO”

TESIS

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA

PRESENTA:

ING. VICENTE FLORES OLVERA

TUTOR PRINCIPAL  
DRA. MAYRA ELIZONDO CORTÉS  
FACULTAD DE INGENIERÍA

MÉXICO, D.F., AGOSTO 2014



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**JURADO ASIGNADO:**

**PRESIDENTE:** DRA. FLORES DE LA MOTA IDALIA

**SECRETARIO:** M. I. RIVERA COLMENERO JOSÉ ANTONIO

**VOCAL:** DRA. ELIZONDO CORTÉS MAYRA

**1ER. SUPLENTE:** DRA. MONROY LEÓN COZUMEL ALLANEC

**2DO. SUPLENTE:** M. I. CHÁVEZ RODRÍGUEZ NORMA ELVA

Lugar o lugares donde se realizó la tesis:

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

CIUDAD UNIVERSITARIA, MÉXICO, D.F.

**TUTOR DE TESIS:**

**DRA. ELIZONDO CORTÉS MAYRA**

---

FIRMA

## **AGRADECIMIENTOS:**

A Dios doy gracias por concederme tantas cosas y que hoy me permiten cerrar un ciclo más en mi vida.

A mi alma mater, la facultad de ingeniería que me ha cobijado en su seno de la sabiduría y del conocimiento que me ha permitido ser más crítico, científico y mejor persona.

A mí estimada Dra. Mayra Elizondo Cortés le dedico este esfuerzo que no pudo ser concretado sin su incondicional apoyo y confianza; le estoy agradecido por su paciencia e infinita tolerancia para culminar este trabajo.

Para profesores, compañeros y amigos, mis infinitas gracias porque sin su apoyo y presencia, no hubiera alcanzado este sueño al que todos aspiramos en algún momento de nuestras vidas.

Al Jurado Asignado mi agradecimiento por su apoyo, paciencia y tolerancia.

A mi familia y amigos gracias por el apoyo incondicional y porque siempre se mantuvieron pacientes a mi lado a lo largo de todo este tiempo para alcanzar este sueño.

A mis padres les doy gracias por su cariño y conocimiento que me transmitieron, para ser una mejor persona durante todos estos años.

Muy especial agradecimiento a mis dos motores que son el motivo que impulsa mi vida.

Un agradecimiento especial para el CONACYT por el apoyo que recibí en mi incursión en el posgrado.



# Índice General.

Índice General.....	i
Índice de figuras:.....	iv
Índice de tablas:.....	vi
Resumen.....	vii
Abstract.....	viii
Introducción.....	ix
Objetivo.....	x
Justificación.....	x
Capítulo 1: Descripción logística del pedido.....	1
1.1 Descripción funcional de un sistema logístico de distribución.....	1
1.2 Cadena de suministro.....	6
1.3 Modelos para mejorar el flujo de bienes y servicios.....	9
1.4 El procesamiento de un pedido.....	10
1.4.1 Elementos típicos del procesamiento del pedido.....	11
1.5 El proceso logístico asociado al pedido.....	13
1.5.1 Recepción del pedido.....	13
1.5.2 Autorización del pedido.....	13
1.5.3 Verificación de existencia.....	14
1.5.4 Apartado de los artículos.....	14
1.5.5 Notificación de artículos faltantes.....	14

1.5.6. Almacén. ....	15
Capítulo 2: Marco de referencia. ....	19
2.1 Estado del arte.....	19
2.2 Marco teórico. ....	24
2.2.1 El algoritmo de optimización por colonia de hormigas (ACO). ....	26
2.2.3 Construcción del algoritmo optimización de colonia de hormigas (ACO) .....	30
2.2.4 Características de la hormiga artificial.....	32
2.2.5 Modo de funcionamiento y estructura genérica de un algoritmo ACO. ....	33
2.3 Estrategia de investigación. ....	37
2.3.1 Ant System (AS).....	37
Capítulo 3 Diseño e implementación del algoritmo Ant System aplicado. ....	45
3.1 Implementación del <i>scheduling</i> adaptado a órdenes de entrega. ....	45
3.2 Metodología a seguir para resolver un problema mediante ACO (ant system). .....	47
3.3 Pseudocódigo ANT SYSTEM adaptado a la asignación de órdenes. ....	49
3.3.1 Desarrollo del aplicativo computacional. ....	49
3.4. Diseño experimental. ....	54
Capítulo 4. Ejemplo de aplicación. ....	56
4.1 Estudio.....	58
4.2 Resultados.....	61
Conclusiones:.....	71

Bibliografía .....	74
Anexo A.....	76
Anexo B.....	83



## Índice de figuras:

Figura 1 Proceso logístico de distribución. ....	2
Figura 2 Flujo de bienes y servicios a los clientes.....	4
Figura 3 Proceso total del pedido.....	12
Figura 4 Levantamiento del pedido. ....	13
Figura 5 Autorización del pedido.....	13
Figura 6 Verificación de la existencia. ....	14
Figura 7 Apartado de la mercancía solicitada. ....	14
Figura 8 Notificación de faltantes. ....	15
Figura 9 Grafo de construcción. ....	21
Figura 10 Rutas y Los rastros de feromona dejadas por las hormigas. ....	27
Figura 11 Comportamiento de las Hormigas ante un Obstáculo y su evolución. ..	28
Figura 12 Rastro de feromona (variable en el tiempo). ....	29
Figura 13 Diagrama de flujo del algoritmo Ant System. ....	40
Figura 14 Grafo ejemplo utilizado para la asignación de recursos sobre una colonia de hormigas aplicado a un proceso logístico con feromonas sobre los rastros.....	43
Figura 15 Representación general de un proceso logístico de distribución. ....	45
Figura 16 Diagrama de flujo del algoritmo Ants System Adaptado. ....	51
Figura 17 Histograma en el modo de preasignamiento.....	57
Figura 18 Parámetros iniciales del algoritmo.....	59
Figura 19 Parámetros iniciales del algoritmo.....	60

Figura 20	Tiempos de resurtimiento para 7 componentes. ....	61
Figura 21	Stock de salida. ....	63
Figura 22	Datos parciales generados para corrida 1.....	63
Figura 23	Stock de Salida para corrida 2. ....	64
Figura 24	Datos parciales generados corrida 3.....	66
Figura 25	Histograma de resultados para el algoritmo Ant System Aplicado. ....	67
Figura 26	Datos parciales generados corrida 3.....	67
Figura 27	Datos parciales generados corrida 3.....	68
Figura 28	Datos cruzados con el método de preasignamiento y el algoritmo Ant System. ....	69
Figura 29	Grafica ventana orden deseada para Ant System.....	70
Figura 30	Grafica ventana orden deseada para preasignamiento.....	71

## Índice de tablas:

Tabla 1 Algoritmos ACO disponibles para el problema scheduling. ....	20
Tabla 2 Función de demanda.....	57
Tabla 3 Descripción de los parámetros iniciales. ....	62
Tabla 4 Descripción de las condiciones iniciales utilizadas.....	62
Tabla 5 Descripción de los parámetros iniciales. ....	64
Tabla 6 Descripción de las condiciones iniciales utilizadas.....	64
Tabla 7 Descripción de los parámetros iniciales. ....	65
Tabla 8 Descripción de las condiciones iniciales utilizadas.....	65
Tabla 9 Cuadro comparativo de cumplimientos. ....	69

## Resumen

Los problemas de asignación de recursos son muy importantes ya que en la vida diaria tienen una alta aplicación en diversos campos de la industria, en donde existe un conjunto de tareas(o recursos) que deben ser organizadas de tal forma que el orden con que estas se desarrollen permita ahorrar tiempo, dinero y esfuerzo.

Desde hace varios años la complejidad de estos problemas ha llamado la atención de ingenieros y científicos, interesados en encontrar métodos que ofrezcan en un tiempo razonable buenas soluciones. Tales métodos proponen formas de solucionar el problema de manera indirecta desde diversos enfoques, estos métodos proponen dar una garantía de resolver en su totalidad cualquier instancia del problema antes mencionado. Entre estos métodos podemos encontrar las llamadas metaheurísticas, que tienen su inspiración en diversas áreas como la química, la física, la biología, entre otras.

Del área biológica podemos encontrar diversas formas de vida que atrapan nuestra atención debido a la simbiosis que muestran para el desarrollo de sus colectividades y en la cual, el hombre ha observado, estudiado y descubierto los “comportamientos inteligentes” de diversas especies para aplicarlos en la resolución de problemas de la vida diaria.

En esta tesis se presenta una variante del algoritmo *Ant System (AS)* basado en el comportamiento de forrajeo<sup>1</sup> de las hormigas. El algoritmo propuesto es desarrollado para resolver el problema de asignación de recursos.

El manejo de las reglas y la aplicación de unos nuevos conocimientos biológicos sobre el comportamiento para localizar los caminos más cortos entre el nido de la colonia de hormigas y la fuente de alimentos, son características que distinguen este trabajo.

---

<sup>1</sup> Forrajeo: búsqueda de alimento basado en comportamiento.

En esta tesis se reporta una variante realizada que dio lugar a la generación de un nuevo algoritmo para la asignación de recursos.

## **Abstract**

The resource allocation issues are very important because in everyday life have a high application in various industrial fields, where there is a set of tasks (or resources) to be organized so that the order in which these are developed to save time, money and effort.

For several years the complexity of these problems has attracted the attention of engineers and scientists, interested in finding methods in a reasonable time offer good solutions. Such methods propose ways to solve the problem indirectly from different approaches, these methods give a guarantee propose fully resolve any instance of the problem mentioned above. These methods can find metaheuristics calls, which are inspired by diverse areas as chemistry, physics, biology, among others.

Biological area can find various forms of life that catch our attention due to the symbiosis that show the development of their communities and in which man has observed, studied and discovered the "intelligent behaviors" of various species for application in the solving everyday problems.

This thesis presents a variant of the algorithm Ant System (AS) based on foraging behavior (foraging) ants. The proposed algorithm is developed to solve the problem of resource allocation.

The management of the rules and the application of new biological knowledge about the behavior to find the shortest paths between the nest of the ant colony and source of food, are characteristics that distinguish this work.

This thesis reports the variant on resulting in the generation of a new algorithm.

## Introducción

En los últimos años ha habido muchos cambios: la apertura económica, los procesos de privatización y la desregulación de los mercados permitieron la conformación de un escenario altamente competitivo que llevó al replanteo de muchas prácticas productivas y comerciales.

La aparición de empresas globales e internacionales que operan con un alto nivel de eficiencia y el ingreso de productos a menores costos que los producidos localmente, ha exigido cambios estructurales en muchas compañías manufactureras y de servicios. Es así que el análisis de valor, el enfoque en el cliente, la reingeniería, la introducción de cambios tecnológicos en los procesos productivos, la implementación de programas de aseguramiento de la calidad, la incorporación de nuevas tecnologías de información, las alianzas estratégicas, la capacitación de los recursos humanos y el uso de innovadoras herramientas gerenciales, se han convertido en prácticas necesarias para aumentar la competitividad.

La nueva realidad competitiva presenta un campo de batalla en donde la flexibilidad, la velocidad de llegada al mercado y la productividad serán las variables claves que determinarán la permanencia de las empresas en los mercados. Y es aquí donde la logística juega un papel crucial, a partir del manejo eficiente del flujo de bienes y servicios hacia el consumidor final.

## Objetivo

Diseñar e implementar un algoritmo de optimización por colonia de hormigas para la minimización de tiempos de respuesta en el cumplimiento de órdenes de un proceso logístico.

## Justificación

Dentro del proceso logístico, específicamente en cuanto a la distribución de artículos o insumos, es importante la reducción de inventario dentro de los almacenes de distribución o bien cómo se desarrolla en esta tesis, en las órdenes de trabajo, con el objeto de minimizar al máximo costos, por otro lado aplicar un método no tan tradicional como es la asignación dinámica de recursos, en este caso utilizando el algoritmo *Ant System*, como una manera de proyectar demanda sobre los distribuidores y lograr una máxima eficacia en cuanto a los tiempos de respuesta que se esperan obtener de ellos.

## Capítulo 1: Descripción logística del pedido.

### 1.1 Descripción funcional de un sistema logístico de distribución.

En un concepto claro, la logística es un término que frecuentemente se asocia con la distribución y transporte de productos terminados; sin embargo, ésta es una apreciación parcial de la misma, ya que la logística se relaciona con la administración del flujo de bienes y servicios, desde la adquisición de las materias primas e insumos en su punto de origen, hasta la entrega del producto terminado en el punto de consumo.<sup>2</sup>

De esta forma, todas aquellas actividades que involucran el movimiento de materias primas, materiales y otros insumos forman parte de los procesos logísticos, al igual que todas aquellas tareas que ofrecen un soporte adecuado para la transformación de dichos elementos en productos terminados: las compras, el almacenamiento, la administración de los inventarios, el mantenimiento de las instalaciones y maquinarias, la seguridad y los servicios de planta (suministros de agua, gas, electricidad, combustibles, aire comprimido, vapor, etc.). Por dicha razón, la logística no debe verse como una función aislada, sino como un proceso global de generación de valor para el cliente, esto es, un proceso integrado de tareas que ofrezca una mayor velocidad de respuesta al mercado, con costos mínimos. (Figura 1).

---

<sup>2</sup> Ballou, administración de la cadena de suministro, Pearson, 2005.



## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

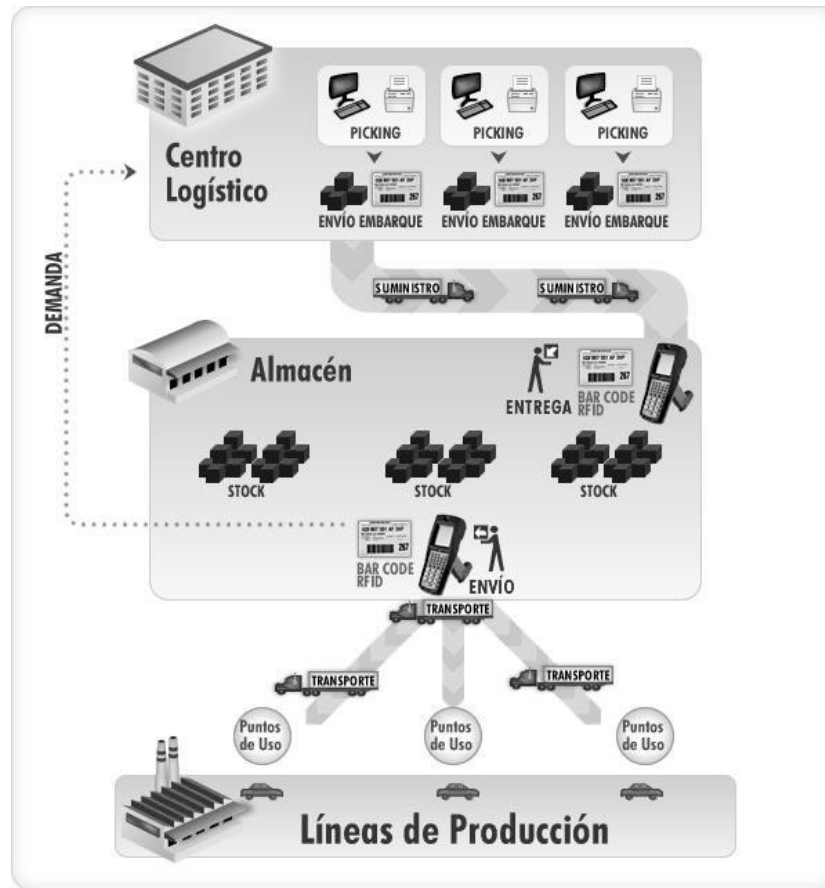


Figura 1 Proceso logístico de distribución. <sup>3</sup>

Como todo sistema, su análisis y la comprensión del mismo pueden obtenerse a partir del estudio de sus partes componentes. De esta forma, podemos abordar el sistema logístico considerando los siguientes subsistemas:

- **Logística de abastecimiento**, que agrupa las funciones de compras, recepción, almacenamiento y administración de inventarios, e incluye actividades relacionadas con la búsqueda, selección, registro y seguimiento de los proveedores.
- **Logística de planta**, que abarca las actividades de mantenimiento y los servicios de planta (suministros de agua, luz, combustibles, etc.), así como también la seguridad industrial y el cuidado del medio ambiente.

<sup>3</sup> Crespón Castro & Auxiliadora, Administración de la Cadena de Suministro. 2003

- **Logística de distribución**, que comprende las actividades de expedición y distribución de los productos terminados a los distintos mercados, constituyendo un nexo entre las funciones de producción y de comercialización.

Los subsistemas de Abastecimiento y de Servicios de Planta se agrupan bajo la denominación de *Logística de Producción*, ya que ambos se relacionan íntimamente con las tareas propias de la fabricación de bienes y/o prestación de servicios.

En el proceso de abastecimiento-producción-distribución, la empresa productora del bien final se convierte en cliente de las empresas proveedoras y éstas, a su vez, son clientes de otras compañías que los abastecen. Paralelamente, la empresa fabricante del producto de consumo final actúa como proveedora de las compañías mayoristas y/o comercios minoristas y que es motivo de análisis en esta tesis.

Las particularidades de cada uno de los subsistemas y sus problemáticas asociadas se presentan bajo la referencia de un concepto clave en la temática logística: la administración de la cadena de suministro.<sup>4</sup>

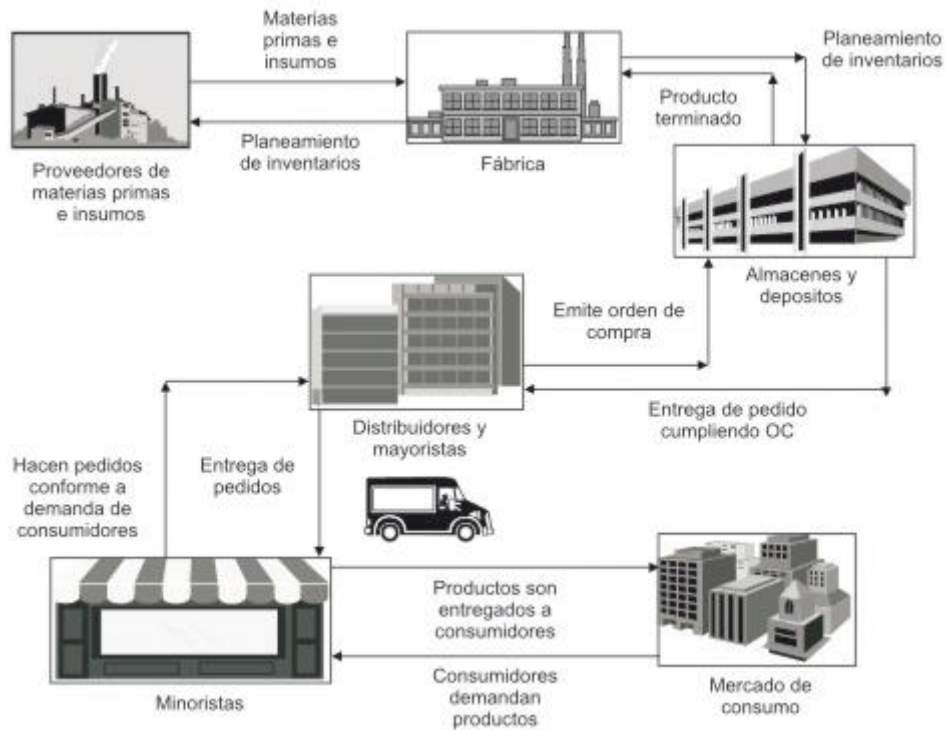
Como es sabido, la producción es un subsistema dinámico de la organización, que transforma los recursos a medida que fluyen a través de las distintas etapas del proceso:

- En una compañía manufacturera, las materias primas, materiales e insumos son adquiridos a proveedores, almacenándose hasta el momento de su utilización en el proceso productivo. Los materiales fluyen a lo largo de este proceso, hasta ser transformados en productos finales que serán almacenados en un depósito hasta su posterior distribución en el mercado. (figura 2).

---

<sup>4</sup> Juan Pablo Antun Callaba, logística internacional" instituto de ingeniería UNAM 2004.

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO



**Figura 2 Flujo de bienes y servicios a los clientes<sup>5</sup>**

- En una empresa de servicios, pueden existir diferentes tipos de flujos: de materiales, de documentos y/o de personas. Los servicios de reparación, en general, son ejemplos en donde los flujos de materiales son los que prevalecen. Un despacho jurídico, un estudio contable o una oficina de rentas realizan actividades principalmente relacionadas con documentos, por lo que el flujo de documentación es el preponderante en estos casos.

El manejo adecuado de los flujos de bienes y servicios es de importancia crítica, no solamente para lograr una reducción en los costos asociados a los procesos de abastecimiento, producción y distribución, sino también para ofrecer una rápida respuesta a los requerimientos de los clientes. Veamos algunos ejemplos:

<sup>5</sup> Gómez Acosta, M. I. & Acevedo Suárez, J. A. (2001). Logística moderna y la competitividad empresarial. Centro de Estudio Tecnología de Avanzada (CETA) y Laboratorio de Logística y Gestión de la Producción (LOGESPRO). Ciudad de la Habana.

- Cuando un material está mal o no bien especificado, se pueden abrir órdenes de compra que resultarán en abastecimientos inadecuados para su utilización en los procesos de fabricación, trayendo como consecuencia un mayor retraso en la producción y, por consiguiente, el incumplimiento en las fechas de entrega prometidas.
- El almacenamiento es otra de las actividades logísticas clave que pueden afectar el rendimiento de los procesos y la atención a los clientes: si no se cumplen las condiciones de seguridad y mantenimiento necesarias para resguardar el inventario, pueden producirse deterioros importantes en la calidad de las materias primas y los materiales, lo que conducirá a mayores costos por reproceso o deshechos. De la misma forma, condiciones inadecuadas en el almacenamiento de los inventarios pueden conducir a mayores costos por pérdida de material (roturas en el caso de elementos frágiles, mermas en el caso de sustancias líquidas o gaseosas, etc.). Ambas situaciones incidirán negativamente en el nivel de servicio al cliente.
- Un diseño descuidado en el flujo de los procesos y de las capacidades de los centros de trabajo, redundará en mayores costos por ineficiencias (mayores distancias a recorrer, trayectorias inadecuadas, mayores tiempos de procesos, cuellos de botella, capacidad ociosa y entregas no cumplimentadas a tiempo).
- La utilización de transportes inadecuados para el traslado de los materiales en planta puede traducirse en mayores costos por roturas y/o afectar el tiempo de entrega total del proceso (*lead time*).
- Los pedidos de los clientes pueden ser distribuidos rápidamente si se poseen grandes cantidades de *stock* de productos terminados, pero esto significa mantener altos costos de inmovilización de capital, con sus riesgos asociados (pérdidas en concepto de roturas, obsolescencia y robos). Por ello, es

necesario diseñar un proceso logístico que ofrezca respuestas rápidas sin incurrir en altos costos.

- El control del área del taller es otro de los problemas típicos que pueden presentarse en las empresas. En efecto, la inexactitud de los datos o su falta de oportunidad llevan a tomar decisiones erróneas de producción, con variadas consecuencias: agotamiento de existencias o inventarios excesivos, fallas en las fechas de entrega de los pedidos, costeos incorrectos.

Las tareas de almacenamiento y los traslados innecesarios de materias primas, materiales, productos en proceso y productos finales, son actividades que generan un gran porcentaje de los costos y, sin embargo, no agregan valor para el cliente. Es tarea de la logística, eliminar todas aquellas actividades que comprometen costos sin agregar valor, con el fin de aumentar la eficiencia del sistema y ofrecer una rápida velocidad de respuesta a los requerimientos de los clientes.

## 1.2 Cadena de suministro.

El objetivo de la cadena de suministro es la transformación eficiente de insumos en productos terminados, para ser consumidos en el extremo final del proceso. A partir de la coordinación del flujo de bienes y servicios *entre todas las entidades participantes*, y como resultado de una *estrecha colaboración* entre los mismos, se produce una agilización del proceso que da como resultado un aumento de valor para el cliente: mayor flexibilidad, precios reducidos y menores tiempos de entrega; en resumen, una mejor calidad de servicio.

La administración de la *cadena de suministro* debe tender a crear lazos más fuertes entre las empresas para potenciar el valor generado a través de todo el proceso.

Seguidamente, y a los fines de proporcionar un mayor detalle acerca del concepto y funcionamiento de la *cadena de suministro*, se hará referencia a la importancia de considerar a los proveedores, productores, operadores logísticos, mayoristas, minoristas y clientes dentro de un mismo proceso.

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

---

En varias oportunidades se ha expresado que no es posible pensar en una adecuada gestión logística si no se tienen en cuenta los requerimientos de los consumidores.

Muchas veces las empresas apuntan de manera inadecuada a aspectos que no son suficientemente valorados por el cliente; por eso, es de suma importancia conocer el mercado al que cada compañía se dirige, y concentrar los esfuerzos de diferenciación en aquello que realmente ofrezca un valor superior desde su punto de vista. Desde esta nueva concepción, el cliente pasa a ser un socio en el proceso de prestación de un servicio o en la generación de un producto. Un socio que aporta distintas experiencias, actitudes, emociones; un socio que tiene diferentes necesidades y que puede sugerir ideas sobre cómo satisfacerlas adecuadamente.

Por otra parte, todo el proceso de transformación, desde las materias primas hasta la distribución del producto final, culmina cuando el consumidor dispone del bien o hace uso del servicio. De esta manera, el cliente juega dos importantes papeles: es tanto el que inicia como el que finaliza este proceso, por lo que puede ser considerado como el eslabón que permite dar forma a la cadena.

