

Optimización de sistemas a medida mediante el algoritmo Cross-Entropy*

Pedro L. González Rodríguez¹, José M. Framiñán Torres¹, Paz Pérez González¹, Rafael Ruiz Usano¹

¹ Dpto. de Organización Industrial y Gestión de Empresas. Escuela Superior de Ingenieros de Sevilla. Camino de los descubrimientos, s/n, 41092 Sevilla. pedroluis@esi.us.es, jose@esi.us.es, pazperez@esi.us.es, usano@esi.us.es.

Resumen

Uno de los mecanismos de control de la producción más recientes son los sistemas de control de la producción a medida basados en tarjetas, los cuales han demostrado su efectividad en gran número de casos. La optimización de este tipo de sistemas puede ser formulada como un problema de optimización combinatoria. Por otra parte, el método cross-entropy (CE) ha mostrado ser útil en una gran diversidad de problemas, incluyendo los problemas de optimización combinatoria del tipo NP-hard. Sin embargo no hay muchos trabajos acerca del método CE en optimización combinatoria. En este trabajo abordamos la optimización de un sistema de control de la producción a medida basado en tarjetas empleando el algoritmo CE bajo optimización multicriterio. Para optimizar estos sistemas proponemos un algoritmo, el cual es probado en un escenario sacado de una fábrica de componentes de automoción en España.

Palabras clave: Optimización, sistemas de control de la producción, optimización multicriterio, cross-entropy.

1. Introducción

Durante los últimos años, en el contexto del control de la producción, los sistemas pull han sido estudiados por los investigadores y exitosamente aplicados en la práctica. Desde el primer sistema Kanban de la factoría de Toyota (Monden, 1983) han surgido diferentes sistemas pull, también denominados “basados en tarjetas” Gershwin (2000). Respecto a los sistemas basados en tarjetas, el sistema más general, hasta ahora, es el denominado sistemas a medida basado en tarjetas, denominados en terminología anglosajona “Customized token-based systems”, CTBS (ver Gaury, *et al.*, 2000, or Gaury, *et al.*, 2001 o González, 2006).

CTBS intentan controlar la cantidad máxima de trabajo entre cada par de estaciones por medio de tarjetas, de manera similar al sistema Kanban, el cual intenta controlar la cantidad máxima de trabajo en cada estación. Parece que estos sistemas alcanzan mejores resultados que otros sistemas pull existentes (ver por ejemplo Gaury, 2000), siendo su principal desventaja su correcta parametrización, es decir, determinar los bucles de control que deben ser implementados así como el número de tarjetas para cada bucle de control. Este problema puede ser formulado como un problema combinatorio de optimización. Además, es usual

* Este trabajo se deriva de la participación de sus autores en un proyecto de investigación con financiación procedente del Ministerio de Educación y Ciencia (Proyecto DPI2004-02902) y con cofinanciación proveniente del Fondo europeo de desarrollo regional (FEDER).

considerar más de un único objetivo simultáneamente, tales como la tasa de salida, el inventario medio en proceso o el nivel de servicio, por lo que el problema puede ser tratado como un problema multicriterio. El método Cross-Entropy, CE, (Rubinstein, 1997) ha demostrado ser útil en una gran diversidad de problemas combinatorios del tipo NP-hard (Alon, *et al.*, 2005; De Boer, *et al.*, 2005; Rubinstein, 1999; Rubinstein and Kroese, 2004). Sin embargo, que sepamos, no hay referencias respecto a su uso en optimización multicriterio. En este trabajo abordamos la optimización de CTBS empleando el método CE para optimización multicriterio.

El resto del trabajo está organizado en las siguientes partes. En la sección 2 se describen los CTBS. La sección 3 se dedica a describir brevemente el método Cross-Entropy, CE, aplicado a optimización combinatoria. En la sección 4 se propone un método, basado en CE, para la optimización de los CTBS en problemas multicriterio. Para mostrar el funcionamiento del método descrito en la sección 4, en la sección 5 se muestra la aplicación a una línea de producción real. Las conclusiones y las futuras líneas de investigación se muestran en la sección 6.

