UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA
CAMPUS D'ALCOI

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Alcoy

Diseño, implementación y validación de un algoritmo para la resolución del Problema de Programación de Lote Económico.

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Organización y Logística

AUTOR/A: Cerdá Leal, José Antonio

Tutor/a: Poler Escoto, Raúl

Cotutor/a: Andrés Navarro, Beatriz

CURSO ACADÉMICO: 2021/2022

Resumen

El objetivo del TFM es el diseño e implementación de un algoritmo novedoso para la resolución del Problema de Programación de Lote Económico (Economic Lot Scheduling Problem - ELSP). El ELSP es un problema común en muchas industrias y trata sobre la programación de varios productos diferentes a través de una sola instalación de producción donde los tiempos de preparación y producción y las tasas de demanda de cada producto son diferentes. El algoritmo se implementará en el lenguaje de programación Python y se realizarán experimentos con datos sintéticos de problemas generados aleatoriamente, para demostrar su eficacia en la resolución del ELSP y eficiencia computacional.

Palabras clave

ELSP; algoritmo; Python; secuenciación; lote económico, heurística, metaheurística.

Resum

L'objectiu del TFM és el disseny i l'implementació d'un nou algoritme per a la resolució del Problema de Programació de Lot Econòmic (Economic Lot Scheduling Problem - ELSP). L'ELSP és un problema comú a moltes indústries i tracta sobre la programació de diversos productes diferents a través d'una sola instal·lació de producció on els temps de preparació i producció i les taxes de demanda de cada producte són diferents. L'algoritme s'implementarà en el llenguatge de programació Python i es realitzaran experiments amb dades sintètiques de problemes generats aleatòriament, per demostrar l'eficàcia en la resolució de l'ELSP i l'eficiència computacional.

Paraules clau

ELSP; algoritme; Python; secuenciación; quantitat económica, heurística, metaheurística.

Abstract

The objective of the TFM is the design and implementation of a novel algorithm for solving the Economic Lot Scheduling Problem (ELSP). ELSP is a common problem in many industries and deals with the scheduling of several different products through a single production facility where the setup and production times and demand rates for each product are different. The algorithm will be implemented in the Python programming language and experiments will be carried out with synthetic data of randomly generated problems, to demonstrate its effectiveness in solving the ELSP and computational efficiency.

keywords

ELSP; algorithm; Python; sequencing; economic lot, heuristic, metaheuristic.

Contenido

Resumen 1
Palabras clave 1
Resum 2
Paraules clau 2
Abstract 3
keywords 3
Contenido 4
Índice de ilustraciones 5
Índice de gráficos 5
Índice de tablas 5
Introducción 6
Estado del arte 6
Desarrollo del EOQ 6
Estrategias de resolución ELSP 8
Metaheurísticas 11
Modelado 16
Modelo general 17
Aplicación de la Solución Independiente 18
Aplicación del Ciclo Común 20
Desarrollo del algoritmo propio 20
Heurística 20
Metaheurística (Algoritmo Genético) 25
Test 26
Conclusiones 29
Referencias 30

Índice de ilustraciones

Ilustración 1 Secuencia Recocido Simulado 12
Ilustración 2 Secuencia Algoritmo Genético 13
Ilustración 3 ejemplo de población AG 14
Ilustración 4 Ejemplo reproducción y mutación AG 16
Ilustración 5 Secuencia heurística 22

Índice de gráficos

Gráfico 1 SI artículo 1 19
Gráfico 2 SI artículo 2 19
Gráfico 3 SI artículo 3 19
Gráfico 4 SI todos los artículos 19
Gráfico 5 solución CC 20
Gráfico 6 Solución CC con demanda aleatoria 25

Índice de tablas

I Nomenclatura EOQ 7
II Ejemplo reparto porcentual inversamente proporcional 15
III Notación modelo general ELSP 17
IV Datos ejemplo SI 18
V Datos ejemplo CC 20
VI Simbología algoritmo propio 21
VII Datos test 4 artículos 27
VIII Resultados test 4 artículos 27
IX Datos test 10 artículos 28
X Resultados test 10 artículos 28

Introducción

Para satisfacer rápidamente las necesidades de demanda, las compañías suelen disponer de mercancía almacenada que esperan vender. A partir de este supuesto nace la teoría de gestión de inventarios, un área de indudable importancia económica para la industria, donde se desarrollan reglas y modelos que puede usar la dirección de operaciones para minimizar los costes relacionados con el almacenamiento de mercancías además de satisfacer las necesidades de los clientes.