El consumidor es el que determina qué empresas satisfacen adecuadamente sus necesidades; de aquí que el enfoque en el cliente sea una condición indispensable para permanecer y tener éxito en las exigentes condiciones de los mercados actuales.

Los proveedores condicionan, en mucho, las actividades de cualquier organización y, por lo tanto, su nivel de servicio y rentabilidad. Por ejemplo, no se puede pensar en brindar un producto de calidad superior si la de sus componentes no los son; el costo de un bien está afectado por el costo de sus materias primas, materiales y otros insumos adquiridos a los proveedores; la velocidad de llegada al mercado depende de la rapidez de respuesta de los proveedores, ya que el tiempo de ciclo de éstos limita el del productor.

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

---

Para algunas empresas, la solución a estos condicionamientos está en mantener grandes inventarios, realizar innumerables inspecciones a los artículos adquiridos y tener muchos proveedores como salvaguarda ante eventuales incumplimientos. Estas prácticas sólo llevan a incurrir en variadas ineficiencias, entre las que se pueden nombrar los mayores costos por mantenimiento de *stocks*, riesgos por pérdidas, roturas u obsolescencia, los recursos destinados a actividades que no agregan valor para el cliente, como las inspecciones y la sub-optimización de las compras, de aquí estableceremos una parte del objetivo de esta tesis sobre el cumplimiento en los tiempos de respuesta sobre los recursos destinados para el cumplimiento de órdenes.

Por otro lado, para otras compañías, sin embargo, la gran interdependencia que existe con sus proveedores se toma como una gran oportunidad para implementar mejoras, aumentar las ganancias y posibilitar la llegada al mercado con bienes y/o servicios de nivel superior. Para ello, establecen alianzas con sus proveedores, realizan planificaciones conjuntas y colaboran mutuamente en la optimización de los procesos de ambas empresas.

Este cambio en las relaciones proveedor-cliente no es fácil, requiere tiempo, esfuerzo y compromiso por parte de cada una de las personas de ambas empresas. Pero también es cierto que, cuando se logra realmente, la integración permite crear ventajas competitivas duraderas. El estrechamiento de las relaciones proveedor-cliente no sólo ofrece la posibilidad de eliminar costos y reducir tiempos, sino la oportunidad de generar un modelo más eficiente de respuesta al consumidor.

Los distintos tipos de productos, las diferentes exigencias en los tiempos de entrega, la variedad de órdenes solicitadas, los múltiples segmentos que una empresa puede servir, implican consideraciones logísticas diferentes para cada uno de ellos. Es común que los procesos de abastecimiento no sean los mismos para los diferentes tipos de insumos, materias primas o materiales utilizados para la fabricación de distintos tipos de productos; de la misma forma, la demanda de los bienes a través de diversos canales también exige diferenciación en cuanto a la composición de los

pedidos -cantidad y variedad-, los tipos de embalajes a utilizar, el tipo de transporte, y los tiempos de llegada al mercado.

La integración de mejoras y la interacción entre proveedores y clientes ofrecen algunas de las siguientes ventajas de la cadena integrada de suministros

- Flujo ágil de productos y servicios.
- Menor stock en toda la cadena.
- Reducción de costos por ineficiencias.
- Disminución del tiempo de entrega (*lead time*) total.
- Plazos de entrega confiables.
- Mejor calidad de servicio.
- Mayor disponibilidad de bienes.
- Mayor confiabilidad de los pronósticos de demanda.
- Relaciones más confiables con los socios de la cadena.
- Sinergia entre los mismos.
- Flexibilidad en la capacidad de respuesta.
- Fortalecimiento de la posición en el mercado.

### **1.3 Modelos para mejorar el flujo de bienes y servicios.**

Algunas de las acciones que tienden a agilizar en forma económica el flujo logístico ya se han visto, sin embargo, a fin de ofrecer una mayor claridad, y a manera de síntesis, se enumeran aquí algunas de las prácticas indispensables para su optimización:

- Enfoque en el cliente.
- Adecuar la producción en función de la demanda.
- Coordinar las actividades de marketing y producción.
- Adecuar las compras de acuerdo a las necesidades de producción.
- Balancear el ciclo total de abastecimiento-producción entrega.



- Crear y mantener relaciones de mayor colaboración con proveedores y clientes.
- Establecer canales diferenciados para diferentes clientes.
- Optimizar la recepción de mercaderías.
- Eliminar controles innecesarios.
- Reducir inventarios.
- Disminuir los tiempos muertos.
- Minimizar manipulaciones de materias primas, materiales, productos en proceso y finales.
- Utilizar equipos y herramientas adecuadas para el manejo y transporte de materiales.
- Reducir distancias.
- Optimizar el *layout*<sup>6</sup> de almacenes y de planta.
- Señalizar los almacenes.
- Registrar adecuadamente los sitios de almacenaje de cada materia prima, repuesto y producto terminado.

#### 1.4 El procesamiento de un pedido.

En la administración de la cadena de suministro, la logística se puede definir como un subproceso de la cadena de suministros que se ocupa del planeamiento, la dirección, y del control del almacenaje de mercancías entre el punto de la fabricación y el punto de la consumo. En el pasado, las mercancías fueron producidas, almacenadas y después entregadas al pedido. Hoy en día, muchas compañías no trabajan con *stocks*, están usando centros de distribución.<sup>7</sup> Las mercancías se transportan de los surtidores a estos centros de distribución, se almacenan, y después se envían a los consumidores. La carencia del almacenaje puede aumentar el plazo de expedición, pero reduce considerablemente el volumen

---

<sup>6</sup> Chase Richard y Aquilano Nicholas, Gestión de la producción y dirección de operaciones, Mc Graw Hill. 2001.

<sup>7</sup> Juan Pablo Antun Callaba, Logística Internacional" Instituto de Ingeniería UNAM 2004

de capital invertido y aumenta la flexibilidad de la cadena de suministro. La cuestión clave es entregar las mercancías en tiempo reduciendo al mínimo los *stocks*. Las mercancías se deben entregar en la fecha correcta (no antes o no después) para asegurar la satisfacción de los consumidores.

El pedido es quizás la meta para un proceso productivo después de cubrir las mejoras en el flujo de bienes y servicios, el objetivo es el tiempo requerido para completar las actividades sobre el pedido, en el, está el núcleo del servicio al cliente y el crecimiento de las industrias. Se ha estimado que las actividades propias en cuanto a la preparación, transmisión, entrada y levantamiento de un pedido representan del 50% al 70% del tiempo total del ciclo del pedido de muchas industrias. Por lo tanto, si se tiene que dar un alto nivel de servicio al cliente mediante tiempos cortos y consistentes con la fase del pedido, es importante que se administren de manera adecuada esas actividades.

Con el tiempo, el costo de tener información a tiempo y confiable en toda la cadena ha descendido de manera notable, mientras que los costos de mano de obra y de materiales ha aumentado. Debido a ello se ha usado la información para reponer inventarios, reduciendo así los costos de la logística.

El procesamiento del pedido está representado por el número de actividades incluidas en el ciclo del pedido al cliente específicamente, incluyen la preparación, la transmisión, la entrada, el surtido y el informe sobre el estado del pedido. El tiempo requerido para completar cada actividad depende del tipo de petición implicada.<sup>8</sup>

#### **1.4.1 Elementos típicos del procesamiento del pedido.**

El proceso de control de pedidos actualmente se realiza como se muestra en la figura 3 observándose las siguientes etapas:

1. Recepción del pedido.

---

<sup>8</sup> Ballou, administración de la cadena de suministro, Pearson, 2005.

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

2. Autorización del pedido por parte del supervisor de ventas.
3. Verificación de existencia de los artículos para surtir el pedido.
4. Apartado de los artículos existentes.
5. Notificación de artículos faltantes.
6. Facturación del pedido.
7. Gestión de empaqueo y embarque.

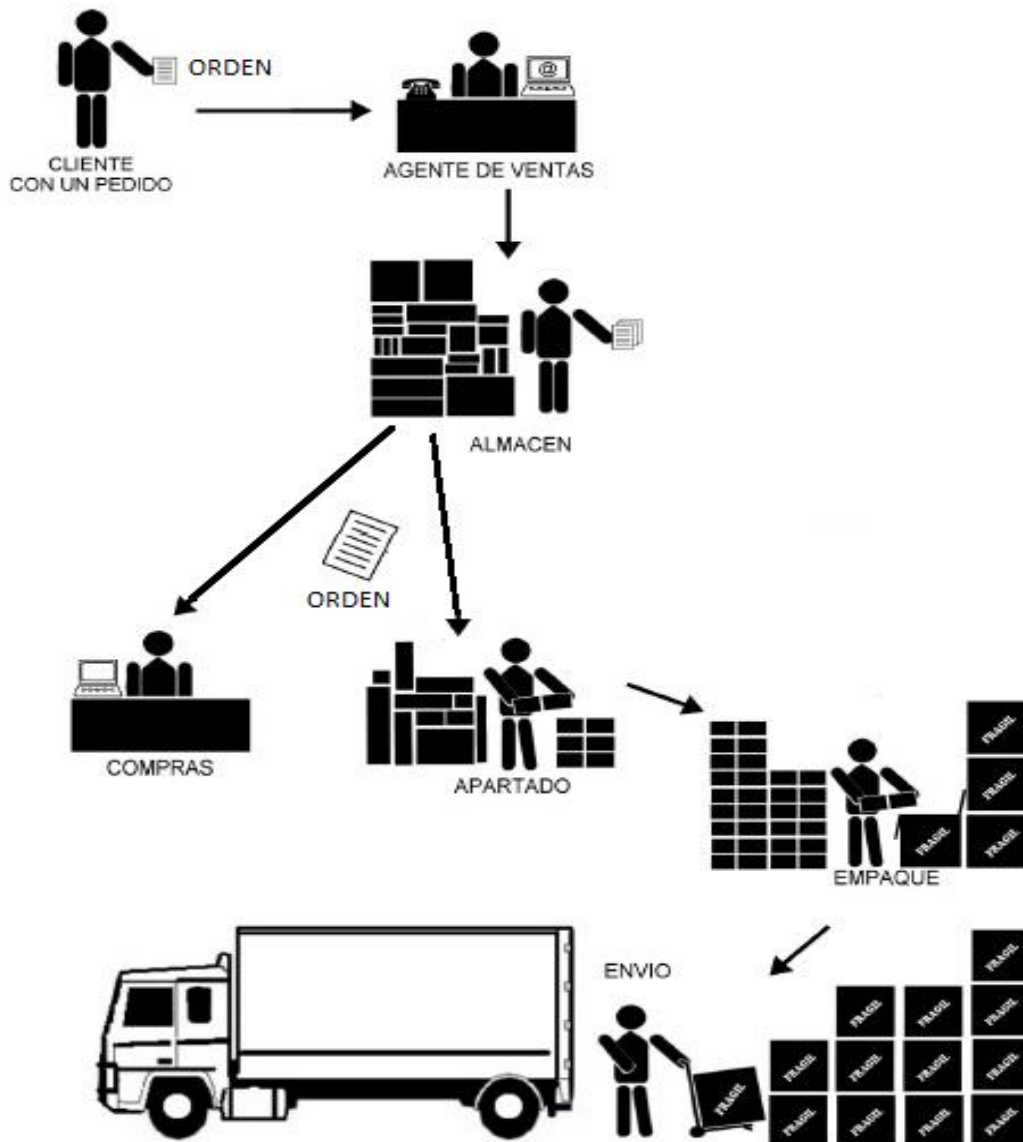


Figura 3 Proceso total del pedido.<sup>9</sup>

<sup>9</sup> Juan Manuel Molina Monroy. Sistema de control de pedidos de clientes, tesis UNAM. 2005.

## 1.5 El proceso logístico asociado al pedido.

### 1.5.1 Recepción del pedido.

Dependiendo del cliente, la recepción del pedido puede realizarse a través de una llamada telefónica, por medio de un documento en papel o un archivo electrónico, (Excel, Word, etc....).

Luego de ser recibido el pedido por parte de algún agente del departamento de ventas, éste llena el formato de solicitud de pedido. En el cual se especifica entre otras cosas, el precio asignado al cliente. Figura 4.



Figura 4 Levantamiento del pedido.<sup>10</sup>

### 1.5.2 Autorización del pedido.

Los pedidos se agrupan y se entregan al supervisor de ventas (figura 5), quien se encarga de revisar los precios, descuentos y demás información relevante. Una vez validada la información se autoriza y notifica al almacén por medio de una relación de los pedidos generados.

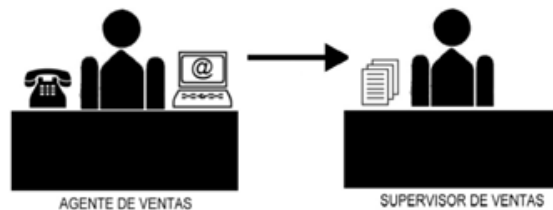


Figura 5 Autorización del pedido.<sup>11</sup>

<sup>10</sup> Juan Manual Molina Monroy. Sistema de control de pedidos de clientes, tesis UNAM. 2005.

<sup>11</sup> Juan Manual Molina Monroy. Sistema de control de pedidos de clientes, tesis UNAM. 2005.

### 1.5.3 Verificación de existencia.

El almacén recibe la información y la compara con su reporte de existencias para después elaborar dos informes, uno con la información por pedido de la mercancía existente en el momento para ser facturada y la otra sobre la mercancía que se requiere.



Figura 6 Verificación de la existencia.<sup>12</sup>

### 1.5.4 Apartado de los artículos.

Con la relación de mercancía requerida se hace el apartado físico de la mercancía. Como se muestra en la figura, aquí es donde se verifica el nivel de inventario (*stock*).

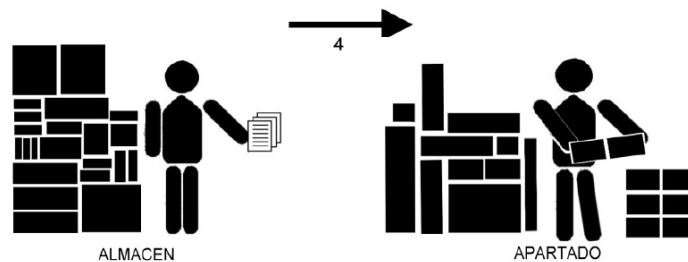


Figura 7 Apartado de la mercancía solicitada.<sup>13</sup>

### 1.5.5 Notificación de artículos faltantes.

Con la relación de mercancía faltante se notifica al supervisor de ventas quien a su vez le notifica al cliente, así también, se notifica al departamento de compras para

<sup>12</sup> Juan Manual Molina Monroy. Sistema de control de pedidos de clientes, tesis UNAM. 2005.

<sup>13</sup> Juan Manual Molina Monroy. Sistema de control de pedidos de clientes, tesis UNAM. 2005.

la gestión correspondiente. Este informe es elaborado en un archivo para ser cargado al sistema de compras existente. Como se observa en la figura 8.

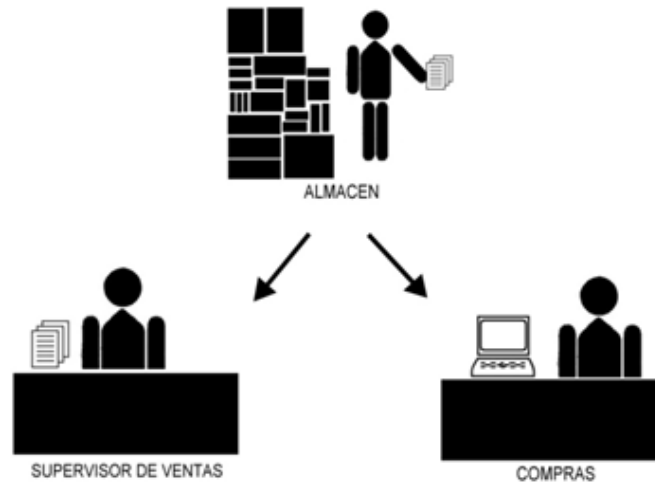


Figura 8 Notificación de faltantes<sup>14</sup>.

#### 1.5.6. Almacén.

En el almacén después de cotejar la información de los pedidos con las existencias se generan un informe de pedidos a registrar, con el que se desprende la siguiente información:

- Numero de pedido.
- Cliente.
- Fecha de alta del pedido.
- Fecha de entrega.
- Autorización.
- Descripción del artículo.
- Cantidad Pedida.
- Cantidad a entregar.
- artículos surtidos.
- artículos sin surtir.

---

<sup>14</sup> Juan Manual Molina Monroy. Sistema de control de pedidos de clientes, tesis UNAM. 2005.

Si bien el objetivo de la tesis no es desarrollar un sistema que trate de resolver en mayor medida todos los problemas involucrados con las funciones logísticas de cada uno de los departamentos que conforman a una organización, nos remitiremos solamente a cubrir al departamento que involucra el aprovisionamiento y el resurtimiento de inventario como es el almacén. Y los movimientos que de ahí surgen.

En un análisis preliminar sobre el proceso que tiene el pedido, desde que se solicita, hasta que se surte y en ese sentido, cómo afecta directamente al inventario, aquí es donde de una manera inicial podemos identificar que es ineludible encontrar un método que asigne ciertos elementos que conforman una orden (cliente), con la restricción que cada cliente asigna a cada pedido, como es la solicitud de un día de surtimiento deseado, ya que sincronizar, la asignación de elementos y/o componentes con restricciones temporales, impone una gran gasto de recursos, con esto, podemos compaginar la asignación de tareas de una máquina, contra la asignación de elementos en ordenes con restricciones temporales, ya que estos representan en un sentido, parte de la división de problemas NP-duro<sup>15</sup> debido a la limitación de los recursos, la cantidad de restricciones que pueden llegar a tener y la forma en que estas son aplicadas sobre un problema de calendarización (*scheduling*).

Los problemas de asignación de órdenes están asociados en organizar una secuencia de recursos limitados y restricciones impuestas en un periodo de tiempo especificado. Cabe señalar que cada instancia presenta restricciones propias a su instauración, aunque generalmente el problema considera el siguiente conjunto de restricciones: asignación de recursos y asignación de tiempo. Por otro lado, tomando como base las diversas complejidades de los algoritmos, donde en primer lugar encontramos la complejidad P que son algoritmos que encuentran siempre una solución a un problema en tiempo polinomial. La complejidad NP de un algoritmo, encuentra una solución a un problema en tiempo polinomial usando una

---

<sup>15</sup> Steven Cook, "The Complexity of Theorem Proving Procedures", University of Toronto, 1974.

computadora no determinística. Encontramos dos tipos clásicos: La complejidad NP-Completo, en estos no existen algoritmos de solución exacta que entreguen respuesta en un tiempo razonable, de ahí se derivan los problemas de búsqueda NP-duros, donde no hay algoritmo de tiempo polinomial que lo resuelva, aquí es donde se encuentran los problemas de tipo de asignación.

Existen diversas técnicas para la resolución de problemas de asignación de recursos, las cuales se clasifican en dos grandes grupos:

1. **Técnicas tradicionales:** programación entera, programación lineal, etc.
2. **Técnicas no tradicionales:** Algoritmos heurísticos o meta-heurísticos que proveen de una buena solución en un tiempo moderado y que suelen proveer de mejores resultados en instancias grandes del problema.

Si bien sabido que estos últimos no necesariamente producen la solución óptima del problema, hay que tomar en cuenta que cuando el espacio de búsqueda del problema es enorme, como ocurre en este caso, los métodos exactos de resolución suelen ser demasiado costosos, ya sea en tiempo de ejecución o en recursos computacionales. En esta tesis nos centramos en el problema de asignación de recursos, el objetivo, por lo tanto, será diseñar estrategias y cauces para obtener buenas soluciones en un tiempo razonable. Para ello se deberán estudiar aspectos heurísticos como son, estructuras de vecindad, propiedades del camino crítico, constructores de planificación, y en general encontrar una mejor configuración para el método utilizado.

Las aportaciones principales de esta tesis serán el estudio entre las características del *scheduling clásico* y el diseño de una serie de estructuras de vecindad con sus respectivas condiciones de factibilidad y de mejora, y otras consideraciones sobre cómo realizar una búsqueda local de la forma más eficiente posible, aquí aplicaremos el reciente algoritmo de optimización por colonia de hormigas (*Ant system*).



La presente tesis está organizada de la siguiente forma, en este capítulo, se planteó la logística relacionada con las órdenes en una estructura de distribución comercial, en el capítulo 2, se dará un marco teórico y la complejidad que conlleva el *scheduling* en la solución de problemas de asignación de recursos y la forma en que se pretende adaptar el algoritmo por optimización de colonias de hormigas, para el capítulo 3, abordaremos una introducción al algoritmo, destacando su principal variante *ants system* y la aplicación del algoritmo, utilizado, insertando su principal aporte, la restricción jerárquica de órdenes, y por último en el capítulo 4, se presentara un ejemplo de aplicación, con el cual se tomara el problema de una empresa y la utilidad del método ante este tipo de problemas.

## Capítulo 2: Marco de referencia.

### 2.1 Estado del arte.

Dentro del área de la ingeniería y las matemáticas aplicadas, la búsqueda de la mejor solución a un problema de entre todo un conjunto de posibles soluciones, es llamada Optimización. Dentro de esta área podemos encontrar una diversidad de problemas que podemos clasificar en dos grupos principales de acuerdo al tipo de solución que se trata de encontrar: Optimización Numérica y Optimización Combinatoria. Por ejemplo: asumiendo que las variables son continuas en las funciones, el problema se considera de optimización numérica, en cambio, si las variables son de naturaleza discreta, el problema de encontrar soluciones óptimas es conocido como optimización combinatoria. En esta tesis, nos ocuparemos de un problema de optimización combinatoria, por lo cual nuestro análisis se centrará en este tema. La optimización combinatoria incluye un conjunto de problemas cuyo objetivo es localizar la combinación adecuada que minimice (o maximice) una función llamada función objetivo y que además la solución cumpla con las restricciones particulares del problema.

Los problemas de asignación de recursos son un tipo de problema perteneciente al área de la optimización combinatoria que ha atraído el interés de muchos investigadores por la alta complejidad de los mismos y la cantidad de recursos consumidos en la búsqueda de buenas soluciones. Básicamente, estos problemas constan de un conjunto de actividades ( $a$ ) que deben ser procesadas en un conjunto finito de recursos disponibles ( $r$ ) (bajo ciertas restricciones), para lo cual, el tamaño de un problema está dado por  $a \times r$ , de tal forma que el número de soluciones al problema crece en forma exponencial a medida que se aumentan actividades, recursos o ambos. En este contexto, se buscará encontrar una solución utilizando una técnica heurística (*Ant System*), que permita cumplir las restricciones marcadas y que contemple las limitaciones sobre los recursos con que se cuenta, y también las prioridades que pudiera haber sobre la distribución jerárquica de órdenes.

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

En esta tesis abordaremos el cumplimiento en la asignación de órdenes, utilizando el concepto de *scheduling*<sup>16</sup> con la idea de utilizar un algoritmo heurístico llamado *Ant Colony Optimization* en su variante *Ant System*, que pudiera analizar, modificar y depurar la manera en que se realiza la distribución de los recursos logísticos existentes dentro de una estructura productiva.

Originalmente la asignación de recursos, se puede conceptualizar como una forma de *scheduling*, que consiste en asignar una serie de tareas a una serie de recursos con capacidad a un costo total mínimo. Cada tarea debe ser asignada a un solo recurso que requiera una cierta cantidad de esta capacidad.

Algunos problemas de programación juegan un papel central en la investigación de ACO, y muchos diferentes tipos de problemas de programación han sido atacados con algoritmos ACO (tabla 1). El rendimiento, sin embargo, varía de problema a problema. El problema de programación de proyectos limitados a recursos, el ACO es uno de los enfoques con mejores resultados.

**Tabla 1 Algoritmos ACO disponibles para el problema scheduling.**

Problemas	Referencias
JSP( job shop problema)	Colomi, Dorigo, Maniezzo & Trubian (1994)
OSP (open shop problema)	Pfahring (1996); Blum(2003)
PFSP (permutation flow shop problema)	Stutzle (1997, 1998)
SMTTP (single-machine total tardiness problema)	Bauer, Bullheimer, Hartl, Strauss (2000)
SMTWTP( single-machine total weighted tardiness problem)	Stutzle, Dorigo(2000)
RCPSP (Resource_constrained Project scheduling problem)	
GSP (group shop scheduling problem)	Merkle, Middendorf (2002)
SMTTPSDST (single-machine total tardiness problema with sequence dependent setup times)	Blum(2002), Gagné, Price, Gravel (2002)

Dentro de la literatura existen trabajos, que tratan el objeto de estudio de esta tesis, podemos mencionar algunos:

En *Ant colony Optimization (ACO)*<sup>17</sup>. La tarea de problemas de asignación es la asignación de un conjunto de elementos (objetos, actividades, etc) a un número determinado de recursos (lugares, agentes, etc) con algunas restricciones.

<sup>16</sup> Un *schedule* es una secuenciación de horarios de ejecución, definido en términos de atributos particulares, como fecha y hora de inicio, periodicidad, fecha y hora de finalización, etc.

<sup>17</sup> Marco Dorigo, Thomas Stutzle, *Ant Colony Optimization*, 2004, the MIT Press.

Las asignaciones pueden, en general, representarse como una asignación de un conjunto  $i$  a un conjunto  $j$ , y la función objetivo para reducir al mínimo es una función de las asignaciones hechas.