2. Sistemas de control de la producción a medida

2.1. Sistemas basados en tarjetas

En los sistemas basados en tarjetas, como el Kanban o Conwip, el flujo de trabajos a través de la línea se realiza por medio de tarjetas kanban. En un sistema Kanban hay un parámetro k_i para cada estación i , la cual representa la cantidad máxima de trabajo permitida en la correspondiente estación. Por tanto, para una línea formada por n estaciones, el sistema Kanban Simple requiere el establecimiento de n parámetros (ver Figura 1).

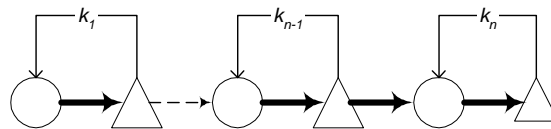


Figure 1. Sistema de control de la producción Kanban Simple

El sistema Conwip emplea una único tipo de tarjetas k para todas las estaciones de la línea. Por tanto, la cantidad máxima de trabajo es limitada por el número de tarjetas, siendo tan solo necesario establecer un único parámetro (ver Figura 2).

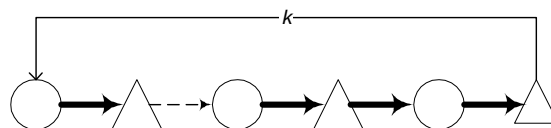


Figura 2. Sistema de control de la producción Conwip

Los sistemas Kanban se caracterizan porque emplean información local sobre el estado de cada estación, mientras que el sistema Conwip, emplea información global de la línea completa. Ambos sistemas son casos extremos de sistemas basados en tarjetas. En medio de estos existen una gran variedad de mecanismos basados en tarjetas, tales como Stock Base, Kanban Doble, Kanban Extendido, Kanban Generalizado, Kanban Genérico o el sistema híbrido Kanban-Conwip, entre otros. Hay numerosas referencias respecto a las comparaciones de sistemas basados en tarjetas (ver por ejemplo Framinan *et al.*, 2003). Los resultados

muestran que no hay ningún sistema que mejore al resto en todos los posibles escenarios. Por tanto podría ser de interés diseñar un sistema *ad-hoc* basado en tarjetas para una determinada línea en su respectivo entorno, en vez de tratar de implementar uno de los sistemas existentes sin garantía de que en dicho escenario pueda alcanzar unos resultados deseables. Esta es la filosofía que siguen los CTBS, la cual se explica en mayor detalle en la siguiente sección.

2.2. Sistemas a medida basados en tarjetas (CTBS)

Si consideramos todos los posibles bucles de control entre todas las estaciones del sistema obtendremos el sistema más general basado en tarjetas, tal como se muestra en la Figura 3. Este tipo de sistemas se conoce con el nombre de sistema a medida basado en tarjetas o *Customized Token Based System* (CTBS). Cada sistema existente basado en tarjetas es una combinación particular de los CTBS. Por ejemplo, si sólo consideramos el bucle de control entre la primera y última estación, estableciendo un elevado número de tarjetas para el resto de bucles, obtenemos el sistema Conwip.

Así, un CTBS considerará inicialmente todas las posibles combinaciones de bucles entre estaciones. Por tanto el proceso de optimización debe simultáneamente eliminar aquellos bucles que no son útiles para el flujo de trabajos y establecer el adecuado número de tarjetas para el resto de bucles. Hay que tener en cuenta que usualmente son necesarios menos de la mitad de los bucles iniciales para un determinado escenario (Gaury, 2000).

