ALGORITMOS GENÉTICOS APLICADOS EN LOS SISTEMAS DE PRODUCCIÓN TIPO PULL

Genetic Algorithms Applied to Pull Production Systems

RESUMEN
Este artículo presenta el desarrollo de un algoritmo genético basado en información obtenida en la simulación de un caso hipotético de un sistema de manufactura tipo *Pull*. El objetivo del algoritmo fue determinar el número de Kanbans de producción en celdas de manufactura. Se utilizó un "metamodelo auxiliar" para expresar la relación funcional entre la cantidad de *Kanbans* y el porcentaje promedio de demanda satisfecha ("% de Throughput") el cual fue incorporado en el algoritmo genético. El algoritmo genético logró una configuración de *kanbans* de producción que produjo un 88,04% de *Throughput* y un nivel de inventario en proceso moderado.
PALABRAS CLAVES: Algoritmo Genético, Demanda satisfecha, Kanbans, Sistema Pull, Metamodelo.

ABSTRACT
This article show the development of Genetic Algorithmic based in the information obtained through the simulation of a hypothetical case of a Pull Manufacturing System. Determination of the number of Production Kanbans in each Manufacturing Cells is the goal of the algorithm here presented. An auxiliary mathematical model "Metamodel" is used as the functional relation between the quantity of "Production Kanbans" and the "% of Throughput" and then incorporated in the genetic algorithm. Finally, the Genetic Algorithmic achieved a production kanban configuration that produces an 88.04% of Throughput and a moderate work in process.
KEYWORDS: Efficiency, Genetic Algorithm, Kanban, Metamodel, Pull System, Throughput.

SERGIO FERNÁNDEZ HENAO
Magíster en Investigación Operativa y Estadística.
Ingeniero Industrial.
Universidad Tecnológica de Pereira
Docente Asistente.
sfernandez@utp.edu.co

JOSÉ SOTO MEJÍA
Ph. D. Ingeniería de Computación
Magíster en Investigación Operativa y Estadística.
Docente Titular
Universidad Tecnológica de Pereira
jomejia@utp.edu.co

1. INTRODUCCIÓN.

Un sistema de manufactura tipo "Pull", se apoya en gran medida por el flujo de las tarjetas "Kanban" en cada estación de trabajo, para autorizar la producción respectiva. Es conocido que dicho sistema de manufactura tiene entre sus objetivos principales la disminución de inventario en proceso "WIP" y lograr a la vez tiempos razonables de entrega de pedidos "Lead Time"[1].

Para dicha programación de Kanbans, el Sistema de Producción Toyota ha elaborado ciertas formulas matemáticas para indicar el número de estas tarjetas que debe ser asignado para determinado proceso productivo [2]. Sin embargo, se ha observado que la tarea de asignar el número de kanbans a dicho proceso productivo en donde se tiene varias células de trabajo y a la vez varios productos, se convierte en una labor de gran dificultad para el personal encargado de su planeación. Esto debido a que a la hora de realizar la correspondiente asignación, se genera un considerable número de alternativas de solución que en la mayoría de las veces es imposible contemplarlas en su totalidad. Por ejemplo, si se tuviera un sistema de manufactura con tan sólo 12 artículos para producir y si se asume que deben pasar únicamente por tres procesos productivos, se debe asignar 12 valores por cada centro de trabajo, lo que genera un total de 36 asignaciones de Kanbans de producción.

Adicionalmente, si se permitiese un rango posible de asignación del número de Kanbans de [1 a 6][1], se tendrían 6^{36} (36 variables de decisión con 6 posibles valores de asignación), lo que equivale a "$1.03 \text{ X } 10^{28}$" combinaciones posibles de asignación de Kanbans.

Esta astronómica cantidad de alternativas de solución genera un problema de explosión combinatorial. Un computador con un procesador que opere a una velocidad de una Giga[2], podría analizar $3.15 \text{ X } 10^{16}$ alternativas por año, es decir, que para poder analizar todas las alternativas, $1.03 \text{ X } 10^{28}$, el procesador tardaría algo más de 22 edades del universo[3].