Para aplicar la metaheurística ACO a problemas de asignación, un primer paso es asignar el problema a un gráfico de construcción<sup>18</sup>,  $G_C = (C, L)$ , donde  $C$  es el conjunto de componentes y  $L$  es el conjunto de conexiones que conecta plenamente al gráfico. Este procedimiento de construcción permite a las hormigas realizar paseos por la construcción en el gráfico que corresponde a la asignación de artículos a los recursos. En la figura 9, es un ejemplo de una representación de un grafo para un problema con 3 actividades y 3 procesos cada uno. Este grafo tiene un tiempo mínimo y máximo para la terminación de sus actividades. Los arcos están etiquetados con este costo de la tarea (tiempo), así  $t_{ab}$  es la tarea en el que  $a$  es la tarea asignada y  $b$  es el proceso a realizarse y  $R_c$  es el retraso derivado por la ejecución del proceso, así, este nos servirá como base para compaginar la asignación de recursos con la realización de actividades del *Scheduling* clásico.

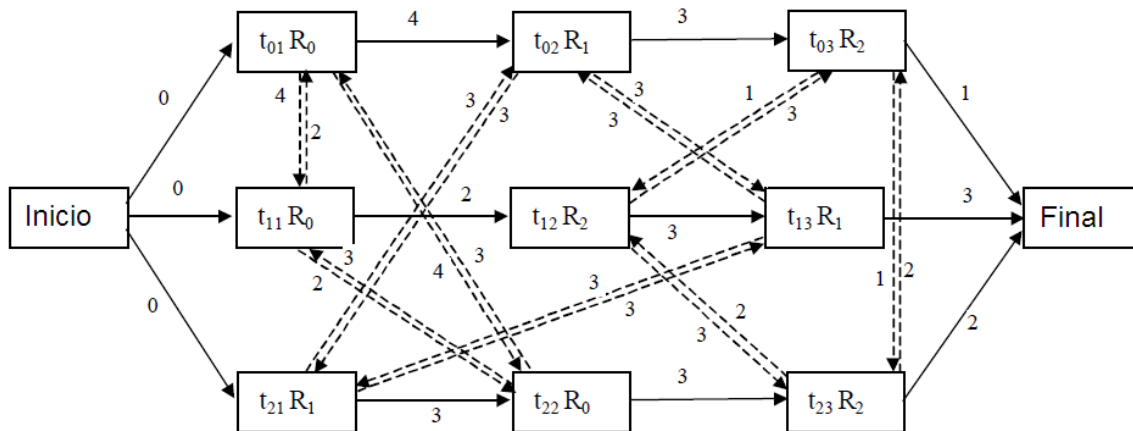


Figura 9 Grafo de construcción<sup>19</sup>.

<sup>18</sup> Representación gráfica sobre posibles relaciones siguiendo una ruta preestablecida y basada en decisiones algorítmicas.

<sup>19</sup> Varela R. Blanco j. Combinación de algoritmos evolutivos y técnicas heurísticas para problemas de Scheduling. Universidad de Oviedo. 2001.

Para la aplicación práctica de la metaheurística ACO a problemas de asignación es conveniente distinguir entre dos tipos de decisiones. La primera se refiere a la asignación de los artículos, es decir, el orden en que los diferentes artículos son asignados a los recursos. La segunda decisión se refiere a la asignación actual, es decir, la elección del recurso al que se asigna un elemento. Rastros de feromonas e información heurística pueden estar asociados con ambas decisiones. En el primer caso, los rastros de feromona y la información heurística se pueden utilizar para decidir sobre el orden de asignación correspondiente.

En el segundo caso, el rastro  $\tau_{ij}$  de feromona y la información heurística  $\eta_{ij}$  asociada con el par  $(i, j)$ , donde  $i$  es un elemento y  $j$  un recurso, determina la conveniencia de asignación de elemento  $i$  al recurso  $j$ .

Todos los algoritmos ACO para problemas de asignación tienen que tomar estas dos decisiones en cuenta. En todas las aplicaciones de ACO a problemas de asignación, los rastros de feromonas se utilizan únicamente para una de estas dos decisiones. Típicamente, los rastros de feromonas se refieren a la manera en que se realiza el ciclo de asignación, en ese sentido, sobre la decisión en qué orden se realizara la asignación, y en esta parte, la mayoría de los algoritmos aprovechan una rutina heurística o bien una forma aleatoria para lograr dicha asignación.

La asignación de actividades se ha tratado en muchos campos y con igual número de técnicas, tanto exactas como heurísticas, en este último caso, quizás el medio teórico que ha tenido un mayor estudio, con aceptables resultados, sobre otros métodos.

En “*Optimization of logistic processes in supply chains using metaheuristics*”,<sup>20</sup> la programación de los procesos logísticos modelado por la llegada y la terminación de procesos es un problema combinatorio. Considerando el problema de la asignación dinámica de los componentes a las órdenes, en ese sentido la idea es

---

<sup>20</sup> Carlos A. Silva, Thomas A. Runkler, Optimization of logistic processes in supply-chains using metaheuristics, Siemens AG, Lisboa Portugal, 2003.

asignar agentes individuales a las órdenes y dejar a la población de agentes interactivamente encontrar una solución organizada. La interacción entre los agentes es observada intercambiando la información sobre cantidad, fecha deseada y fecha de arribo. En un problema logístico de calendarización, el número de los agentes implicados y la cantidad de información que tenga que ser intercambiada es muy grande. Los algoritmos multiagente basados en insectos sociales pueden evitar esta complejidad. Los insectos sociales han llamado la atención de diversos investigadores debido al alto nivel de la estructuración que las colonias pueden alcanzar, especialmente cuando se compara con la simplicidad relativa de los individuos. Las hormigas son un ejemplo de insectos sociales. Aunque las hormigas son animales muy simples, sin capacidades especiales y casi ciegas, son capaces de establecer las trayectorias más cortas de la ruta de sus colonias a las fuentes de alimento y de nuevo a los hormigueros.

En “*A logistic process scheduling problem: genetic algorithms or ant colony optimization*”,<sup>21</sup> con el fin de mejorar la competitividad y productividad, la mayoría de las empresas hoy en día están organizadas como cadenas de producción: una red mundial de socios externos (proveedores, almacenes y centros de distribución) a través del cual se adquieren materias primas, transforman en productos y entregados a clientes. En este contexto la logística es un subconjunto de la cadena de suministro que recoge la mercancía comprada a los proveedores y los envía al exterior a diferentes clientes en el tiempo, reduciendo al mínimo el tiempo de entrega y los costos de almacenamiento. Por lo tanto, el control de los procesos de la logística consiste en optimizar un problema de programación. Hoy en día, los metaheurística como algoritmos genéticos o, más recientemente, la optimización de colonia de hormiga (ACO),<sup>22</sup> se consideran que son técnicas de programación muy potentes. Estas técnicas son muy atractivas para diseñar un método de optimización para una nueva aplicación en la calendarización, debido a

---

<sup>21</sup> C.A. Silva, J.M. Sousa. *A logistic process scheduling problem: genetic algorithms or ant colony optimization*, Instituto Superior Tecnico, Lisboa Portugal, 2004.

<sup>22</sup> Marco Dorigo, Thomas Stutzle, *Ant Colony Optimization*, 2004, the MIT Press.

que son fáciles de implementar y han demostrado ser eficientes para encontrar soluciones óptimas para diferentes aplicaciones.

Dentro del análisis que se pretende construir en cuanto a la distribución de órdenes a componentes, esta tesis esta insertada en un problema clásico de *Job Shop* con lo cual se pretende retomar el algoritmo optimización por colonia de hormigas (ACO) en su variedad *Ant System*. El problema del *Job Shop Scheduling*, corresponde a un tipo de problema de planificación de tareas, y es un problema de optimización. La función del *scheduling* es la asignación de recursos limitados a tareas a lo largo del tiempo y tiene como finalidad la optimización de uno o más objetivos. En donde uno de los objetivos principales es la disminución de tiempos en la planificación de las tareas. En el caso concreto de esta tesis, el objetivo es encontrar una distribución óptima de las órdenes que ingresan al área de inventario, con la idea de entregar estas, considerando la disponibilidad de los elementos que la integran. Así, nos apoyaremos en dos particularidades principales que tienen las mismas, la primera el día deseado de entrega, y segundo el día de entrada, estas dos condiciones afectarán de manera directa a la parte de la demanda de estos componentes, esto arrojará datos importantes acerca de los tiempos en que se tendría que considerar del resurtimiento de los mismos y el manejo que la organización le dará a este factor, otro elemento a considerar es la importancia que le damos a una orden sobre otra, aun cuando ésta esté lista para ser entregada, esto es un extra que cada organización aporta sobre el valor que cada cliente tiene, con esto aseguraremos con absoluta certeza que las entregas serán surtidas en tiempo y forma, algo que es una aportación importante dentro de este estudio.

## 2.2 Marco teórico.

Cuando se pretende obtener resultados a corto plazo es importante identificar qué tipo de problema se plantea solucionar, y cuál será la metodología que se seguirá para alcanzar tal fin, en cuanto al planteamiento que se definió en el apartado anterior, este es un problema de optimización combinatoria. Tales problemas aparecen en situaciones que involucran elecciones entre alternativas discretas y

solucionarlos equivale a encontrar una solución óptima entre esas opciones. Por otro lado, la optimalidad debe ser entendida de acuerdo a cierto criterio de costo que nos provea de una medida cuantitativa de la calidad de cada solución. Es decir, estos problemas residen en encontrar en un conjunto de esquemas permitidos, aquel que tenga un costo asociado óptimo. La naturaleza de estos problemas hace que rara vez sean tratados teóricamente. Por lo tanto, nos vemos motivados a eludir el cálculo directo y considerar la utilización de computadoras y algoritmos específicos que nos auxilien en nuestra tarea. Pero aquí aparece una dificultad imprevista, la cual radica en el tiempo que demora la computadora en encontrar la solución puesto que, por desgracia, muchos problemas de optimización combinatoria pertenecen a la clase *NP-duro*, antes mencionado, es decir, el tiempo de cómputo que se requiere para resolver uno de estos problemas se incrementa conforme crece el tamaño del problema presentando una dependencia funcional tal que no admite ser acotada por un polinomio.

En consecuencia, nuestra expectativa se centra en torno a algoritmos que encuentran soluciones casi óptimas en tiempos de aplicación razonables. Es decir, pactamos comprometer la calidad de la solución a cambio de obtener el acceso a un resultado en tiempo viable.

La literatura distingue entre dos clases principales de algoritmos de aproximación: constructivos (*constructives*) y de búsqueda local (*local search*). Nos detendremos principalmente sobre estos últimos.

Hay muchas aplicaciones particulares de búsqueda local pero todas descansan sobre la utilización del concepto de vecindad (*neighborhood*), el cual es usado para guiar la búsqueda iterativa hacia buenas soluciones.

El uso de la búsqueda local en problemas de optimización combinatoria se remonta a 1950 cuando surge el primer algoritmo de intercambio de arcos (*edge-exchange*) para resolver el problema del agente viajero (*travelling salesman problem*). En los años subsiguientes el alcance de la búsqueda local se amplió hasta incluir aplicaciones a problemas de toda índole.



Muchas variantes de búsqueda local han sido propuestas basadas en analogías con procesos de la naturaleza, como *simulated annealing* (recocido simulado), *genetic algorithms* (algoritmos genéticos) y algunos ejemplos de *neural networks* (redes neuronales). El incremento de recursos de cómputo y la sofisticación de las estructuras de datos han posibilitado el empleo de estos algoritmos para resolver complejos problemas de la vida real. Se han modelado matemáticamente y obtenido consistentes resultados teóricos en cuanto a su complejidad y eficiencia. El ejemplo más difundido es el *simulated annealing*, modelado a partir de cadenas de Markov.

### 2.2.1 El algoritmo de optimización por colonia de hormigas (ACO).

Las hormigas constituyen la familia *Formicidae*, orden *Hymenoptera*. El nombre científico de la hormiga cosechera roja (o agrícola) es *Pogonomyrmex barbatus*.

La comunicación entre hormigas es muy eficaz y se realiza sobre todo por medios táctiles y químicos, aunque algunas especies emplean mecanismos vibratorios e incluso auditivos. Casi siempre una hormiga ‘exploradora’ alerta a la colonia y la dirección en que se lanzan sus excitadas compañeras de hormiguero puede verse afectada por uno u otro medio, dependiendo de la especie. Por ejemplo, en el caso de la hormiga faraón, común en las cocinas, la exploradora, al regresar al nido, libera un reguero de secreciones químicas hasta el alimento; de esta forma, la excitación de la exploradora es más considerable cuanto mayor sea la concentración de alimento descubierto y, por consiguiente, mayor es el número de compañeras que se ven estimuladas a seguirla<sup>23</sup>.

El algoritmo hormiga se ha venido proponiendo como un modelo computacional que reemplaza los métodos tradicionales haciendo énfasis en el control, pre-programación y centralización de procesos aplicables en automatización de eventos, y distribución de funciones entre otras.

---

<sup>23</sup> Dorigo M. T. Maniezzo. The Ant System: optimization by a colony of cooperating agents. Université Libre de Bruxelles, 1996.

Una investigación particularmente exitosa conocida como optimización de la colonia de hormigas se dedica a la optimización de problemas discretos. Se ha aplicado exitosamente en gran número de problemas difíciles combinatorios como el del agente viajero o la asignación cuadrática.

Las hormigas han sido de gran interés de estudio por tratarse de insectos sociales, viven en colonias y su comportamiento está dirigido más a la supervivencia de la colonia que al del individuo en particular. Otro importante comportamiento de estos insectos tiene que ver con su aprovisionamiento de alimento, específicamente de cómo van identificando la ruta más corta entre las fuentes de alimento y su nido sin utilizar ayudas visuales. Mientras caminan desde la fuente hasta el nido y viceversa, las hormigas depositan en el suelo una sustancia llamada feromona formando en este camino un rastro de feromonas. *Al comienzo de cada búsqueda generalmente, si no encuentra ningún rastro de feromona, una hormiga se mueve de manera básicamente aleatoria, pero cuando existe feromona depositada, las hormigas pueden olerla y tienen mayor tendencia a seguir el rastro cuando escogen su ruta, tendiendo a escoger, probablemente la ruta marcada por la mayor concentración de feromona.* Figura 10.

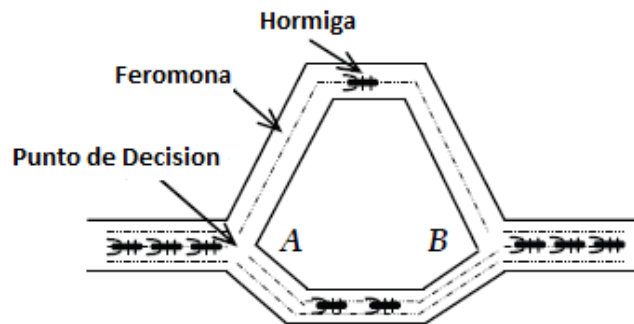


Figura 10 Rutas y Los rastros de feromona dejadas por las hormigas<sup>24</sup>.

El rastro de feromonas permite a las hormigas encontrar el camino más corto de regreso al alimento o al nido, incluso también puede guiar a otros miembros del nido

<sup>24</sup> Enxiu Chen, Xiyu Liu. *Multi-Colony Ant Algorithm*. School of Management & Economics, Shandong Normal University, Jinan, Shandong, China

a la fuente de alimento. Esto ocurre gracias a que al ser los caminos más cortos, las hormigas que los siguen consiguen encontrar la comida más rápidamente, por lo que comienzan su viaje de retorno antes. Entonces, en el camino más corto habrá un rastro de feromona ligeramente superior y, por lo tanto, las decisiones de las siguientes hormigas estarán dirigidas en mayor medida a dicho camino. Además, este camino recibirá una proporción mayor de feromona por las hormigas que vuelven por él que por las que vuelven por el camino más largo. Este proceso finaliza haciendo que la probabilidad de que una hormiga escoja el camino más corto aumente progresivamente y que al final el recorrido de la colonia converja al más corto de todos los caminos posibles. Cuando se han armado muchas rutas, la colonia de hormigas es capaz de aprovechar el rastro de feromonas dejado por otros individuos para descubrir la ruta más corta. Por ejemplo en el caso de un obstáculo, que se presenta en el camino de las hormigas de la comida al nido, estas bordean indistintamente los dos lados del obstáculo buscando el rastro del antiguo camino de feromona. Figura 11.

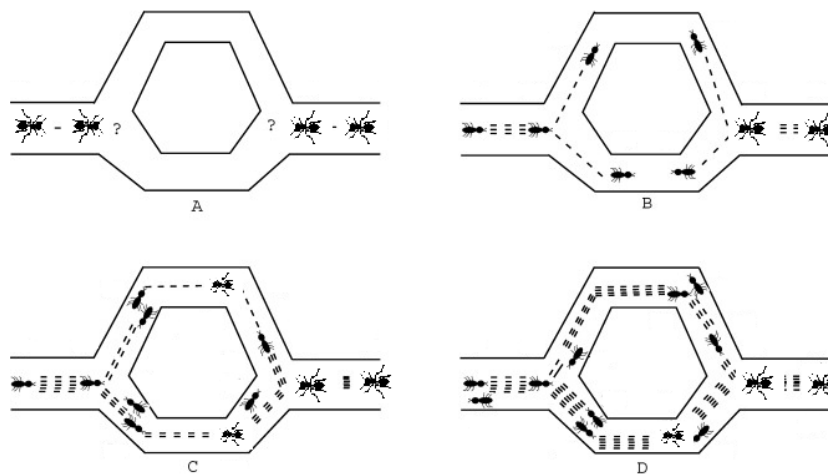


Figura 11 Comportamiento de las Hormigas ante un Obstáculo y su evolución<sup>25</sup>.

Pero las hormigas que escogen la ruta corta alrededor del obstáculo reconstruyen el sendero de feromona más rápido. Esto se debe a que la ruta corta recibe una

<sup>25</sup> Surendar Kumar. Application of ant colony, genetic algorithm and data mining-based techniques for scheduling. Mechanical Engineering Department, NIT, Warangal, AP, India.

mayor cantidad de feromona por unidad de tiempo ( $t_1$ ,  $t_2$ ,  $t_3$ ), y por tanto más hormigas toman la ruta más corta. Figura 12.

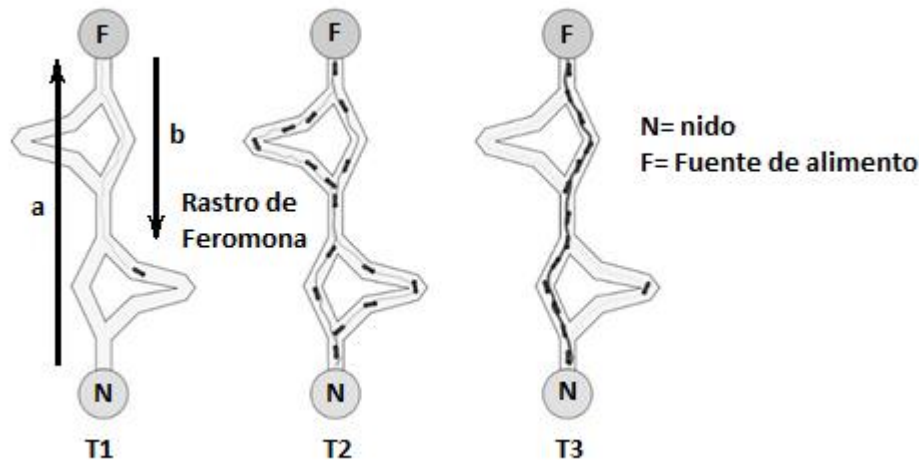


Figura 12 Rastro de feromona (variable en el tiempo).<sup>26</sup>

Esta convergencia se complementa con la acción del entorno natural que provoca que la feromona se evapore transcurrida un cierto tiempo. Así, los caminos menos prometedores pierden progresivamente feromona porque son visitados cada vez por menos hormigas. Sin embargo, algunos estudios biológicos han demostrado que los rastros de feromona son muy persistentes, la feromona puede permanecer desde unas pocas horas hasta varios meses dependiendo de varios aspectos distintos, como la especie de las hormigas, el tipo de suelo, lo que provoca una menor influencia del efecto de la evaporación en el proceso de búsqueda del camino más corto. Numerosos experimentos muestran que, debido a la gran persistencia de feromona, es difícil que las hormigas “olviden” un camino que tiene un alto nivel de feromona aunque hayan encontrado un camino aún más corto. Hay que tener en cuenta que si se traslada este comportamiento directamente al ordenador para diseñar un algoritmo de búsqueda, podemos encontrarnos con que se quede rápidamente estancado en un óptimo local.

<sup>26</sup> Enxiu Chen, Xiyu Liu. *Multi-Colony Ant Algorithm*. School of Management & Economics, Shandong Normal University, Jinan, Shandong, China

Este tipo de comportamiento permite describir las bondades del mecanismo de optimización por el cual cada individuo de la colonia hace una pequeña contribución. Los modelos computacionales han venido desarrollándose simulando procesos de búsqueda de alimento; los resultados han sido satisfactorios, mostrando que un simple modelo de probabilidad es suficiente para justificar los complejos y variados modelos colectivos.

Comencemos por llamar hormiga a un agente actor, del algoritmo hormiga que vamos a explicar, que al igual que en una colonia verdadera cada agente representa un individuo de la colonia.

En las hormigas los efectos de esa reacción pueden tomarse como un nuevo estímulo significativo tanto para el insecto que lo produce, como para otros individuos de la colonia<sup>27</sup>.

La producción de un nuevo estímulo significativo como consecuencia de la reacción a otro estímulo significativo, determina una fórmula de coordinación de actividades y puede ser interpretada como una manera de comunicación indirecta.

### 2.2.3 Construcción del algoritmo optimización de colonia de hormigas (ACO)

Los algoritmos de ACO son esencialmente algoritmos constructivos: esto es, en cada iteración del algoritmo, cada hormiga construye una solución al problema recorriendo un grafo de construcción. Cada arco del grafo, que representa los posibles pasos que la hormiga puede dar, tiene asociada dos tipos de información que guían el movimiento de la hormiga:

- Información heurística, que mide la preferencia heurística de moverse desde el nodo  $i$  hasta el nodo  $j$ , o sea, de recorrer el arco o tramo  $a_{ij}$ . Se nota por  $\eta_{ij}$ . Las hormigas no modifican esta información durante la ejecución del algoritmo.

---

<sup>27</sup> Pierre-Paul Grassé en 1946 observó que los insectos son capaces de responder a los llamados “estímulos significativos (stigmergy)” mecanismos ambientales para coordinar el trabajo de los actores independientes. Tratado de zoología. (*Traité de Zoologie*) 1948.

- Información de los rastros de feromona artificiales, que mide la “deseabilidad aprendida” del movimiento de  $i$  a  $j$ . Imita a la feromona real que depositan las hormigas naturales. Esta información se modifica durante la ejecución del algoritmo dependiendo de las soluciones encontradas por las hormigas. Se denota por  $\tau_{ij}$ .

En esta sección se presentan los pasos que llevan a las hormigas reales al ACO. A partir de ahora debe tenerse en cuenta que los algoritmos ACO presentan una doble perspectiva:

- Por un lado, son una abstracción de algunos patrones de comportamiento naturales relacionados con el comportamiento que permite encontrar el camino más corto. Por otro lado, incluyen algunas características que no tienen una contrapartida natural, pero que permiten que se desarrollen algoritmos para obtener buenas soluciones al problema que se pretende resolver (por ejemplo, el uso de información heurística que guíe el movimiento de las hormigas).

Los problemas de optimización combinatoria pueden representarse como una función de la forma  $G = (N, A)$ , donde  $A$  es el conjunto de arcos que conectan un conjunto de componentes  $N$ . Donde  $G$  se denomina función de construcción  $G$  Por tanto, tenemos que:

- Los componentes  $\eta_i$  son los nodos del problema.
- Los estados  $d$  (y por tanto las soluciones  $S$ ) se corresponden con caminos en la gráfica solución, esto es, secuencias de nodos o arcos.
- Los arcos de la gráfica,  $a_{ij}$  son conexiones/transiciones que definen la estructura del vecindario. Así,  $d2 = \langle d1, S \rangle$  será vecino de  $d1$  si el nodo  $i$  es la última componente de  $d1$  y el arco  $a_{ij}$  existe en el gráfico.

- Deben existir los costos explícitos  $C_{ij}$  asociados con cada arco; y
- Los componentes o las conexiones deben tener asociados rastros de feromona  $\tau$ , que representan un tipo de memoria indirecta y a largo plazo del proceso de búsqueda, como valores heurísticos  $\eta$  que representan la información heurística disponible en el problema a resolver.

#### 2.2.4 Características de la hormiga artificial.

La hormiga artificial es un agente computacional simple que intenta construir soluciones posibles al problema explotando los rastros de feromona disponibles y la información heurística. Sin embargo, en algunos problemas, puede también construir soluciones no válidas que podrán ser penalizadas dependiendo de lo inadecuado de la solución. Tiene las siguientes propiedades:

- Busca soluciones válidas de costo mínimo para el problema a solucionar.
- Tiene una memoria que almacena información sobre el camino seguido hasta el momento, esto es, almacena la secuencia generada. Esta memoria puede usarse para (i) construir soluciones válidas, (ii) evaluar la solución generada, y (iii) reconstruir el camino que ha seguido la hormiga.
- Tiene un estado inicial, que normalmente se corresponde con una secuencia unitaria y una o más condiciones  $\tau$  de parada asociadas.
- Comienza en el estado inicial y se mueve siguiendo estados válidos, construyendo la solución asociada incrementalmente.
- Cuando está en un estado  $dr = \langle di - 1, i \rangle$  (es decir, actualmente está localizada en el nodo  $i$  y ha seguido previamente la secuencia  $di - 1$ , puede moverse a cualquier nodo  $j$  de su vecindario posible  $N(i)$ , definido como:

$$N(i) = \{s | (a_{ij} \in A) \text{ y } (\langle \delta_i, s \rangle \in \tilde{\Delta})\}$$

- El movimiento se lleva a cabo aplicando una regla de transición, que es función de los rastros de feromona que están disponibles localmente, de los valores heurísticos de la memoria privada de la hormiga y de las restricciones del problema;

- Cuando durante el procedimiento de construcción una hormiga se mueve desde el nodo  $i$  hasta la  $j$ , puede actualizar el rastro de feromona  $\tau_{ij}$ , asociado al arco  $a_{ij}$ .