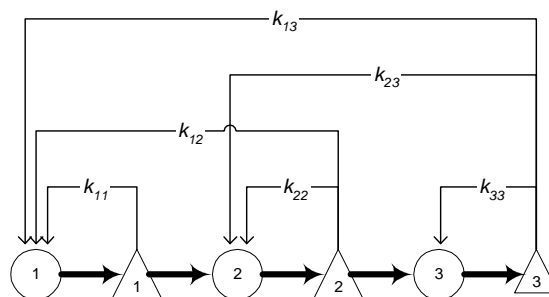


Figura 3. CTBS de 3 estaciones antes de la optimización.

2.3. Espacio de búsqueda

Las ventajas del empleo de sistemas a medida basados en tarjetas en vez de intentar detectar y optimizar los mejores sistemas pull “tradicionales” ha sido discutido y probado en diferentes trabajos (Gaury, 2000 y González 2006). Sin embargo el principal problema asociado a los CTBS radica en su optimización. Ésta puede ser formulada como encontrar el valor entero más adecuado para cada bucle de control de acuerdo a un conjunto de objetivos. Si inicialmente consideramos todos los posibles bucles de tarjetas entre cada par de estaciones, el número de total de bucles, L , en una línea formada por n estaciones es:

$$L = n(n+1)/2 \quad (1)$$

El espacio total de búsqueda viene dado por todas las combinaciones de valores posibles en los bucles. Para un número máximo fijo, K_{\max} , para todos los bucles, la expresión de dicho espacio de búsqueda es:

$$(K_{\max})^L \quad (2)$$

En la Figura 4 se puede observar el crecimiento del espacio de búsqueda con el número de estaciones para un número máximo fijo de tarjetas, $K_{\max} = 10$. La representación se ha hecho en escala logarítmica en base 10 con objeto de facilitar su interpretación.

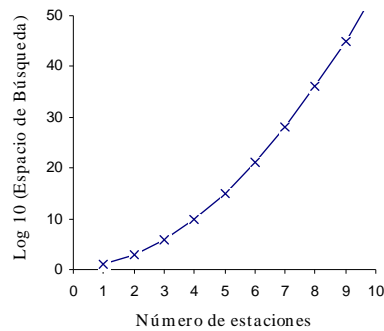


Figura 4. Espacio de búsqueda

Como se puede observar, el tamaño del espacio de búsqueda hace el problema intratable por medio de un proceso de enumeración completa, debido a que la manera usual de obtener las soluciones es por medio de simulación de eventos discretos. Los métodos alternativos, como el modelado matemático o teoría de colas, pueden resultar útiles sólo para casos muy simplistas y por tanto no es aplicable en la mayor parte de modelos basados en entornos reales. Por tanto, los métodos heurísticos pueden ser un método práctico para dicho problema de optimización.

2.4. Propiedades estructurales

Aunque las propiedades estructurales no son objeto de estudio en este trabajo, es importante introducir su concepto, con el objeto de clarificar el contenido de secciones posteriores. En 4.3 se ha expuesto el gran tamaño del espacio de búsqueda en los CTBS. Por tanto es importante investigar la posibilidad de reducción del espacio total de búsqueda. Gaury (2000) propone dos propiedades estructurales en los CTBS para reducir dicho espacio, generalizadas por González (2006). Dichas propiedades se denominan estructurales, debido a que no hay ninguna dependencia con las características particulares de la línea, tales como los tiempos de proceso, existencia de averías en las máquinas, tiempos de set-up o tipo de demanda, sino sólo dependen del número de estaciones en la línea. Por tanto es posible detectar si una solución particular es una solución estructural o no.

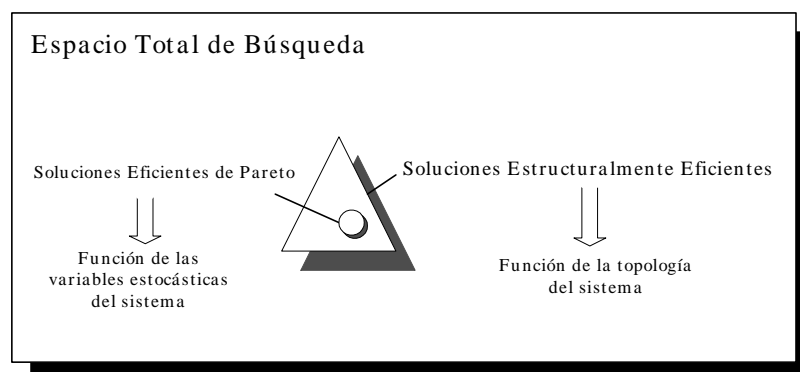


Figura 5. Partición del espacio de soluciones.