Fecha de Recepción: 27 de Abril de 2011
Fecha de Aceptación: 30 de Agosto de 2011

[1] Este rango indica la cantidad de unidades autorizadas para producir del producto i en el centro de trabajo j, según el número de Kanbans.
[2] Una Giga de velocidad en un procesador equivale a realizar $1 \text{ X } 10^{9}$ operaciones por segundo.
[3] La edad del universo está contemplada en 14.500 millones de años, es decir "$1.45 \text{ X } 10^{10}$" años.

En el presente trabajo se muestran las ventajas de utilizar la Metaheurística de Algoritmos genéticos para conseguir encontrar una combinación apropiada de número de kanbans, en un tiempo razonable.

Una combinación apropiada de número de kanbans que deben circular en las correspondientes estaciones de trabajo, intenta garantizar un equilibrio entre el inventario total en proceso "WIP" y el porcentaje promedio de demanda satisfecha "% de Throughput", para todos los tipos de productos solicitados [3].

Con base en la información obtenida a través de la simulación de un caso hipotético de un sistema de manufactura tipo *Pull,* ilustrado en la figura 1, se desarrolló un modelo matemático auxiliar (Metamodelo) que permitió determinar la relación funcional existente entre la cantidad de "Kanbans de Producción" (asignados a las estaciones de trabajo) y el porcentaje promedio de demanda satisfecha para todos sus productos ("% de Throughput").

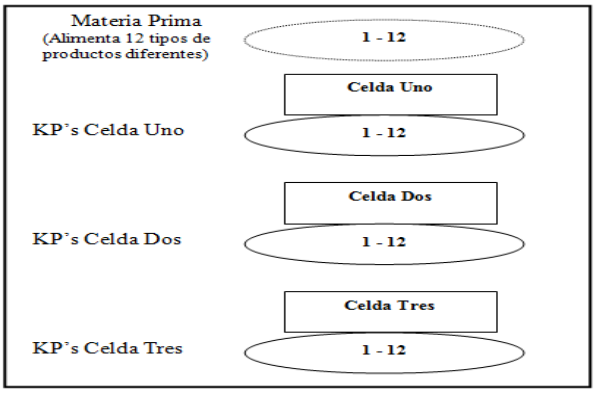

Figura 1. Línea de manufactura controlada por Kanbans de Producción.

La figura 1 anterior ilustra 3 celdas de manufactura, cada una controlada por Kanbans, con un rango posible de 1 a 6 (KPs) y en las cuales se procesan 12 tipos diferentes de productos.

Mediante la simulación del modelo anterior, con solo tres niveles (uno, tres y seis kanbans), donde cada nivel se encarga de autorizar la fabricación de una, tres o seis unidades del respectivo artículo, se consiguió ajustar el siguiente modelo, ver detalles del proceso en [4].

$$\mathbf{Ln(Y)} = 2{,}46 + 0{,}0133\mathbf{TK_1} + 0{,}0213\mathbf{TK_2} + 0{,}0354\mathbf{TK_3} - 0{,}000166\mathbf{TK_1^2} - 0{,}000238\mathbf{TK_2^2} - 0{,}000360\mathbf{TK_3^2} + 0{,}000003\ \mathbf{TK_1TK_2TK_3} \quad (1)$$

Donde "**TK1**, **TK2** y **TK3"** corresponde al total de Kanbans permitido en la celda respectiva, cuando se alimenta el modelo de manera sucesiva con los doce artículos y "**Y"**, representa la variable de desempeño (porcentaje promedio de demanda satisfecha "% Throughput").

La anterior ecuación 1, permite evaluar la variable de desempeño, "**Y"**, para asignaciones dadas del número de kanbans en cada una de las tres celdas, pero no establece la asignación óptima de kanbans en cada celda (mayor % de *Throughput* y menor nivel de inventario en el sistema).