Al proceso descrito se le llama, actualización en línea de los rastros de feromona paso a paso.

- El procedimiento de construcción acaba cuando se satisface alguna condición de parada.
- Una vez que la hormiga ha construido la solución, puede reconstruir el camino recorrido y actualizar los rastros de feromona de los arcos/componentes visitados(as) utilizando un proceso llamado actualización en línea *a posteriori*. Este es el único mecanismo de comunicación entre las hormigas, utilizando la estructura de datos que almacena los niveles de cada arco/componente (memoria compartida).

### **2.2.5 Modo de funcionamiento y estructura genérica de un algoritmo ACO.**

Como se ha mencionado, el modo de operación básico de un algoritmo de ACO es como sigue: las  $m$  hormigas (artificiales) de la colonia se mueven, concurrentemente y de manera asíncrona, a través de los estados adyacentes del problema. Este movimiento se realiza siguiendo una regla de transición que está basada en la información local disponible en las componentes (nodos). Esta información local incluye la información heurística y memorística (rastros de feromona) para guiar la búsqueda. Al moverse por el espacio de construcción, las hormigas construyen incrementalmente soluciones.

Opcionalmente, las hormigas pueden depositar feromona cada vez que crucen un arco (conexión) mientras que construyen la solución (actualización en línea paso a paso de los rastros de feromona). Una vez que cada hormiga ha generado una solución, ésta se evalúa y puede depositar una cantidad de feromona que es función



de la calidad de su solución (actualización en línea de los rastros de feromona). Esta información guiará la búsqueda de las otras hormigas de la colonia en el futuro.

Para la solución de los algoritmos aquí trabajados no se usará actualización en línea, ya que no genera mayor beneficio.

El modo de operación genérico de un algoritmo de ACO incluye dos procedimientos adicionales, la evaporación de los rastros de feromona y las acciones de cualidades atribuibles tan solo a hormigas artificiales y que se crean según las características de cada problema. La evaporación de feromona la lleva a cabo el entorno y se usa como un mecanismo que evita el estancamiento en la búsqueda y permite que las hormigas busquen y exploren nuevas regiones del espacio. Algunas de estas cualidades son por ejemplo observar la calidad de todas las soluciones generadas y depositar una nueva cantidad de feromona adicional sólo en las transiciones/componentes asociadas a algunas soluciones, o aplicar un procedimiento de búsqueda local a las soluciones generadas por las hormigas antes de actualizar los rastros de feromona. En ambos casos, estas funciones reemplazan la actualización en línea a posteriori de feromona y el proceso pasa a llamarse actualización fuera de línea de rastros de feromona.

La estructura de un algoritmo de ACO genérico es como sigue:

1. Procedimiento Metaheurística ACO.
2. Inicialización de parámetros
3. Mientras (criterio de terminación no satisfecho).
4. Programación de actividades
5. Generación de Hormigas y actividad.
6. Evaporación de Feromona.
7. Acciones de optimización adicional. {opcional}
8. fin Programación de actividades
9. fin mientras
10. fin Procedimiento.

1. Procedimiento Generación de Hormigas y actividad()
  2. repetir en paralelo desde  $k = 1$  hasta  $m$  (numero hormigas)
  3. Nueva Hormiga( $k$ )
  4. fin repetir en paralelo
  5. fin Procedimiento
- 
1. Procedimiento Nueva Hormiga(id Hormiga)
  2. Inicializa hormiga(id Hormiga)
  3.  $L =$  actualiza memoria hormiga()
  4. mientras (estado actual  $\neq$  estado objetivo)
  5.  $P =$  calcular probabilidades de transición( $A, L, W$ )
  6. Siguiete estado = aplicar política decisión( $P, W$ )
  7. Mover al siguiente estado(siguiente estado) si (actualización feromona en línea paso a paso)
  8. Depositar feromona en el arco visitado () fin si
  9.  $L =$  actualizar estado interno()
  10. fin mientras si (actualización feromona en línea a posteriori)
  11. para cada arco visitado
  12. Depositar feromona en el arco visitado()
  13. fin para fin si
  14. Liberar recursos hormiga (id Hormiga)
  15. fin Procedimiento.

El primer paso incluye la inicialización de los valores de los parámetros que se tienen en consideración en el algoritmo. Entre otros, se deben fijar el rastro inicial de feromona asociado a cada transición, que es un valor positivo pequeño, normalmente el mismo para todas las componentes-conexiones, el número de hormigas en la colonia,  $m$ , y los pesos que definen la proporción en la que afectarán la información heurística y memorística en la regla de transición probabilística.

El procedimiento principal de la metaheurística ACO controla, mediante el programa base, la planificación de las tres componentes mencionadas en esta sección: (i) la

generación y puesta en funcionamiento de las hormigas artificiales, (ii) La evaporación de feromona, y (iii) las acciones de optimización propias. La implementación de este programa determinará la sincronía existente entre cada una de las los tres componentes.

Como se ha comentado antes, varias componentes son o bien opcionales, o bien dependientes estrictamente del algoritmo *Ant Colony (AC)* específico, por ejemplo cuándo y cómo se deposita la feromona. Generalmente, la actualización en línea paso a paso de los rastros de feromona y la actualización en línea *a posteriori* de los rastros de feromona son mutuamente excluyentes y no suelen estar presentes a la vez ni faltar ambas al mismo tiempo (si las dos faltan, sol las acciones de optimización propias las que actualizan los rastros de feromona).

Por otro lado, hay que remarcar que el procedimiento actualiza memoria hormiga () se encarga de especificar el estado inicial desde el que la hormiga comienza su camino y, además de almacenar la componente correspondiente en la memoria de la hormiga *L*. La decisión sobre cuál será dicho nodo depende del algoritmo específico (puede ser una elección aleatoria o una fija para toda la colonia, o una elección aleatoria o fija para cada hormiga, etc.).

Por último, cabe comentar que los procedimientos “calcular probabilidades de transición y aplicar política decisión” tienen en consideración el estado actual de la hormiga, los valores actuales de la feromona visibles en dicho nodo y las restricciones del problema para establecer el proceso de transición probabilístico hacia otros estados válidos.

## 2.3 Estrategia de investigación.

### 2.3.1 Ant System (AS)

El AS, desarrollado por Dorigo y Maniezzo<sup>28</sup>, fue el primer algoritmo de ACO. Inicialmente, se presentaron 3 variantes distintas: *AS-densidad*, *AS-cantidad* y *AS-ciclo*, que se diferenciaban en la manera en que se actualizaban los rastros de feromona. En los dos primeros, las hormigas depositaban feromona mientras que construían sus soluciones, con la diferencia de que la cantidad de feromona depositada en el *AS-densidad* es constante, mientras que la depositada en *AS-cantidad* dependía directamente de la deseabilidad heurística de la transición  $ij$ .

Por último, en *AS-ciclo*, la deposición de feromona se lleva a cabo una vez que la solución está completa (actualización en línea *a posteriori* de feromona). Esta última variante era la que obtenía mejores resultados y es por tanto la que se conoce como AS en la literatura (y en el resto de este trabajo).

El AS se caracteriza por el hecho de que la actualización de feromona se realiza una vez que todas las hormigas han completado sus soluciones, y se lleva a cabo como sigue: primero, todos los rastros de feromona se reducen en un factor constante, implementándose de esta manera la evaporación de feromona. A continuación, cada hormiga de la colonia deposita una cantidad de feromona que es función de la calidad de su solución.

Inicialmente, el AS no usaba ninguna acción del demonio<sup>29</sup>, pero es relativamente fácil, por ejemplo, añadir un procedimiento de búsqueda local para refinar las soluciones generadas por las hormigas.

---

<sup>28</sup> Dorigo M. , T. Maniezzo. The Ant System: optimization by a colony of cooperating agents. Université Libre de Bruxelles ,1996.

<sup>29</sup> las acciones del demonio, utilizadas para implementar tareas desde una perspectiva global que no pueden llevar a cabo las hormigas, por ejemplo, observar la calidad de todas las soluciones generadas y depositar una nueva cantidad de feromona adicional en las transiciones asociadas a

Las soluciones en el AS se construyen como sigue. En cada paso de construcción, una hormiga  $k$  escoge ir al siguiente nodo con una probabilidad que se calcula como:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{r \notin \Gamma} \tau_{ir}^\alpha \cdot \eta_{ij}^\beta} & \text{si } j \notin \Gamma \\ 0 & \text{de otra manera} \end{cases}$$

Donde  $\tau(ij)$  es la cantidad de rastro de feromona en el tramo  $(ij)$ , y  $\eta(ij) = \frac{1}{d_{ij}}$  es la visibilidad en función de la heurística de decisión, igual al inverso de la distancia entre las ciudades  $i$  y  $j$ ,  $\alpha$  es un parámetro que estima la importancia relativa de uso del sendero en función de la feromona depositada y  $d$  de cercanía en términos de visibilidad,  $Q$  es un valor escogido al azar que representa la cantidad de feromona que se repartirá al final de cada tour tasada por la distancia total del recorrido. Si  $\alpha = 0$ , aquellos nodos con una preferencia heurística mejor, tienen una mayor probabilidad de ser escogidos, haciendo al algoritmo muy similar a un algoritmo voraz probabilístico clásico (con múltiples puntos de partida en caso de que las hormigas estén situadas en nodos distintos al comienzo de cada iteración). Sin embargo, si  $\beta = 0$ , sólo se tienen en cuenta los rastros de feromona para guiar el proceso constructivo, lo que puede causar un rápido estancamiento, esto es, una situación en la que los rastros de feromona asociados a una solución son ligeramente superiores que el resto, provocando por tanto que las hormigas siempre construyan las mismas soluciones, normalmente óptimos locales. Por tanto, es preciso establecer una adecuada proporción entre la información heurística y la información de los rastros de feromona.

Como se ha dicho, la deposición de feromona se realiza una vez que todas las hormigas han acabado de construir sus soluciones. Primero, los rastros de feromona asociados a cada arco se evaporan reduciendo todos los rastros de feromona en un factor constante:

---

algunas soluciones. Evelyn Menéndez Alonso. Metaheurística de Optimización mediante Colonias de Hormigas y Aplicaciones. 2001

$$\tau(ij) \leftarrow (1 - \rho)\tau_{ij}$$

Donde  $\rho \in (0,1)$  es la tasa de evaporación. El siguiente paso de cada hormiga es recorrer de nuevo el camino que ha seguido (el camino está almacenado en su memoria local (lista tabú  $LT_k$ ) y deposita una cantidad de feromona  $\Delta\tau^k_{ij}$  en cada conexión por la que ha viajado:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}, \quad \forall a_{ij} \in S_k$$

Donde  $\Delta\tau^k_{ij} = f(C(S_k))$ , es decir, la cantidad de feromona que se deposita depende de la calidad  $C(S_k)$  de la solución  $S_k$  construida por la hormiga  $k$ .

Para resumir la descripción del AS, mostramos a continuación el procedimiento Nueva Hormiga para este algoritmo de ACO en particular:

1. Procedimiento Nueva Hormiga (id Hormiga)
2.  $k = id$  Hormiga;  $r =$  generar estado inicial;  $S_k = i$
3.  $LT_k = i$
4. Mientras (estado actual= estado objetivo)
5. Para cada  $S \in Nk(r)$  hacer  $p^k(i, j) = \left\{ \frac{[\tau(i, j)]^k [\eta(i, j)]^\beta}{\sum_{j \in LT_k} [\tau(i, j)]^k [\eta(i, j)]^\beta} \right\}$
6. Siguiendo esta aplicar política de decisión ( $P, Nk(i)$ )
7.  $i =$  siguiente est;  $S_k = \langle S_k, i \rangle$
8. ---
9.  $LT_k = LT_k \cup i$
10. Fin mientras {se ejecuta el procedimiento Evaporación de feromona () se lanza y evapora la feromona en cada arco  $a_{ij}$ :  $\tau_{ij} = \{(1 - \rho) + \tau(i, j)\}$
11. Para cada arco  $a_{ij} S_k$  hacer
12.  $\tau(ij) = \tau(ij) + f(c(s_k))$
13. Fin para
14. Liberar recursos hormiga (id Hormiga)
15. Fin Procedimiento.

La línea 8 vacía se incluye para hacer notar que no existe una actualización de feromona en línea paso a paso y que, antes de la línea 12, se debe haber aplicado la evaporación de feromona. De hecho, este es un ejemplo donde la construcción Programación de actividades interfiere con el funcionamiento de los procedimientos fundamentales de la metaheurística ACO, tal como se indicaba anteriormente.

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

En la figura 13, se muestra el diagrama de flujo del comportamiento del algoritmo *Ant System*:

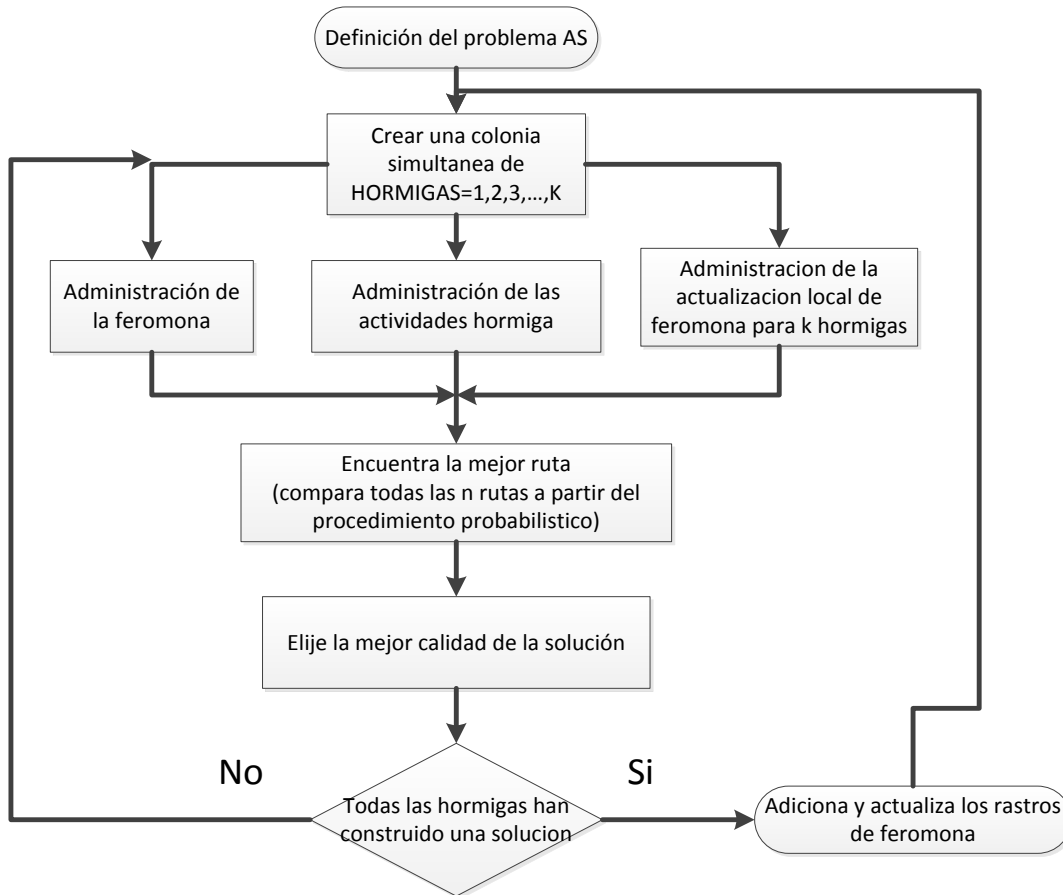


Figura 13 Diagrama de flujo del algoritmo *Ant System*<sup>30</sup>.

Una vez que se ha analizado e identificado las fortalezas del algoritmo, éste ya es susceptible de ser adaptado al problema de asignación de órdenes.

En el problema planteado, hay dos sistemas de entidades diversas: las órdenes y los componentes. Cada tipo de componente está ligado a un tipo de alimento y cada orden a un nido. Los nidos pueden requerir diversos tipos de alimento en diversas cantidades, así las órdenes requieren diversos tipos de componentes en diversas cantidades. Los diversos nidos pueden requerir el mismo tipo de alimento, mientras que diversas órdenes pueden requerir tipos comunes de componentes.

<sup>30</sup> Marco Dorigo, Thomas Stutzle, *Ant Colony Optimization*, 2004, the MIT Press.

Conceptualmente, el trabajo de las hormigas de esta colonia es muy diferente del trabajo del forrajeo descrito antes, aquí, hay  $m$  hormigas, una por fuente de alimento, El trabajo de estas hormigas es distribuir la cantidad de alimento  $q$  en las fuentes del alimento a  $n$  nidos, De aquí, la hormiga tiene que elegir con una cierta probabilidad  $p$  cuál es el nido a visitar primero y mientras que al visitar el nido una feromona  $\tau$  es depositada durante este trayecto, desde el nido hasta la fuente alimento. Cada viaje es completado cuando todas las hormigas de cada una de las fuentes de alimento han visitado todos los nidos. Cada hormiga entrega una cantidad  $q_j^i$  de la cantidad total  $q^i$  de componentes  $i \in \{1, \dots, m\}$  a una orden  $j \in \{1, \dots, n\}$ . Puesto que hay varios nidos a visitar, la hormiga elige la trayectoria a un nido particular con la probabilidad:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{r \notin \Gamma} \tau_{ir}^\alpha \cdot \eta_{ij}^\beta} & \text{si } j \notin \Gamma \\ 0 & \text{de otra manera} \end{cases}$$

Donde  $\tau_{ij}$  es la cantidad de feromona que conecta el tipo componente  $i$  a la orden  $j$ ,  $\eta_{ij}$  es la función de la visibilidad y  $\Gamma$  es la lista tabú. La lista tabú, es la lista de las órdenes que la hormiga  $k$  ha visitado ya, o las órdenes que no necesitan el tipo componente  $i$  (así la visita puede ser evitada). La función de la visibilidad  $\eta_{ij}$  se define de la manera siguiente:

$$\eta_{ij} = e^{d_j}$$

Donde  $d_j$  es retraso de la orden es decir la diferencia entre el día real y la fecha deseada, es aquí donde junto a criterio y el razonamiento de prioridad antes mencionado, (saldrán aquellas órdenes que tengan un índice de jerarquía específico, que se coloca cuando se ha anexado sobre la llegada de la misma). El objetivo de la primera función es dar a las hormigas una cierta información sobre cómo está retrasada una orden, puesto que la función objetivo del problema es no solamente satisfacer el número más alto de órdenes por día, sino también



entregarlas en el día correcto, no antes y no después. Las órdenes que se deben entregar con una fecha máxima vigente y tienen visibilidad muy cercana o igual a  $\eta_{ij}=1$ . Las órdenes que todavía pueden esperar tendrán una visibilidad más baja, y las órdenes que se retrasan ya tendrán un valor más alto, así que las hormigas se alimentarán preferiblemente de órdenes retrasadas. Los rastros de feromona indican las tentativas anteriores de otras hormigas. Los parámetros  $\alpha$  y  $\beta$  miden la importancia relativa del rastro de feromona (experiencia) con la visibilidad (conocimiento), respectivamente. Después de elegir la orden para visitar, la hormiga depositará feromona en el rastro que conecta el tipo de componentes  $i$  y las órdenes  $j$ . En este punto, la hormiga puede encontrar dos diversas situaciones: 1) si la cantidad de componentes  $q^i$  del tipo componente  $i$  no es bastante para satisfacer la cantidad  $q_j^i$  necesarias para la orden  $j$ , la hormiga no debe reforzar esta trayectoria; 2) si existen los suficientes componentes  $q^i$  para entregar a la orden  $j$ . La hormiga debe reforzar esta trayectoria; 3) la jerarquía de las mismas órdenes. La actualización local de la concentración del feromona entonces está dada por:

$$\delta_{ij}^k(t) = \begin{cases} \tau_c & \text{si } q_j^i \leq q^i \\ 0 & \text{de otra manera} \end{cases}$$

Donde  $\tau_c$  es una constante. De esta manera, las feromonas se actualizan localmente con la información sobre la posibilidad de satisfacción o no de la orden  $j$  con los componentes  $i$ . Después de que la hormiga visite la orden  $j$ . Debemos decrementar  $q^i$  por  $q_j^i$  antes que la siguiente hormiga del recurso  $i$  visite la orden  $j + 1$ . Obsérvese también, que las hormigas entregan siempre todos los  $q_j^i$  componentes que necesita la orden  $j$ , para evitar la situación donde todas las órdenes tienen parte de los componentes necesarios, pero ninguna tienen la cantidad completa. En el final de un viaje completo tendremos:

$$\Delta\tau_{ij} = \sum_{k=1}^m \delta_{ij}^k$$

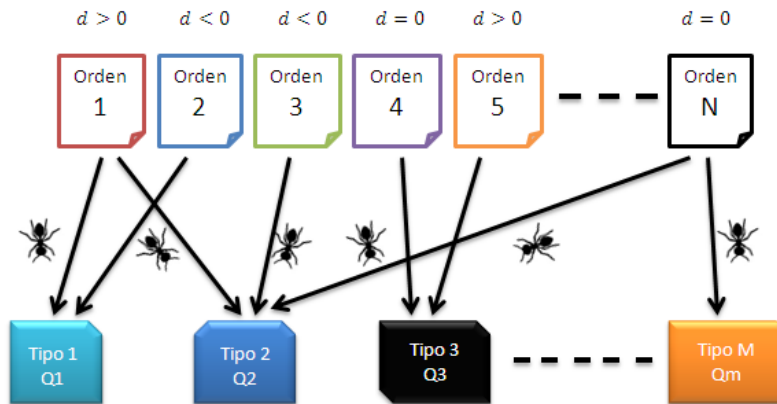
## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

En este punto, cuando todas las hormigas han entregado todos los componentes posibles, la solución se puede evaluar usando un índice del funcionamiento.

$$Z = \sum_{j=1}^n a_j \quad \text{donde } \begin{cases} a_j = 1 & \text{si } d_j = 0 \\ a_j = 0 & \text{de otra manera} \end{cases}$$

Donde  $n$  es el número de órdenes y  $d_j$  es el retraso de la orden  $j$ . Este índice se utiliza para actualizar globalmente las feromonas.

La Figura 14,<sup>31</sup> representa esquemáticamente el modelo planteado con  $n$  entradas (órdenes) y  $m$  elementos que conforman a la orden (componentes).



**Figura 14 Grafo ejemplo utilizado para la asignación de recursos sobre una colonia de hormigas aplicado a un proceso logístico con feromonas sobre los rastros.**

En cada viaje  $N$  del algoritmo (con  $t + n$  iteraciones) a  $z$  se calcula y se almacena en el sistema  $Z = z(n), \dots, z(t + 1)$ . Si  $z(t + n)$  es más alto que el anterior  $z \in Z$  significa que la solución actual ha mejorado, así que las feromonas usadas deben ser acentuadas. Si es peor, deben ser reducidas. Esto es hecho de nuevo por la actualización global de la feromona, donde  $(1 - \rho)$  el coeficiente de la evaporación evita que la solución se estanque. De esta manera, la actualización del feromona es la siguiente:

<sup>31</sup> Carlos A. Silva, Thomas A. Runkler, Optimization of logistic processes in supply-chains using metaheuristics, Siemens AG, Lisboa Portugal, 2003.

$$\tau_{ij}(t+n) = \begin{cases} \tau_{ij}(t) \times \rho + \Delta\tau_{ij} & \text{si } z(t+n) \geq \max Z \\ \tau_{ij} \times \rho & \text{de otra manera} \end{cases}$$

Este marco particular del algoritmo de las colonias de hormigas se puede también poner en ejecución usando el algoritmo general ACO descrito con anterioridad.

## Capítulo 3 Diseño e implementación del algoritmo Ant System aplicado.

### 3.1 Implementación del *scheduling* adaptado a órdenes de entrega.

En la gestión de cadenas de suministro, la logística se puede definir como el subproceso del proceso de la cadena de suministro que se ocupa con la planificación, el manejo y el control del almacenamiento de las mercancías entre el punto de producción y el punto de consumo. En el pasado, se producen los bienes, y se almacenan para ser luego ser entregadas bajo demanda. Hoy en día, en muchas empresas no funcionan estas acciones, utilizando en su lugar los centros de cross-docking.<sup>32</sup> Los productos son transportados desde los proveedores de estos centros cross-docking, almacenados, y luego se envían a los clientes. La falta de almacenamiento puede incrementar el tiempo de entrega, pero reduce considerablemente el volumen de capital invertido y aumenta la flexibilidad de la cadena de suministro. La cuestión clave es entregar las mercancías en el tiempo, reduciendo al mínimo los stocks. Las mercancías deben ser entregadas en la fecha correcta (no antes o después) con el fin de asegurar la satisfacción de clientes.

