

Aplicación de inteligencia distribuida a la disminución del efecto bullwhip

David de la Fuente García¹, Jesús Lozano Mosterín¹, Isabel Fernández Quesada¹,
Alejandro Garmón Izquierdo¹

¹ Dpto. de Admón. de Empresas y Contabilidad. Escuela Politécnica Superior de Ingeniería de Gijón. Campus de Viesques, 33204 Gijón (Asturias). david@uniovi.es, lozano@epsig.uniovi.es, ifq@uniovi.es.

Resumen

La disminución del tamaño de los almacenes y de los niveles de productos en stock, fórmulas que podrían abaratar los costes logísticos a lo largo de una cadena o red de suministro, encuentran un obstáculo importante en el denominado efecto bullwhip, que asimismo transmite gran incertidumbre en el área de producción. Las soluciones tradicionales abarcan desde el uso de las tecnologías de la información para el intercambio de datos sobre la demanda, hasta la centralización del control de los inventarios. A diferencia de estas estrategias, la solución propuesta en este artículo podría ser utilizada independientemente por los distintos miembros de la cadena, presentando las ventajas de la previsión de la demanda mediante algoritmos genéticos, la metodología de Box-Jenkins y el uso de medidores del efecto bullwhip, y la potencia de análisis de las redes neuronales artificiales y las herramientas de simulación, todo ello coordinado mediante sistemas multi-agente que añaden al conjunto capacidades de planificación, aprendizaje y comunicación. A lo largo del artículo se estudia el efecto bullwhip mediante inteligencia distribuida, logrando disminuirlo en diferentes series temporales a las que se aplica el algoritmo genético.

Palabras clave: Efecto bullwhip, sistemas multi-agente, logística, secuenciación

1. Introducción

El trabajo, está dividido en cinco apartados: En el primero, se realiza una somera descripción del efecto bullwhip y se presenta la estructura en red de agentes, y el modelo general que los soporta. En el segundo apartado se describen los algoritmos genéticos como dispositivos que aportarán inteligencia al modelo. En el tercer apartado se exponen los ejemplos numéricos y en el último apartado se presentan las conclusiones.

1.1. Antecedentes y efecto bullwhip.

El estudio del efecto bullwhip, también llamado efecto látigo o efecto Forrester, ha sido una fructífera subárea de investigación en el estudio de la logística. El primer precedente formal del efecto bullwhip se atribuye al propio Forrester (1961), y fue ampliado con posterioridad por Serman (1989, 1992, 1995), con el famoso ejemplo del juego de la cerveza que se practica en muchas Escuelas de Negocios. Según Serman el efecto bullwhip se origina desde la solución no óptima adoptada por los miembros de la cadena de suministro al considerar sus estrategias, no como un conjunto, sino como una suma de individualidades; de esta manera, las variaciones registradas en la demanda de los consumidores finales se trasladan en el sentido inverso de la cadena de suministro, multiplicando sus efectos en cada cambio de nivel. Kaminsky y Simchi-Levi (1998) desarrollaron diferentes versiones informáticas del

juego de la cerveza ilustrando tres efectos interrelacionados en la cadena de suministro: el efecto bullwhip, el efecto centralización y el efecto tiempo de suministro.

Diferentes estudios económicos ponen en cuestión las razones por las que se produce dicho efecto al tomar decisiones racionales para evitar la ruptura de stocks, y al intentar mitigar la incertidumbre de la demanda (Blanchard (1983), Blinder (1986), West (1986), Kahn (1987)). Lee, Padmanabhan y Whang (1997) estudiaron el problema, aplicando modelos analíticos, y concluyeron en la existencia de cuatro razones fundamentales por las que se producía el efecto bullwhip: errores en las previsiones, temor a faltas de productos, inadecuada lotificación y fluctuaciones en los precios.

En trabajos recientes Cachon y Lariviere (1999) estudiaron el efecto de la escasez; Kelle y Milne (1999) y Cachon (1999) estudiaron el efecto de la notificación; Drezner et al. (1996), Baganha y Cohen (1998), Graves (1999), Chen et al. (2000), Aviv (2001, 2003) y Dejonckheere et al (2003) estudiaron los errores de demanda. Sin embargo, los autores más prolíficos en este tema y en relación con el efecto bullwhip son Disney y Towill, (2001, 2002, 2003a, 2003b, 2004a, 2004b, 2004c). El primero, realizó su tesis doctoral sobre la metodología VMI: Virtual Managed Inventory (2001), ampliando posteriormente el estudio del efecto bullwhip desde el punto de vista de la teoría de control, junto con otros autores (ver Disney y Grubbstrom (2004), Chiderhouse et al. (2004), Dejonckheere et al. (2004), Disney et al. (2004), Hosoda y Disney S. M. (2005)).

En la estructura típica de distribución logística (Figura 1), el flujo de información no fluye aguas arriba entre todos los niveles, sino sólo entre cada dos niveles consecutivos (Sternan, 1989). Algunos casos específicos, como cuando existe realimentación de la información de los puntos de venta a todos los niveles de la cadena de suministro, han sido estudiados con simulación (Mason-Jones, 1998) o con teoría de control y transformada z (Dejonckheere et al., 2003).

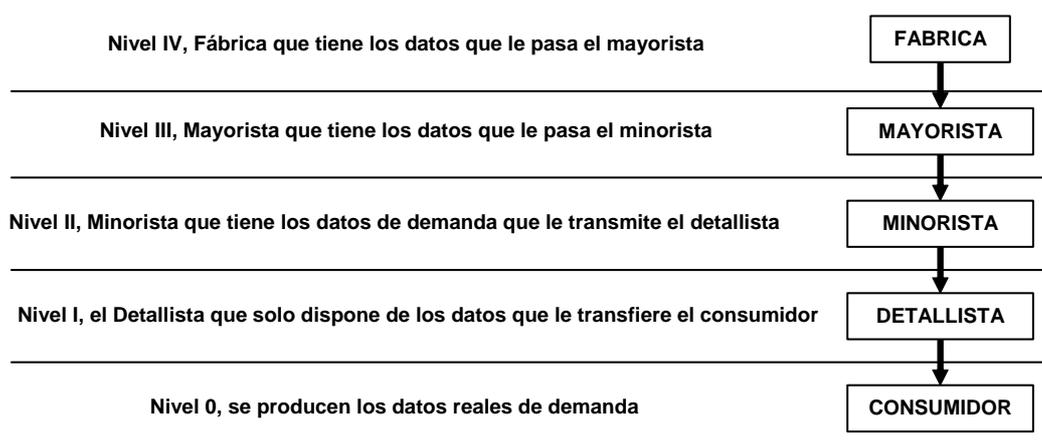


Figura 1. Cuatro escenarios de la cadena de suministros.