Para encontrar la asignación óptima se tendría que realizar una evaluación exhaustiva de todas las combinaciones posibles de diferente nivel de número de kanbans en cada una de las celdas de manufactura para cada uno de los 12 productos. Como ya se mencionó esto tardaría más de 22 edades del universo.

Para explorar este gran espacio de búsqueda se utilizará la metaheuristica conocida con el nombre de algoritmo genético que en un tiempo razonable deberá encontrar una solución de buena calidad [5].

2. CONCEPTOS GENERALES.

2.1. Algoritmo Genético.

En síntesis los Algoritmos Genéticos trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor ó puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma.

Este cruce producirá nuevos individuos (Descendientes de los anteriores), los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones.

De esta manera se produce una nueva población de posibles soluciones, la cual, reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así, a lo largo de las generaciones las buenas características se propagan a través de la población. Esto favorece el cruce de los individuos mejor adaptados y van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el Algoritmo Genético ha sido bien diseñado, la población convergerá hacia una solución óptima del problema [6].

3. APLICACIÓN DEL ALGORITMO GENÉTICO.

La figura 2, presenta la estructura que sigue el algoritmo desarrollado e implementado en este caso de estudio [7].

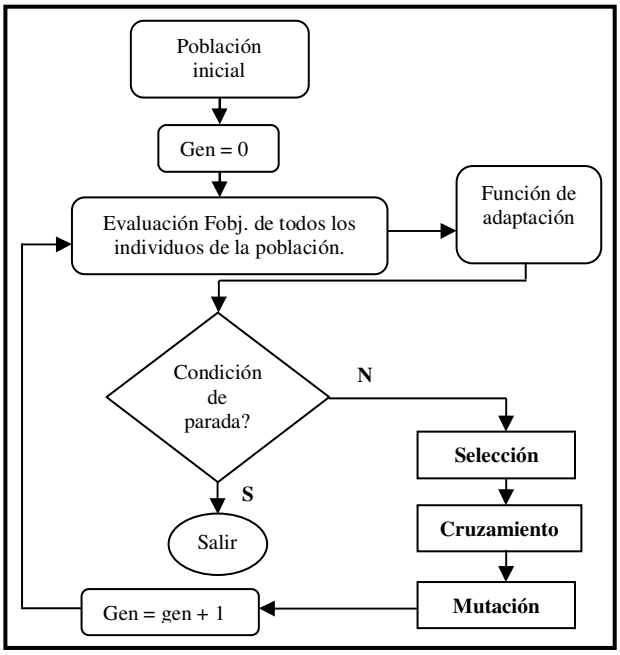

Figura 2. Algoritmo genético con sus operadores de evolución[4].

3.1. Población inicial y codificación de la información. La población inicial es una matriz generada de manera aleatoria, en la cual, la cantidad de filas es controlada por un parámetro denominado "tampoblacion" (Tamaño de la población) y la cantidad de columnas es controlada por el parámetro "numgenes" (Número de genes en cada cromosoma). Este parámetro a su vez, está controlado por la cantidad de celdas de manufactura "nummaquinas" y la cantidad de artículos a procesar "numproductos".

Es así, como cada vector de la matriz que forma la población inicial, será una alternativa de solución. Esta alternativa indica en bloques de tres, la cantidad de "Kanbans de Producción" que se debe asignar para el producto i en la celda de manufactura j. Por ejemplo, si se cuenta con tres celdas de manufactura y dos productos a procesar, una alternativa de solución de dicha matriz podrá tomar la siguiente forma:

3 1 4 3 5 6

Dicho vector indica que para el primer producto se debe asignar tres kanbans de producción en la primera celda de manufactura, un kanban en la segunda celda y cuatro kanbans en la última celda; para el segundo producto su asignación será de tres kanbans en la primera celda, cinco kanbans en la segunda y seis en la última celda de manufactura.