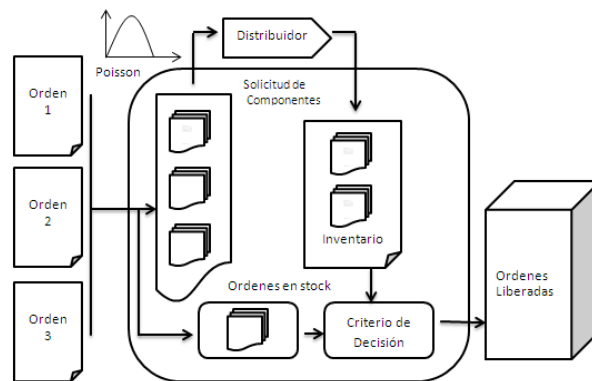


Figura 15 Representación general de un proceso logístico de distribución.<sup>33</sup>

<sup>32</sup> En Logística el **Cross-docking** corresponde a un tipo de preparación de pedido (una de las funciones del almacén logístico) sin colocación de mercancía en stock (inventario), ni operación de picking (recolección). Permite transitar materiales con diferentes destinos o consolidar mercancías provenientes de diferentes orígenes

<sup>33</sup> Carlos A. Silva, Thomas A. Runkler, Optimization of logistic processes in supply-chains using metaheuristics, Siemens AG, Lisboa Portugal, 2003.

En la Figura 15 se muestra el proceso de llegada de las ordenes (Poisson). Un posible algoritmo tiene que decidir qué bienes son entregado y a que clientes. Si las órdenes de los consumidores son conjuntos de diferentes componentes, el método más común es preasignar los componentes a las órdenes. En esta estrategia, todos los componentes son asignados a las órdenes en el momento cuando los componentes se ordenan de los proveedores. Pueden por tanto, no ser utilizados por cualquier otra orden. En esta estrategia de programación estática resulta ser pobre, Por lo tanto, una estrategia de asignación dinámica permite el intercambio de componentes entre las órdenes.

La asignación dinámica consiste en la distribución de la disposición componentes a las órdenes. Esto se puede hacer con una lista ordenada de los pedidos. Si el primer orden de la lista se puede suministrar con los componentes disponibles, los componentes se toman de la lista activa de valores y la entrega. Entonces, el algoritmo de asignación dinámica procede sobre las órdenes siguientes en la lista. La forma más fácil de ordenar esta lista es utilizar un First In First Served (FIFS), cuando las órdenes se ordenan por fecha de pedido. Puesto que ambos principios utilizan una lista de pedidos únicos, estos métodos de programación se llaman enfoques dinámicos centralizados.<sup>34</sup>

En el problema de asignación de pedidos (ordenes compra) dentro de la industria, se debe de considerar dos factores importantes dentro del surtimiento de órdenes:

- El nivel de stock.
- El día de entrega deseado por el cliente.
- La categoría (jerarquía) dada por la distribución de órdenes.

El nivel de stock o inventario afectan directamente al segundo factor, ya que si no existiese este, no estaríamos en capacidad de generar un nivel satisfactorio de entregas, acorde con las restricciones marcadas por los clientes, y que se diseñan,

---

<sup>34</sup> Peter s. Sapaty, Ruling Distributed Dinamic Worlds.. Wiley. instituto de maquinaria matemática y sistemas, Ukania.2008.

en cuanto al surtimiento de las ordenes, en este sentido podemos acotar el problema inicialmente basándonos en los siguientes criterios para encontrar la mejor solución:

1. Considerar con prioridad extrema el surtimiento de la orden completa (componentes) en el día deseado.
2. Construir el algoritmo que satisfaga la restricción anterior, así también que permita distribuir de mejor manera el inventario existente y permita realizar un pronóstico a partir de la demanda estimada en futuros plazos.

### **3.2 Metodología a seguir para resolver un problema mediante ACO (ant system).**

Observando las aplicaciones actuales de la ACO, se pueden identificar algunas directivas sobre como atacar problemas utilizando esta metaheurística. Estas directivas se pueden resumir en las seis tareas de diseño que se enumeran a continuación:

- Representar el problema como un conjunto de componentes y transiciones o a través de un grafo ponderado que será recorrido por las hormigas para construir soluciones.
- Definir de manera apropiada el significado de los rastros de feromona  $\tau$ , esto es, el tipo de decisión que inducen. Éste es un paso crucial en la implementación de un algoritmo ACO y, a menudo, una buena definición de los rastros de feromona no es una tarea trivial. De hecho, normalmente implica tener un buen conocimiento del problema que se quiere solucionar.
- Definir de manera apropiada la preferencia heurística de cada decisión que debe tomar una hormiga mientras que construye una solución, es decir, definir la información heurística  $ij$  asociada a cada componente o

transición. Hay que remarcar que la información heurística es crucial para un buen rendimiento si no existen o no pueden ser aplicados algoritmos de búsqueda local.

- Si es posible, implementar una búsqueda local eficiente para el problema que se desea solucionar, porque los resultados de muchas aplicaciones del ACO a problemas de optimización combinatoria NP-duros demuestran que el mejor rendimiento se alcanza cuando se complementa con optimizaciones locales.
- Escoger un algoritmo ACO específico y aplicarlo al problema que hay que solucionar, teniendo en cuenta, obviamente, todos los aspectos previamente comentados.
- Refinar los parámetros del algoritmo ACO. Un buen punto de partida para la especificación de los valores de los parámetros es usar configuraciones de parámetros que han demostrado ser buenas cuando se aplicaban en el algoritmo ACO a problemas similares o a una gran variedad de problemas distintos. Otra posible alternativa para evitar el esfuerzo y tiempo necesarios para esta tarea es utilizar procedimientos automáticos de refinamiento de parámetros.

Debe quedar claro que los pasos descritos sólo dan una idea a general sobre la implementación del algoritmo ACO. Además, en muchas ocasiones la implementación es un proceso iterativo, donde cuando se obtiene una visión más en profundidad del problema y del comportamiento del algoritmo, deben revisarse algunas de las decisiones iniciales. Por último, se debe insistir en el hecho de que probablemente los pasos más importantes son los cuatro primeros, ya que una elección poco acertada en esos punto no suele poder arreglarse con un buen refinamiento de parámetros.

### 3.3 Pseudocódigo ANT SYSTEM adaptado a la asignación de órdenes.

#### 3.3.1 Desarrollo del aplicativo computacional.

Dentro del análisis para la concepción del algoritmo, la manera más sencilla de entender los diferentes pasos del algoritmo es mediante un diagrama conceptual del algoritmo Ant system adaptado. Figura 15.

Algunas variables con las que se puede jugar para optimizar tiempos de ejecución son:

- Número de hormigas ( $m$ ).
- Número de ciclos (*días de prueba*).
- Cantidad máxima para la búsqueda por hormiga ( $k$ ).  
(Se expresa como un radio de avance sobre las órdenes)
- Restricciones por jerarquía (prioridad).
- Tiempo de recarga del stock.

Tan solo se cambia el valor máximo de exploración (cantidad de órdenes), ya que para las exploraciones subsiguientes, el área de búsqueda irá disminuyendo conforme vaya mejorando la función objetivo.

El esquema general del algoritmo es como se describe a continuación.

Inicialización:

- 1 Establecer para cada par  $(i, j)$ :  $\tau_{ij} = \tau_0$ .
- 2 Set  $N = 1$  y definir un  $N_{max}$ .
- 3 Colocar las hormigas  $m$ .
- 4 Mientras que  $N \leq N_{max}$ .
- 5 Construir un recorrido completo.
- 6 Para  $i = 1$  a  $n$ .
- 7 Para  $k = 1$  a  $m$ .
- 8 Seleccionar el siguiente nodo utilizando para el algoritmo general:



$$p_{ij}^k(t) = \begin{cases} f(\tau_{ij}, \eta_{ij}) & \text{si } j \notin \Gamma \\ 0 & \text{de otra manera} \end{cases} \quad \text{O bien para el algoritmo } scheduling$$

$$p_{ij}^k(t) = p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{r \in \Gamma} \tau_{ir}^\alpha \cdot \eta_{ij}^\beta} & \text{si } j \notin \Gamma \\ 0 & \text{de otra manera} \end{cases}$$

9 Actualizar localmente  $\tau_{ij}(t)$  utilizando para el algoritmo general:

$$\delta_{ij}^k = \tau_c \quad \text{o bien para el algoritmo } scheduling:$$

$$\delta_{ij}^k(t) = \begin{cases} \tau_c & \text{si } q_j^i \leq q^i \\ 0 & \text{de otra manera} \end{cases}$$

10 Actualizar la lista tabú  $\Gamma$

11 Analizar las soluciones

Para  $k = 1$

12 Cálculo del Índice de Rendimiento para el algoritmo general:

$Z: (L_k)$  O bien para el algoritmo *scheduling*:

$$Z = \sum_{j=1}^n a_j \quad \text{donde } \begin{cases} a_j = 1 & \text{si } d_j = 0 \\ a_j = 0 & \text{de otra manera} \end{cases}$$

13 Actualizar globalmente  $\tau_{ij}(t + n)$  usando para el algoritmo general:

$$\tau_{ij}(t + n) = \tau_{ij}(t) \times \rho + \Delta\tau_{ij} \quad \text{O bien para el algoritmo } scheduling:$$

$$\tau_{ij}(t + n) = \begin{cases} \tau_{ij}(t) \times \rho + \Delta\tau_{ij} & \text{si } z(t + n) \geq \max Z \\ \tau_{ij} \times \rho & \text{de otra manera} \end{cases}$$

14. Actualizar localmente el nivel de stock con las restricciones asignadas.

15. Fin Procedimiento.

En la figura 16, se muestra un diagrama de flujo que describe al algoritmo *Ant system* modificado, insertando los cambios pertinentes al objetivo trazado originalmente.

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

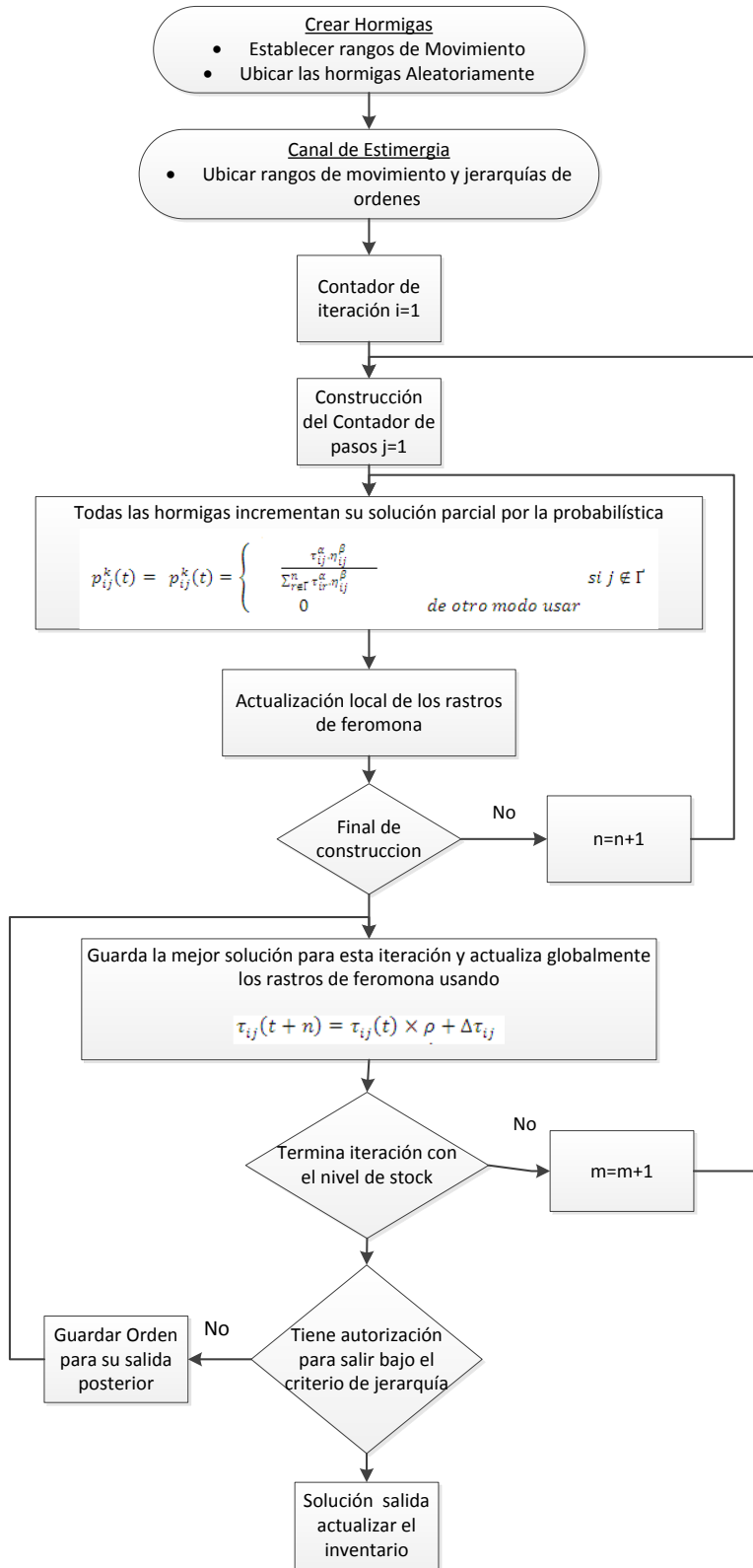


Figura 16 Diagrama de flujo del algoritmo Ants System Adaptado.

Cada paso se trabaja de la siguiente forma:

- a) Inicializar variables.
- b) Crear hormigas: las hormigas se crean al comienzo del programa y se asume que vivirán hasta encontrar la mejor solución. No mueren ni se crean nuevas hormigas.

Recuérdese que este algoritmo simula la forma en que las hormigas exploran el entorno, después de abandonar el nido, por ello la exploración inicial (sin conocimiento de fuentes de alimento) se debe hacer sobre todo el espacio a recorrer; por lo tanto las hormigas se deben ubicar aleatoriamente sobre la superficie a explorar. En este caso se dieron valores aleatorios a las posiciones de las órdenes.

- c) Ubicar las hormigas aleatoriamente, sobre las órdenes.
- d) Crear índice: aunque esta parte del programa no aparece en el algoritmo, es importante crear los índices antes de comenzar los ciclos, esto para no hacer dos funciones diferentes como por ejemplo, almacenar las tareas ya realizadas.

Los índices son simplemente una matriz que contiene información acumulada de las posiciones o coordenadas de las hormigas a través de los ciclos. Solo almacena los mejores ciclos y se actualiza llamando a dicha función y se almacena en la matriz acumulada. Es importante tener mucho cuidado al diseñar el almacenamiento de los mensajes ya que son el 50% de la opción de destino, van a emular la persistencia de feromona en los mejores destinos y la evaporación de resultados intrascendentes.

Para el programa la actualización de los mensajes se hace al terminar cada ciclo de iteraciones. La función se lleva una matriz filtrada del mejor resultado que encontró cada hormiga y los mensajes actualmente almacenados, para ordenarlos y

almacenar los mejores. Este filtro consiste en comparar el resultado que encontró la hormiga  $j$  y compararlo con el que obtuvo en el ciclo anterior, sólo almacenará su resultado si el que obtuvo es mejor que el anterior. Así se evita convergencia hacia los índices y se le da mayor acción a la estimergia.<sup>35</sup>

- e) Comienza el “for” del número de ciclos que se va a ejecutar el programa.
- f) Establecer rangos de movimiento: se estableció a partir de la cantidad de órdenes ingresadas por día. Debe ser directamente proporcional a la exponencial del número de hormigas (preferiblemente) por lo tanto se debe actualizar en cada ciclo. No es necesario hacerlo inmediatamente después de mover cada hormiga. Se determina con la función distancia y es almacenada. La función distancia le asigna a cada hormiga un porcentaje máximo de exploración que disminuye a medida que va encontrando un mejor resultado.

En teoría comienza el ciclo por hormiga, pero como requiere información de la colonia se calcula primero la información recabada por la colonia para cada hormiga. Evalúa la función de estimergia.

- g) Una vez actualizados los rastros de las  $m$  hormigas, se puede iniciar con el ciclo para mover a la hormiga, for  $j = 1:m$ .

Otra diferencia con el algoritmo original, es que la hormiga no va a hacer las dos acciones de estimergia e índice, sino que también comparará la jerarquía que ofrece la orden, para evaluar su factibilidad para su liberación.

- h) Movimiento hacia el destino: para este punto se debe tener en cuenta que se trata de una función continua, así que una forma de discretizarla es evaluar la diferencia entre el día deseado y el día que actualmente corre

---

<sup>35</sup> La Estimergia es la comunicación entre las hormigas por medio del depósito de feromona en el terreno sobre el que se mueven.

dentro del sistema, y evaluar la función a cada incremento sobre el delta de la actualización del rastro. Si se encuentra una mejor aptitud se almacena.

- i) Al terminar el avance, el mejor resultado encontrado por la hormiga  $j$  se almacenará en la matriz, que al terminar el ciclo tendrá acumulado el mejor resultado de las  $m$  hormigas.
- j) Ahora se debe actualizar la matriz de restricciones para elegir la orden con mayor nivel prioritario de manera óptima, una forma es como se dijo anteriormente evitar almacenar resultados repetidos o inferiores generados por la misma hormiga, para lo cual se compara su mejor resultado con el resultado obtenido por la misma hormiga en un ciclo anterior; si el resultado es menor al anterior, se almacenará un valor de -1 en la nueva matriz, de lo contrario el valor obtenido.
- k) Una vez que se ha finalizado el ciclo es importante indagar si hemos alcanzado las fechas señaladas para el aprovisionamiento del inventario, para generar la posibilidad del surtimiento de nuevas órdenes.
- l) Fin del ciclo.

### 3.4. Diseño experimental.

Las pruebas preliminares fueron desarrolladas en PC con Sistema Operativo Windows XP, procesador Intel de 1.7 GHz y 1GB de memoria RAM.

Para la construcción del algoritmo se utilizó el lenguaje Java:

- Características del lenguaje Java.

Java es un lenguaje de programación originalmente desarrollado por James Gosling de Sun Microsystems y publicado en el 1995 como un componente fundamental de

la plataforma Java de Sun Microsystems. El lenguaje deriva mucho de su sintaxis de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Java es un lenguaje de programación de propósito general, concurrente, basado en clases, y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

Para la implantación conceptual del algoritmo, se logró conseguir una distribución más o menos congruente con el objetivo principal planteado en esta tesis.

## Capítulo 4. Ejemplo de aplicación.

En este capítulo se presentaran los resultados experimentales obtenidos al aplicar el algoritmo implementado sobre la situación de una empresa que llamaremos “Money”, donde su campo de trabajo es la distribución y comercialización de equipos para la industria bancaria y soporte a los mismos. En este trabajo, nos focalizamos más que nada en el análisis computacional de los datos de la empresa. En la actualidad en el departamento de soporte se trabaja a través del mantenimiento a los equipos que se distribuyen, vislumbrando que estos equipos tienen prioridad por el área donde se trabajan, en ese sentido es importante contemplar que el nivel de inventario deberá ser suficiente para cumplir las expectativas de los clientes y también los compromisos temporales adquiridos para la reparación de estos equipos, en este departamento para realizar el aprovisionamiento de sus refacciones, Money aplica una asignación dinámica (preasignamiento), de tipo “FIFO”, donde su inventario se actualiza una vez que el nivel baja por debajo de ciertos límites de manera global (número de artículos totales, independientemente del tipo). Esto representa un gran problema por no contemplar la demanda, ni llevar un registro preciso y detallado de cuales refacciones tienen una gran solicitud, es aquí donde se aplicará el algoritmo propuesto, con el objeto de poder corregir estas deficiencias.

Con base en el modelo planteado el siguiente paso será la creación del programa que ejemplifica dichas consideraciones sobre el algoritmo *ANT SYSTEM* modificado.

Como un primer paso necesitamos delimitar el tiempo de proyección a un periodo determinado a través de la demanda, datos proporcionados por la empresa. Además se trabaja sobre órdenes de trabajo, Esta empresa utiliza muchos componentes, en un primer acercamiento solo se utilizarán 7 componentes de una población 24 elementos para esta primera versión, cada uno de estos componentes tendrán cierto tiempo de reabastecimiento con el objeto de indagar precisamente cuando y en qué momento se llevaría a cabo esta acción, seguido a esto y tomando

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

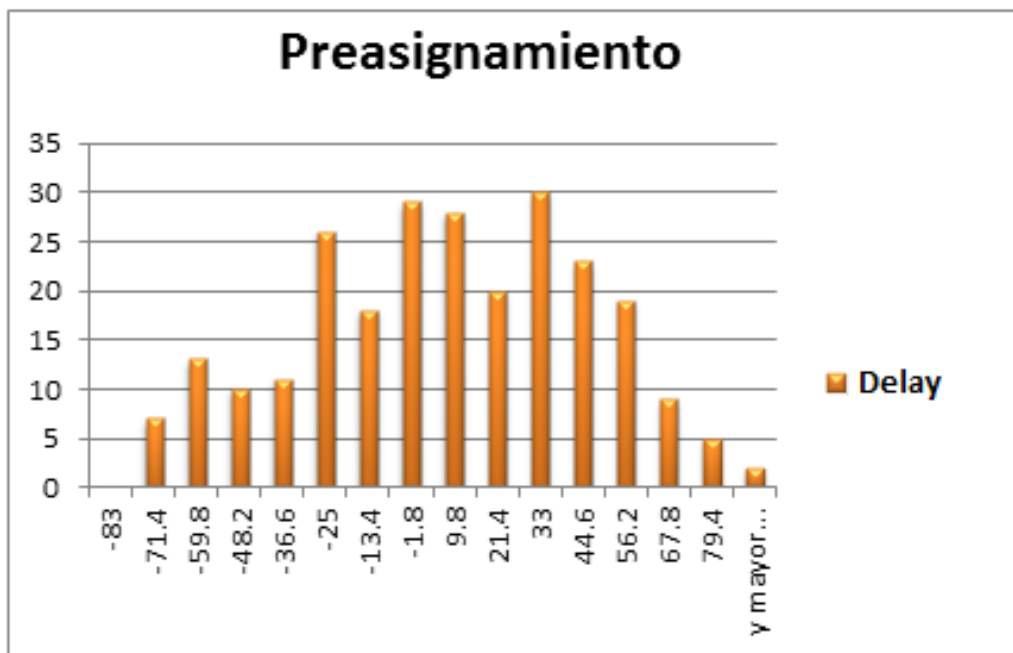
como referencia la demanda de órdenes para este mismo periodo, y como resultado, determinar un parámetro de la efectividad que tiene la forma en que se han estado trabajando las ordenes.

**Tabla 2 Función de demanda.**

<b>Demanda</b>	<b>Mes 1</b>	<b>Mes 2</b>	<b>Mes 3</b>	<b>Mes 4</b>	<b>Mes 5</b>
<b>Comp1</b>	7	4	5	12	1
<b>Comp2</b>	7	5	1	3	6
<b>Comp3</b>	4	2	6	2	2
<b>Comp4</b>	2	4	8	6	1
<b>Comp5</b>	11	3	4	8	4
<b>Comp6</b>	8	5	1	4	2
<b>Comp7</b>	4	9	2	9	8

Dentro de las consideraciones atribuidas al proceso en los capítulos anteriores se describió los pasos para la instauración del algoritmo desarrollado con el objeto de ser implantado (creado) mediante el lenguaje Java.

En la Tabla 2 los datos se muestran con la información recabada de la empresa Money, con el siguiente historial. (Anexo A).



**Figura 17 Histograma en el modo de preasignamiento.**



En la Tabla 2 se observa la distribución que guardan las órdenes liberadas basándose en la metodología de preasignamiento de la empresa “Money”, en la figura 17, se observa que no existe una tendencia clara acerca del retraso

(delay  $d$ ), ya que si bien la minoría de las ordenes liberadas se encuentran en la porción central, estas no conservan una predisposición clara que indique que esta metodología puede ser aplicable de manera general para la mayoría de las situaciones aplicables a la empresa. Este es un aspecto importante que se pretende corregir con la implantación del algoritmo sobre la presente tesis.

#### 4.1 Estudio.

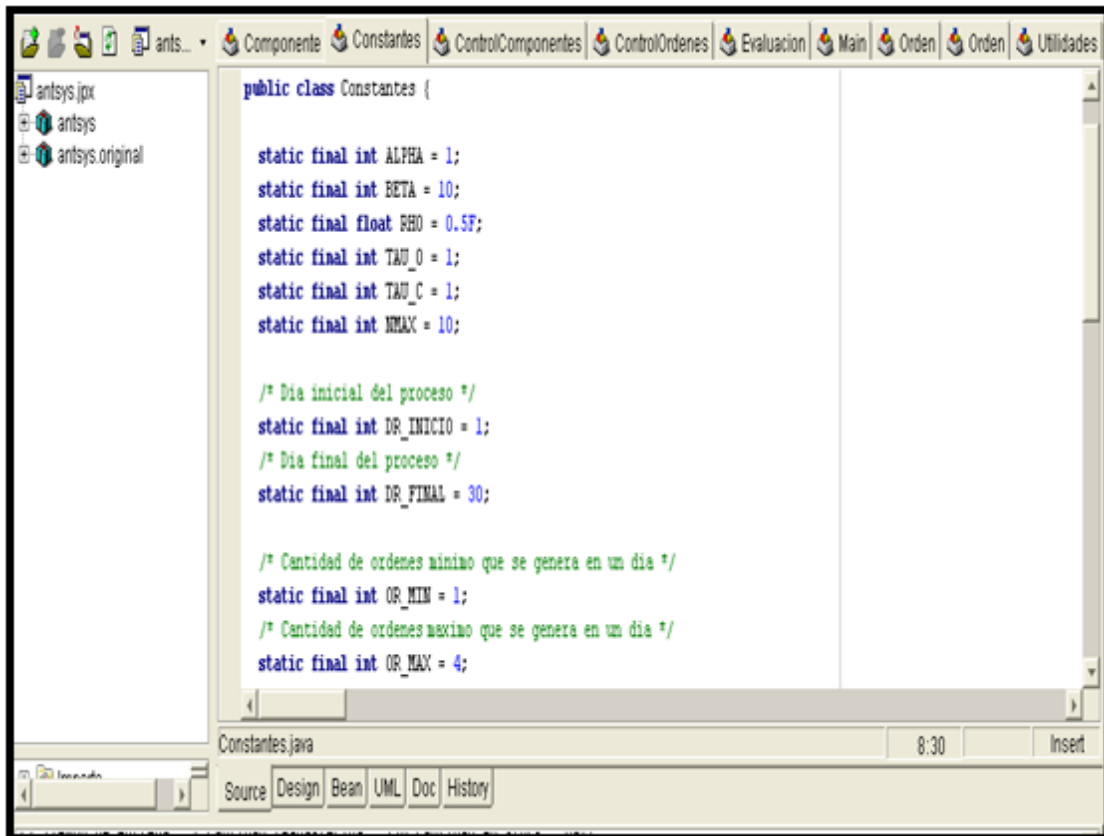
En este problema, hay cuatro diversos sistemas de entidades: las órdenes, los componentes, el día sugerido de salida de la orden y la prioridad de la misma. El algoritmo creado pretende obtener una programación de salida de órdenes, considerando que cada orden está conformada de varios tipos de componentes, si se pretende minimizar el inventario (*stock*) por los efectos que este tiene, Será necesario empezar con una cierta cantidad de componentes, de varios tipos, claro que esto limita la capacidad de entrega de cierta cantidad de órdenes, por lo cual hace necesario que los componentes cada cierto tiempo, sean restablecidos (a través de un proveedor) con la posibilidad de verificar la disponibilidad de la liberación de las ordenes que se han retenido y de estar listas para su entrega oportuna en tiempo y forma.