En definitiva, las soluciones tradicionales al efecto bullwhip abarcan desde el uso de las tecnologías de la información para el intercambio de datos sobre la demanda, hasta la centralización del control de los inventarios, pasando por estrategias que retrasan la diferenciación de los productos a los últimos niveles de la cadena de suministro.

1.2. Modelos multi-agentes para la disminución del efecto bullwhip.

A diferencia de las soluciones tradicionales, la solución propuesta* en este artículo podría ser utilizada independientemente por los distintos miembros de la cadena, presentando las ventajas de la previsión de la demanda mediante algoritmos genéticos, la metodología de Box-Jenkins (1970) y el uso de medidores del efecto bullwhip, y la potencia de análisis de las redes neuronales artificiales y las herramientas de simulación, todo ello coordinado mediante sistemas multi-agente que añaden al conjunto capacidades de planificación, aprendizaje y comunicación (Figura 2 y Figura 3).

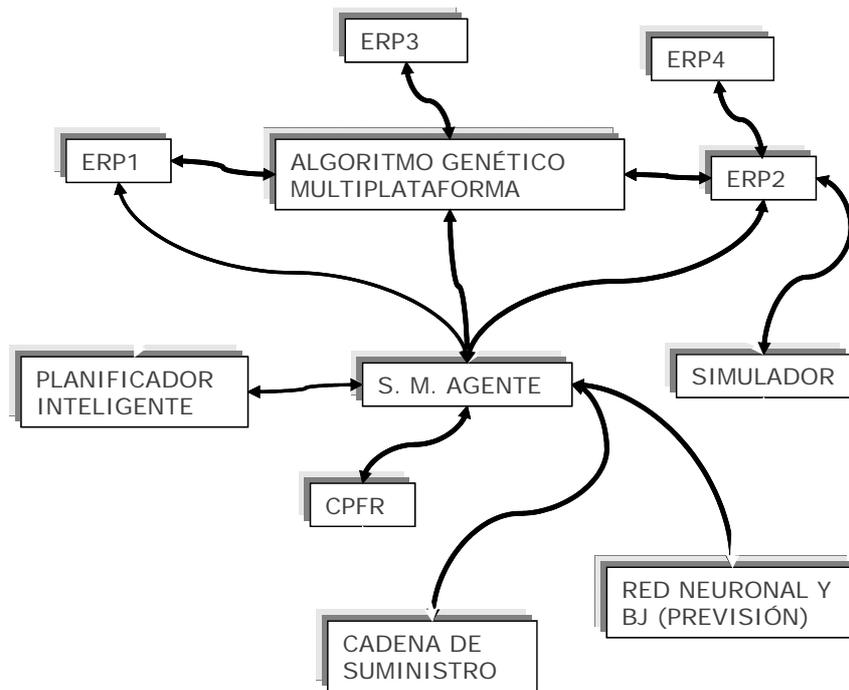


Figura 2. Modelo propuesto

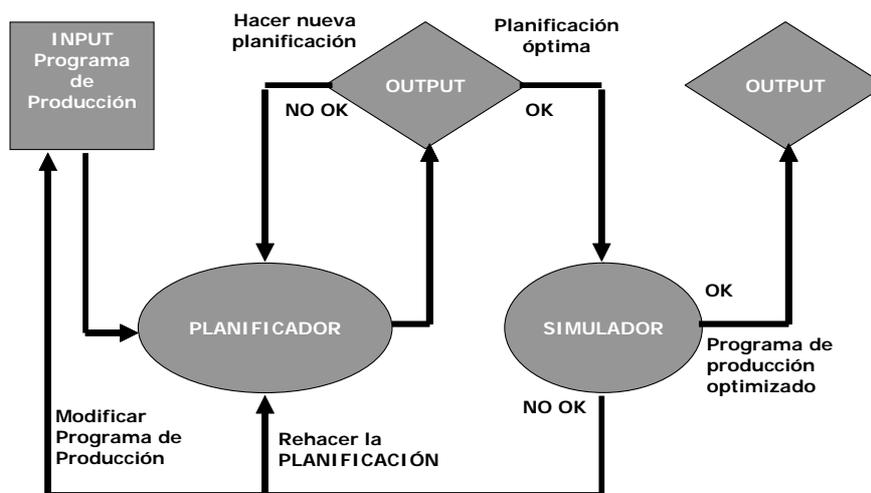


Figura 2. Planificador-Simulador

* Este trabajo se deriva de la participación de sus autores en un proyecto de investigación financiado por la CECA, titulado "Optimization of stocks management and production scheduling by simulation of the Continuous Casting Rolling and Finishing departments (SIMUSTEEL)".

En la figura 4 se representa el esquema general de multi-agentes para la cadena de suministro en estudio. Se debe de entender que el concepto de agente se aplicará, además de a un algoritmo genético, para utilizar otra información proveniente de otras herramientas, para decidir en conjunto cuanto se debe pedir, tener en stock, etc. Estas herramientas podrían ser, por ejemplo, redes neuronales, Box-Jenkins u otros métodos para hacer previsiones, o herramientas de simulación para validar los envíos o recepciones u obtener reglas. Además la utilización de la arquitectura Corba permite resolver el algoritmo genético en diferentes plataformas, y el lenguaje KQML permite la comunicación entre los agentes, pudiendo llegar a acuerdos sobre cuanto enviar, pedir, etc.