Los modelos, programas o simulaciones que representan la realidad son herramientas de apoyo que emplean los departamentos de ingeniería para la ayuda de toma de decisiones. Por un lado, la creación y generación del modelo ayuda al equipo desarrollador a tener una visión real sobre los problemas que se presentan, y por otro, el modelo permite simular posibles situaciones futuras para anticiparse con éxito a los acontecimientos.

En este trabajo se pretende desarrollar una heurística genérica basada en la teoría de inventarios reforzada con una metaheurística que aporte una solución al problema de programación del lote económico ELSP *(Economic Lot Scheduling Problem)*.

Estado del arte

El ELSP o problema de programación de lote económico se lleva analizando formalmente desde la década de los años cincuenta (Rogers 1958). La revisión de la literatura que se muestra en este apartado sigue la evolución del modelo general que nace a partir del famoso modelo de lote económico o EOQ (Harris 1913). A partir de este sistema, se presentan distintas estrategias que pretenden fijar tiempos de ciclo para establecer patrones repetitivos de control de pedidos, eso sí, bajo condiciones de demanda ideales. Posteriormente se desarrollan distintos algoritmos hasta construir una heurística capaz de aportar una solución a este tipo de problemas, con el añadido de contemplar la demanda como una variable aleatoria.

Desarrollo del EOQ

Si se disponen de conocimientos básicos de teoría de inventarios, lo más probable es que nos venga a la cabeza alguna versión del modelo de lote económico de pedido EOQ que se puede emplear para la toma de decisiones cuando la demanda es periódica, continua y conocida. El modelo de Harris se centra entonces en calcular el tamaño de lote de cada producto en cada pedido, así como en conocer la periodicidad de abastecimiento para reducir al máximo los costos de inventario. Para ello, en el modelo EOQ se plantea calcular los costos de gestión de inventarios como la suma de los costos de almacenamiento y los costos de emisión de pedidos, bajo la premisa de contar con una demanda constante y continua.

El modelo EOQ básico es óptimo siempre y cuando se analice una situación ideal de demanda y plazos de entrega constantes para poder realizar pedidos continuos y repetitivos. Es difícil que se den este tipo de condiciones en el mundo empresarial, pero este modelo actúa como referencia para distintas situaciones además de ser la base de muchas investigaciones dentro de la teoría de lotificación de pedidos.

Para aplicar correctamente el EOQ, de antemano se debe asumir que la demanda es determinística y ocurre a una tasa constante, que los costes de lanzamiento de pedido independientemente de la cantidad son constantes, que el tiempo de recepción de los pedidos es nulo y los costes de almacenamiento unitarios por unidad de tiempo son constantes, además de que no se permite la escasez de unidades.

En la I Nomenclatura EOQ se muestra la nomenclatura empleada durante la explicación del modelo EOQ.

I Nomenclatura EOQ

Simbología	Descripción
D	Demanda unitaria por unidad de tiempo
k	Costo de lanzamiento de pedido de q unidades
p	Costo individual del producto
h	Costo de mantener una unidad de producto durante un periodo de tiempo
q	Tamaño de lote
T	Periodo de tiempo entre pedidos.

A partir de estos supuestos, el modelo define que el costo total de inventario $CT(q)$ es:

$$CT(q) = \frac{kD}{q} + pD + \frac{hq}{2} \quad (1)$$

Para calcular el tamaño de lote que minimiza los costos, se establece $CT'(q) = 0$

Como resultado, se define que el tamaño de lote que minimiza los costes de aprovisionamiento y almacenamiento corresponden con el resultado de la ecuación (2):

$$q^* = \sqrt{\frac{2kD}{h}} \quad (2)$$

A partir de la ecuación (2) se observa que el precio unitario del producto p no influye en q^* debido a que el tamaño de cada pedido no modifica el precio del artículo (restricción en el planteamiento inicial del problema).