Ahora la siguiente restricción contemplada es cuál de estas órdenes, que están listas para salir, tendrán prioridad sobre otras, esto estará en términos de una designación que se le hace a cada una de las ordenes al momento de ingresar al sistema, esto es el día deseado que cada cliente le da a su orden con el objeto de aproximarnos de manera correcta a un retardo cero y no retardar la calendarización, y en tercer punto una vez que se han liberado es imprescindible elegir de las restantes aquella que está más próxima a salir, una vez que se ha actualizado el nivel de inventario en un tiempo posterior y existen suficientes elementos

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

(componentes) para cumplir esta última restricción. Por otro lado, cada tipo de componente está ligado a un tipo de fuente de alimento, hablando ya del comportamiento con las hormigas y de cada orden para un nido. Los nidos pueden requerir diversos tipos de alimento en diversas cantidades, así las órdenes requieren diversos tipos de componentes en diversas cantidades. Diversos nidos pueden requerir el mismo tipo de alimento, mientras que diversas órdenes pueden requerir tipos comunes de componentes. Conceptualmente, el trabajo de las hormigas de esta colonia es muy diferente del trabajo de búsqueda del alimento (forrajeo) descrito antes.

Partiremos de un supuesto ideal de aproximadamente 30 días de análisis como se muestra Figura 18, dentro de las condiciones preestablecidas en el programa.

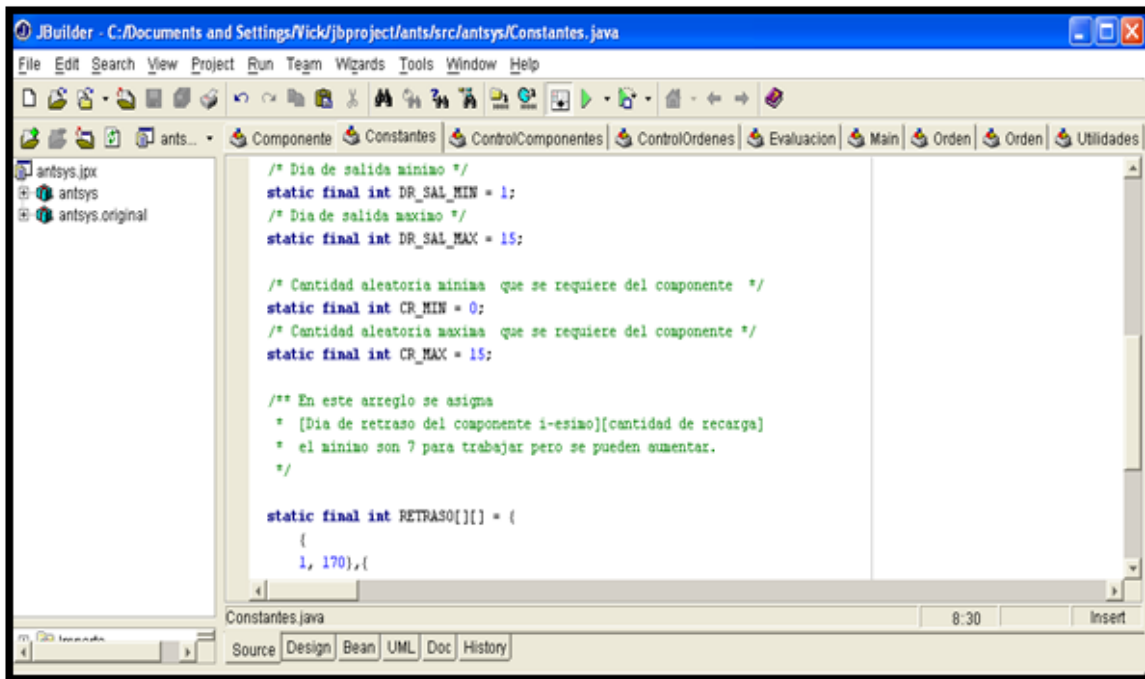


```
public class Constantes {  
  
    static final int ALPHA = 1;  
    static final int BETA = 10;  
    static final float RHO = 0.5F;  
    static final int TAU_0 = 1;  
    static final int TAU_C = 1;  
    static final int NMAX = 10;  
  
    /* Dia inicial del proceso */  
    static final int DR_INICIO = 1;  
    /* Dia final del proceso */  
    static final int DR_FINAL = 30;  
  
    /* Cantidad de ordenes minimo que se genera en un dia */  
    static final int OR_MIN = 1;  
    /* Cantidad de ordenes maximo que se genera en un dia */  
    static final int OR_MAX = 4;  
  
}
```

Figura 18 Parámetros iniciales del algoritmo.

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

En la figura 19, se observa que se puede delimitar el inicio y el termino del proceso a evaluar, el siguiente parámetro a considerar son la capacidad por orden de entrega, cada una de ellas está conformada de 7 elementos o componentes, con la posibilidad de poderse ampliar o reducir, cada orden tiene un frecuencia de aparición diaria de 1 a 4 órdenes tomadas de la muestra en la empresa, que al igual se puede modificar, otra característica importante a resaltar es la cantidad mínima y máxima que se requiere de un componente o elemento dado generado para surtirse en el día propuesto; Este parámetro es de suma importancia ya que se requiere delimitar cual es la demanda que se tiene para un periodo propuesto de cada una de las ordenes.



```
/* Dia de salida minimo */
static final int DR_SAL_MIN = 1;
/* Dia de salida maximo */
static final int DR_SAL_MAX = 15;

/* Cantidad aleatoria minima que se requiere del componente */
static final int CR_MIN = 0;
/* Cantidad aleatoria maxiaa que se requiere del componente */
static final int CR_MAX = 15;

/** En este arreglo se asigna
 * [Dia de retraso del componente i-esimo][cantidad de recarga]
 * el minimo son 7 para trabajar pero se pueden aumentar.
 */

static final int RETRASO[][] = {
    {
        1, 170},{
```

Figura 19 Parámetros iniciales del algoritmo.

Y por último tenemos los periodos en qué cada componente es solicitado al proveedor para ser regenerado el nivel de stock en una fase, estos periodos pueden ser ajustados, así también las cantidades solicitadas, esto se observa en la Figura 20.

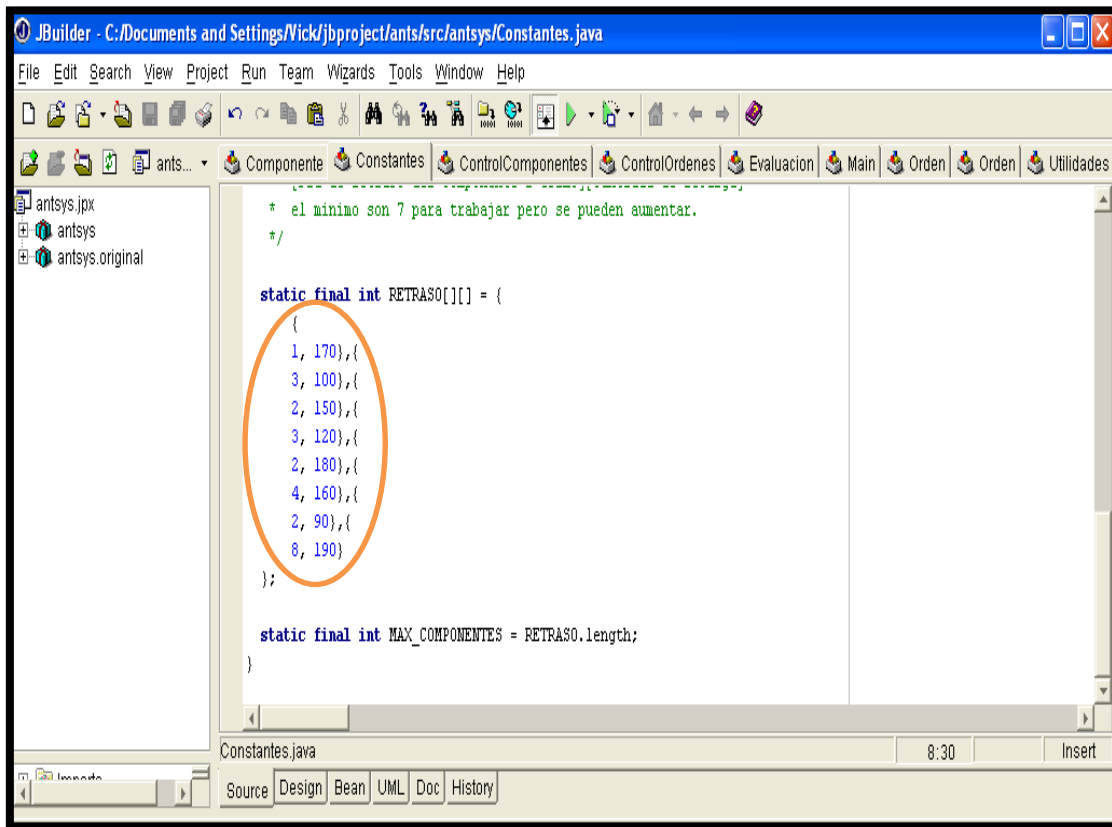


Figura 20 Tiempos de resurtimiento para 7 componentes.

## 4.2 Resultados

En la fase de análisis de resultados, se realizaron tres corridas para distintos escenarios con el objeto de verificar la viabilidad del algoritmo propuesto bajo las restricciones de la entrada y un análisis de sensibilidad con respecto a los parámetros inherentes al algoritmo en este caso  $\beta$ . Que como se ha comentado este parámetro es el que modifica la convergencia del mismo. En esta etapa se ha modificado el parámetro  $\beta$  (20) sobre la función de probabilidad, con lo cual se busca identificar el comportamiento del algoritmo con respecto a la convergencia del tiempo de resultados.

Corrida 1.

Los parámetros del algoritmo Ant System Tabla 3.

**Tabla 3 Descripción de los parámetros iniciales.**

$\beta$	20
$\rho$	0.5
$\tau$	1
$\alpha$	1
$N_{max}$	10

Los parámetros de restricciones se observan en la Tabla 3.

**Tabla 4 Descripción de las condiciones iniciales utilizadas.**

Intervalo periodo	Inicio	1	
	Final	30	
Cantidad componentes	Mínimo	0	
	Máximo	15	
Resurtimiento	Componentes	Días	cantidad
	C1	7	17
	C2	3	10
	C3	2	5
	C4	3	12
	C5	2	18
	C6	4	16
	C7	2	9
	C8	8	19
Cantidad ordenes	Mínimo	0	
	Máximo	4	

En la Figura 12 Se observan los resultados obtenidos.

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

```

Dia deseado :0
Dia de entrada :7
Dia de salida :25
C1 : 2 C2 : 5 C3 : 10 C4 : 9 C5 : 9 C6 : 11 C7 : 0 C8 : 3
Numero de orden :7
Dia deseado :5
Dia de entrada :1
Dia de salida :29
C1 : 8 C2 : 11 C3 : 11 C4 : 1 C5 : 0 C6 : 0 C7 : 8 C8 : 13
Numero de orden :32
Dia deseado :24
Dia de entrada :11
Dia de salida :29
C1 : 11 C2 : 11 C3 : 3 C4 : 11 C5 : 12 C6 : 9 C7 : 14 C8 : 0
Componentes en el dia : 30
C1 Tiempo de entrega : 7 Cantidad reabastecida : 17 Cantidad en stock : 0
C2 Tiempo de entrega : 3 Cantidad reabastecida : 10 Cantidad en stock : 39
C3 Tiempo de entrega : 2 Cantidad reabastecida : 5 Cantidad en stock : 5
C4 Tiempo de entrega : 3 Cantidad reabastecida : 12 Cantidad en stock : 37
C5 Tiempo de entrega : 2 Cantidad reabastecida : 18 Cantidad en stock : 177
C6 Tiempo de entrega : 4 Cantidad reabastecida : 16 Cantidad en stock : 14
C7 Tiempo de entrega : 2 Cantidad reabastecida : 9 Cantidad en stock : 59
C8 Tiempo de entrega : 8 Cantidad reabastecida : 19 Cantidad en stock : 5
Process finished.

```

Figura 21 Stock de salida.

Como se observa en la corrida generada a través del algoritmo se produce un ordenamiento en términos del criterio del día deseado y la suficiencia del stock a través del aprovisionamiento que se produce a partir del tiempo indicado (30 días) en las restricciones. También se muestra la cantidad de componentes incorporados a la orden solicitada para el día de salida como medida clave, en cuanto al tiempo este se estabilizó en 2 min 35 seg.

```

Numero de orden :17
Dia deseado :13
Dia de entrada :6
Dia de salida :21
C1 : 0 C2 : 2 C3 : 13 C4 : 3 C5 : 9 C6 : 14 C7 : 6 C8 : 10
Numero de orden :11
Dia deseado :11
Dia de entrada :4
Dia de salida :22
C1 : 13 C2 : 7 C3 : 2 C4 : 14 C5 : 10 C6 : 12 C7 : 1 C8 : 4
Numero de orden :20
Dia deseado :8
Dia de entrada :7
Dia de salida :25
C1 : 2 C2 : 5 C3 : 10 C4 : 9 C5 : 9 C6 : 11 C7 : 0 C8 : 3
Numero de orden :7
Dia deseado :5
Dia de entrada :1
Dia de salida :29
C1 : 8 C2 : 11 C3 : 11 C4 : 1 C5 : 0 C6 : 0 C7 : 8 C8 : 13
Numero de orden :32
Dia deseado :24
Dia de entrada :11
Dia de salida :29
C1 : 11 C2 : 11 C3 : 3 C4 : 11 C5 : 12 C6 : 9 C7 : 14 C8 : 0

```

Figura 22 Datos parciales generados para corrida 1

Corrida 2.

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

Parámetros del algoritmo Ant System:

**Tabla 5 Descripción de los parámetros iniciales.**

$\beta$	10
$\rho$	0.5
$\tau$	1
$\alpha$	1
$N_{max}$	10

Parámetros de restricciones:

**Tabla 6 Descripción de las condiciones iniciales utilizadas.**

Intervalo periodo	Inicio	Final	
		1	
		30	
Cantidad componentes	Mínimo	Máximo	
		0	
		15	
Resurtimiento	Componentes	Días	cantidad
	C1	7	17
	C2	3	10
	C3	12	5
	C4	13	12
	C5	8	18
	C6	14	16
	C7	20	9
	C8	6	19
Cantidad ordenes	Mínimo	Máximo	
		0	
		4	

Resultados obtenidos:

		Ordenes Liberadas							
Numero de orden :10	Día deseado :4	Día de entrada :3	Día de salida :4						
Numero de orden :5	Día deseado :9	Día de entrada :1	Día de salida :14						
Numero de orden :2	Día deseado :8	Día de entrada :1	Día de salida :21						
Numero de orden :21	Día deseado :22	Día de entrada :7	Día de salida :25						
Numero de orden :7	Día deseado :8	Día de entrada :1	Día de salida :27						
Componentes en el día : 30									
Numero de orden :10	C1 : 7	C2 : 3	C3 : 0	C4 : 11	C5 : 0	C6 : 10	C7 : 1	C8 : 11	
Numero de orden :5	C1 : 0	C2 : 0	C3 : 3	C4 : 9	C5 : 8	C6 : 3	C7 : 0		
Numero de orden :2	C1 : 2	C2 : 2	C3 : 4	C4 : 0	C5 : 6	C6 : 8	C7 : 9		
Numero de orden :21	C1 : 6	C2 : 14	C3 : 5	C4 : 3	C5 : 7	C6 : 0	C7 : 2	C8 : 7	
Numero de orden :7	C1 : 1	C2 : 11	C3 : 0	C4 : 8	C5 : 12	C6 : 8	C7 : 3	C8 : 2	
Cantidad reabastecida :									
C1 Tiempo de entrega : 7	Cantidad reabastecida : 17		Cantidad en stock : 69						
C2 Tiempo de entrega : 3	Cantidad reabastecida : 10		Cantidad en stock : 80						
C3 Tiempo de entrega : 12	Cantidad reabastecida : 5		Cantidad en stock : 3						
C4 Tiempo de entrega : 13	Cantidad reabastecida : 12		Cantidad en stock : 5						
C5 Tiempo de entrega : 8	Cantidad reabastecida : 18		Cantidad en stock : 39						
C6 Tiempo de entrega : 14	Cantidad reabastecida : 16		Cantidad en stock : 19						
C7 Tiempo de entrega : 20	Cantidad reabastecida : 9		Cantidad en stock : 3						
C8 Tiempo de entrega : 6	Cantidad reabastecida : 19		Cantidad en stock : 94						

**Figura 23 Stock de Salida para corrida 2.**

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

Para esta corrida a diferencia de la anterior se han incrementado los días de Resurtimiento para los componentes, y como se podría esperar la cantidad de órdenes liberadas se ha reducido notoriamente, aunque si bien existen en stock cierta cantidad considerable de algunos componentes para un tipo, hay otros que debido a su escasa presencia no fue posible la liberación de suficientes órdenes.

Corrida 3:

Esta tercera corrida se ha incrementado la cantidad de unidades en cada componente.

Parámetros del algoritmo Ant System:

**Tabla 7 Descripción de los parámetros iniciales.**

$\beta$	10
$\rho$	0.5
$\tau$	1
$\alpha$	1
$N_{max}$	10

Parámetros de restricciones:

**Tabla 8 Descripción de las condiciones iniciales utilizadas.**

Intervalo periodo	Inicio	1	
	Final	30	
<b>Cantidad componentes</b>	Mínimo	0	
	Máximo	30	
<b>Resurtimiento</b>	Componentes	Días	cantidad
	C1	2	173
	C2	3	100
	C3	2	50
	C4	3	120
	C5	2	180
	C6	4	160
	C7	2	90
	C8	8	190
<b>Cantidad ordenes</b>	Mínimo	0	
	Máximo	2	



## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

Resultados obtenidos:

En este tercer caso, dentro de las restricciones se ha reducido notoriamente los días de Resurtimiento de los componentes y se han incrementado sustancialmente la cantidad de los mismos, así también se han aumentado la demanda de componentes como resultado se observa que ha habido una cantidad mucho mayor de órdenes surtidas durante el mismo periodo de tiempo, y se nota que la cantidad de componentes ha tenido un notorio incremento, con lo cual, se hace pesar en que el incremento no es lo ideal para el resultado que se espera encontrar. Todo esto se observa en las tablas parciales (hasta el día 14) que a continuación se presentan.

Ordenes Liberadas			
Numero de orden :8	Día deseado :4	Día de entrada :2	Día de salida :4
Numero de orden :11	Día deseado :6	Día de entrada :4	Día de salida :6
Numero de orden :2	Día deseado :8	Día de entrada :1	Día de salida :8
Numero de orden :17	Día deseado :9	Día de entrada :8	Día de salida :9
Numero de orden :5	Día deseado :9	Día de entrada :1	Día de salida :9
Numero de orden :3	Día deseado :10	Día de entrada :1	Día de salida :10
Numero de orden :7	Día deseado :12	Día de entrada :1	Día de salida :12
Numero de orden :21	Día deseado :13	Día de entrada :10	Día de salida :13
Numero de orden :10	Día deseado :13	Día de entrada :3	Día de salida :13
Numero de orden :25	Día deseado :14	Día de entrada :12	Día de salida :14
Numero de orden :19	Día deseado :14	Día de entrada :9	Día de salida :14
Numero de orden :14	Día deseado :14	Día de entrada :6	Día de salida :14
Numero de orden :12	Día deseado :14	Día de entrada :4	Día de salida :14

Numero de orden :8	C1 : 5	C2 : 8	C3 : 17	C4 : 16	C5 : 17	C6 : 13	C7 : 12	C8 : 18
Numero de orden :11	C1 : 11	C2 : 1	C3 : 2	C4 : 1	C5 : 6	C6 : 12	C7 : 7	C8 : 7
Numero de orden :2	C1 : 2	C2 : 2	C3 : 4	C4 : 0	C5 : 6	C6 : 8	C7 : 9	
Numero de orden :17	C1 : 19	C2 : 8	C3 : 2	C4 : 5	C5 : 11	C6 : 10	C7 : 6	C8 : 11
Numero de orden :5	C1 : 0	C2 : 0	C3 : 3	C4 : 9	C5 : 8	C6 : 3	C7 : 0	
Numero de orden :3	C1 : 1	C2 : 3	C3 : 2	C4 : 7	C5 : 8	C6 : 10	C7 : 0	
Numero de orden :7	C1 : 13	C2 : 15	C3 : 1	C4 : 16	C5 : 7	C6 : 11	C7 : 16	C8 : 2
Numero de orden :21	C1 : 0	C2 : 7	C3 : 19	C4 : 8	C5 : 17	C6 : 10	C7 : 4	C8 : 10
Numero de orden :10	C1 : 19	C2 : 11	C3 : 15	C4 : 8	C5 : 15	C6 : 8	C7 : 10	C8 : 20
Numero de orden :25	C1 : 9	C2 : 14	C3 : 7	C4 : 15	C5 : 20	C6 : 3	C7 : 1	C8 : 18
Numero de orden :19	C1 : 10	C2 : 12	C3 : 19	C4 : 9	C5 : 16	C6 : 5	C7 : 0	C8 : 12
Numero de orden :14	C1 : 4	C2 : 5	C3 : 19	C4 : 0	C5 : 2	C6 : 19	C7 : 13	C8 : 5
Numero de orden :12	C1 : 2	C2 : 18	C3 : 15	C4 : 5	C5 : 2	C6 : 3	C7 : 19	C8 : 18

Componentes en el día : 30		
C1 Tiempo de entrega : 2	Cantidad reabastecida : 173	Cantidad en stock : 2385
C2 Tiempo de entrega : 3	Cantidad reabastecida : 100	Cantidad en stock : 730
C3 Tiempo de entrega : 2	Cantidad reabastecida : 50	Cantidad en stock : 389
C4 Tiempo de entrega : 3	Cantidad reabastecida : 120	Cantidad en stock : 880
C5 Tiempo de entrega : 2	Cantidad reabastecida : 180	Cantidad en stock : 2490
C6 Tiempo de entrega : 4	Cantidad reabastecida : 160	Cantidad en stock : 876
C7 Tiempo de entrega : 2	Cantidad reabastecida : 90	Cantidad en stock : 989
C8 Tiempo de entrega : 8	Cantidad reabastecida : 190	Cantidad en stock : 392

Figura 24 Datos parciales generados corrida 3

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

De las corridas anteriores podemos concluir que el algoritmo y su programación nos ayuda para tener una aproximación acerca de las predicciones que se tendrían que hacer para generar un resurtimiento apropiado a los niveles de stock fijados en un principio dentro de la empresa.

Una vez evaluados los tres escenarios anteriores, se procede a capturar los datos proporcionados por la empresa, con los cuales se elabora el siguiente histograma, figura 16.

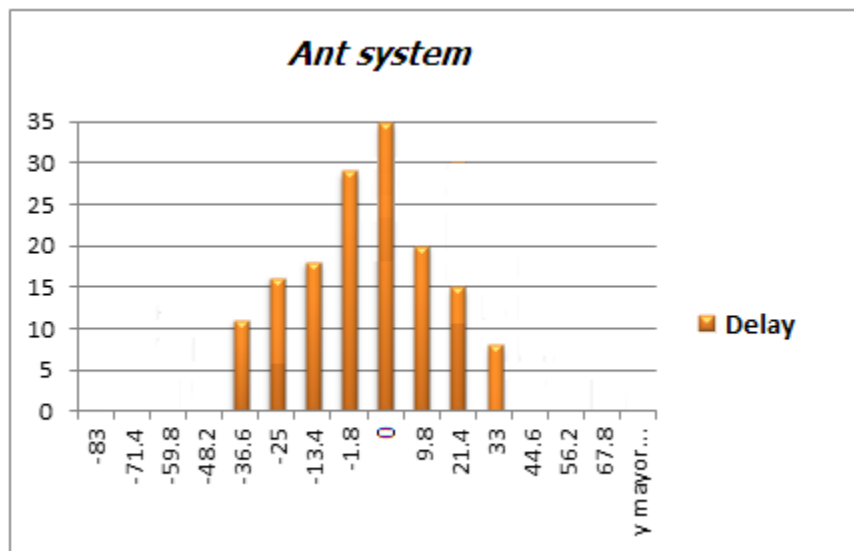


Figura 25 Histograma de resultados para el algoritmo Ant System Aplicado.