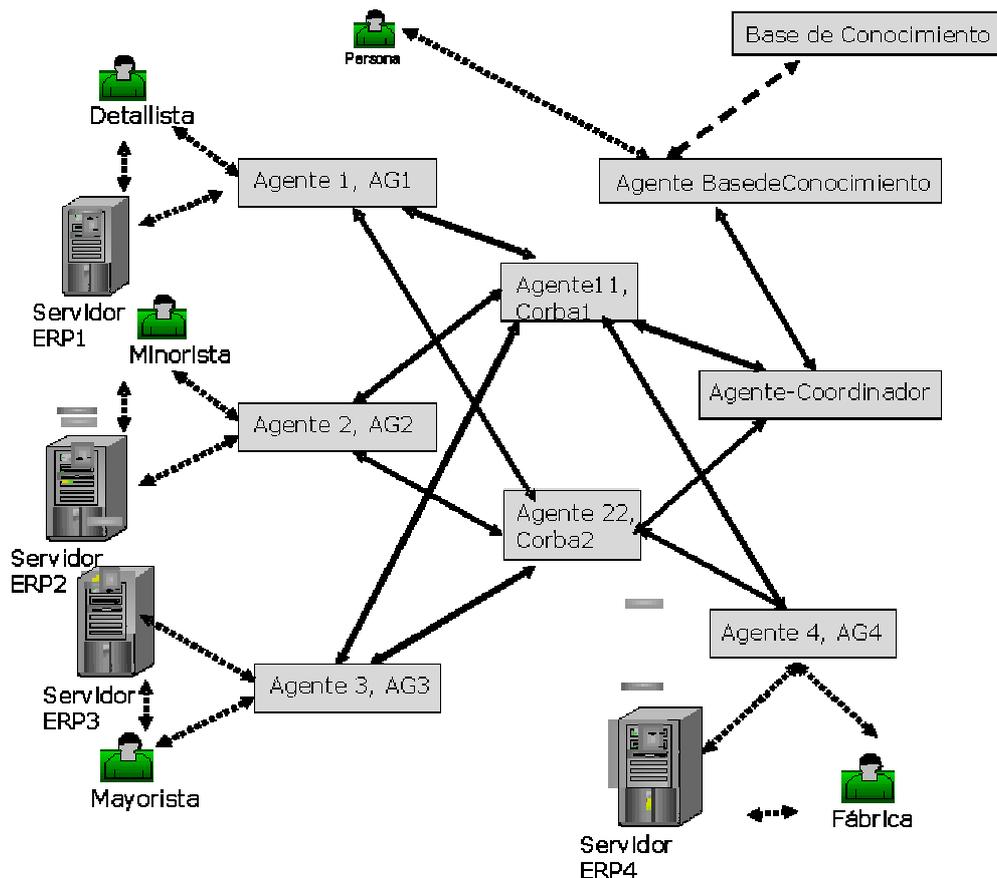


Figura 4. Red con 4 niveles.

2. Algoritmos genéticos.

El esquema evolutivo del ciclo típico de un Algoritmo Genético (Figura 5) es el siguiente, partiendo de una población inicial de N posibles soluciones del problema propuesto, siendo N el tamaño de la población. En este caso, un elemento de la población tomaría la forma (1):

$$X^k = (X_{11}^k, \dots, X_{1L}^k, X_{21}^k, \dots, X_{2L}^k, \dots, X_{M1}^k, \dots, X_{ML}^k) \quad k = 1, \dots, n \quad (1)$$

donde k indica el nivel de la cadena de suministro; en nuestro caso n llega a cuatro. El índice, que varía de 1 a M , será las diferentes características que queremos medir en un determinado nivel como puede ser el stock inicial, lo que se pide, etc. Y el índice en el que varía L representa los periodos temporales.

En general, el conjunto de genes (elementos) para cada cromosoma (individuo) es seleccionado del siguiente modo:

- El primer gen es seleccionado al azar.
- Posteriormente el resto de genes son también elegidos al azar pero de tal forma que verifiquen las restricciones de la demanda del modelo. A continuación se efectúa una comprobación para verificar si se cumplen las condiciones de parada.

Después se hace la selección de la población N , eligiendo solamente $N \cdot G$ elementos, siendo G el índice de cruce. Serán seleccionados los $N \cdot G$ elementos de la población que presenten un valor de f (función de fitness) más alto. En la etapa reproductiva se efectúan operaciones de cruce y de mutación. La operación de cruce es lineal; de este modo, se elige dos padres (p_1 y p_2) de tamaño N' al azar y se generan dos hijos. El hijo con mejor "fitness" es seleccionado.

Por otra parte, la operación de mutación (cuya frecuencia depende del parámetro M) consiste en el intercambio de dos componentes elegidas al azar de un hijo, también seleccionado al azar. La fase reproductiva, debe de aplicarse tantas veces como sea necesario para obtener $(1-G) \cdot N$ hijos.

Así tenemos Algoritmos Genéticos en cada nivel, con individuos binarios (0,1) modificados con el método de ruleta de 2 puntos, con un índice de cruce de 0,9, con un índice de mutación de 0,01 y 500 individuos.

