UNIVERSIDAD DE CUENCA

Facultad de Ciencias Químicas

Carrera de Ingeniería Industrial

Optimización de la cadena de suministro mediante el uso de un algoritmo genético basado en clusterización

Trabajo de titulación previo a la obtención del título de Ingeniero Industrial

Modalidad: Proyecto de Investigación

Autor:

Nelson Bladimiro Berrezueta Guamán

C.I. 0301834982

nelson.berrezueta@gmail.com

Directora:

Ing. Diana Carolina Jadán Avilés, M.Sc.

C.I. 0104236971

Cuenca, Ecuador

27-febrero-2020

"Optimización de la cadena de suministro mediante el uso de un algoritmo genético basado en clusterización"

Berrezueta Nelson[1], Jadán Diana[2]

[1] Universidad de Cuenca, nelson.berrezueta@ucuenca.edu.ec

[2] Facultad de Ciencias Químicas, Universidad de Cuenca, diana.jadan@ucuenca.edu.ec

Fecha de entrega: 27 de febrero de 2020.

RESUMEN

Esta investigación propone k-NSGA-II, un algoritmo basado en computación evolutiva que mezcla propiedades de micro algoritmos y fundamentos de inteligencia artificial como la clusterización. El objetivo de k-NSGA-II es optimizar procesos en entornos reales y apoyar eficientemente a la toma de decisiones en organizaciones.

Los cambios desarrollados se realizaron en NSGA-II, un algoritmo genético multiobjetivo basado en la no dominancia de sus resultados. La funcionalidad de k-NSGA-II se verificó mediante pruebas de rendimiento comparándolo con NSGA-II y µ-NSGA-II. Estas pruebas se realizaron en distintas funciones objetivo y en un caso de estudio, el mismo que se fundamentó en la optimización de la producción y distribución de productos. Los objetivos fueron la minimización de residuos y maximización del beneficio obtenido mediante las ventas.

k-NSGA-II se utilizó para optimizar las funciones generando resultados interesantes para la empresa. También fue más útil con respecto a NSGA-II ya que generó un número reducido de soluciones precisas que el analista pudo revisar rápidamente antes de tomar una decisión, en comparación con NSGA-II que trabaja con conjuntos de 200 soluciones o µ-NSGA-II que no presentó soluciones que puedan ser útiles para la empresa.

El algoritmo k-NSGA-II presenta una innovación con respecto a NSGA-II al ser un micro algoritmo preciso cuyas soluciones son de gran utilidad para la toma de decisiones en entornos de problemas reales. Mejora el tiempo de evaluación y evita la fatiga del analista al no presentar una gran cantidad de resultados que muchas veces no son revisados.

Palabras clave: Cadena de suministro. k-NSGA-II. Inteligencia artificial. Algoritmo genético. Micro algoritmo.

ABSTRACT

This research proposes k-NSGA-II, an algorithm based on evolutionary computing that mixes properties of micro-algorithms and artificial intelligence fundamentals such as clustering. The objective of k-NSGA-II is to optimize processes in real environments and to efficiently support decision making in organizations.

The changes developed were made in NSGA-II, a multiobjective genetic algorithm based on the non-dominance of its results. The functionality of k-NSGA-II was verified by performance tests comparing it with NSGA-II and µ-NSGA-II. These tests were performed on different objective functions and on a case study, which was based on the optimization of product production and distribution. The objectives were to minimize waste and maximize profit through sales.

k-NSGA-II was used to optimize the functions generating interesting results for the company. It was also more useful with respect to NSGA-II as it generated a small number of accurate solutions that the analyst could review quickly before making a decision, compared to NSGA-II which works with sets of 200 solutions or µ-NSGA-II which did not present solutions that could be useful to the company.

The k-NSGA-II algorithm presents an innovation with respect to NSGA-II as it is a precise micro algorithm whose solutions are very useful for decision making in real problem environments. It improves the evaluation time and avoids the analyst's fatigue because it does not present a great amount of results that many times are not analysed.

Keywords: Supply chain. k-NSGA-II. Artificial intelligence. Genetic algorithm. Micro algorithm.

Índice del Trabajo

Índice

1.

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Nelson Bladimiro Berrezueta Guamán en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Optimización de la cadena de suministro mediante el uso de un algoritmo genético basado en clusterización", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 27 de febrero de 2020

Nelson Bladimiro Berrezueta Guamán

C.I: 0301834982

Cláusula de Propiedad Intelectual

Nelson Bladimiro Berrezueta Guamán, autor del trabajo de titulación "Optimización de la cadena de suministro mediante el uso de un algoritmo genético basado en clusterización", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 27 de febrero de 2020

Nelson Bladimiro Berrezueta Guamán

C.I: 0301834982

2. INTRODUCCION

Los cambios sociales, económicos y culturales que están afectando a la sociedad en los últimos años han provocado que las necesidades de los consumidores cambien. Estos cambios han puesto en jaque a las empresas, que para seguir siendo competitivos, han buscado mejorar sus cadenas de suministro (SC, por sus siglas en inglés) con el objetivo de reducir costes, incrementar la satisfacción del cliente, optimizar el uso de recursos y construir nuevos ingresos (EOI, 2010). Las SC se definen como un macro proceso que transforma materia prima en productos que son enviados a los consumidores (Beamon, 1998). Algunos de los procesos que conforman una SC son la logística, la gestión de inventarios, compras y adquisiciones, planeación de la producción y relaciones tanto dentro como fuera de empresa (Arshinder et al., 2008).