Esto es un aspecto importante debido a que no es interesante explorar las soluciones no estructuralmente eficientes, ya que las soluciones eficientes de Pareto están incluidas en el conjunto de soluciones estructuralmente eficientes (ver Figura 5).

Las propiedades estructurales tratan de detectar para una solución específica aquellos bucles de control que no afectan al funcionamiento del sistema, debido a que no limitan el flujo de trabajos a lo largo de la línea. Por tanto las propiedades estructurales pueden ser empleadas de dos maneras diferentes: la primera para simplificar una cierta solución y la segunda para generar soluciones estructuralmente eficientes. Estos aspectos serán tenidos en cuenta en el proceso de optimización descrito en la sección 4. Como se ha mencionado en la subsección 2.1, se ha observado que, en general, en un sistema optimizado sólo serán necesarios menos de la mitad de los bucles, con lo que los sistemas optimizados se tornan más prácticos y aplicables en entornos reales.

3. El método Cross-Entropy (CE)

3.1. Descripción

Muchos problemas de investigación operativa tratan de resolver problemas complicados de optimización. Generalmente este tipo de problemas son de complejidad conocida y de carácter estático. Sin embargo, hay otro tipo de problemas combinatorios donde el valor de la función objetivo ha de ser estimado, ya que no se conoce su valor exacto, debido a factores de ruido o al carácter estocástico de los sistemas. Como sabemos, la simulación de eventos discretos representa una forma habitual de estimación de dicho valor. El método denominado “Cross-Entropy”, CE, (ver por ejemplo Rubinstein y Kroese, 2004)), proporciona un método simple y eficiente para resolver este tipo de problemas. El método CE es un proceso iterativo donde cada iteración se compone de dos fases:

Fase 1. Generar un conjunto aleatorio de datos (trayectorias, vectores,...) de acuerdo a un mecanismo específico.

Fase 2. Actualizar los parámetros del mecanismo de generación aleatoria basado en los datos para producir un “mejor” conjunto en la siguiente iteración.

Es posible demostrar que este método alcanza el óptimo global para ciertos casos. Otro aspecto importante del método CE es que provee un método unificado para los problemas de simulación y optimización.

3.1. El método CE general

Consideremos el siguiente problema general de maximización. Sea una cierta función $S(x)$ sobre todos los valores de x pertenecientes a un conjunto \mathcal{X} . El objetivo es encontrar el valor máximo de $S(x)$ y el valor de x para el cual se consigue. Denotemos el valor máximo de la función como γ^* , por tanto:

$$S(x^*) = \gamma^* = \max_{x \in \mathcal{X}} S(x) \quad (3)$$

El punto de partida del algoritmo CE es enlazar el problema de optimización, definido en (3), con un problema de estimación. Consideremos un conjunto de funciones de densidad (pmf)

$\{f(\cdot; \nu), \nu \in V\}$ en \mathcal{X} , parametrizadas para los valores de ν obtenidos del sistema real (mediante simulación). Enlazamos el problema de optimización definido en (3) con el problema de estimación por medio de (4):

$$l(\gamma) = P_\nu(S(X) \geq \gamma) = E_\nu I_{\{S(X) \geq \gamma\}} \quad (4)$$

Donde ν es un parámetro conocido - $X \in \mathcal{X}$ tiene una pmf $f(\cdot; \nu)$ - y γ un escalar desconocido. Consideramos el criterio de el mayor es el mejor para calificar los eventos de interés. Para estimar este evento el método CE genera una secuencia iterativa de tuplas $(\hat{\gamma}_t, \hat{\nu}_t)$ las cuales convergen rápidamente a la vecindad del valor óptimo (γ^*, ν^*) .