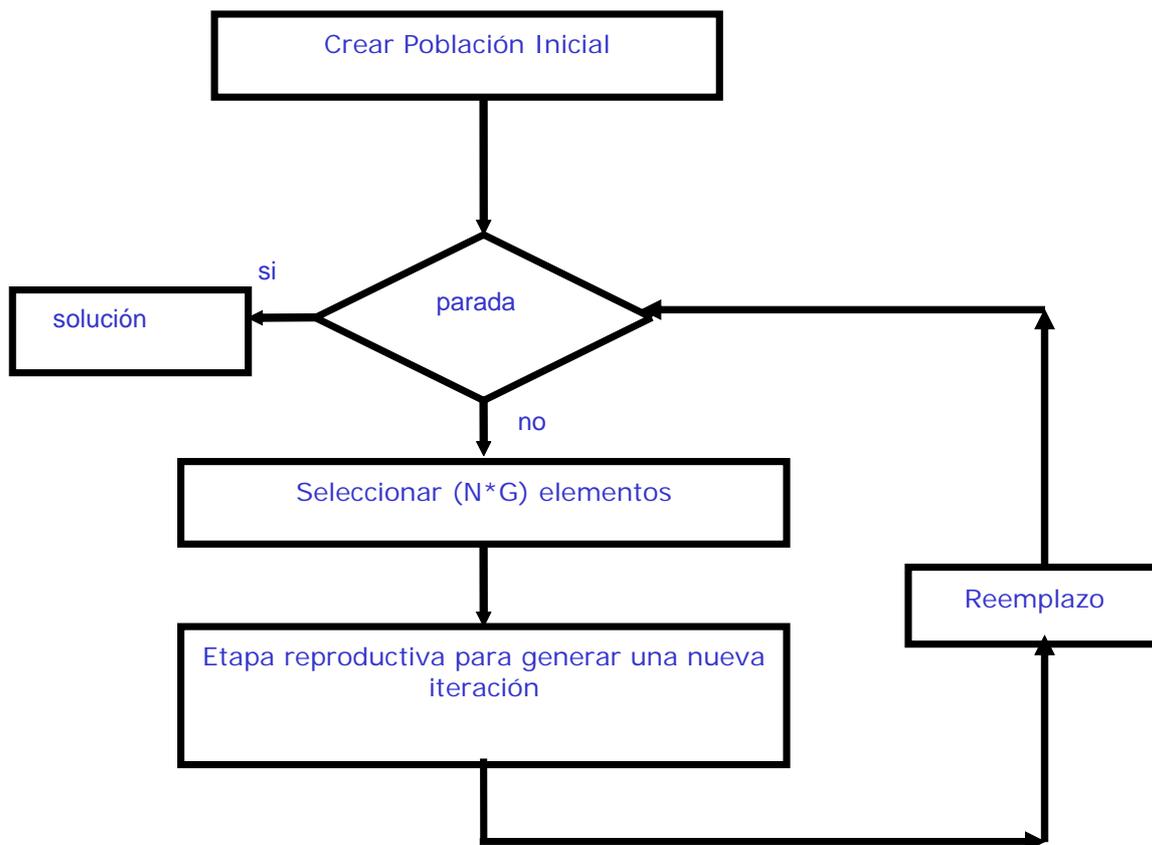


Figura 5. Bucle de un algoritmo genético.

A continuación (Figura 6) se presenta el pseudocódigo de las funciones de fitness para cada nivel.

DETALLISTA

```
sub fitdetall {
  w = .5+rand(0:1);
  media=0;
  for z (p-1..período actual p){ # bucle desde z=p-1 hasta z=p
    x=demanda[z]-invfdetall[z]+rstockdetall[z];
    y = abs(w*x); # valor absoluto
    media += y; # acumulado
  }
  detall=media/2; # lo que pide al minorista
  return 1/detall; # cuanto menos, mejor por eso se toma la inversa}

```

MINORISTA

```
sub fitmino {
  w = .5+rand(0:1);
  media=0;
  for z (p-1..p){
    x=pidedetall[z]-invfmino[z]+rstockmino[z];
    y = abs(w*(x+detall)/2);
    media += y;
  }
  mino=media/2;
  return 1/mino;
}

```

MAYORISTA

```
sub fitmayo {
  w = .5+rand(0:1);
  media=0;
  for z (p-1..p){
    x=pidemino[z]-invfmayo[z]+rstockmayo[z];
    y = abs(w*(x+detall)/2);
    media += y;
  }
  mayor=media/2;
  return 1/mayor;
}

```

FABRICA

```
sub fitfabr {
  w = .5+rand(0:1);
  media=0;
  for z (p-1..p){
    x=pidemayo[z]-invffabr[z]+rstockfabr[z];
    y = abs(w*(x+detall)/2);
    media += y;
  }
  proc=media/2;
  return 1/proc;
}

```

Figura 6. Pseudocódigo de las funciones de fitness para cada nivel.

En la tabla 1 se representa cómo se introducen los datos de partida para el Algoritmo Genético que se propone. Los resultados se ampliarán en el último apartado donde se describen las

aplicaciones numéricas, dichos datos provienen del ejemplo del juego de la cerveza de Stermán (1989).

Tabla 1. Introducción de los datos de partida para el Algoritmo Genético.

PERIODO 1	Periodo 2	Periodo 3	Periodo 4	PERIODO 5
Demanda de 1 = 95	80	80	85	Demanda de 5 = 90
Inv. inicial detalle 1 = 100	95	80	65	Inv. inicial detalle 5 = 40
Sirve detallista 1 = 95	80	80	65	Sirve detallista 5 = 40
Rotura de stock 1 =			20	Rotura de stock 5 = 50
Inv. Final detalle 1 = 5	15	0	0	Inv. final detalle 5 = 0
Pide detallista 1 = 90	65	80	85	Pide detallista 5 = 90
Inv. inicial minor. 1 = 100	90	65	40	Inv. inicial minor. 5 = 60
Sirve minorista 1 = 90	65	65	40	Sirve minorista 5 = 60
Rotura de stock 1 =		15	45	Rotura de stock 5 = 30
Inv. final minor. 1 = 10	25	0	0	Inv. final minor. 5 = 0
Pide minorista 1 = 80	40	80	85	Pide minorista 5 = 90
Inv. inicial mayor. 1 = 100	80	40	60	Inv. inicial mayor. 5 = 80
Sirve mayorista 1 = 80	40	40	60	Sirve mayorista 5 = 80
Rotura de stock 1 =		40	25	Rotura de stock 5 = 10
Inv. final mayorista 1 = 20	40	0	0	Inv. final mayorista 5 = 0
Pide mayorista 1 = 60	0	80	85	Pide mayorista 5 = 90
Inv. inicial fábrica 1 = 100	60	60	80	Inv. inicial fábrica 5 = 85
Sirve fábrica 1 = 60	0	60	80	Sirve fábrica 5 = 85
Rotura de stock 1 =		20	5	Rotura de stock 5 = 5
Inv. final fábrica 1 = 40	60	0	0	Inv. final fábrica 5 = 0
Procesa fábrica 1 = 20	0	80	85	Procesa fábrica 5 = 90

Y en la tabla 2 se presenta la solución después de utilizar el algoritmo llamado 1-1 (Stermán 1989) y el Algoritmo Genético que se ha presentado. En este caso se tiene el resultado con 10 datos. Puede comprobarse que en este caso, las soluciones que dan ambos algoritmos son similares y con elevados niveles de error.

Tabla 2. Soluciones del algoritmo 1-1 y del algoritmo genético.