Gracias a la gestión de la cadena de suministro (SCM, por sus siglas en inglés) se puede hablar de un método que ayuda a mejorar y conocer si las decisiones tomadas son las correctas o no (Fabbe- Costes y Jahre, 2008). Los modelos usados para la SCM son la referencia de operaciones de la cadena de suministro (SCOR, por sus siglas en inglés) y el Cuadro de Mando Integral (BSC, por sus siglas en inglés). El modelo SCOR fue creado por el Supply Chain Council para gestionar las SC tanto a nivel estratégico, táctico y operacional. Este modelo fue creado específicamente para este propósito por lo que sus procesos: planificación, aprovisionamiento, fabricación, distribución y devolución, son los adecuados para esta tarea (Giannakis, 2011). Por otro lado, el BSC es un modelo general que colabora en la gestión mediante la implantación de métricas financieras y no financieras para conseguir los objetivos estratégicos marcados (Cho et al., 2012). Otros modelos son los matemáticos, que se utilizan para realizar optimizaciones o simulaciones de la realidad de las empresas. Este tipo de modelos tienen una estructura similar a la mostrada en la Ecuación 1, que minimiza o maximiza una serie de funciones sujetas a restricciones de igualdad o desigualdad (Srinivas y Deb, 1994).

$$\begin{aligned} &\text{Maximizar / Minimizar} \quad f_i(x) \quad i = 1,2,\dots,N \qquad (1)\\ &\text{Sujeto a:} \qquad g_j(x) \leq 0 \; j = 1,2,\dots,J \\ &\qquad\qquad\quad h_k(x) = 0 \; k = 1,2,\dots,K \end{aligned}$$

La mayoría de los modelos de optimización usados en empresas se centran en la fase de producción y planeación del transporte con el objetivo de minimizar costos y maximizar los beneficios (Mula et al., 2010). Los modelos más utilizados para estas tareas son la programación lineal, programación lineal entera mixta y multiobjetivo. Debido a la complejidad del entorno de las SC de los últimos tiempos se ha extendido el uso de métodos no exactos como la programación matemática difusa, estocástica o modelos heurísticos y metaheurísticos (Mula

et al., 2010). Los métodos metaheurísticos son procedimientos iterativos que combinan distintos conceptos para explorar y explotar inteligentemente un espacio de búsqueda (Francisco Herrera, 2009). Dentro de estos algoritmos se ubican los algoritmos genéticos, búsqueda tabú, colonia de hormigas o enjambre de partículas, entre otros.

Los algoritmos genéticos (GA, por sus siglas en inglés) se desarrollaron de la mano de John Holland en la década de los 60 imitando el proceso de selección natural descrito por Darwin. El algoritmo utiliza una población de individuos que contiene información genética que es evaluada por una función objetivo. A dicha población se le aplican operadores genéticos como el cruce y la mutación para generar descendencia y así explorar el campo de búsqueda. Los individuos mejor adaptados o lo que es lo mismo, que mejor satisfacen el modelo matemático serán parte del resultado óptimo (Pavez, Rafael A., 2014).

Una desventaja del algoritmo propuesto por Holland es su ineficacia y alto consumo de recursos para generar una solución, por este motivo se desarrollaron otros algoritmos en base a su idea. Srinivas y Deb (1994) desarrollan el algoritmo genético de clasificación no dominado (NSGA, por sus siglas en inglés) para hacer al GA más eficiente y potente con el fin de realizar optimización multiobjetivo. Aunque su desarrollo es bueno, Deb, Agrawal, Pratap, y Meyarivan (2000) proponen el NSGA-II que es aún más rápido y eficiente que su antecesor, convirtiéndose en uno de los algoritmos de búsqueda más utilizados (Memari et al., 2017). Por la popularidad que adquiere el NSGA-II diversos investigadores se proponen mejorarlo mediante la modificación de algunos de sus parámetros ya que entienden que se puede conseguir mejores resultados mediante la aplicación de nuevas técnicas desarrolladas en los últimos años (Bandyopadhyay y Bhattacharya, 2013; Memari et al., 2017). Una de las modificaciones en este tipo de algoritmos ha sido la reducción de las poblaciones mediante la hipótesis de que el algoritmo será igual de preciso en su resultado pero con menor uso de recursos en las evaluaciones (Kalmannjjee Krishnakumar, 1990). Aunque también se propusieron otros cambios más profundos como el cambio de los operadores genéticos e incluso hibridaciones con otros algoritmos (Coello y Toscano, 2001; Konchada et al., 2016; Llamas et al., 2015; Toscano y Coello, 2003; Zhuang-Cheng Liu et al., 2011).

Otro fundamento de la inteligencia artificial son los métodos de agrupación como el k-means, un algoritmo de aprendizaje automático no supervisado que permite clasificar puntos en grupos. Donde *k* representa un número de agrupaciones predefinidos por el analista, y *means* hace referencia a que el criterio utilizado para evaluar la solución es que la suma de las distancias euclidianas a un punto central sea el menor posible. Este algoritmo fue desarrollado por MacQueen en el año 1967 (Kassambara, 2018; Macqueen, 1967).

Con los antecedentes mencionados, en este estudio se desarrolló k-NSGA-II, un micro algoritmo basado en NSGA-II y clusterización mediante k-means para optimizar varios objetivos de una SC. Para ello se estudiaron los micro algoritmos y las modificaciones que se pueden realizar al NSGA-II para convertirlo en micro algoritmo y comparar su rendimiento frente al NSGA-II y el micro-NSGA-II (µ-NSGA-II) propuesto por Konchada, Pragada, Chintada y Bhimuni (2016). La hipótesis se fundamentó en que el uso de micro algoritmos es suficiente en entornos reales de optimización como son las SC de empresas ya que los analistas no necesitan 100 soluciones o más como entregan los actuales algoritmos evolutivos, sino que es suficiente entregar un conjunto de menos de 20 resultados para que sean evaluados e implementados para la resolución del problema.

3. MATERIALES Y MÉTODOS

El algoritmo propuesto se evaluó con diferentes funciones para comprobar su rendimiento. Estas funciones pertenecen a un set predefinido utilizado para este tipo de estudios, además del modelo matemático de una empresa de la ciudad de Cuenca (Azuay) que se dedica a la producción y venta de muebles con estructura metálica. Por razones de confidencialidad con la empresa, no se menciona mayor detalle de la misma. El estudio de optimización mediante k-NSGA-II se centró en los procesos de producción y distribución de su cadena de suministro. La parte de producción se centra en la optimización del uso de recursos y de la generación de desperdicios, mientras que en la parte de distribución se maximiza el beneficio obtenido por la colocación y venta de productos en los distintos puntos de venta. A continuación, se describen las funciones que modelan el caso de estudio.