Para estimar las soluciones elite, se establece una fracción ρ para cada muestra. Dicha fracción nos da un valor del umbral de γ . El número de muestras para cada iteración es dada por el valor N . Dicho valor se asume predefinido, aunque hoy en día existen ciertas técnicas para ajustar dicho valor dinámicamente (ver Rubinstein y Kroese, 2004). El parámetro ρ se establece usualmente el rango $0.01 \leq \rho \leq 0.03$ para sistemas donde el número de parámetros a ser estimados, n , es mayor que 100. En el caso de que $n < 100$ se suele establecer aproximadamente mediante el valor $\rho \approx \ln(n)/n$.

El método CE estándar, para un problema de maximización consiste en un proceso iterativo. En primer lugar es necesario establecer el parámetro $\hat{\nu}_0$ de manera uniforme. El proceso iterativo es el siguiente:

1. Actualización del parámetro γ_t . Para un valor fijo de ν_{t-1} , γ_t es el $(1-\rho)100\%$ -percentil de $S(X)$ bajo ν_{t-1} . Es decir γ_t satisface $P_{\nu_{t-1}}(S(X) \geq \gamma_t) \geq \rho$, donde $X \sim f(\cdot; \nu_{t-1})$. Un estimador simple de γ_t , $\hat{\gamma}_t$, se puede obtener por medio de una muestra aleatoria $\{X^{(1)}, \dots, X^{(N)}\}$ de la pmf $f(\cdot; \nu_{t-1})$, obteniendo las respuestas $S(X^{(i)})$ para todos los i y ordenándolos desde el menor al mayor $S_{(1)} \leq \dots \leq S_{(N)}$ y obteniendo el valor $\hat{\gamma}_t$ como el $(1-\rho)100\%$ percentil, de manera que $\hat{\gamma}_t = S_{\lceil (1-\rho)N \rceil}$.

2. Actualización del parámetro ν_t . Para un valor fijo de γ_t y ν_{t-1} , derivamos ν_t mediante la solución de (5):

$$\max_{\nu} D(\nu) = \max_{\nu} E_{\nu_{t-1}} I_{\{S(X) \geq \gamma_t\}} \log f(X; \nu) \quad (5)$$

Dados $\hat{\gamma}_t$ y ν_{t-1} y empleando un muestre es posible determinar el valor de $\hat{\nu}_t$ de la solución del programa (6):

$$\max_{\nu} \hat{D}(\nu) = \max_{\nu} \frac{1}{N} \sum_{i=1}^N I_{\{S(X^{(i)}) \geq \hat{\gamma}_t\}} \log f(X^{(i)}; \nu) \quad (6)$$

Una buena práctica es alisar el parámetro ν , con el objeto de evitar que las probabilidades de algunos de los componentes $\hat{\nu}_{t,j}$ de $\hat{\nu}_t$ tiendan a cero en iteraciones iniciales. Por tanto es usual es usual alisar el valor de ν , según la expresión (7).

$$\hat{v}_t = \alpha \tilde{v}_t + (1-\alpha) \hat{v}_{t-1} \quad (7)$$

Donde α es el factor de alisamiento, \tilde{v}_t es el vector parámetro solución de (6). Usualmente los valores de α están comprendidos entre 0.7 y 1.