INVENTARIOS FINALES, CASO 1-1	INVENTARIOS FINALES, CASO GA
Media demanda 86	Media demanda 86
Máximo 95 mínimo 80	Máximo 95 mínimo 80
Varianza demanda 34	Varianza demanda 34
Media detallista 4	Media detallista 6
Máximo 15 mínimo 0	Máximo 25 mínimo 0
Varianza detallista 34	Varianza detallista 94
Media minorista 7	Media minorista 8.8
Máximo 25 mínimo 0	Máximo 44 mínimo 0
Varianza minorista 96	Varianza minorista 309.76

Media mayorista 12	Media mayorista 25
Máximo 40 mínimo 0	Máximo 85 mínimo 0
Varianza mayorista 256	Varianza mayorista 1140
Media fábrica 20	Media fábrica 39.8
Máximo 60 mínimo 0	Máximo 75 mínimo 0
Varianza fábrica 640	Varianza fábrica 879.76
Media procesamiento 55	Media procesamiento 113.4
Máximo 90 mínimo 0	Máximo 199 mínimo 20
Varianza procesamiento 1400	Varianza procesamiento 5195.44
ROTURA DE STOCK TOTAL 95	ROTURA DE STOCK TOTAL 31

La tabla 2 será el punto de partida del agente para decidir como planifica sus envíos, los niveles de stock, etc. Esta información también la tendrán los diferentes miembros de la cadena de suministro y como cada uno conoce sus costes reales (para resolver el algoritmo genético no se han utilizado costes, como se puede ver en las funciones de fitness) decidirá lo que más le convenga.

Se puede adelantar aquí que los resultados mediocres que se presentan en la tabla 2, son ampliamente mejorados en otras ejecuciones, o cuando se superan los 20 datos.

Recientemente muchos autores han utilizado la ecuación siguiente (2) como medida del efecto látigo, donde ORATE se refiere a los pedidos de los ofertantes (minoristas fabrica etc. en general en este artículo se tomará la fábrica ya que es la mas alejada del consumidor y este coeficiente es mayor), y CONS representa las ventas del consumidor final.

$$RV_{ORATE} = \frac{\sigma^2_{ORATE} / \mu_{ORATE}}{\sigma^2_{CONS} / \mu_{CONS}} = \frac{\sigma^2_{ORATE}}{\sigma^2_{CONS}} \quad (2)$$

Con el siguiente ratio de variación RV (3), se pueden medir fácilmente las fluctuaciones en el inventario real.

$$RV_{INVR} = \frac{\sigma^2_{INVR} / \mu_{INVR}}{\sigma^2_{CONS} / \mu_{CONS}} = \frac{\sigma^2_{INVR}}{\sigma^2_{CONS}} \quad (3)$$

El software utilizado fue el lenguaje Perl (versión 5.8.6) que soporta tanto programación procedural como orientada a objetos, y cuyo intérprete se puede ejecutar en otros sistemas. De este programa se han utilizado subrutinas de algoritmos genéticos en Linux. Además se utiñizó la arquitectura Corba, que surge del consorcio Object Management Group Inc. (<http://www.omg.org>) fundado en 1989, con motivo del desarrollo de estándares para la reutilización, portabilidad e interoperatividad de software de objetos en entornos heterogéneos y distribuidos. Este consorcio creo una arquitectura llamada OMA (Object Management Architecture), del cual Corba es una parte. Al construirlo de esta forma ha permitido que esta arquitectura pueda ser ejecutada en cualquier plataforma.

3. Resultados numéricos.

A continuación se presentan las pruebas realizadas numéricamente y gráficamente. El hardware empleado fue un PC CPU AMD Duron 800 Mhz 512 Mb RAM, y el software empleado: Sistema Operativo: Linux Debian; perl v5.8.4 (con los paquetes de AI (inteligencia artificial)::Genetic, Bundle::Math::Random); gnuplot v4.0, y prototipo Perl propio de algoritmos genético. Además se realizaron simulaciones con Arena 8.0 en un ordenador con Sistema Operativo Windows XP para ilustrar y comprobar algunos cálculos (Figuras 7 y 8).

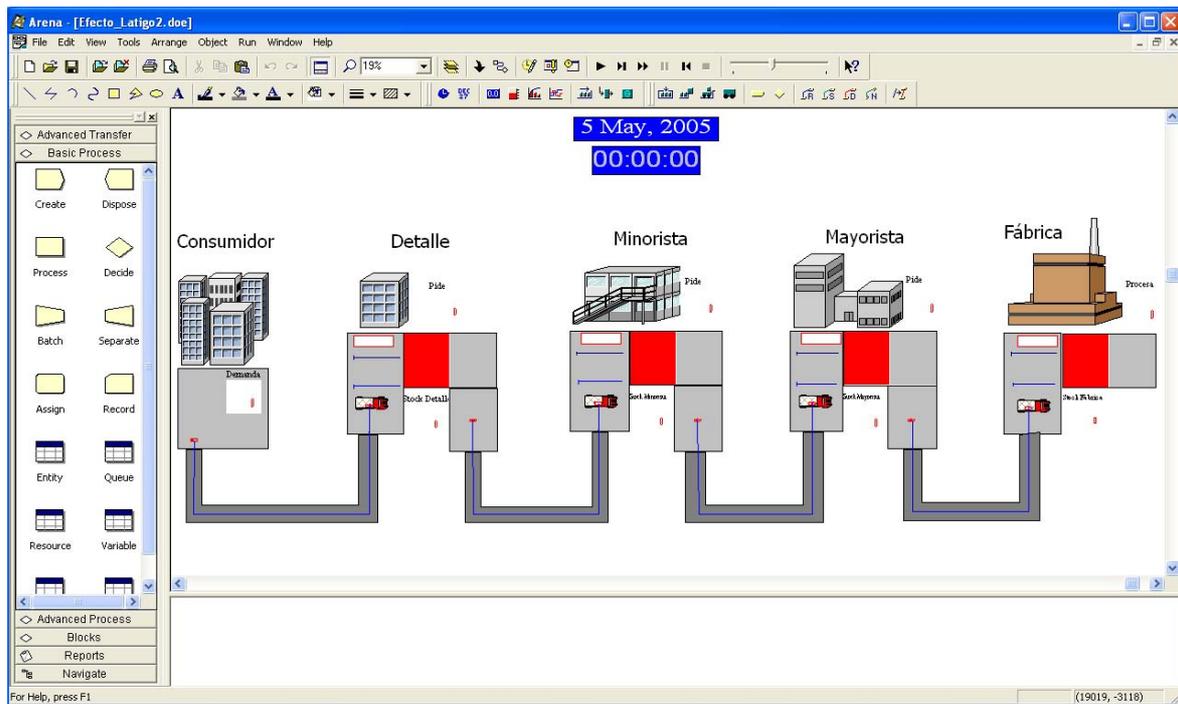


Figura 7. Simulación en Arena 8.0 de cadena de suministro.