La optimización de la producción se centra en el proceso de corte de materia prima, la misma que se compone principalmente de tuberías, perfiles y tubos metálicos que deben ser cortados para después ser doblados y soldados, creando así una estructura para cada uno de los productos manufacturados. Este proceso de corte se dividió en dos subprocesos. El primero que, mediante una función propia, genera patrones de corte para una materia prima en específico según las necesidades de producción; y el segundo subproceso que, mediante otra función, determina el número de veces que un patrón se debe repetir para cumplir con la demanda al menor costo y desperdicio posible.

La primera función objetivo genera patrones de corte mediante permutaciones, es decir, se crea un abanico de posibilidades de corte en función de N_k. La Ecuación 2, que calcula dicho valor, indica el número de cortes de longitud l_i que pueden realizarse en una materia prima de longitud L_i en función de la demanda d_i (Gracia, 2010). A través de N_k se genera una matriz con las permutaciones de las longitudes demandadas tomadas de N_k en N_k veces.

$$N_k = \left[\frac{L_k}{\left(\frac{\sum_i d_i * l_i}{\sum_i d_i} \right)} + 0.5 \right] \qquad (2)$$

Una vez obtenida la matriz de patrones completa, la matriz de cortes se determina mediante la restricción que define la Ecuación 3. Esta restricción verifica que la suma de los cortes de longitud l_i realizados por cada patrón a_i no superen la longitud de la materia prima L_i. En caso de superar dicha longitud, la función anula los últimos cortes hasta cumplir con la restricción. Anular cortes, hace referencia a que se colocan ceros en cada fila de la matriz, comenzando desde la derecha hasta que se cumpla la restricción. La Ecuación 3, en resumen, permite depurar la matriz original mostrando solamente aquellas que cumplan con la restricción definida (Gracia, 2010; Ochoa, 2014).

$$\sum_{i=1} l_i a_i \leq L_i \qquad (3)$$

La matriz de patrones generada se muestra en la Tabla 1 que contiene 9 patrones de corte para 4 demandas de cortes diferentes para una longitud de 2398 y 5850 milímetros. La función asignó un 1 si el corte pertenece al patrón y un 0 si no. Esta matriz pudo ser optimizada mediante el uso del algoritmo genético de manera que los valores 1 puedan tomar otros valores, validando de igual manera la restricción de la Ecuación 3. La matriz optimizada se muestra en la Tabla 2, donde se observa que los patrones número 3, 4 y 6 pudieron aceptar el doble y el triple de cortes de las demandas 1930.5 y 1949.805 milímetros.

Tabla 1. ***Matriz de patrones de corte (Elaboración propia)***

Patrón	Longitud Materia Prima	Longitud de las demandas			
		1930.5	1949.805	3451.5	4909.32
1	2398	1	0	0	0
2	2398	0	1	0	0
3	5850	1	0	0	0
4	5850	1	1	0	0
5	5850	1	0	1	0
6	5850	0	1	0	0
7	5850	0	1	1	0
8	5850	0	0	1	0
9	5850	0	0	0	1

Tabla 2. ***Matriz de patrones de corte optimizados (Elaboración propia)***

Patrón	Longitud Materia Prima	Longitud de las demandas			
		1930.5	1949.805	3451.5	4909.32
1	2398	1	0	0	0
2	2398	0	1	0	0
3	5850	3	0	0	0
4	5850	2	1	0	0
5	5850	1	0	1	0
6	5850	0	3	0	0
7	5850	0	1	1	0
8	5850	0	0	1	0
9	5850	0	0	0	1

Una vez generada la matriz de patrones de corte optimizados (Tabla 2) se define la siguiente parte del problema de corte mediante la Ecuación 4. El objetivo de esta ecuación es minimizar el residuo generado por los patrones de corte *c*, cortados por una frecuencia *x*. Es decir, el número de veces *x* que se corta a cada uno de los patrones *c*, cumpliendo las restricciones de las Ecuaciones 5 y 6. La Ecuación 5 verifica que la suma de los cortes producidos por los patrones de corte a_{ij} por la frecuencia de cortes aplicados a cada patrón x_j satisfacen la cantidad de cortes demandados d_i. La Ecuación 6, en cambio, confirma que la frecuencia de cortes aplicados a cada patrón x_j no supere la materia prima e_j disponible en inventario. Este conjunto de 6 ecuaciones modeló la producción de la empresa.

$$z_{entero} = \min\{cx : x \in \mathrm{Z}_{+}^{\eta}\} \qquad (4)$$

$$\sum_{j=1}^{\eta} a_{ij} x_j \geq d_i \qquad (5)$$

$$\sum_{j=1}^{\eta} x_j \leq e_j \qquad (6)$$

Gráficamente el problema se resume en la Figura 1, donde se encuentran las materias primas de longitud *L*, los cortes de demanda *l* y la matriz de patrones *a*. Las frecuencias de corte de cada patrón *x* por el número de veces que se corta *l* debe satisfacer la demanda *d*. Se puede observar cómo cada patrón de corte *a*, satisface la restricción de la longitud de la materia prima *L*. Así mismo las zonas blancas son el desperdicio generado por cada patrón de corte.

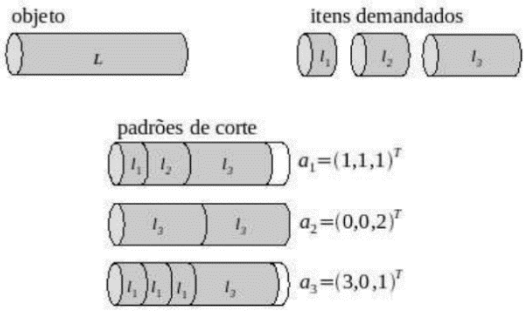

Figura 1. Detalle gráfico del problema de corte (Pinto, 2008). L es la longitud de la materia prima. l es la longitud de los cortes demandados. Y a es un patrón de corte. Las zonas en blanco muestran la cantidad de desperdicio generado por cada patrón.