De igual forma, si se desea conocer el tiempo que debe transcurrir entre la realización de distintos pedidos se trabaja con la misma ecuación (2) aplicando un cambio de variable, quedando la siguiente expresión (3):

$$CT(T) = \frac{k}{T} + \frac{Dht}{2} \quad (3)$$

Para conocer el tiempo entre pedidos que minimiza la función se establece $CT'(T) = 0$

Quedando como resultado (4):

$$T^* = \sqrt{\frac{2k}{hd}} \quad (4)$$

La suposición de recibir los pedidos de forma instantánea queda lejos de la realidad. En el modelo EOQ de tasa continua se pretende eliminar la asunción de recibir la materia prima de forma inmediata. En esta versión del modelo se plantea la existencia de un tiempo entre la primera recepción del pedido y la última. Por este motivo se añade al modelo una nueva variable de velocidad r que determinará la cantidad de unidades recibidas por unidad de tiempo. Teniendo en cuenta que la recepción de los artículos no se genera de forma instantánea y que la demanda es constante y continua, el inventario máximo se verá reducido en comparación al EOQ básico siguiendo el siguiente supuesto $DT(1-\frac{D}{r})$ quedando la expresión de costes totales (5):

$$CT(T) = \frac{k}{T} + \frac{Dht}{2}(1-\frac{D}{r}) \qquad (5)$$

Si se desea conocer el periodo de tiempo entre pedidos que minimiza la ecuación de costos, al igual que ocurría con anterioridad, se aplica $CT'(T) = 0$ y se obtiene (6):

$$T^* = \sqrt{\frac{2k}{hd(1-\frac{D}{r})}} \qquad (6)$$

Estrategias de resolución ELSP

El modelo EOQ básico y sus variantes, buscan de manera matemática minimizar los costes de inventarios al realizar pedidos a proveedores externos. Pero ¿qué ocurre si el proveedor de varios materiales es una sección interna de la empresa que necesita fabricar dichos productos, y que además se crean en una sola instalación? En la aplicación del EOQ base no se contempla la necesidad de compartir recursos de producción, de modo que, al aplicarlo a la pregunta anterior, gran parte de las soluciones que se obtienen bajo estas condiciones no son factibles por la saturación de recursos productivos en ciertos periodos de tiempo.

Para poder disponer de soluciones factibles bajo el enfoque de compartir recursos productivos para la fabricación de distintos artículos, es necesario combinar la teoría de inventarios con métodos de secuenciación.

El tipo de problema que se analiza durante la extensión de este trabajo pertenece a lo que en la literatura se conoce como secuenciación del lote económico, o por sus siglas en inglés ELSP (economic lot scheduling problem). El ELSP se centra en la programación de la producción de varios artículos diferentes en una única instalación o máquina en la que simultáneamente solo se puede fabricar un único producto, con el objetivo de satisfacer la demanda al cliente minimizando los costes relacionados con la gestión de inventarios. Para aplicar la versión más conocida y simple del ELSP se deben asumir las siguientes restricciones:

- La máquina produce un único producto simultáneamente
- La capacidad de producción es limitada pero suficiente para satisfacer a la demanda
- Existe un tiempo y costo de *setup* constante para cada lanzamiento de lote de producto.
- Los costes de tiempo y setup son independientes de la secuencia
- La demanda de cada producto es conocida y constante a lo largo del horizonte de planificación
- El ratio de producción de cada producto es conocido y contante
- Los costes de inventario son proporcionales a los niveles de stock.

Para realizar la programación de la producción y el control de inventarios, se deben tener en cuenta dos aspectos fundamentales:

- Dimensionamiento de lotes: unidades a producir en cada tirada de fabricación de un único producto
- Programación: Cuándo se deben producir los lotes para entregar a tiempo minimizando el costo de almacenamiento.