Con base a lo anterior, se puede observar dos características importantes para ser tenidas en cuenta en la programación de este algoritmo. La primera es que el tipo de codificación utilizado en este algoritmo Genético es de tipo "Entero". La segunda es que varios genes pueden tener el mismo valor, ya que cada uno indica la cantidad de kanbans a asignar, por lo que se puede dar el caso en donde una alternativa de solución asigne cantidades iguales de kanbans a sus productos en las correspondientes celdas de manufactura.

El valor de la asignación de kanbans para este estudio, se limita a un rango de mínimo un kanban y máximo seis kanbans por cada producto en su correspondiente celda de manufactura. Por tal motivo cada gen de las alternativas de solución solo puede tomar valores entre dicho rango. Esto se logra utilizando un parámetro llamado "numkanban" el cual controla el cumplimiento de dicho rango al momento de generar el valor aleatorio.

A continuación se presenta un esquema de pseudo código del algoritmo para formar la población inicial.

```
tampoblacion=50;
numproductos=12;
nummaquinas=3;
numgenes=(nummaquinas*numproductos);
numkanban=6;
poblacion=zeros (tampoblacion, numgenes);
for i=1 :tampoblacion
   for j=1 :numgenes
      poblacion(i,j)=round(rand*(numkanban-1)+1);
   end
end
poblacion
```

Esta codificación indica que se va a generar una matriz llamada "poblacion", la cual se conforma por 50 filas y 36 columnas, en donde cada celda contiene un valor aleatorio entre 1 y 6. De esta manera, se genera una población inicial con 50 alternativas de solución para 12 productos en sus tres celdas de manufactura.

Para que la metaheuristica pueda encontrar soluciones mejores que las existentes en la población actual, es necesario modificar los individuos a través de la aplicación de estrategias de selección de los mejores.

Esta metaheurística utiliza como estrategia inicial de este proceso, la selección de los dos mejores individuos de la población inicial para que jueguen el rol de padres de otros individuos (soluciones).

3.2. Selección de Padres.

[4] Granada, M. (2009), "Algoritmos Evolutivos y Técnicas Bioinspiradas: De la teoría a la práctica" Universidad Tecnológica de Pereira. Cap. 3 Pág. 82.

Para este proceso se debe plantear en primera instancia una función objetivo que mida la calidad de los individuos. Como función objetivo se tomó como base la evaluación de la variable de desempeño (porcentaje promedio de demanda satisfecha "% Throughput") dada en la ecuación 1. Para evaluar la calidad de un individuo de la población la ecuación mencionada fue ajustada para que contemplase penalizaciones que limitaran el crecimiento del número de kanbans en el sistema, y por ende el inventario en proceso.

La siguiente ecuación 2, muestra la función objetivo ya ajustada, la cual se busca maximizar.

$$\text{Valor Función objetivo} = (W_0 * Y) - (W_i * P) \qquad (2)$$

Donde:

W_0= Vector de Pesos de importancia asignado a la variable dependiente (porcentaje promedio de demanda satisfecha "% Throughput").
Y= Escalar, Porcentaje promedio de la demanda satisfecha (% Throughput) encontrado a través de la ecuación 1.
W_i= Vector de pesos de penalización para el total de Kanbans asignados en la celda de manufactura i.
P= Población Total o matriz que contiene el total de Kanbans asignados a cada celda de manufactura para el total de alternativas exploradas.

De esta manera, se evalúa cada alternativa de solución a través de su correspondiente valor de la función objetivo. La estructura de la ecuación número (2), tiene la finalidad de encontrar un equilibrio entre el número total de kanbans asignados a cada celda de manufactura y el porcentaje promedio de demanda satisfecha. Esto se puede observar, ya que la ecuación 1, permite que a mayor cantidad de Kanbans, mayor sea el porcentaje de Throughput; sin embargo la función objetivo planteada (ecuación 2), controla el crecimiento del número de kanbans mediante las penalizaciones agregadas (vector **W_i**).