Por otro lado, el problema de distribución maximiza el beneficio obtenido por las ventas. La función objetivo mostrada en la Ecuación 7 se define a través de las Ventas Brutas (*VB*) menos los costos de Producción (*CTP*), de Inventario (*CTI*) y Distribución (*CTD*).

$$\max (VB - CTP - CTI - CTD) \qquad (7)$$

En donde, las ventas brutas (Ecuación 8) son el resultado del beneficio bruto ganado por las ventas en los centros *c* por cada producto *k*. El costo total de producción (Ecuación 9) es igual a la cantidad de producción de *k* (PR_k) por el costo de producción (CP_k). El costo total de inventario (Ecuación 10) se calcula como el interés generado del inventario mantenido (Inv_k) por el costo del producto (CP_k) durante un periodo definido (T_k) a una tasa de interés definida (I_a). El costo total de distribución (Ecuación 11) es igual al costo de transporte del producto *k* (CT_k) por la cantidad de producto distribuido al centro *c* ($PD_{c,k}$). El inventario (Ecuación 12) sigue las políticas de restar el producto distribuido PD_k y sumar el inventario inicial $Inv_{0,k}$ y la cantidad de producción PR_k. Estas ecuaciones están sujetas a las restricciones de la Ecuación 13 para el cumplimento de la demanda y la Ecuación 14 para cumplimiento del inventario disponible.

$$VB = \sum_{c,k} PVP_k * PV_{ck} \qquad (8)$$

$$CTP = \sum_k PR_k * CP_k \qquad (9)$$

$$CTI = \sum_k CP_k * Inv_k * T_k * I_a \qquad (10)$$

$$CTD = \sum_{c,k} CT_k * PD_{c,k} \qquad (11)$$

$$Inv_k = PR_k + Inv_{0,k} - PD_k \qquad (12)$$

$$\sum_{c,k} PD_{c,k} \geq \sum_{c,k} D_{c,k} \qquad (13)$$

$$PR_k + Inv_{\ k} \geq PD_k \qquad (14)$$

Para el desarrollo de las funciones objetivo y la evaluación del algoritmo, se utilizó el software R versión 3.6.1 junto con el entorno de desarrollo RStudio Version 1.2.1335 en un equipo con un procesador Intel Core i7-7500U a 2.70GHz con 16 GB de memoria RAM bajo Windows 10. Se utilizó también la implementación del algoritmo NSGA-II en R mediante los paquetes "*nsga2R*" elaborado por Ching-Shih (2015), "*mco*" elaborado por Olaf et al. (2014) y el algoritmo "*k-means*" implementado dentro del paquete Stats de la base de R. Con el fin de eliminar incertidumbres, cada función se evaluó 15 veces por algoritmo, siendo el promedio de las evaluaciones el resultado a tener en cuenta, al igual que Sarrafha et al. (2014) realizaron en su estudio.

Las métricas de rendimiento que se utilizaron para evaluar la factibilidad del algoritmo respecto a NSGA-II y micro-NSGA-II son las siguientes:

- *Tiempo de procesamiento (CPU):* mide el tiempo, en segundos, que cada algoritmo toma para evaluar las funciones y entregar los resultados. El algoritmo es mejor mientras menor tiempo tome para evaluar cada función (Sarrafha et al., 2014).
- *Hipervolumen (HV)*: su valor muestra el espacio o hiperespacio que las soluciones generadas abarcan respecto a un punto definido. Mientras mayor es este valor, mejor es la calidad del conjunto de soluciones ofrecidas (Fonseca et al., 2006; Riquelme et al., 2015).
- *Dispersión Generalizada (Δ):* mide la distribución y el rango abarcado por las soluciones en el espacio, es decir, la diversidad de las soluciones. Este indicador muestra que el algoritmo es mejor mientras más alto sea este valor (Riquelme et al., 2015).
- *Distancia Generacional (DG)*: determina la distancia entre el conjunto de soluciones ofrecido y la frontera de Pareto real de la función evaluada. Un valor bajo indica que la solución es próxima a la solución real (Riquelme et al., 2015).

Para conseguir comportamientos similares, se utilizaron los siguientes parámetros para cada algoritmo, según lo mostrado en la Tabla 3. Entre paréntesis se muestran los valores utilizados para las funciones del caso de estudio. Fuera del paréntesis es el valor del parámetro que se utilizó en las funciones de prueba.

Tabla 3. ***Parámetros de los algoritmos genéticos utilizados (Elaboración propia)***

Parámetro	**Algoritmo**		
	NSGA-II	**µ-NSGA-II**	**k-NSGA-II**
Número de generaciones	200	200	200
Número de individuos (población)	200	12	200

Selección por torneo	2	2	2
Coeficiente de cruce	0.7 (0.9)	0.7 (0.9)	0.7 (0.9)
Índice de distribución de cruce	5 (500)	5 (500)	5 (500)
Coeficiente de mutación	0.2 (0.01)	0.2 (0.01)	0.2 (0.01) / 0
Índice de distribución de mutación	10 (50)	10 (50)	10 (50) / 0

Entre paréntesis se muestran los valores utilizados para las funciones del caso de estudio. Fuera del paréntesis se muestra el valor del parámetro que se utilizó en las funciones de prueba. El coeficiente e índice de mutación para µ-NSGA-II y k-NSGA-II es cero.

Las funciones de prueba corresponden al conjunto del *ZDT test suit* desarrollado por Zitzler que, según Huband et al. (2006), es uno de los test de rendimiento más utilizados para evaluar problemas multiobjetivo en algoritmos evolutivos. De este conjunto se utilizaron en específico las siguientes funciones: ZDT1, ZDT2 y ZDT3. Las características de cada una son las mostradas en la Tabla 4.

Tabla 4. ***Características de las funciones ZDT1, ZDT2 y ZDT 3 (Elaboración propia)***

Característica	ZDT1, ZDT2 y ZDT3	Caso de Estudio
Número de variables	30	19
Número de objetivos	2	3
Límite superior	1	500
Límite inferior	0	0

Las características para las funciones de prueba fueron las mismas. En cambio, en el caso de estudio el número de variables sufrió cambios según el periodo analizado.

El algoritmo propuesto se basa en la hibridación de otros dos algoritmos ampliamente conocidos. Por un lado, el NSGA-II y por otro el método de clusterización *k-means*. El proceso con el que el algoritmo se ejecuta se define de la siguiente manera:

Paso 1: Se inicia ejecutando el algoritmo NSGA-II durante un número determinado de generaciones con una población de menos de 300 individuos.