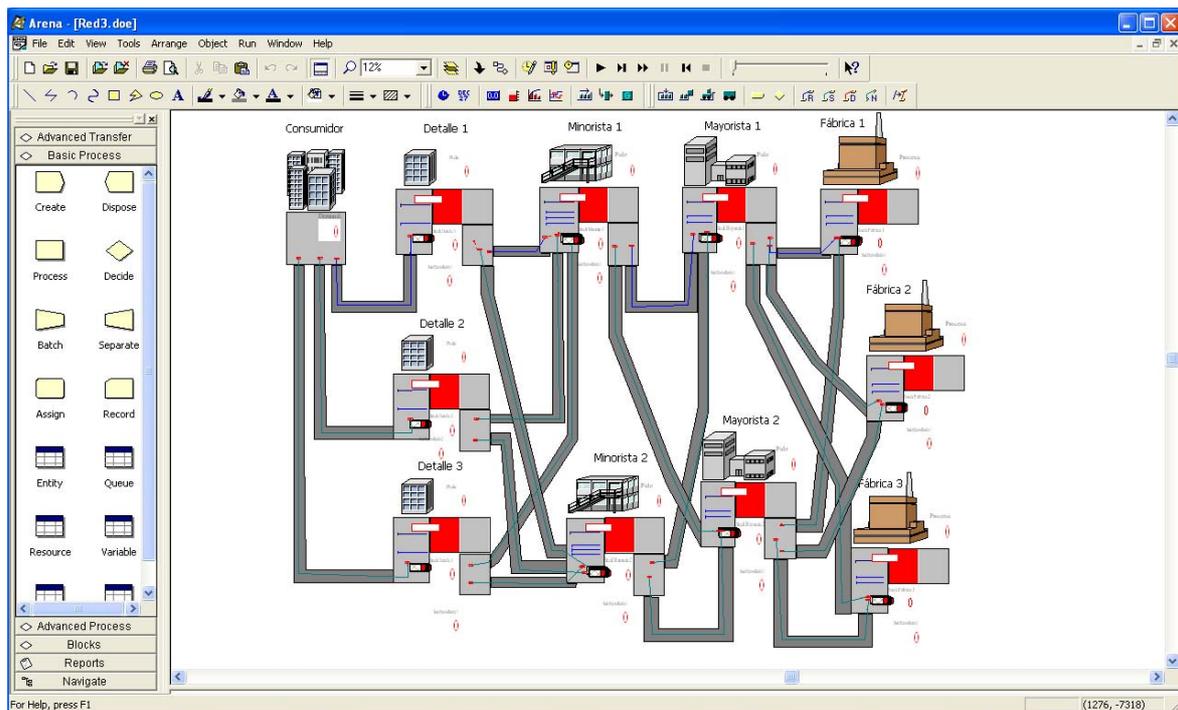


Figura 8. Prototipo de simulación en Arena 8.0 de red de suministro.

Ahora se detallan numéricamente algunas pruebas realizadas utilizando como simulador de la demanda una distribución Normal, se puede notar la mejora en la varianza al aplicar los algoritmos genéticos (tabla 3) aunque en general este método es bastante lento, y en un procesador de AMD Duron de 800Mhz puede tardar más de 30 minutos en una resolución de 20 períodos, este tiempo no es constante sino que puede haber gran diferencia con otra resolución en las mismas condiciones. El tiempo de resolución del Modelo 1-1 es como cabría esperar despreciable.

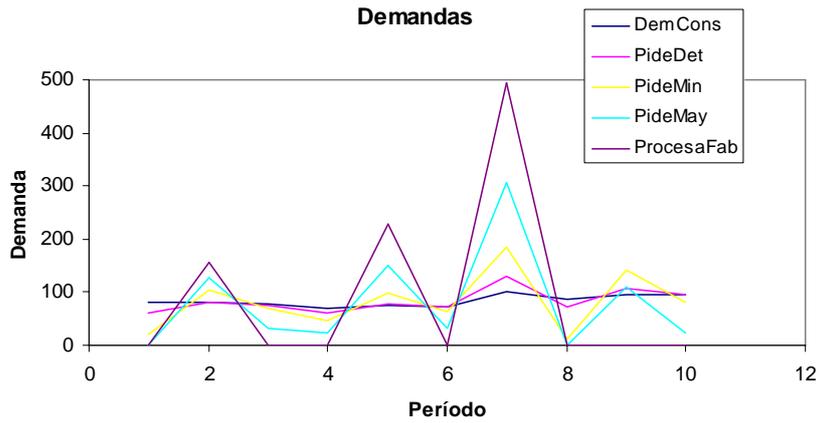
En la tabla 3, se presentan por columnas los siguientes valores: Los diferentes tipos de series. El número de casos utilizado en cada una de ellas. El tipo de modelo utilizado, que pueden ser dos: el de equilibrar (el llamado Modelo 1-1), o el algoritmo genético AG. La columna 4 representa las varianzas de la demanda de la fábrica y del consumidor final. La quinta es el tiempo que se tardó en encontrar la solución con un ordenador normal. La última columna es el medidor del efecto látigo RV_{ORATE} .

Tabla 3. Comparación entre el modelo 1-1 y AG1.