El resumen del algoritmo CE es el siguiente:

1. Seleccionar algún \hat{v}_0 . Establecer el contador de iteraciones $t=1$.
2. Generar un conjunto de soluciones $\{X^{(1)}, \dots, X^{(N)}\}$ mediante la función de probabilidad $f(\cdot; v_{t-1})$ y calcular el $(1-\rho)100\%$ -percentil, $\hat{\gamma}_t$ de las respuestas ordenadas ascendientemente.
3. Emplear $\{X^{(1)}, \dots, X^{(N)}\}$ para resolver (6). Denotar la solución por \tilde{v}_t .
4. Alisar el vector \hat{v}_t : $\hat{v}_t = \alpha \tilde{v}_t + (1-\alpha) \hat{v}_{t-1}$
5. Si algún $t \geq d$ y $\hat{\gamma}_t = \hat{\gamma}_{t-1} = \dots = \hat{\gamma}_{t-d}$ FIN, en otro caso establecer $t = t+1$ e ir al paso 2

4. Algoritmo de optimización para sistemas a medida

4.1. Introducción

En la sección 3 hemos descrito el algoritmo general para problemas de optimización. Sin embargo este algoritmo debe ser adaptado a nuestro problema. En la subsección 4.2 se describe la manera de implementar el algoritmo para la optimización de los CTBS, mientras que en la sección 4.3 se describe el caso multicriterio.

4.2. CE aplicado a la optimización de los CTBS

En la sección 2 se han descrito los CTBS. Estos sistemas se caracterizan por un vector de valores enteros que simbolizan el número de tarjetas. Cada permutación en el vector de tarjetas producirá diferentes valores de la tasa de salida/nivel de servicio y del inventario en proceso en el sistema. Por tanto, el problema puede ser interpretado como un problema de optimización combinatoria.

El algoritmo principal de CE se ha modificado, resultando el siguiente:

0. Inicialización. Primero establecemos una matriz de probabilidades $P = |P_{ij}|$, donde i es el índice para los bucles de control y j es el índice para el valor de las tarjetas, tal que $1 \leq i \leq n \frac{(n+1)}{2}$, donde n es el número de estaciones en la línea y para cada j , tal que $1 \leq j \leq K_{\max} \cup \infty$. El valor infinito significa que el bucle de control no tiene efecto en el funcionamiento del sistema (ver referencia a propiedades estructurales en sección 2). Por tanto, P_{ij} es la probabilidad de que el bucle i tome el valor j . La matriz P se inicializa uniformemente al valor $1/(K_{\max} + 1)$, el cual significa que un bucle tiene la misma probabilidad de ser asignado cualquier número de tarjetas, o en caso de infinito de no ser considerado. Con la matriz de probabilidad es necesario generar un conjunto de soluciones $\{K^{(1)}, \dots, K^{(N)}\}$. Dicho proceso es realizado generando sólo soluciones estructuralmente eficientes.

1. Actualización de γ_t . Dada una cierta función objetivo, se calcula el $(1-\rho)100\%$ percentil y γ_t se estima como $\hat{\gamma}_t = S_{[(1-\rho)N]}$.

2. Actualización de pmf. Con el mismo conjunto de soluciones se calcula la matriz de probabilidades $Q = |Q_{ij}|$, empleando la expresión (8) (ver Rubinstein y Kroese, 2004 para más detalles):

$$Q_{ij} = \frac{\sum_{h=1}^N I_{\{S(K^{(h)}) \geq \gamma\}} I_{\{S(K^{(h)}) \in \tilde{K}_{ij}\}}}{I_{\{S(K^{(h)}) \geq \gamma\}}} \quad (8)$$

La matriz almacena el número de veces que una solución $S(K^{(i)}) \geq \gamma_t$ y que el bucle i tiene asignado el valor j , convirtiendo los valores a probabilidades sobre el conjunto de soluciones. Posteriormente la matriz se alisa mediante la expresión (9).

$$P_t = \alpha Q_t + (1-\alpha) P_{t-1} \quad (9)$$

El algoritmo se resume como sigue:

1. Establecer la matriz P_0 a su valor inicial $1/(K_{\max} + 1)$
2. Generar un conjunto de soluciones $\{K^{(1)}, \dots, K^{(N)}\}$ estructuralmente eficientes, empleando las propiedades estructurales y las probabilidades $P_{ij,t-1}$. Calcular el $(1-\rho)100\%$ -percentil $\hat{\gamma}_t$.
3. Calcular la matriz de probabilidades $Q_{ij,t}$ empleando las soluciones $\{K^{(1)}, \dots, K^{(N)}\}$ y la expresión (8).
4. Alisar la matriz $P_t = \alpha Q_t + (1-\alpha) P_{t-1}$
5. Si algún $t \geq d$ y $\hat{\gamma}_t = \hat{\gamma}_{t-1} = \dots = \hat{\gamma}_{t-d}$ FIN, en otro caso establecer $t = t + 1$ e ir al paso 2.