Paso 2: Con las soluciones generadas en el Paso 1, se ejecuta el algoritmo de clusterización k-means, con k = 10.

Paso 3: Se ejecuta nuevamente NSGA-II a cada conjunto de soluciones agrupadas por k-means por un número determinado de generaciones. La particularidad en este caso es que los coeficientes de mutación son nulos para evitar saltos de la descendencia fuera de los límites de cada población, como indica Kalmannjjee Krishnakumar (1990).

Paso 4: De cada una de las poblaciones entregadas en el Paso 3 se selecciona al individuo mejor evaluado, además de los individuos límites que se generaron en el Paso 1. El conjunto de individuos seleccionados es el mostrado como resultado final.

Es necesario indicar que el número de generaciones definido para el algoritmo se divide en las dos fases del mismo (Paso 1 y Paso 3). Un porcentaje mayor para la primera evaluación, y el restante para la evaluación de cada clúster.

4. RESULTADOS

Los indicadores calculados en cada uno de los algoritmos se muestran en la Tabla 5. Cada fila de la tabla corresponde a un indicador y cada columna a un algoritmo. El algoritmo µ-NSGA-II demostró ser el más rápido en evaluar cada una de las funciones con respecto a los otros dos algoritmos, sin embargo, fue el peor en acercarse a la frontera de Pareto real *(DG)*, en donde NSGA-II demostró generar las soluciones más próximas. El algoritmo propuesto, k-NSGA-II, cumplió con las expectativas mostrando un tiempo menor a NSGA-II, mejor proximidad a la frontera real de Pareto con respecto a µ-NSGA-II y mejor dispersión de individuos, es decir, con menor o igual cantidad de soluciones, abarcó el mismo o más espacio de búsqueda con respecto a los otros dos algoritmos.

Tabla 5. ***Indicadores de la evaluación de algoritmos (Elaboración propia)***

Función	Indicador	NSGA-II	µ-NSGA-II	k-NSGA-II
ZDT1	CPU	49.24	1.48	47.79
	HV	0.6639	0.6127	0.6021
	Δ	0.0780	0.5696	0.7040
	DG	14.2954	856.0283	115.0343
ZDT2	CPU	64.87	2.06	66.09
	HV	0.2865	0.1521	0.2192
	Δ	0.1946	0.8247	0.0467
	DG	26.0722	141.5237	184.9772
ZDT3	CPU	53.67	1.29	42.29
	HV	1.0431	0.99	0.9552
	Δ	0.0642	0.4589	0.6629
	DG	23.1247	2122.6595	303.0768
CASO DE ESTUDIO	CPU	161.16	10.76	125.73
	HV	1,13E+11	1,26E+10	1,3612E+12
	Δ	0.8089	0.933	0.905
	DG	0.4541	183.5044	6.08

ZDT1, ZDT2, ZDT3 y CASO DE ESTUDIO son las funciones que fueron evaluadas por cada algoritmo. CPU es el indicador de tiempo de procesamiento. HV es el hipervolumen generado por las soluciones. Δ es la dispersión generalizada de las soluciones y DG es la distancia generacional de la solución con respecto a la frontera de Pareto real.

La evaluación del caso de estudio con cada uno de los algoritmos mostró resultados satisfactorios para k-NSGA-II, como se observa en la Tabla 6. El algoritmo k-NSGA-II generó

menor desperdicio que sus contrapartes y produjo mayor cantidad de unidades demandadas. Por otro lado, µ-NSGA-II entregó resultados totalmente diferentes que NSGA-II y k-NSGA-II.

Tabla 6. ***Evaluación del caso de estudio por NSGA-II, µ-NSGA-II y k-NSGA-II (Elaboración propia)***

	Scrap (metros)	Materia Prima (unidades)	Insatisfacción (unidades)	Sobreproducción (unidades)
NSGA-II	166	350	0	91
µ-NSGA-II	2280	1566	0	2197
k-NSGA-II	144	402	0	205

Scrap en metros es la cantidad de desperdicio generado por el problema de corte. Materia Prima indica el número de unidades de materia prima que deben utilizarse para la producción. Insatisfacción muestra las unidades de demanda que no fueron cubiertas. Sobreproducción son las unidades de demanda que se generaron adicionales a lo solicitado.

La Figura 2 muestra la distribución de las soluciones de la evaluación del caso de estudio para cada uno de los algoritmos. A primera vista se distingue la diferencia en la cantidad de puntos para NSGA-II (Figura 2a) y los micro algoritmos, µ-NSGA-II (Figura 2b) y k-NSGA-II (Figura 2c). Se observó la separación entre los puntos y la escala de los ejes para cada uno de ellos, siendo µ-NSGA-II el que más alejado de los valores aceptables presentó.