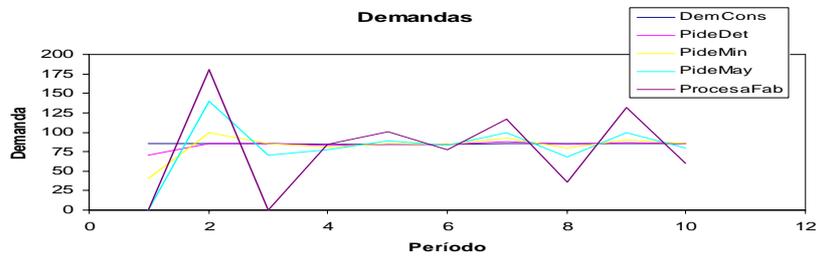
Serie	Número de casos	Modelo	$\frac{\sigma^2_{ORATE}}{\sigma^2_{CONS}}$	Tiempo	$RV_{ORATE} = \frac{\sigma^2_{ORATE}}{\sigma^2_{CONS}}$
N(85,10) Prueba 1	10	Modelo1-1	17.652,16	1 ms	28,9
		AG1	609,89	25 minutos	
N(85,10) Prueba 2	15	Modelo1-1	13.536,16	1 ms	5,3
		AG1	2.536,81	20 minutos	
N(85,10) Prueba 3	25	Modelo1-1	16.175,61	1 ms	2,5
		AG1	6.410,81	22 minutos	
N(85,10) Prueba 1	10	Modelo1-1	21.716,64	1 ms	11,2
		AG1	1.930,06	18 minutos	
N(85,5) Prueba 4	20	Modelo1-1	7.909,84	1 ms	7,01
		AG1	1.127,44	8 minutos	
N(85,1) Prueba 5	20	Modelo1-1	2.223,49	1 ms	2,23
		AG1	996,09	5 minutos	
N(85,0.5) Prueba 6	10	Modelo1-1	1.805,98	1 ms	1,82
		AG1	987,59	2 minutos	
N(85,15) Prueba 7	20	Modelo1-1	32.648,21	1 ms	12,49
		AG1	2.613,84	40 minutos	

Como se puede ver el resultado presentado en la tabla 3, demuestra la eficacia del algoritmo genético frente al modelo 1-1, en todos los casos; además cuando la σ es pequeña, por ejemplo 0,5, la varianza de los dos modelos es también pequeña pero siempre superior la del modelo sin inteligencia. El medidor del efecto látigo es de 28,9 para la prueba 1 y de 1,82 para la prueba 6.

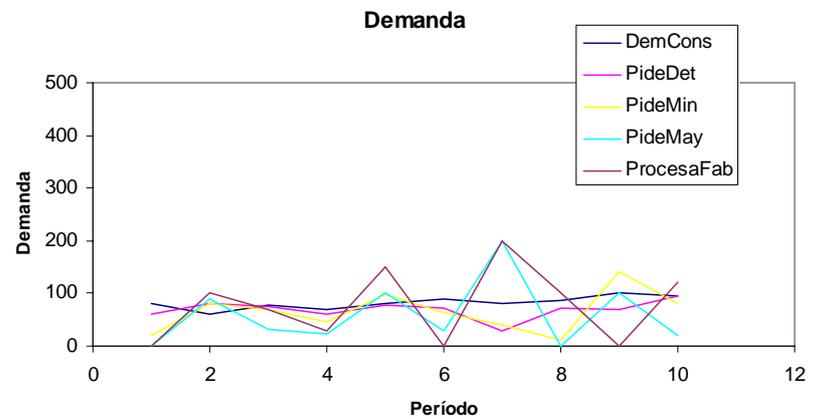
En las gráficas siguientes se presenta la evolución de la demanda con los dos métodos, prueba 1 y 6 (Gráficas 1 y 2, Gráficas 3 y 4). En la prueba 1, la varianza es mucho mayor, no obstante el suavizado que realiza el Algoritmo Genético es espectacular (Gráficas 1 y 2) ya que en algunos puntos, baja a producciones cercanas a las 0 unidades cuando la demanda real es de aproximadamente 100, y el AG no supera las 110 unidades.



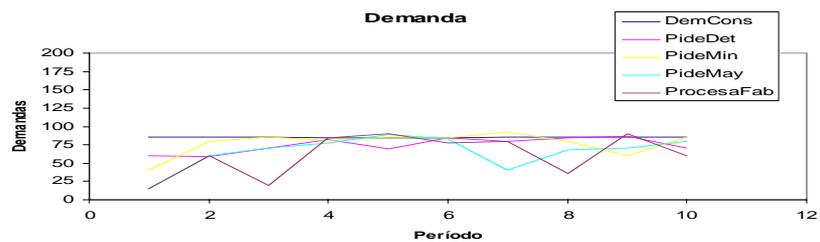
Gráfica 1. Demanda de todos los niveles con el método 1-1, con una $N(85,10)$.



Gráfica 2. Demanda de todos los niveles con el AG, con una $N(85,10)$.



Gráfica 3. Demanda de todos los niveles con el método 1-1, con una $N(85,0.5)$.



Gráfica 4. Demanda de todos los niveles con el AG, con una $N(85,0.5)$.

En la tabla 4 se presentan los resultados de los dos métodos Modelo 1-1, o AG (como en la mayoría de los cálculos es mejor el genético).

Tabla 4. Resultados en la reducción del efecto bullwhip.

Modelo 1-1	Algoritmo genético
Media fábrica 20.9	Media fábrica 21
Máximo 100 mínimo 0	Máximo 90 mínimo 0
Varianza fábrica 1028.89	Varianza fábrica 906
Inv. inicial fábr. 10 = 96	
Sirve fábrica 10 = 95	
Rotura de stock 10 =	
Inv. final fábr. 10 = 1	
Procesa fábrica 10 = 94	

4. Conclusiones.

El presente trabajo presenta una aplicación de la metodología multi-agente para disminuir el efecto bullwhip en una red de suministro. Una parte de los agentes es un algoritmo genético, que se ejecuta en varias plataformas diferentes y que según los métodos de previsión que le resulten más adecuado tomará la decisión de fabricar enviar o recibir determinadas cantidades de producto. Se ha planteado el caso de cuatro niveles de cadena de suministro y se ha comprobado que se disminuye el efecto bullwhip al utilizar herramientas avanzadas de previsión.