4.3. CE multicriterio

Gaury (2000) muestra tres diferentes funciones objetivos para los casos en que es deseable alcanzar una determinada tasa de salida/nivel de servicio con el menor inventario en proceso, mediante un enfoque lexicográfico. Sin embargo, algunas veces es deseable conocer el conjunto de soluciones eficientes de Pareto. Por ello, en esta subsección se describe un método eficiente para obtener la frontera de Pareto, basado en el método CE descrito en la última subsección.

El procedimiento consiste en los siguientes pasos:

1. Inicialización.

Establecer el contador de iteraciones $t = 0$

Establecer el máximo número de iteraciones T

Establecer la tasa de salida/nivel de servicio máximo τ

Seleccionar alguna solución inicial K_0 , y establecer su salida como tasa de salida/nivel de servicio objetivo, θ .

2. Comenzar el algoritmo CE descrito en 4.2 con el objetivo de alcanzar una tasa de salida/nivel de servicio, θ -de la solución K_t , con el mínimo inventario en proceso, actualizando el conjunto de soluciones eficientes. Las funciones objetivo de Gaury (2000) pueden ser empleadas en este paso. Si la solución K_t pertenece al conjunto de soluciones de eficientes Pareto, marcar dicha solución como visitada.

3. Si $t \geq T$ y $\gamma_t \geq \tau$ o todo el set de Pareto está marcado como visitado, FIN. En otro caso seleccionar la primera solución del set de Pareto no marcada y establecer la tasa de salida/nivel de servicio como θ . Establecer $t = t + 1$ e ir al paso 2.

Este algoritmo hace un uso iterativo del método CE, por medio de una búsqueda intensiva de las mejores soluciones en cada nuevo punto en el conjunto de soluciones eficientes de Pareto.

5. Experiencia computacional

Con el objeto de probar la eficiencia de la algoritmo descrito en 4.3, realizamos un experimento basado en un modelo son datos de una factoría real de una planta de componentes de automoción de España. Aunque los datos detallados aportados por el departamento de control de la producción de la fábrica no son aportados en el trabajo, por razones de confidencialidad, vamos a resumir las principales características de la línea: el experimento se centra en una línea en serie de 10 estaciones. Los tiempos de proceso se pueden considerar deterministas, pero los procesos están afectados por diferentes causas de variabilidad. El modelo considera 13 tipos de averías importantes, donde los tiempos entre fallos están distribuidos exponencialmente, mientras que los tiempos de reparación siguen una distribución del tipo Erlang. Además consideramos 23 paradas programadas para labores de mantenimiento. Su frecuencia duración se pueden considerar deterministas. Finalmente consideramos 15 microparadas con frecuencia exponencial y duración determinista. Con el objeto de obtener soluciones fiables, se han simulado 40 días de producción y se han realizado 10 replicados. En la Figura 6 se muestran las soluciones eficientes de Pareto para el CTBS empleando la metodología CE muticriterio propuesta. Además se muestran los resultados para el mismo sistema operando con un sistema Conwip.