Órdenes Liberadas											
Numero de orden :14	Dia deseado :5	Dia de entrada :4	Dia de salida :5	C1 : 2	C2 : 8	C3 : 3	C4 : 2	C5 : 4	C6 : 5	C7 : 4	C8 : 7
Numero de orden :19	Dia deseado :8	Dia de entrada :6	Dia de salida :8	C1 : 0	C2 : 4	C3 : 4	C4 : 0	C5 : 0	C6 : 7	C7 : 8	C8 : 6
Numero de orden :11	Dia deseado :8	Dia de entrada :3	Dia de salida :8	C1 : 5	C2 : 1	C3 : 8	C4 : 5	C5 : 8	C6 : 4	C7 : 4	C8 : 8
Numero de orden :2	Dia deseado :8	Dia de entrada :1	Dia de salida :8	C1 : 2	C2 : 2	C3 : 4	C4 : 0	C5 : 6	C6 : 8	C7 : 9	
Numero de orden :5	Dia deseado :9	Dia de entrada :1	Dia de salida :9	C1 : 0	C2 : 0	C3 : 3	C4 : 9	C5 : 8	C6 : 3	C7 : 0	
Numero de orden :3	Dia deseado :10	Dia de entrada :1	Dia de salida :10	C1 : 1	C2 : 3	C3 : 2	C4 : 7	C5 : 8	C6 : 10	C7 : 0	
Numero de orden :8	Dia deseado :10	Dia de entrada :2	Dia de salida :11	C1 : 2	C2 : 3	C3 : 6	C4 : 4	C5 : 0	C6 : 5	C7 : 6	C8 : 2
Numero de orden :25	Dia deseado :13	Dia de entrada :9	Dia de salida :13	C1 : 8	C2 : 1	C3 : 2	C4 : 4	C5 : 2	C6 : 6	C7 : 1	C8 : 4
Numero de orden :29	Dia deseado :12	Dia de entrada :10	Dia de salida :15	C1 : 7	C2 : 1	C3 : 7	C4 : 2	C5 : 7	C6 : 4	C7 : 7	C8 : 5
Numero de orden :15	Dia deseado :12	Dia de entrada :4	Dia de salida :17	C1 : 2	C2 : 7	C3 : 6	C4 : 4	C5 : 0	C6 : 2	C7 : 3	C8 : 2
Numero de orden :13	Dia deseado :14	Dia de entrada :3	Dia de salida :19	C1 : 3	C2 : 0	C3 : 4	C4 : 2	C5 : 5	C6 : 4	C7 : 3	C8 : 6
Numero de orden :32	Dia deseado :18	Dia de entrada :11	Dia de salida :19	C1 : 7	C2 : 2	C3 : 1	C4 : 0	C5 : 2	C6 : 6	C7 : 8	C8 : 8
Numero de orden :1	Dia deseado :14	Dia de entrada :1	Dia de salida :21	C1 : 3	C2 : 5	C3 : 5	C4 : 8	C5 : 2	C6 : 6	C7 : 8	
Numero de orden :4	Dia deseado :15	Dia de entrada :1	Dia de salida :23	C1 : 11	C2 : 0	C3 : 5	C4 : 6	C5 : 0	C6 : 0	C7 : 0	
Numero de orden :22	Dia deseado :16	Dia de entrada :7	Dia de salida :25	C1 : 7	C2 : 1	C3 : 3	C4 : 2	C5 : 8	C6 : 6	C7 : 7	C8 : 8
Numero de orden :34	Dia deseado :19	Dia de entrada :13	Dia de salida :25	C1 : 7	C2 : 5	C3 : 2	C4 : 4	C5 : 5	C6 : 8	C7 : 2	C8 : 6
Numero de orden :37	Dia deseado :26	Dia de entrada :14	Dia de salida :26	C1 : 5	C2 : 4	C3 : 0	C4 : 7	C5 : 7	C6 : 8	C7 : 5	C8 : 4
Numero de orden :28	Dia deseado :17	Dia de entrada :10	Dia de salida :27	C1 : 1	C2 : 0	C3 : 3	C4 : 0	C5 : 7	C6 : 1	C7 : 1	C8 : 3
Numero de orden :45	Dia deseado :21	Dia de entrada :18	Dia de salida :27	C1 : 2	C2 : 8	C3 : 2	C4 : 5	C5 : 3	C6 : 8	C7 : 8	C8 : 6
Numero de orden :40	Dia deseado :20	Dia de entrada :15	Dia de salida :29	C1 : 2	C2 : 4	C3 : 3	C4 : 0	C5 : 5	C6 : 1	C7 : 8	C8 : 2
Numero de orden :12	Dia deseado :22	Dia de entrada :3	Dia de salida :29	C1 : 1	C2 : 2	C3 : 1	C4 : 2	C5 : 3	C6 : 6	C7 : 5	C8 : 3
Numero de orden :44	Dia deseado :24	Dia de entrada :17	Dia de salida :29	C1 : 3	C2 : 8	C3 : 1	C4 : 3	C5 : 5	C6 : 3	C7 : 0	C8 : 6
Numero de orden :47	Dia deseado :22	Dia de entrada :19	Dia de salida :31	C1 : 7	C2 : 3	C3 : 4	C4 : 5	C5 : 0	C6 : 4	C7 : 8	C8 : 5
Numero de orden :30	Dia deseado :29	Dia de entrada :10	Dia de salida :31	C1 : 5	C2 : 1	C3 : 1	C4 : 3	C5 : 4	C6 : 1	C7 : 8	C8 : 4
Numero de orden :67	Dia deseado :31	Dia de entrada :27	Dia de salida :31	C1 : 1	C2 : 8	C3 : 0	C4 : 4	C5 : 2	C6 : 3	C7 : 8	C8 : 8
Numero de orden :39	Dia deseado :24	Dia de entrada :15	Dia de salida :33	C1 : 4	C2 : 1	C3 : 3	C4 : 4	C5 : 3	C6 : 8	C7 : 1	C8 : 1
Numero de orden :50	Dia deseado :25	Dia de entrada :20	Dia de salida :33	C1 : 5	C2 : 6	C3 : 2	C4 : 3	C5 : 2	C6 : 0	C7 : 6	C8 : 6
Numero de orden :41	Dia deseado :33	Dia de entrada :16	Dia de salida :33	C1 : 5	C2 : 4	C3 : 0	C4 : 3	C5 : 5	C6 : 0	C7 : 3	C8 : 0

Figura 26 Datos parciales generados corrida 3.

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

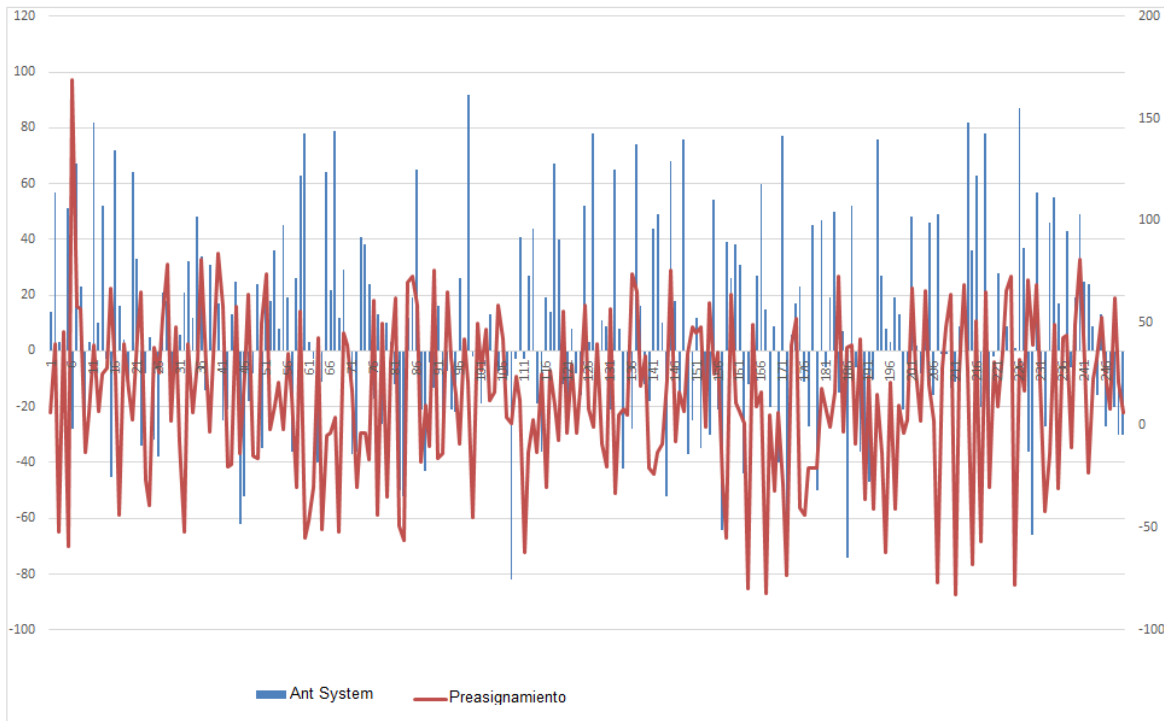
Numero de orden :63	Dia deseado :27	Dia de entrada :25	Dia de salida :61	C1 : 2	C2 : 4	C3 : 7	C4 : 3	C5 : 3	C6 : 3	C7 : 1	C8 : 3
Numero de orden :73	Dia deseado :57	Dia de entrada :31	Dia de salida :61	C1 : 6	C2 : 5	C3 : 0	C4 : 5	C5 : 0	C6 : 7	C7 : 5	C8 : 6
Numero de orden :119	Dia deseado :61	Dia de entrada :53	Dia de salida :61	C1 : 3	C2 : 5	C3 : 0	C4 : 2	C5 : 6	C6 : 7	C7 : 2	C8 : 8
Numero de orden :38	Dia deseado :34	Dia de entrada :14	Dia de salida :63	C1 : 5	C2 : 7	C3 : 3	C4 : 6	C5 : 8	C6 : 5	C7 : 0	C8 : 2
Numero de orden :103	Dia deseado :54	Dia de entrada :45	Dia de salida :65	C1 : 8	C2 : 7	C3 : 6	C4 : 3	C5 : 8	C6 : 6	C7 : 6	C8 : 0
Numero de orden :70	Dia deseado :31	Dia de entrada :29	Dia de salida :67	C1 : 8	C2 : 4	C3 : 5	C4 : 3	C5 : 2	C6 : 1	C7 : 5	C8 : 7
Numero de orden :36	Dia deseado :36	Dia de entrada :13	Dia de salida :67	C1 : 0	C2 : 5	C3 : 1	C4 : 2	C5 : 6	C6 : 1	C7 : 1	C8 : 5
Numero de orden :81	Dia deseado :65	Dia de entrada :35	Dia de salida :67	C1 : 5	C2 : 4	C3 : 0	C4 : 4	C5 : 7	C6 : 4	C7 : 6	C8 : 2
Numero de orden :69	Dia deseado :34	Dia de entrada :28	Dia de salida :69	C1 : 0	C2 : 3	C3 : 3	C4 : 0	C5 : 7	C6 : 6	C7 : 5	C8 : 4
Numero de orden :82	Dia deseado :51	Dia de entrada :36	Dia de salida :69	C1 : 2	C2 : 0	C3 : 1	C4 : 6	C5 : 1	C6 : 8	C7 : 3	C8 : 1
Numero de orden :133	Dia deseado :70	Dia de entrada :63	Dia de salida :71	C1 : 6	C2 : 4	C3 : 5	C4 : 7	C5 : 8	C6 : 3	C7 : 6	C8 : 0
Numero de orden :27	Dia deseado :37	Dia de entrada :9	Dia de salida :73	C1 : 1	C2 : 3	C3 : 5	C4 : 7	C5 : 2	C6 : 2	C7 : 6	C8 : 4
Numero de orden :71	Dia deseado :40	Dia de entrada :30	Dia de salida :73	C1 : 3	C2 : 7	C3 : 1	C4 : 8	C5 : 7	C6 : 7	C7 : 7	C8 : 2
Numero de orden :31	Dia deseado :39	Dia de entrada :11	Dia de salida :75	C1 : 4	C2 : 4	C3 : 2	C4 : 3	C5 : 1	C6 : 2	C7 : 5	C8 : 3
Numero de orden :88	Dia deseado :40	Dia de entrada :38	Dia de salida :75	C1 : 3	C2 : 4	C3 : 3	C4 : 1	C5 : 8	C6 : 3	C7 : 4	C8 : 6
Numero de orden :102	Dia deseado :49	Dia de entrada :44	Dia de salida :77	C1 : 7	C2 : 1	C3 : 5	C4 : 8	C5 : 4	C6 : 2	C7 : 6	C8 : 2
Numero de orden :83	Dia deseado :40	Dia de entrada :36	Dia de salida :79	C1 : 2	C2 : 1	C3 : 3	C4 : 2	C5 : 4	C6 : 6	C7 : 6	C8 : 8
Numero de orden :60	Dia deseado :40	Dia de entrada :24	Dia de salida :79	C1 : 3	C2 : 0	C3 : 2	C4 : 4	C5 : 0	C6 : 7	C7 : 4	C8 : 5
Numero de orden :35	Dia deseado :40	Dia de entrada :13	Dia de salida :81	C1 : 3	C2 : 8	C3 : 1	C4 : 2	C5 : 5	C6 : 4	C7 : 3	C8 : 6
Numero de orden :104	Dia deseado :51	Dia de entrada :46	Dia de salida :81	C1 : 3	C2 : 3	C3 : 2	C4 : 3	C5 : 8	C6 : 1	C7 : 4	C8 : 1
Numero de orden :130	Dia deseado :63	Dia de entrada :61	Dia de salida :81	C1 : 5	C2 : 3	C3 : 1	C4 : 1	C5 : 2	C6 : 4	C7 : 6	C8 : 1
Componentes en el día : 90											
C1 Tiempo de entrega : 1 Cantidad reabastecida : 17 Cantidad en stock : 1248 C2 Tiempo de entrega : 3 Cantidad reabastecida : 10 Cantidad en stock : 44 C3 Tiempo de entrega : 2 Cantidad reabastecida : 5 Cantidad en stock : 26 C4 Tiempo de entrega : 3 Cantidad reabastecida : 12 Cantidad en stock : 106 C5 Tiempo de entrega : 1 Cantidad reabastecida : 18 Cantidad en stock : 1350 C6 Tiempo de entrega : 2 Cantidad reabastecida : 16 Cantidad en stock : 424 C7 Tiempo de entrega : 2 Cantidad reabastecida : 9 Cantidad en stock : 103 C8 Tiempo de entrega : 6 Cantidad reabastecida : 19 Cantidad en stock : 38											

**Figura 27 Datos parciales generados corrida 3.**

En las figuras 26 y 27, Vemos los resultados conseguidos, se observa una marcada diferencia con el método anterior, se puede ver el reordenamiento que han sufrido las órdenes, en concreto, sobre el retraso de las órdenes, en algunos puntos, la función objetivo ha sido totalmente mejorada, y en otros se ha igualado sus resultados, con esto es factible mediante la manipulación de los parámetros iniciales, estimar con qué frecuencia tendría que refrescar el aprovisionamiento del inventario, esto como una característica importante y que deriva de la estructura misma del algoritmo.

A continuación se presenta la figura 28, se muestra la tendencia positiva que tiene el algoritmo Ant system con respecto al algoritmo de asignación, en ella la predisposición de resultados en el caso del Ant System (Azul) es tendiente a una disminución de los retrasos en la distribución de órdenes.

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO



**Figura 28 Datos cruzados con el método de preasignamiento y el algoritmo Ant System.**

Como una justificación a la utilización del algoritmo como una herramienta de decisión se ofrece la siguiente Tabla 9 que fortalecen las conclusiones anteriores.

**Tabla 9 Cuadro comparativo de cumplimientos.**

	<b>Porcentaje en el cumplimiento dentro de la ventana del día deseado.</b>	<b>Porcentaje anterior-posterior al día deseado.</b>	<b>Media porcentual.</b>
<b>Preasignamiento</b>	48%	46%	47%
<b>Ant System</b>	63%	43%	53%

## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

En esta tabla se tomó como base la frecuencia entre el día deseado y el cumplimiento en abastecer la orden. Se puede observar que el algoritmo tiene un índice de efectividad superior al de preasignamiento (63% contra 48%), es decir este algoritmo si bien no alcanzo un porcentaje alto cercano al 100%, se consiguió un índice bastante aceptable, algo que se pretendía mejorar desde un principio y que en determinado momento pudo haberse afectado por los índices o parámetros  $\beta$  y  $\rho$  incluidas dentro del mismo algoritmo.

Por otro lado en la misma tabla nos arroja sin embargo un porcentaje mejor en cuanto al surtimiento de las mismas por parte del preasignamiento sobre el algoritmo Ant System, esto es debido al factor humano, ya que cabe hacer notar que aquí se consideran situaciones como pueden ser prioridades de un cliente sobre otro, o falta de resurtimiento del nivel de stock aplicable a terceros.

Finalmente dentro de la media porcentual se observa que de manera global el algoritmo Ant System presenta un porcentaje levemente superior sobre el algoritmo de preasignamiento y como tal reafirmando la utilidad práctica del algoritmo.

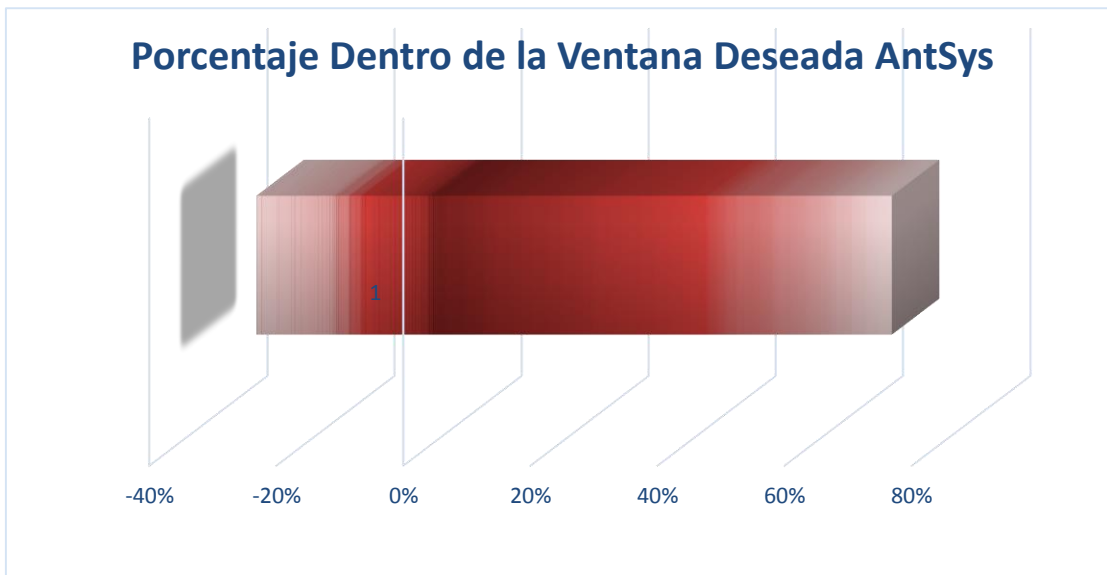
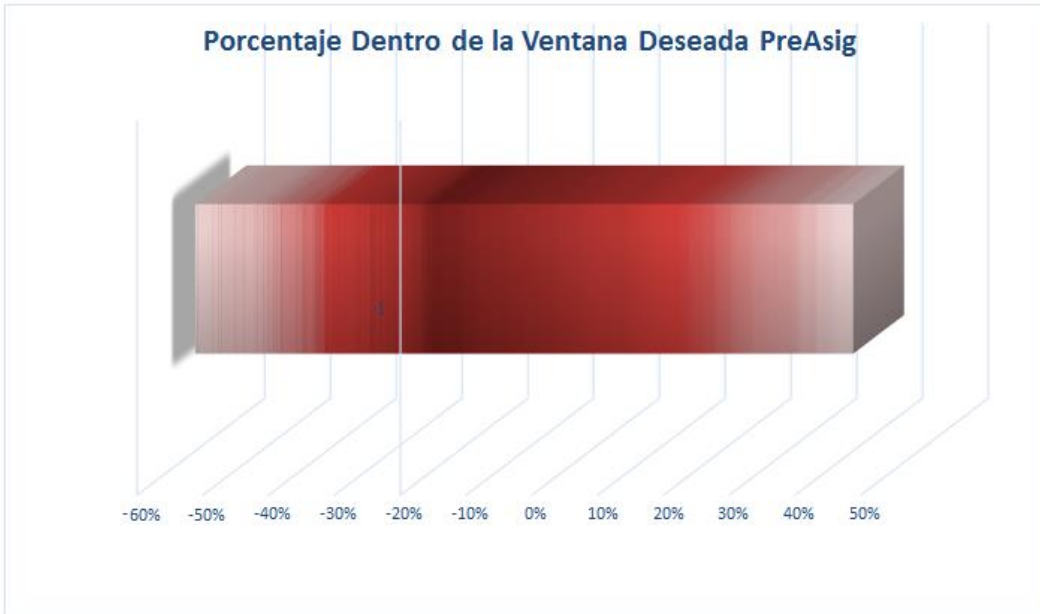


Figura 29 Grafica ventana orden deseada para Ant System



**Figura 30 Grafica ventana orden deseada para preasignamiento**

De la figura 29 y 30 se observa la tendencia que guarda el algoritmo Ant System contra el método de preasignamiento, todo esto plasmado en la tabla 9.

## Conclusiones:

En este trabajo se presenta un algoritmo de colonia de hormigas modificado (*Ant System*) para la optimización de problemas logísticos. Una asignación distribuida dinámicamente de los componentes de las órdenes se consigue por medio el intercambio de información simple entre los agentes que interactúan en el sistema. La feromonas utilizadas por las hormigas para determinar la asignación representa información distribuida por los intentos anteriores de asignación, así como la información sobre la fecha de entrega deseada y disponibilidad de los componentes para cumplir con los pedidos. La comparación entre este algoritmo y el otro método de programación (*Preasignamiento*) ha demostrado que el algoritmo de hormigas (*Ant System* modificado) ofrece una mejora para resolver los problemas de tipo *scheduling* en el caso de asignación de recursos.

El ejemplo simplificado ilustra un proceso logístico sobre el caso de la compañía llamada "Money", y muestra que el algoritmo de colonias de hormigas es una mejor alternativa en comparación con procesos paralelos al método de clasificación actualmente utilizado, ya que mediante la categorización de la función objetivo, se estableció una mejora en cuanto a la diferencia de los retrasos de las ordenes obtenidas en la distribución de órdenes, mediante esta técnica.

Otra importante conclusión que se desprende de los resultados es la utilidad como medio de pronóstico para la interpretación de la demanda futura que pudiera tener cualquier sistema.

En cuanto a la implementación del algoritmo La optimización de Colonia de Hormigas (*Ant System*), demuestra ser un potente instrumento para aplicar en los problemas de ingeniería industrial llegando a los siguientes resultados:

- Aunque en principio parezca complicado, resulta realmente sencillo su implementación;



- Queda demostrada la versatilidad de aplicación del método de optimización, tan solo con la modificación de pequeños parámetros en la función de la probabilidad;
- Se deja la vía trazada para la implementación de nuevos estudios, que como se demostró no se restringe a un solo campo de uso;
- Que el mayor incremento en el número de evaluaciones y por ende en el tiempo de ejecución se presenta cuando se aumenta el número de hormigas, más que aumentando el número de ciclos, sin embargo también se mejoran los resultados al tener un mayor número de hormigas;
- Por lo anterior es importante hacer varias pruebas para determinar un número de ambas técnicas que optimice los resultados y el tiempo de ejecución;
- Dada la magnitud del campo de exploración y los relativamente cortos tiempos de ejecución, se puede aumentar el número de iteraciones a fin de mejorar los resultados.
- En términos de la autonomía que se buscaba lograr del algoritmo, no se contempló prioridades en cuanto a las órdenes-clientes o bien al fallo en cuanto al resurtimiento, y por tanto un buen punto a refinar de esta tesis.



## Bibliografía

- Brian Kernighan, D. R. (2001). *El lenguaje de programación C*. Mexico D.F.: Prentice Hall.
- callaba, j. p. (2004). *logística internacional*. Mexico D.F.: Instituto de Ingeniería.
- Carlos A. Silva, T. A. (2003). *Optimization of logistic Processes in Supply Chains Using Metaheuristics*. Lisboa Portugal: Siemens AG Ed.
- Chase Richard, A. N. (2001). *Gestión de la producción y dirección de operaciones*. Mc Graw Hill.
- Cook, S. (1974). *The Complexity of Theorem Proving Procedures*. Toronto.
- Enxiu Chen, X. L. (2009). *Multi-Colony Ant Algorithm*. Shandong China: Shandong Normal University.
- Fabio Vicetini, S. P. (2003). *Algoritmos Heurísticos y el Problema del Job Shop Scheduling*. Buenos Aires, Argentina: Facultad de Ciencias Exactas y Naturales.
- G.L. Gambardella, D. M. (1996). *Ant colonies for the traveling salesman problem*. Boston M.: MIT.
- Grasse, P. P. (1948). *Traité de Zoologie*. Paris Francia: Pariss Ed.
- Kumar., S. (2010). *Application of Ant Colony Genetic Algorithm and Data Mining Based Techniques for Scheduling*. Warangal, India: Mechanical Engineering Department NIT.
- Maniezzo, D. M. (1996). *The Ant System: Optimization by Colony of Cooperating Agents*. Bruxelles Belgium: University Libre Of Bruxelles.
- Monroy, J. M. (2005). *Sistema de Control de Pedidos*. Mexico D.F.: Tesis UNAM.