Referencias

- Abraham, B.; Ledolter, J. (1983). *Statistical Methods for Forecasting*. New York Wiley.
- Aviv Y., A. (2003). "Time-series framework for supply chain inventory management". *Operations Research*, 51 (2) 210–227.
- Aviv Y., A. (2002). "Gaining benefits from joint forecasting and replenishment processes". *Manufacturing and Service Operations Management*, 4 (1) 55–74.
- Aviv Y., A. (2001). "The effect of collaborative forecasting on supply chain performance". *Management Science*, 47 (10) 1326–1343.
- Baganha M., M.; Cohen, M. (1998). "The stabilizing effect of inventory in supply chains". *Operations Research*, 46 572–583.
- Blanchard, O. J. (1983). "The production and inventory behavior of the American automobile industry". *Journal of Political Economy*, 91 365–400.
- Blinder, A. S. (1986). "Can the production smoothing model of inventory behavior be saved?". *Quarterly Journal of Economics*, 101 (3) 431–454.
- Box, G. E. P.; Jenkins, G. M. (1970). *Time Series Analysis: Forecasting and Control*. San Francisco. Holden Day.
- Cachon, G. P. (1999). "Managing supply chain demand variability with scheduled ordering policies". *Management Science*, 45 (6) 843–856.
- Chen, F.; Drezner, Z.; Ryan, J.; Simchi-Levi, D. (2002). "Quantifying the bullwhip effect in a simple supply chain: the impact of forecasting, lead-times, and information". *Management Science*, 46 (3) 436–443.
- Childerhouse, P.; Disney, S. M.; Towill, D. R. (2004). "Tailored toolkit to enable seamless supply chains". *International Journal of production Research*, 42 (17) 3627-3646.
- Dejonckheere, J.; Disney, S. M.; Lambrecht, M. R.; D.R. Towill, D. R. (2003). "Measuring and avoiding the bullwhip effect: a control theoretic approach". *European Journal of Operational Research*, 147 (3) 567–590.

- Dejonckheere, J.; Disney, S. M.; Lambrecht, M. R.; Towill, D. R. (2004). "The impact of information enrichment on Bullwhip effect in supply chain: A control engineering perspective". *European Journal of Operational Research*, 153 727-750.
- Disney, S. M. (2001). *The production and inventory control problem in vendor manager inventory supply chain*. Ph. D. Thesis, Cardiff Business School, Cardiff University, UK.
- Disney, S. M.; Towill, D. R. (2002). "Transfer function analysis of forecasting induced bullwhip in supply chain". *International Journal of Production Economics*, 78 133-144.
- Disney, S. M.; Towill, D. R. (2003a). "The effect of Vendor Managed Inventory (VMI) dynamics on the Bullwhip effect in supply chain". *International Journal of Production Economics*, 85 199-215.
- Disney, S. M.; Towill, D. R. (2003b). *On the Bullwhip and inventory variance produced by an ordering policy*. *Omega*, 31 157-167.
- Disney, S. M.; Grubbstrom, R. W. (2004). "Economic consequences of a production and inventory control policy". *International Journal of Production Research*, 42 (17) 3419-3431.
- Disney, S. M.; Naim, M. M.; Potter, A. (2004). "Assessing the impact of e-business on supply chain dynamics". *International Journal Economics*, 89 109-118.
- Disney, S. M.; Towill, D. R.; Velde, W. (2004). "Variance amplification and the golden ratio in production and inventory control". *International Journal of Production Economics*, 90 295-309.
- Drezner, Z.; Ryan, J.; Simchi-Levi, D. (1996). *Quantifying the bullwhip effect: the impact of forecasting, leadtime and information*. Working Paper, Northwestern University, Evanston, IL.
- Forrester, J. W. (1961). *Industrial dynamics*, MIT Press, Cambridge, MA.
- Cachon, G. P.; Lariviere, M. A. (1999). "Capacity choice and allocation: strategic behavior and supply chain performance". *Management Science*, 45 (8) 1091-1108.
- Graves, S. C. (1999). "A single-item inventory model for a nonstationary demand process". *Manufacturing & Service Operations Management*, 1 50-61.
- Hosoda, T.; Disney, S. M. *On variance amplification in a tree echelon supply chain with minimum square error forecasting*. *Omega* in press.
- Kahn, J. (1987). "Inventories and the volatility of production". *The American Economic Review*, 77 667-679.
- Kaminsky, P.; Simchi-Levi, D. (1998). *A new computerized beer game: a tool for teaching the value of integrated supply chain management, POMS series in technology and operation management, Global supply chain and technology management*, USA, 1.
- Kelle P.; Milne, A. (1999). "The effect of (s,S) ordering policy on the supply chain". *International Journal of Production Economics*, 59 113-122.
- Lee, H.L.; Padmanabhan, V.; Whang, S. (1997). "The bullwhip effect in supply chains", *Sloan Management Review*, Spring, pp. 93-102.
- Makridakis, S. (1993). "Accuracy Measures: Theoretical and Practical Concerns". *International Journal of Forecasting*, 9 527-529.
- Reilly, D. P. (1980). *Experience with an automatic Box-Jenkins Modeling in Time Series Analysis*, O. D. Anderson, Amsterdam: North-Holland.
- Slutsky, E. (1937). "The summation of random causes as the source of cyclic processes". *Econometrica*, 5 105-46.
- Sterman, J.D. (1989). "Modelling managerial behaviour: misperceptions of feedback in a dynamic decision making experiment", *Management Science*, Vol. 35 , No. 3, pp. 321-339.
- Sterman, J.D. (1992). "Teaching Takes Off - Flight Simulators for Management Education. The Beer Game", *OR/MS Today*, 19 (5), pp. 40-44.

- Sterman, J.D. (1995). "The beer distribution game. In: J. Heineke and L. Meile, Editors", *Games and exercises for operations management*, Prentice Hall, Englewood Cliffs, NJ, pp. 101-112.
- West, K. D. (1986). "A variance bounds test of the linear quadratic inventory model". *Journal of Political Economy*, 94 (4) 374-401.
- Wold, H. (1938). *A study in the analysis of stationary time series*. Estocolmo: Almqvist & Wiksell.
- Yule, G.U. (1926). "Why do we sometimes get nonsense-correlations between time series? A study in sampling and the nature of time series". *Journal of Royal Statistical Society*, 89 1-64.