Para la elección de los padres, se realiza una selección por torneo, en donde las parejas seleccionadas aleatoriamente son sometidas a un proceso de comparación del valor de la función objetivo obtenida con cada alternativa de solución. En este proceso se utiliza un criterio de maximización, ya que se busca encontrar aquella solución que logre un importante porcentaje de demanda satisfecha sin tener que elevar los niveles de inventario en proceso. El elitismo prevalece en este proceso ya que aquellas alternativas de solución con menor valor de la función objetivo se irán eliminando de la población solución y las de mejor desempeño se irán replicando en las próximas generaciones.

La estrategia anterior de selección padres no tiene la capacidad de crear nuevas soluciones dentro de la población, solamente pueden hacer copias de las buenas soluciones a expensas de soluciones de menor calidad. La creación de nuevas soluciones es una tarea realizada por otras estrategias cuyo objetivo es incorporar diversidad en el proceso de optimización. La incorporación de estas estrategias en la metaheuritica de los algoritmos genéticos es conocida con el nombre de operadores de cruzamiento y de mutación.

3.3. Operador de cruzamiento y recombinación.

Para realizar este proceso de variación, se establecieron tres puntos de corte y se utilizó un método de recombinación "Binario", ya que, este tipo de codificación permite tener genes de un mismo cromosoma con valores iguales. Situación que se ajusta al sistema analizado, debido a que los productos pueden tener la misma cantidad de Kanbans de Producción en las diferentes celdas de manufactura.

Para indicar la posición de los puntos de corte se generan números aleatorios entre la cantidad de genes que compone el vector de solución. En la figura 3, se observa la forma como funciona este operador generando para una pareja de vectores dos nuevos descendientes.

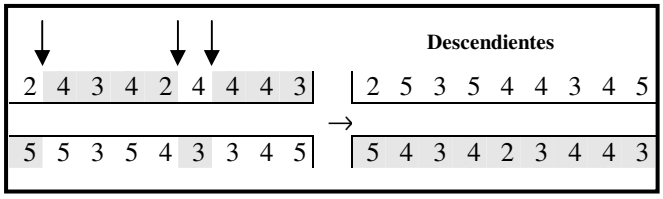

Figura 3. Cruzamiento multipunto con tres puntos de corte.

De esta manera se aplica el proceso de recombinación en donde cada vez que se genera un descendiente denominado "Hijo", es comparado con el vector de la población que obtuvo la peor función objetivo (ecuación 2). El mejor de los dos queda ubicado en la matriz que conforma la población para continuar con el torneo de selección y el perdedor es eliminado de dicha población.

3.4. Operador de mutación.

Para ejecutar esta parte del Algoritmo Genético se uso un valor de 0.3 como la tasa de mutación y se le asignó a un parámetro llamado "tasamutacion". Es así, como después de haber realizado la Selección, el Cruzamiento y la Recombinación respectiva, se genera un número aleatorio entre 0 y 1 para ser comparado con la tasa de mutación. Si el valor generado está por debajo de dicha tasa, se inicia el proceso de Mutación.

La Mutación para este caso de estudio opera a través de la generación de dos números aleatorios (Éstos están en el rango del número de asignaciones de kanbans de Producción establecidas) con los que se identifica una porción del cromosoma o alternativa de solución, para ser rotada provocando un nuevo individuo. Este nuevo individuo mutado reemplaza el antiguo y entra a ser parte

de la población para ser comparado el valor de su función objetivo con las demás. Cabe anotar que este proceso se hace con cada descendiente para saber si se muta o no, y así, permitir una mayor exploración del espacio de soluciones.