Teniendo en cuenta estas características básicas en la programación de la producción, este tipo de problema se clasifica dentro de la teoría de la complejidad computacional como NP-duro debido a la relación que existe al incrementar el número de artículos a fabricar con el crecimiento del número de combinaciones posibles. Es por ello por lo que para obtener una solución suficientemente buena y de forma relativamente rápida, se emplean algoritmos y modelos informáticos.

En la literatura aparecen distintas aproximaciones para la resolución del ELSP al igual que ocurre con el EOQ, las variantes más sencillas del modelo aportan soluciones asumiendo un mayor número de restricciones.

Solución independiente

Como se ha comentado con anterioridad, en el planteamiento del ELSP se deben tener en cuenta dos parámetros fundamentales como son el dimensionamiento de lotes y la programación de estos. Las soluciones más sencillas del ELSP se centran en determinar tiempos de ciclo de cada artículo, es decir, se calcula el número de unidades temporales que existen entre dos periodos de fabricación. A la variante más simple del ELSP se le conoce como solución independiente (SI), y resulta de aplicar el modelo EOQ de tasa continua a cada uno de los productos. Transformando la estrategia del lote económico de pedido, se obtiene la siguiente expresión (7) a partir de esta se define $T^{SI}{}_i$ (8) al tiempo de ciclo óptimo de cada producto i que minimiza los costos de gestión y almacenamiento.

$$CT = \sum_{i=1}^{n} CT_i = \sum_{i=1}^{n} \frac{k_i}{T_i} + \sum_{i=1}^{n} \frac{D_i h_i T_i}{2}\left(1 - \frac{D_i}{r_i}\right) \quad (7)$$

$$T^{SI}{}_i = \sqrt{\frac{2k_i}{h_i D_i (1 - \frac{D_i}{r_i})}} \quad (8)$$

Al aplicar la variante de Solución Independiente se consiguen definir los tiempos de ciclo óptimos de cada artículo, eso sí ignorando por completo las capacidades de los recursos productivos. Al emplear la SI gran parte de los resultados obtenidos pueden ser no factibles debido a la saturación de los recursos productivos en ciertos periodos de tiempo.

En caso de que esta aproximación consiga entregar una solución factible, el costo total del sumatorio de cada uno de los n artículos representará una cota inferior en los costes consiguiendo así la solución óptima al problema.

Ciclo común

En 1962, Hanssmann realiza la primera publicación de la variante Ciclo Común. En este modelo, el autor plantea que para obtener una solución factible se debe definir una secuencia ordenada

de los n productos que se repetirá cíclicamente cada cierto periodo de tiempo T. Con este planteamiento, cada uno de los productos comparte el mismo tiempo de ciclo T.

Al igual que ocurría con el modelo de solución independiente, para conocer la ecuación que define el tiempo T del problema, previamente se debe partir de expresión (9) del costo total.

$$CT = \sum_{i=1}^{n} \frac{k_i}{T} + \sum_{i=1}^{n} \frac{D_i h_i T}{2}\left(1 - \frac{D_i}{r_i}\right) \quad (9)$$

Conocida la ecuación de costo total el tiempo del modelo de ciclo común se obtendrá aplicando la derivada del costo total en función del tiempo dando como resultado (10):

$$T^{CC} = \sqrt{\frac{2\sum_{i=1}^{n} k_i}{\sum_{i=1}^{n} h_i D_i (1 - \frac{D_i}{r_i})}} \quad (10)$$

El método de ciclo común es un modelo sencillo y factible en el que cada uno de los n artículos comparten el mismo tiempo de ciclo T^{CC}. Esta generalidad común a cada uno de los artículos implica que se están aplicando tiempos de ciclo no óptimos generando un mayor costo en la función objetivo del problema hasta el punto en que se suele considerar que con esta solución se consigue una cota superior del problema. Sin embargo, según Jones e Inman (1989) en las situaciones en que el ratio entre el costo de preparación de máquina y el costo de inventario es muy similar en cada uno de los artículos esta solución queda próxima a la óptima.