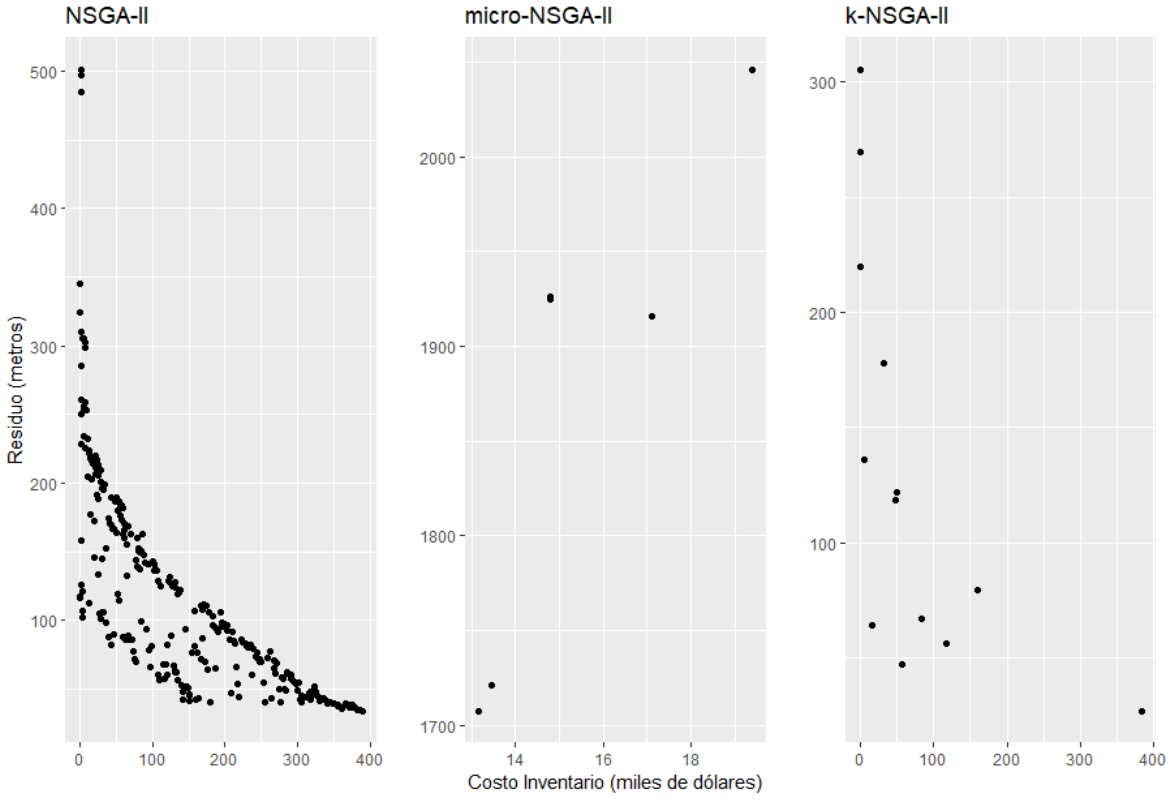

Figura 2. Comparación de las Fronteras de Pareto. Figura 2a (izquierda): Residuo generado contra el costo de inventario generado por NSGA-II. Figura 2b (centro): Residuo generado contra el costo de inventario generado por µ-NSGA-II. Figura 2c (derecha): Residuo generado contra el costo de inventario generado por k-NSGA-II (Elaboración propia).

Los resultados obtenidos para la empresa estudiada con k-NSGA-II en los procesos de producción y distribución mostraron que el modelo generó prácticamente la misma cantidad de desperdicio o residuo que la producción real, sin embargo, se utilizó tan solo la mitad (50.58%) de la materia prima. El beneficio en la distribución también fue superior, obteniendo $133,000 dólares frente a los $73,320 que obtuvo la empresa según los datos entregados por la misma, es decir, un 81.40% más que el beneficio real.

5. DISCUSIÓN

El indicador de tiempo de procesamiento *(CPU)* mostrado en la Tabla 5, empleado para evaluar las funciones tiene una clara ventaja para el µ-NSGA-II, que ocupó solamente el 3% del tiempo de NSGA-II. Repitiendo así los resultados presentados por Konchada et al. (2016), en donde su comparación del micro algoritmo frente a NSGA-II obtuvo una relación de 5 a 12 segundos respectivamente, es decir, menos de la mitad (41.66%) del tiempo tomado por NSGA-II. Sin embargo, k-NSGA-II entregó soluciones más cercanas a la frontera de Pareto después del NSGA-II. Aunque µ-NSGA-II mantenga tiempos mínimos en las evaluaciones, no muestra un acercamiento sustancial a la frontera de Pareto real, es decir, el indicador de Distancia Generacional *(DG)* es muy superior en µ-NSGA-II por lo que las soluciones no fueron lo suficientemente buenas como para ser tomadas en cuenta.

En la Tabla 6 se observa que µ-NSGA-II no entregó suficiente calidad en sus soluciones. Este algoritmo propone que con 1566 unidades de materia prima se satisface la demanda de cortes, pero plantea 2280 metros de desperdicio y 2197 unidades de sobreproducción que, comparando con las soluciones entregadas por los otros dos algoritmos, son valores totalmente inviables. Los algoritmos NSGA-II y k-NSGA-II ofrecieron resultados similares. En cuanto al desperdicio generado, NSGA-II entregó 22 metros (15.27%) más de desperdicio frente a k-NSGA-II. Sin embargo, k-NSGA-II propuso 114 unidades (125%) más de sobreproducción para ser utilizadas en futuras órdenes con respecto a NSGA-II, aunque también consumió 52 unidades más de materia prima.

En la Figura 2 se visualizan las fronteras de Pareto localizadas por cada algoritmo. La Figura 2a muestra los 200 individuos de NSGA-II y una frontera de Pareto definida. Las Figuras 2b y 2c muestran una distribución de menos de 15 puntos que corresponden a las Fronteras de Pareto que entregan los algoritmos µ-NSGA-II y k-NSGA-II respectivamente. Esta línea de investigación tuvo como objetivo demostrar que es dificultoso tomar una solución de una nube de 200 puntos sin un análisis previo. En cambio, µ-NSGA-II y k-NSGA-II al mostrar menor cantidad de posibles soluciones, permiten al analista seleccionar una respuesta y aplicarla en

menor tiempo. Esta hipótesis fue compartida por Coello y Toscano (2001) y Kalmannjjee Krishnakumar (1990) en sus investigaciones.

Una vez visualizada la diferencia entre NSGA-II y sus versiones reducidas, vale la pena comparar los resultados de µ-NSGA-II y k-NSGA-II. En la Figura 2b, se muestran solamente cinco puntos cuando en realidad se evaluaron doce individuos, esto se debió a que los individuos convergieron hacia el conjunto de soluciones mostrado, es decir, hay soluciones repetidas. Además, se pudo observar que la frontera de Pareto entregada por µ-NSGA-II se encuentra en un rango del campo de búsqueda por encima de los $13500 para el costo de inventario y 1700 metros de residuo, es decir, valores muy elevados con respecto a NSGA-II y k-NSGA-II que muestran individuos desde $0.00 en el costo de inventario y menos de 50 metros en el residuo generado. Resumiendo, µ-NSGA-II no muestra una aproximación a la frontera de Pareto real que sea aplicable, en este caso a entornos reales, aunque su tiempo de respuesta sea ínfimo.