3.5. Criterio de parada.

Se estableció un parámetro llamado "maxgen" encargado de identificar la cantidad de ciclos de búsqueda a generar. En la actual implementación se uso un número de generaciones máximo de 300. En cada iteración se buscó obtener la mayor función objetivo, la cual, se fue comparando con cada resultado final hasta elegir la solución de mejor calidad que indicara para cada producto la cantidad de Kanbans de Producción en su correspondiente celda de manufactura, de tal manera que garantizara un equilibrio entre el inventario total en proceso y el porcentaje promedio de demanda satisfecha.

4. RESULTADOS.

Escenario	Alternativas Individuales			Alternativas Totales			Valor F. Objetivo
	Ki1	Ki2	Ki3	TK1	TK2	TK3	
1	3	1	3	36	12	36	166,174
2	3	6	3	36	72	36	51,342
3	6	6	3	72	72	36	53,182
4	3	1	1	36	12	12	132,546
5	3	1	6	36	12	72	0
6	3	3	6	36	36	72	0
7	1	3	1	12	36	12	109,003
8	6	3	6	72	36	72	0
9	1	6	3	12	72	36	0
10	3	3	3	36	36	36	284,504
11	1	3	6	12	36	72	0
12	6	1	6	72	12	72	0
13	6	1	3	72	12	36	0
14	1	1	1	12	12	12	150,214
15	1	1	3	12	12	36	139,615
16	6	6	1	72	72	12	0
17	6	1	1	72	12	12	0
18	6	3	1	72	36	12	0
19	6	6	6	72	72	72	204
20	1	3	3	12	36	36	171,395
21	1	6	6	12	72	72	0
22	3	3	1	36	36	12	117,870
23	1	6	1	12	72	12	0
24	3	6	1	36	72	12	0
25	6	3	3	72	36	36	151,990
26	1	1	6	12	12	72	0
27	3	6	6	36	72	72	0
28	2	3	4	24	36	48	218,386
29	3	5	2	36	60	24	89,305
30	**4**	**4**	**4**	**48**	**48**	**48**	**347,446**
31	3	2	4	36	24	48	213,849
32	2	5	3	24	60	36	120,725

Tabla 1. Estimación de la función objetivo.

En la implementación del algoritmo genético se usaron como parámetros iniciales los valores citados en el numeral 3.1. La tabla 1, presenta la evaluación inicial de la función objetivo, con base a una configuración de kanbans establecida para tres posibles valores (1, 3 y 6) en cada celda para cada producto. Como se puede observar en dicha tabla, se generaron 32 escenarios equivalentes a las 27 posibles combinaciones entre los tres valores mencionados y 5 más escogidas aleatoriamente.

Igualmente, en esta tabla 1 se observa que la mejor asignación, se presenta en el escenario 30 obteniendo un valor de la función objetivo de 347.446. Esta solución indica que se deben asignar cuatro Kanbans de Producción para cada producto en cada celda de manufactura, con lo cual se logra un porcentaje promedio de demanda satisfecha "% Throughput" de 80.76% (valor que se obtiene al evaluar esta combinación de kanbans mediante la ecuación 1). A los valores negativos de la función objetivo se le asignó un valor "cero". (Esto debido a que la penalización sobre el total de Kanbans impide que el porcentaje de demanda satisfecha se incremente desmesuradamente a costa de niveles de inventario altos).

La tabla 2, presenta un resumen de 10 corridas del Algoritmo Genético (cada una con 300 generaciones). En esta tabla 2, se muestra el total de Kanbans "TKi" asignados en cada estación de trabajo y el mejor valor de la función objetivo encontrado tras las 300 generaciones en cada una de las 10 corridas.

Corrida	TK1	TK2	TK3	Mejor F. Objetivo
1	52	49	49	355.19
2	51	48	48	352.25
3	55	52	50	360.49
4	50	47	47	348.90
5	49	46	47	346.43
6	56	53	51	362.37
7	47	49	47	347.70
8	51	49	48	353.08
9	**57**	**53**	**52**	**363.56**
10	54	50	50	358.54

Tabla 2. Resumen de las corridas del algoritmo genético.