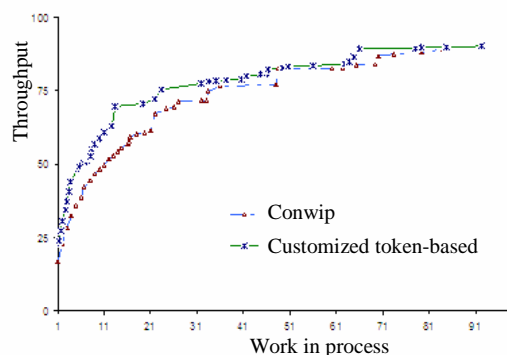


Figura 6. Frontera de Pareto para el CTBS y el sistema Conwip

Actualmente la línea de producción en la factoría es controlada por un sistema Kanban, con un 90% de tasa de salida y un inventario en proceso de 350 trabajos. Por el contrario, el sistema a medida alcanza una tasa de salida similar 90.17% con un inventario de solo 92.33 trabajos. Dichos resultados se alcanzan con el sistema de la Figura 7.

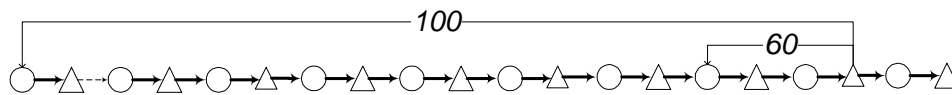


Figura 7. Sistema a medida para un 90.17% de tasa de salida

Es importante resaltar que para esta solución el CTBS es más fácil de controlar e implementar ya que solo depende de 2 parámetros. Sin embargo, en la implementación actual del tipo Kanban necesitan ajustar 10 parámetros.

6. Conclusiones

En este trabajo hemos introducido los CTBS, como una alternativa y eficiente a los mecanismos pull existentes. Hemos descrito brevemente el método CE, así como su adaptación a la optimización de los CTBS para problemas multicriterio. Para probar el funcionamiento del algoritmo de optimización, realizamos una experimentación para un modelo basado en datos reales de una fábrica de componentes de automoción. Los resultados muestran que el CTBS obtenido alcanza mejores resultados que el sistema Conwip para cada combinación de tasa de salida e inventario en proceso. Adicionalmente el sistema mejora con el que actualmente se opera en la fábrica.

Referencias

- Alon, G.; Kroese, D.P.; Raviv, T.; Rubinstein, R.Y. (2005). Application of the Cross Entropy Method to the Buffer Allocation Problem in a Simulation-Based Environment. *Annals of Operations Research*, Vol. 134, No. 1, pp. 137-151.
- De Boer, P-T.; Kroese, D.P.; Mannor, S.; Rubinstein, R.Y. (2005). A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, Vol. 134, No. 1, pp. 19-67.
- Framinan, J.M.; González, P.L.; Ruiz-Usano, R. (2003). The Conwip production control system: review and research issues. *Production Planning and Control*, Vol. 14, No. 3, pp. 255-265.
- Gaury, E.G.A. (2000). *Designing pull production control systems: customization and robustness*. Tesis Doctoral, Ed. Center. Universidad de Tilburg, Holanda.
- Gaury, E.G.A.; Pierreval, H.; Kleijnen, J.P.C. (2000). An evolutionary approach to select a pull system among Kanban, Conwip and Hybrid. *Journal of Intelligent Manufacturing*, Vol. 11, pp. 157-167.
- Gaury, E.G.A.; Pierreval, H.; Kleijnen, J.P.C. (2001). A methodology to customize pull control systems. *Journal of Operational Research Society*, Vol, 52, No. 7, pp. 789-799.
- González Rodríguez, P.L. (2006). *Diseño y Operación de sistemas a medida basados en tarjetas*. Tesis Doctoral. Universidad de Sevilla.
- Monden, Y. (1983). *Toyota production system*. Industrial Engineering and Management Press, Atlanta.
- Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events. *European Journal of Operations Research*, Vol. 99, pp. 89-112.
- Rubinstein, R. Y. (1999). The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, Vol. 2, pp. 127-190.
- Rubinstein, R.Y.; Kroese, D. P. (2004). *The Cross-Entropy Method: A Unified Approach to Monte Carlo Simulation, Randomized Optimization and Machine Learning*. Information Science & Statistics, Springer Verlag, NY.