Descartado µ-NSGA-II por su falta de precisión en los resultados, se analizó la diferencia entre NSGA-II y el micro algoritmo basado en clusterización propuesto. k-NSGA-II mostró un rango de soluciones parecido al generado por NSGA-II, como se puede observar en las Figura 2a y 2c. Ambos algoritmos definieron sus fronteras de Pareto entre $0.00 y $400,000 para el costo de inventario y, entre 0 metros y 500 metros de residuo para NSGA-II, y 0 metros y 300 metros para k-NSGA-II. Es decir, sus fronteras de Pareto son similares tal y como se demostró con el indicador de distancia generacional de la Tabla 5, en donde NSGA-II se aproximó más a la frontera de Pareto real, seguido de k-NSGA-II.

Basados en esta similitud de las fronteras de Pareto, es necesario discutir por qué k-NSGA-II genera más valor que NSGA-II, teniendo en cuenta que el tiempo de procesamiento también fue similar. Para ello se tuvo que entender el contexto para el que tanto NSGA-II como k-NSGA-II fueron pensados. Los modelos matemáticos son complicados de desarrollar, pero demuestran ser precisos y útiles para la gestión de ciertos procesos y la toma de decisiones. Por este motivo, aunque NSGA-II y k-NSGA-II entregan soluciones aproximadas entre sí, k-NSGA-II es más útil que NSGA-II por la cantidad de resultados que entrega, es decir, no es lo mismo tomar una decisión de un conjunto de doscientas posibilidades, que de un conjunto de menos de quince. En el segundo caso el análisis, y en consecuencia la aplicación de la solución es más rápida, optimizando recursos al analista y a la organización que aplique este tipo de técnicas.

Para el caso de estudio, se modelaron matemáticamente los procesos de producción y distribución de la empresa y se optimizaron mediante k-NSGA-II generando resultados aplicables a la producción real para próximos periodos. De esta forma la empresa puede, en base

a este modelo y algoritmo, tomar mejores decisiones de producción, optimizar el uso de recursos e incluso modelar costos que actualmente no se toman en cuenta como son los costos de inventario. Con este tipo de herramientas, el jefe de producción es capaz de decidir en un periodo concreto si es más interesante generar menor desperdicio a consecuencia de incrementar el costo de inventarios por sobreproducción (puntos a la derecha inferior en la Figura 2c. O, al contrario, producir solamente lo demandado, es decir, sin generar inventario de sobreproducción, pero a un costo de desperdicio mayor (puntos a la izquierda en la Figura 2c).

En el modelo de distribución la lógica fue similar. En base a las demandas pronosticadas y el producto en inventario, la persona que analice el envío de productos puede seleccionar rápidamente una respuesta a su problema. k-NSGA-II entrega un conjunto reducido y preciso de soluciones que maximizan el beneficio obtenido o la satisfacción del cliente mediante la colocación de insatisfacciones en los puntos de venta. Es decir, por un lado, se obtiene el máximo beneficio por ventas de productos en un periodo determinado, pero afectando la calidad del servicio al cliente al colocar insatisfacciones en puntos de venta que tienen mayor demanda y que no necesariamente son los que mayor margen de ganancia entreguen. O, por el contrario, maximizar la atención al cliente reduciendo las insatisfacciones, pero afectando al beneficio obtenido. Este tipo de análisis se puede realizar tanto con NSGA-II como con k-NSGA-II ya que ambos entregan soluciones similares. Sin embargo, la desventaja de NSGA-II es que el analista revisa al menos doscientas respuestas para llegar a una conclusión, mientras que con k-NSGA-II en el peor de los casos, revisa quince posibles soluciones.

6. CONCLUSIONES

NSGA-II y k-NSGA-II presentaron resultados similares tanto en los indicadores utilizados como en los análisis para el caso de estudio. Sin embargo, µ-NSGA-II presentó una inexactitud tan grande que no pudo ser tomado en cuenta para aplicarse en entornos reales, aunque el tiempo que toma para evaluar funciones fue muy atractivo. En cambio, k-NSGA-II demostró ser útil y practicable para entornos de producción y distribución reales. Se optimizaron los modelos de los procesos de la cadena de suministro de la empresa consiguiendo resultados interesantes para el uso de recursos y la maximización del beneficio obtenido en ventas a través de la distribución.

k-NSGA-II aportó positivamente a la hipótesis que indicaba que el uso de micro algoritmos es suficiente para entornos reales de optimización. Este algoritmo permitió evaluar y seleccionar una solución para cada uno de los problemas planteados por el caso de estudio, entregando resultados teóricos que ya pueden ser aplicados a la producción y distribución real de la empresa.

El análisis se realizó solamente con una base de productos de la empresa y no con la totalidad que ésta mantiene actualmente, lo que representa una limitación. Por este motivo algunos de los costos y mix de productos utilizados en cada uno de los procesos de producción y distribución no presentaron la realidad como tal. Es posible que los resultados de producción puedan mejorar sustancialmente si se toma en cuenta el total de productos ya que la función de generación de patrones depende de las longitudes de cortes demandadas, siendo que, a mayor cantidad de cortes para una misma materia prima, mejores patrones se generan.

La Ecuación 2, que entrega un número de cortes máximo aceptado *(N_k)*, es sensible a valores altos, por lo que cuando en una demanda existe gran variación entre la longitud de cortes, N_k es menor y en consecuencia los patrones de corte generados toman menos cortes de los que realmente pueden soportar. Por este motivo, una línea de estudio será determinar otro método que sustituya a la Ecuación 2 que no sea sensible a esta variación de longitudes de corte. Otra línea de estudio será la unión de las decisiones tomadas por el analista según las soluciones presentadas por k-NSGA-II contra otras variables externas no evaluadas dentro del modelo. De esta forma, con suficientes datos se puede desarrollar un algoritmo inteligente basado en un problema específico que sea capaz de aprender y entregar mejores respuestas para la toma de decisiones.

7. AGRADECIMIENTO

A los miembros del Grupo de Investigación multidisciplinaria IMAGINE, a la empresa en donde se desarrolló esta tesis y a la DIUC por permitir que este tipo de investigaciones sea posible.