En la tabla 2, se puede observar que la mejor alternativa de solución se da en la novena corrida con un valor de

363.56 superando la solución del escenario 30 referenciado en la tabla 1.

En la tabla 3 siguiente se resume en detalle la asignación de kanbans para cada producto correspondiente a la mejor corrida (la 9) presentada en la tabla 2. Donde $KP_{i1,}$ significa el número de kanbans asignado al producto i-ésimo en la celda de manufactura 1.

Producto_i	KP_{i1}	KP_{i2}	KP_{i3}
1	4	3	5
2	5	5	3
3	3	5	4
4	6	5	4
5	5	4	4
6	6	3	5
7	6	4	5
8	5	5	6
9	4	3	5
10	5	5	3
11	5	5	3
12	3	6	5

Tabla 3. Asignación de kanbans de producción para cada producto.

Con la asignación de Kanbans de Producción de la tabla 3, se obtiene un porcentaje promedio de demanda satisfecha de 88.04%, lo cual, corrobora nuevamente que esta solución es de mejor calidad respecto a la solución de la tabla 1 que generó un porcentaje promedio de demanda satisfecha de 80.76%.

4. CONCLUSIONES

Esta investigación muestra que la utilización de una función analítica de desempeño que ajusta a datos obtenidos mediante experimentos de simulación, aunada a su exploración mediante algoritmos genéticos, es un enfoque apropiado para resolver el problema de la asignación del número óptimo de kanbans en un sistema de producción tipo *pull.*

Es también de interés observar como la función objetivo ajustada presentada en la ecuación dos y posteriormente insertada como criterio de calidad en el Algoritmo Genético, garantiza un equilibrio entre el porcentaje de demanda satisfecha y el nivel de inventario en proceso.

Lo anterior se logra gracias a la competencia de dos factores, uno que intenta incrementar la función objetivo ajustada y el otro que trata de impedir este crecimiento. En el literal a) se describe el factor que tiende a incrementar la función y en el b) el factor que impide su crecimiento:
a) nivel de importancia asignado a la variable dependiente porcentaje promedio de demanda satisfecha "% Throughput". ($\mathbf{W_0}$= Vector de Pesos de importancia).
b) las penalizaciones asignadas a los totales de Kanbans asignados a cada celda de manufactura i ($\mathbf{W_i}$= Vector de pesos de penalización) .

5. BIBLIOGRAFÍA

[1] PHILIPOOM, P.R., REES, L.P., TAYLOR, B.w. and HUANG, P.Y. "An Investigation of the Factors Influencing the Number of Kanbans Required in the Implementation of the JIT Technique with Kanbans", International Journal of production Research, vol. 25, no. 3, 457-472. 1997.

[2] LIKER, Jeffrey. MEIER, David. The Toyota Way Field Book, A Practical Guide For Implementing Toyota's 4Ps. Editorial McGraw Hill. 2006.

[3] RESTREPO C. Jorge, PEREZ V., Paula A., CRUZ T. E. Definición, Clasificación y Aplicación del sistema Kankan. Editorial Papiro. 2007.

[4] FERNÁNDEZ Sergio, SOTO José, Los Metamodelos de Regresión en Simulación con Aplicación en Sistemas de Manufactura. Scientia et Technica, vol 47. Universidad Tecnológica de Pereira., Marzo 2011.

[5] BLUE, C., ROLI, A. "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison", ACM Computing Surveys. 2003.

[6] GOLDBERG, D., DEB, K. "A Comparison of Selection Schemes used in Genetic Algorithms", In Foundations of Genetic Algorithms 1. pp. 69-93. 2004.

[7] GRANADA, M. "Algoritmos Evolutivos y Técnicas Bioinspiradas: De la teoría a la práctica" Universidad Tecnológica de Pereira. 2009.