- R., V. (2001). *Combinacion de Algoritmos Evolutivos y Tecnicas Heuristicas para Problemas de Scheduling*. Oviedo, España: Universidad de Oviedo.
- Ronald, B. (2005). *Administracion de la cadena de suministros*. Mexico D.F.: Pearson.
- Runkler, C. A. (2003). *Optimizacion of Logistic Processes in Suply-Chains Using Metaheuristics*. Lisboa Portugal.
- Sapaty, P. s. (2008). *Ruling Distributed Dinamic Worlds*. Ukrania: Wiley.
- Sousa, C. S. (2004). *A Logistic Process Scheduling Problem: Genetic Algorithms or Ant Colony Optimization*. Lisboa Portugal: Instituto Superior Tecnico.
- Stutzle, M. D. (2004). *Ant Colony Optimization*. Boston : The MIT Press.
- Suarez, G. A. (2001). *logistica moderna y la competitividad empresarial*. Habana: Laboratorio de Logistica y Gestion de la Produccion.
- Zumelzu, V. P. (2006). *Estado del Arte del Job Shop Scheduling Problem*. Colombia: Universidad Colombia .

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

## Anexo A

Tabla de datos recopilada de la empresa “money” para la demanda en un lapso de 3 meses y 250 muestras.

Orden	Día de Entrada	Día Deseado	Día liberado	Diferencia
1	67	19	73	54
2	35	45	36	-9
3	81	60	92	32
4	76	82	80	-2
5	9	49	16	-33
6	31	4	32	28
7	85	22	91	69
8	73	45	80	35
9	48	26	58	32
10	19	46	27	-19
11	71	60	79	19
12	56	5	67	62
13	4	23	7	-16
14	25	30	31	1
15	69	71	77	6
16	76	21	85	64
17	25	43	35	-8
18	59	17	61	44
19	4	57	10	-47
20	25	63	26	-37
21	51	66	62	-4
22	75	53	86	33
23	80	68	82	14
24	18	86	25	-61
25	1	87	11	-76
26	40	33	41	8
27	42	0	49	49
28	89	29	95	66
29	53	7	60	53
30	62	19	66	47
31	90	26	95	69
32	81	22	89	67
33	52	16	58	42
34	41	32	51	19
35	51	8	62	54

ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

36	39	85	45	-40
37	62	42	73	31
38	19	55	28	-27
39	46	0	57	57
40	68	75	70	-5
41	28	47	31	-16
42	79	37	88	51
43	17	75	21	-54
44	14	24	23	-1
45	88	10	94	84
46	36	17	46	29
47	3	2	10	8
48	24	31	30	-1
49	82	86	93	7
50	22	68	26	-42
51	65	60	73	13
52	58	30	64	34
53	7	9	13	4
54	90	38	94	56
55	20	2	32	30
56	44	70	52	-18
57	81	54	87	33
58	18	19	22	3
59	58	6	59	53
60	12	17	19	2
61	6	48	10	-38
62	54	66	64	-2
63	48	31	57	26
64	33	8	40	32
65	42	39	48	9
66	67	19	72	53
67	80	84	85	1
68	1	89	13	-76
69	66	79	76	-3
70	49	86	52	-34
71	60	59	63	4
72	24	9	30	21
73	3	20	14	-6
74	86	13	88	75
75	78	15	81	66
76	90	43	99	56
77	20	41	31	-10

ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

78	38	27	47	20
79	3	21	12	-9
80	40	11	41	30
81	30	49	37	-12
82	19	46	21	-25
83	9	33	17	-16
84	76	60	83	23
85	82	58	94	36
86	83	46	93	47
87	32	87	41	-46
88	28	11	33	22
89	65	46	75	29
90	67	32	79	47
91	33	38	43	5
92	74	11	84	73
93	31	43	42	-1
94	56	88	67	-21
95	23	19	27	8
96	72	54	81	27
97	29	51	36	-15
98	0	72	5	-67
99	22	13	24	11
100	16	47	26	-21
101	30	35	41	6
102	88	9	97	88
103	77	35	88	53
104	8	14	19	5
105	12	81	13	-68
106	70	48	76	28
107	0	30	12	-18
108	67	57	68	11
109	89	62	100	38
110	5	27	6	-21
111	16	88	24	-64
112	61	27	69	42
113	1	53	6	-47
114	42	17	52	35
115	33	74	45	-29
116	4	43	9	-34
117	71	37	82	45
118	80	1	86	85
119	1	43	12	-31

ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL  
CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

120	3	31	7	-24
121	47	12	51	39
122	77	74	81	7
123	36	64	37	-27
124	90	9	100	91
125	36	35	37	2
126	18	3	19	16
127	90	11	102	91
128	80	69	88	19
129	31	79	40	-39
130	79	45	89	44
131	52	84	54	-30
132	71	31	76	45
133	77	22	82	60
134	86	28	92	64
135	65	10	76	66
136	30	55	42	-13
137	9	50	13	-37
138	27	60	29	-31
139	46	19	48	29
140	48	40	60	20
141	26	70	34	-36
142	51	77	58	-19
143	68	16	80	64
144	85	4	86	82
145	90	7	100	93
146	29	51	41	-10
147	62	83	73	-10
148	26	90	28	-62
149	4	87	16	-71
150	85	17	92	75
151	48	77	50	-27
152	88	34	100	66
153	19	48	24	-24
154	55	25	63	38
155	1	34	10	-24
156	34	17	44	27
157	0	23	10	-13
158	17	82	23	-59
159	76	33	78	45
160	89	75	98	23
161	22	11	31	20

ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

162	20	57	25	-32
163	72	55	78	23
164	21	50	31	-19
165	70	8	71	63
166	7	44	14	-30
167	18	7	30	23
168	24	21	26	5
169	25	59	27	-32
170	38	75	43	-32
171	1	65	11	-54
172	29	69	35	-34
173	51	34	55	21
174	59	13	64	51
175	59	37	69	32
176	6	75	9	-66
177	78	42	80	38
178	64	15	65	50
179	18	34	23	-11
180	63	28	65	37
181	26	73	30	-43
182	16	38	26	-12
183	10	21	11	-10
184	77	1	87	86
185	87	18	99	81
186	48	90	58	-32
187	14	68	20	-48
188	39	38	48	10
189	86	37	95	58
190	45	49	54	5
191	64	18	67	49
192	62	52	67	15
193	89	30	101	71
194	10	68	13	-55
195	28	38	33	-5
196	25	51	33	-18
197	29	17	35	18
198	42	64	46	-18
199	36	26	39	13
200	2	55	12	-43
201	7	59	15	-44
202	28	16	31	15
203	33	53	41	-12

ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL  
CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

---

204	88	49	93	44
205	46	41	51	10
206	45	21	47	26
207	6	11	8	-3
208	82	10	86	76
209	78	74	85	11
210	73	21	76	55
211	30	40	33	-7
212	34	69	42	-27
213	33	37	45	8
214	32	61	41	-20
215	72	3	81	78
216	46	46	55	9
217	56	34	68	34
218	4	0	13	13
219	64	23	67	44
220	14	76	16	-60
221	81	28	88	60
222	53	50	62	12
223	8	41	15	-26
224	67	5	71	66
225	19	88	30	-58
226	46	13	55	42
227	15	64	25	-39
228	9	1	16	15
229	85	89	88	-1
230	78	61	89	28
231	41	72	42	-30
232	40	63	47	-16
233	42	72	52	-20
234	8	58	19	-39
235	19	82	29	-53
236	69	1	73	72
237	70	79	75	-4
238	0	79	1	-78
239	1	17	8	-9
240	42	67	52	-15
241	3	65	8	-57
242	70	19	74	55
243	73	82	76	-6
244	81	2	83	81
245	55	45	66	21



## ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

---

<b>246</b>	51	2	62	60
<b>247</b>	85	61	89	28
<b>248</b>	57	26	59	33
<b>249</b>	18	35	28	-7
<b>250</b>	59	43	63	20

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

## Anexo B

Tabla de salida del algoritmo Ant system Modificado.

Numero	de orden:	Ordenes Liberadas
	14	
	Dia deseado:	5
	Dia de entrada :	4
	Dia de salida :	5
	C1 :	2 C2 : 8 C3 : 3 C4 : 2 C5 : 4 C6 : 5 C7 : 4 C8 : 7
	19	
	Dia deseado :	8
	Dia de entrada :	6
	Dia de salida :	8
	C1 :	0 C2 : 4 C3 : 4 C4 : 0 C5 : 0 C6 : 7 C7 : 8 C8 : 6
	11	
	Dia deseado :	8
	Dia de entrada :	3
	Dia de salida :	8
	C1 :	5 C2 : 1 C3 : 8 C4 : 5 C5 : 8 C6 : 4 C7 : 4 C8 : 8
	2	
	Dia deseado :	8
	Dia de entrada :	1
	Dia de salida :	8
	C1 :	2 C2 : 2 C3 : 4 C4 : 0 C5 : 6 C6 : 8 C7 : 9
	5	
	Dia deseado :	9
	Dia de entrada :	1
	Dia de salida :	9
	C1 :	0 C2 : 0 C3 : 3 C4 : 9 C5 : 8 C6 : 3 C7 : 0
	3	
	Dia deseado :	10
	Dia de entrada :	1
	Dia de salida :	10
	C1 :	1 C2 : 3 C3 : 2 C4 : 7 C5 : 8 C6 : 10 C7 : 0
	8	
	Dia deseado :	10
	Dia de entrada :	2
	Dia de salida :	11
	C1 :	2 C2 : 3 C3 : 6 C4 : 4 C5 : 0 C6 : 5 C7 : 6 C8 : 2
	25	
	Dia deseado :	13
	Dia de entrada :	9
	Dia de salida :	13
	C1 :	8 C2 : 1 C3 : 2 C4 : 4 C5 : 2 C6 : 6 C7 : 1 C8 : 4
	29	
	Dia deseado :	12
	Dia de entrada :	10
	Dia de salida :	15
	C1 :	7 C2 : 1 C3 : 7 C4 : 2 C5 : 7 C6 : 4 C7 : 7 C8 : 5
	15	
	Dia deseado :	12
	Dia de entrada :	4
	Dia de salida :	17
	C1 :	2 C2 : 7 C3 : 6 C4 : 4 C5 : 0 C6 : 2 C7 : 3 C8 : 2
	13	
	Dia deseado :	14
	Dia de entrada :	3
	Dia de salida :	19
	C1 :	3 C2 : 0 C3 : 4 C4 : 2 C5 : 5 C6 : 4 C7 : 3 C8 : 6
	32	
	Dia deseado :	18
	Dia de entrada :	11
	Dia de salida :	19
	C1 :	7 C2 : 2 C3 : 1 C4 : 0 C5 : 2 C6 : 6 C7 : 8 C8 : 8

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

---

Numero de orden :1  
Dia deseado :14  
Dia de entrada :1  
Dia de salida :21  
C1 : 3 C2 : 5 C3 : 5 C4 : 8 C5 : 2 C6 : 6 C7 : 8

Numero de orden :4  
Dia deseado :15  
Dia de entrada :1  
Dia de salida :23  
C1 : 11 C2 : 0 C3 : 5 C4 : 6 C5 : 0 C6 : 0 C7 : 0

Numero de orden :22  
Dia deseado :16  
Dia de entrada :7  
Dia de salida :25  
C1 : 7 C2 : 1 C3 : 3 C4 : 2 C5 : 8 C6 : 6 C7 : 7 C8 : 8

Numero de orden :34  
Dia deseado :19  
Dia de entrada :13  
Dia de salida :25  
C1 : 7 C2 : 5 C3 : 2 C4 : 4 C5 : 5 C6 : 8 C7 : 2 C8 : 6

Numero de orden :37  
Dia deseado :26  
Dia de entrada :14  
Dia de salida :26  
C1 : 5 C2 : 4 C3 : 0 C4 : 7 C5 : 7 C6 : 8 C7 : 5 C8 : 4

Numero de orden :28  
Dia deseado :17  
Dia de entrada :10  
Dia de salida :27  
C1 : 1 C2 : 0 C3 : 3 C4 : 0 C5 : 7 C6 : 1 C7 : 1 C8 : 3

Numero de orden :45  
Dia deseado :21  
Dia de entrada :18  
Dia de salida :27  
C1 : 2 C2 : 8 C3 : 2 C4 : 5 C5 : 3 C6 : 8 C7 : 8 C8 : 6

Numero de orden :40  
Dia deseado :20  
Dia de entrada :15  
Dia de salida :29  
C1 : 2 C2 : 4 C3 : 3 C4 : 0 C5 : 5 C6 : 1 C7 : 8 C8 : 2

Numero de orden :12  
Dia deseado :22  
Dia de entrada :3  
Dia de salida :29  
C1 : 1 C2 : 2 C3 : 1 C4 : 2 C5 : 3 C6 : 6 C7 : 5 C8 : 3

Numero de orden :44  
Dia deseado :24  
Dia de entrada :17  
Dia de salida :29  
C1 : 3 C2 : 8 C3 : 1 C4 : 3 C5 : 5 C6 : 3 C7 : 0 C8 : 6

Numero de orden :47  
Dia deseado :22  
Dia de entrada :19  
Dia de salida :31  
C1 : 7 C2 : 3 C3 : 4 C4 : 5 C5 : 0 C6 : 4 C7 : 8 C8 : 5

Numero de orden :30  
Dia deseado :29  
Dia de entrada :10  
Dia de salida :31  
C1 : 5 C2 : 1 C3 : 1 C4 : 3 C5 : 4 C6 : 1 C7 : 8 C8 : 4

Numero de orden :67  
Dia deseado :31  
Dia de entrada :27  
Dia de salida :31  
C1 : 1 C2 : 8 C3 : 0 C4 : 4 C5 : 2 C6 : 3 C7 : 8 C8 : 8

Numero de orden :39  
Dia deseado :24  
Dia de entrada :15  
Dia de salida :33  
C1 : 4 C2 : 1 C3 : 3 C4 : 4 C5 : 3 C6 : 8 C7 : 1 C8 : 1

Numero de orden :50

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

---

Numero de orden :41  
Dia deseado :25  
Dia de entrada :20  
Dia de salida :33  
C1 : 5 C2 : 6 C3 : 2 C4 : 3 C5 : 2 C6 : 0 C7 : 6 C8 : 6

Numero de orden :42  
Dia deseado :33  
Dia de entrada :16  
Dia de salida :33  
C1 : 5 C2 : 4 C3 : 0 C4 : 3 C5 : 5 C6 : 0 C7 : 3 C8 : 0

Numero de orden :49  
Dia deseado :25  
Dia de entrada :17  
Dia de salida :35  
C1 : 5 C2 : 8 C3 : 2 C4 : 2 C5 : 3 C6 : 0 C7 : 5 C8 : 3

Numero de orden :21  
Dia deseado :31  
Dia de entrada :20  
Dia de salida :35  
C1 : 3 C2 : 5 C3 : 1 C4 : 5 C5 : 1 C6 : 8 C7 : 7 C8 : 0

Numero de orden :33  
Dia deseado :25  
Dia de entrada :12  
Dia de salida :37  
C1 : 3 C2 : 2 C3 : 3 C4 : 0 C5 : 3 C6 : 1 C7 : 8 C8 : 2

Numero de orden :58  
Dia deseado :28  
Dia de entrada :9  
Dia de salida :37  
C1 : 8 C2 : 1 C3 : 2 C4 : 3 C5 : 5 C6 : 7 C7 : 5 C8 : 7

Numero de orden :62  
Dia deseado :36  
Dia de entrada :23  
Dia de salida :37  
C1 : 0 C2 : 1 C3 : 0 C4 : 8 C5 : 1 C6 : 6 C7 : 3 C8 : 8

Numero de orden :9  
Dia deseado :12  
Dia de entrada :2  
Dia de salida :41  
C1 : 2 C2 : 5 C3 : 6 C4 : 0 C5 : 2 C6 : 7 C7 : 1 C8 : 0

Numero de orden :46  
Dia deseado :26  
Dia de entrada :1  
Dia de salida :43  
C1 : 7 C2 : 7 C3 : 4 C4 : 2 C5 : 1 C6 : 2 C7 : 3 C8 : 8

Numero de orden :24  
Dia deseado :36  
Dia de entrada :19  
Dia de salida :43  
C1 : 5 C2 : 1 C3 : 1 C4 : 8 C5 : 6 C6 : 2 C7 : 5 C8 : 7

Numero de orden :74  
Dia deseado :31  
Dia de entrada :8  
Dia de salida :45  
C1 : 0 C2 : 6 C3 : 5 C4 : 0 C5 : 4 C6 : 8 C7 : 7 C8 : 4

Numero de orden :23  
Dia deseado :34  
Dia de entrada :31  
Dia de salida :47  
C1 : 5 C2 : 1 C3 : 3 C4 : 2 C5 : 0 C6 : 3 C7 : 6 C8 : 0

Numero de orden :22  
Dia deseado :22

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

---

Numero de orden :64  
Dia de entrada :8  
Dia de salida :49  
C1 : 4 C2 : 1 C3 : 7 C4 : 6 C5 : 1 C6 : 2 C7 : 2 C8 : 3

Numero de orden :76  
Dia deseado :42  
Dia de entrada :26  
Dia de salida :49  
C1 : 8 C2 : 4 C3 : 0 C4 : 1 C5 : 5 C6 : 0 C7 : 1 C8 : 7

Numero de orden :7  
Dia deseado :50  
Dia de entrada :32  
Dia de salida :50  
C1 : 8 C2 : 5 C3 : 0 C4 : 7 C5 : 1 C6 : 7 C7 : 6 C8 : 0

Numero de orden :101  
Dia deseado :28  
Dia de entrada :1  
Dia de salida :51  
C1 : 3 C2 : 2 C3 : 5 C4 : 7 C5 : 4 C6 : 7 C7 : 2 C8 : 8

Numero de orden :91  
Dia deseado :51  
Dia de entrada :44  
Dia de salida :51  
C1 : 1 C2 : 8 C3 : 0 C4 : 2 C5 : 3 C6 : 6 C7 : 3 C8 : 1

Numero de orden :96  
Dia deseado :49  
Dia de entrada :39  
Dia de salida :53  
C1 : 7 C2 : 8 C3 : 1 C4 : 6 C5 : 3 C6 : 5 C7 : 2 C8 : 1

Numero de orden :18  
Dia deseado :52  
Dia de entrada :41  
Dia de salida :53  
C1 : 8 C2 : 8 C3 : 3 C4 : 1 C5 : 8 C6 : 1 C7 : 5 C8 : 0

Numero de orden :17  
Dia deseado :29  
Dia de entrada :6  
Dia de salida :55  
C1 : 2 C2 : 2 C3 : 4 C4 : 7 C5 : 4 C6 : 4 C7 : 0 C8 : 8

Numero de orden :95  
Dia deseado :32  
Dia de entrada :5  
Dia de salida :55  
C1 : 6 C2 : 0 C3 : 2 C4 : 3 C5 : 6 C6 : 0 C7 : 5 C8 : 3

Numero de orden :59  
Dia deseado :54  
Dia de entrada :41  
Dia de salida :55  
C1 : 7 C2 : 4 C3 : 0 C4 : 4 C5 : 8 C6 : 4 C7 : 5 C8 : 8

Numero de orden :63  
Dia deseado :33  
Dia de entrada :23  
Dia de salida :59  
C1 : 8 C2 : 5 C3 : 8 C4 : 6 C5 : 6 C6 : 7 C7 : 4 C8 : 0

Numero de orden :73  
Dia deseado :27  
Dia de entrada :25  
Dia de salida :61  
C1 : 2 C2 : 4 C3 : 7 C4 : 3 C5 : 3 C6 : 3 C7 : 1 C8 : 3

Numero de orden :119  
Dia deseado :57  
Dia de entrada :31  
Dia de salida :61  
C1 : 6 C2 : 5 C3 : 0 C4 : 5 C5 : 0 C6 : 7 C7 : 5 C8 : 6

Numero de orden :38  
Dia deseado :61  
Dia de entrada :53  
Dia de salida :61  
C1 : 3 C2 : 5 C3 : 0 C4 : 2 C5 : 6 C6 : 7 C7 : 2 C8 : 8

Numero de orden :34  
Dia de entrada :14

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

---

Dia de salida :63  
C1 : 5 C2 : 7 C3 : 3 C4 : 6 C5 : 8 C6 : 5 C7 : 0 C8 : 2  
Numero de orden :103  
Dia deseado :54  
Dia de entrada :45  
Dia de salida :65  
C1 : 8 C2 : 7 C3 : 6 C4 : 3 C5 : 8 C6 : 6 C7 : 6 C8 : 0  
Numero de orden :70  
Dia deseado :31  
Dia de entrada :29  
Dia de salida :67  
C1 : 8 C2 : 4 C3 : 5 C4 : 3 C5 : 2 C6 : 1 C7 : 5 C8 : 7  
Numero de orden :36  
Dia deseado :36  
Dia de entrada :13  
Dia de salida :67  
C1 : 0 C2 : 5 C3 : 1 C4 : 2 C5 : 6 C6 : 1 C7 : 1 C8 : 5  
Numero de orden :81  
Dia deseado :65  
Dia de entrada :35  
Dia de salida :67  
C1 : 5 C2 : 4 C3 : 0 C4 : 4 C5 : 7 C6 : 4 C7 : 6 C8 : 2  
Numero de orden :69  
Dia deseado :34  
Dia de entrada :28  
Dia de salida :69  
C1 : 0 C2 : 3 C3 : 3 C4 : 0 C5 : 7 C6 : 6 C7 : 5 C8 : 4  
Numero de orden :82  
Dia deseado :51  
Dia de entrada :36  
Dia de salida :69  
C1 : 2 C2 : 0 C3 : 1 C4 : 6 C5 : 1 C6 : 8 C7 : 3 C8 : 1  
Numero de orden :133  
Dia deseado :70  
Dia de entrada :63  
Dia de salida :71  
C1 : 6 C2 : 4 C3 : 5 C4 : 7 C5 : 8 C6 : 3 C7 : 6 C8 : 0  
Numero de orden :27  
Dia deseado :37  
Dia de entrada :9  
Dia de salida :73  
C1 : 1 C2 : 3 C3 : 5 C4 : 7 C5 : 2 C6 : 2 C7 : 6 C8 : 4  
Numero de orden :71  
Dia deseado :40  
Dia de entrada :30  
Dia de salida :73  
C1 : 3 C2 : 7 C3 : 1 C4 : 8 C5 : 7 C6 : 7 C7 : 7 C8 : 2  
Numero de orden :31  
Dia deseado :39  
Dia de entrada :11  
Dia de salida :75  
C1 : 4 C2 : 4 C3 : 2 C4 : 3 C5 : 1 C6 : 2 C7 : 5 C8 : 3  
Numero de orden :88  
Dia deseado :40  
Dia de entrada :38  
Dia de salida :75  
C1 : 3 C2 : 4 C3 : 3 C4 : 1 C5 : 8 C6 : 3 C7 : 4 C8 : 6  
Numero de orden :102  
Dia deseado :49  
Dia de entrada :44  
Dia de salida :77  
C1 : 7 C2 : 1 C3 : 5 C4 : 8 C5 : 4 C6 : 2 C7 : 6 C8 : 2  
Numero de orden :83  
Dia deseado :40  
Dia de entrada :36  
Dia de salida :79  
C1 : 2 C2 : 1 C3 : 3 C4 : 2 C5 : 4 C6 : 6 C7 : 6 C8 : 8  
Numero de orden :60  
Dia deseado :40  
Dia de entrada :24  
Dia de salida :79

# ACO PARA LA MINIMIZACIÓN DE TIEMPOS DE RESPUESTA EN EL CUMPLIMIENTO DE ÓRDENES EN UN PROCESO LOGÍSTICO

---

C1 : 3 C2 : 0 C3 : 2 C4 : 4 C5 : 0 C6 : 7 C7 : 4 C8 : 5  
Numero de orden :35  
Dia deseado :40  
Dia de entrada :13  
Dia de salida :81  
C1 : 3 C2 : 8 C3 : 1 C4 : 2 C5 : 5 C6 : 4 C7 : 3 C8 : 6  
Numero de orden :104  
Dia deseado :51  
Dia de entrada :46  
Dia de salida :81  
C1 : 3 C2 : 3 C3 : 2 C4 : 3 C5 : 8 C6 : 1 C7 : 4 C8 : 1  
Numero de orden :130  
Dia deseado :63  
Dia de entrada :61  
Dia de salida :81  
C1 : 5 C2 : 3 C3 : 1 C4 : 1 C5 : 2 C6 : 4 C7 : 6 C8 : 1  
Componentes en el dia : 90

---

C1	Tiempo de entrega : 1	Cantidad reabastecida : 17	Cantidad en stock : 1248
C2	Tiempo de entrega : 3	Cantidad reabastecida : 10	Cantidad en stock : 44
C3	Tiempo de entrega : 2	Cantidad reabastecida : 5	Cantidad en stock : 26
C4	Tiempo de entrega : 3	Cantidad reabastecida : 12	Cantidad en stock : 106
C5	Tiempo de entrega : 1	Cantidad reabastecida : 18	Cantidad en stock : 1350
C6	Tiempo de entrega : 2	Cantidad reabastecida : 16	Cantidad en stock : 424
C7	Tiempo de entrega : 2	Cantidad reabastecida : 9	Cantidad en stock : 103
C8	Tiempo de entrega : 6	Cantidad reabastecida : 19	Cantidad en stock : 38