8. BIBLIOGRAFÍA

Arshinder, Kanda, A., & Deshmukh, S. G. (2008). Supply chain coordination: Perspectives, empirical studies and research directions. *International Journal of Production Economics*, *115*(2), 316-335, doi: 10.1016/j.ijpe.2008.05.011

Bandyopadhyay, S., & Bhattacharya, R. (2013). Applying modified NSGA-II for bi-objective supply chain problem. *Journal of Intelligent Manufacturing*, *24*(4), 707-716, doi: 10.1007/s10845-011-0617-2

Beamon, B. M. (1998). Supply chain design and analysis: Models and methods. *International Journal of Production Economics*, *55*(3), 281-294, doi: 10.1016/S0925-5273(98)00079-6

Cho, D. W., Lee, Y. H., Ahn, S. H., & Hwang, M. K. (2012). A framework for measuring the performance of service supply chain management. *Computers & Industrial Engineering*, *62*(3), 801-818, doi: 10.1016/j.cie.2011.11.014

Coello, C. A., & Toscano, G. (2001). A Micro-Genetic Algorithm for Multiobjective Optimization. *Evolutionary Multi-Criterion Optimization*, *1993*, 126-140, doi: 10.1007/3-540-44719-9_9

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. *Parallel Problem Solving from Nature PPSN VI* (Vol. 1917, pp. 849-858). Springer Berlin Heidelberg, doi: 10.1007/3-540-45356-3_83

EOI. (2010). *La gestión de la Cadena de Suministro*. Escuela de Organización industrial. Disponible en: http://www.eoi.es

Fabbe- Costes, N., & Jahre, M. (2008). Supply chain integration and performance: A review of the evidence. *The International Journal of Logistics Management*, *19*(2), 130-154, doi: 10.1108/09574090810895933

Fonseca, C. M., Paquete, L., & Lopez-Ibanez, M. (2006). An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. *2006 IEEE International Conference on Evolutionary Computation*, 1157-1163, doi: 10.1109/CEC.2006.1688440

Francisco Herrera. (2009). *Introducción a los Algoritmos Metaheurísticos*. Universidad de Granada. Disponible en: http://sci2s.ugr.es/sites/default/files/files/Teaching/OtherPostGraduateCourses/Metaheuristicas/Int-Metaheuristicas-CAEPIA-2009.pdf

Giannakis, M. (2011). Management of service supply chains with a service- oriented reference model: The case of management consulting. *Supply Chain Management: An International Journal*, *16*(5), 346-361, doi: 10.1108/13598541111155857

Gracia, C. (2010). *Métodos y algoritmos para resolver problemas de corte unidimensional en entornos realistas. Aplicación a una empresa del sector siderúrgico* [Universidad

Politécnica de Valencia]. Disponible en: https://riunet.upv.es/bitstream/handle/10251/7530/tesisUPV3250.pdf

Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, *10*(5), 477-506, doi: 10.1109/TEVC.2005.861417

Kalmannjjee Krishnakumar. (1990). Micro-genetic algorithms for stationary and non-stationary function optimization. *SPIE Intelligent Control and Adaptive Systems*, 8, doi: 10.1117/12.969927

Kassambara, A. (2018). *Datanovia*. Datanovia. Disponible en: https://www.datanovia.com

Konchada, P. K., Pragada, V. V., Chintada, A. K., & Bhimuni, V. (2016). *Optimization of Machining Parameters using Micro NSGA-II for Al6061-Silver Coated Copper Metal Matrix Composite*. 6.

Llamas, T. J. B., Baltazar, R., Araiza, M. A. C., Frau, D. C., & Rodríguez, V. M. Z. (2015). *Comparison study of micro-algorithms for lighting comfort*. 820007, doi: 10.1063/1.4913026

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. *MULTIVARIATE OBSERVATIONS*, 17.

Memari, A., Abdul Rahim, Abd. R., Hassan, A., & Ahmad, R. (2017). A tuned NSGA-II to optimize the total cost and service level for a just-in-time distribution network. *Neural Computing and Applications*, *28*(11), 3413-3427, doi: 10.1007/s00521-016-2249-0

Mula, J., Peidro, D., Díaz-Madroñero, M., & Vicens, E. (2010). Mathematical programming models for supply chain production and transport planning. *European Journal of Operational Research*, *204*(3), 377-390, doi: 10.1016/j.ejor.2009.09.008

Ochoa, G. (2014). *Optimización de corte de varillas de acero de construcción* [Universidad de Cuenca]. Disponible en: http://dspace.ucuenca.edu.ec/handle/123456789/19839

Pavez, Rafael A. (2014). Estimación de costos para la fabricación de tuberías, utilizando redes neuronales polinomiales con algoritmos genéticos. *Pontifica Universidad Católica de Valparaíso*, 77.

Pinto, T. D. S. (2008). *Uma proposta para resolver o problema de corte de estoque unidimensional com reaproveitamento de sobras por meio de dois objetivos*. Universidade Estadal de Londrina.

Riquelme, N., Von Lucken, C., & Baran, B. (2015). Performance metrics in multi-objective optimization. *2015 Latin American Computing Conference (CLEI)*, 1-11, doi: 10.1109/CLEI.2015.7360024

Sarrafha, K., Kazemi, A., & Alinezhad, A. (2014). *A Multi-objective Evolutionary Approach for Integrated Production- Distribution Planning Problem in a Supply Chain Network*. 14.

Srinivas, N., & Deb, K. (1994). Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, *2*(3), 221-248, doi: 10.1162/evco.1994.2.3.221

Toscano, G., & Coello, C. A. (2003). The Micro Genetic Algorithm 2: Towards Online Adaptation in Evolutionary Multiobjective Optimization. *Evolutionary Multi-Criterion Optimization*, *2632*, 252-266, doi: 10.1007/3-540-36970-8_18

Zhuang-Cheng Liu, Xiao-Feng Lin, Yan-Jun Shi, & Hong-Fei Teng. (2011). A Micro Genetic Algorithm with Cauchy Mutation for Mechanical Optimization Design Problems. *Information Technology Journal*, *10*(9), 1824-1829, doi: 10.3923/itj.2011.1824.1829