Periodo Básico

El modelo del Periodo Básico aparece publicado en 1966 por Bomberger. En la metodología se plantea la opción de definir diferentes tiempos de ciclos de fabricación para cada uno de los n productos, siendo estos múltiplos enteros de un tiempo de ciclo común a cada uno de los artículos.

$$T_i = C_i T^{PB} \quad (11)$$

Quedando la ecuación (12) de coste total como:

$$CT = \sum_{i=1}^{n} \frac{k_i}{C_i T^{PB}} + \sum_{i=1}^{n} \frac{D_i h_i C_i T^{PB}}{2}\left(1 - \frac{D_i}{r_i}\right) \quad (12)$$

Cabe destacar que el tiempo de ciclo del periodo básico, debe de ser suficiente como para abarcar los tiempos de ocupación de máquina del conjunto de cada uno de los productos.

Desde la publicación de este modelo, se han desarrollado distintas metodologías para proceder al cálculo de los tiempos de ciclo, no obstante, el procedimiento más empleado se atribuye a Doll y Whybark (1973) que plantean un proceso iterativo en el que se inicia calculando el tiempo de ciclo óptimo de cada artículo (8) y parte del menor de estos como referencia para la primera iteración del PB. A partir de este resultado, se calculan las constantes de tiempo C_i para cada artículo y se estima un nuevo periodo básico aplicando (13)

$$T^{PB} = \sqrt{\frac{\sum_{i=1}^{n} \frac{k_i}{C_i}}{\sum_{i=1}^{n} \frac{D_i h_i C_i}{2}\left(1 - \frac{D_i}{r_i}\right)}} \qquad (13)$$

Este nuevo periodo básico es la nueva variable del proceso iterativo, de modo que se vuelven a calcular las constantes de tiempo C_i de los n productos. El proceso iterativo finaliza cuando se consigue el mismo valor en dos iteraciones simultaneas quedando definido el T^{PB} del problema.

Metaheurísticas

Cuando se desea aportar una solución a un problema complejo uno de los primeros criterios a seguir es el de definir una serie de restricciones que permitan aislar las variables con más repercusión del sistema analizado, creando así un problema más simple y genérico que se puede emplear como base para un gran número de casos.

Una vez obtenida una respuesta válida a este problema genérico y acotado con gran número de restricciones, se va aumentado la dificultad del procedimiento resolutivo, bien para mejorar la respuesta del algoritmo o bien eliminando restricciones o añadiendo nuevas variables para implementar un nuevo modelo personalizado que tenga una mayor apariencia al problema real.

Las metodologías presentadas anteriormente, son los modelos más empleados por los investigadores como punto de partida para desarrollar nuevas estrategias que aporten solución al ELSP, pero como se ha detallado en apartados anteriores estas soluciones solamente son válidas bajo ciertas restricciones.

Al aplicar el modelo de Ciclo Común en condiciones de demanda estable y constante, el problema de secuenciación de productos queda muy simplificado, aplicando la misma secuencia de producción cada tiempo de ciclo. En caso de eliminar las restricciones que idealizan el comportamiento de la demanda, la variable de secuenciación de productos asume una gran influencia y dificulta la obtención de la solución óptima.

La dificultad de modelar un algoritmo capaz de entregar una solución óptima al problema del ELSP teniendo en cuenta la aleatoriedad de la demanda, invita a los investigadores a emplear otras técnicas de resolución de problemas como son las metaheurísticas.

Una metaheurística, es un procedimiento computacional genérico, es decir, no están diseñados para problemas específicos, de modo que son algoritmos flexibles que se pueden adaptar con relativa facilidad a distintos problemas.

Las metaheurísticas son algoritmos cíclicos que emplean la aleatoriedad como proceso para entregar una solución. A diferencia de las heurísticas o algoritmos específicos donde no está presente la aleatoriedad en la resolución del problema, una metaheurística puede entregar distintas soluciones bajo los mismos datos de entrada.

La mayoría de metaheurísticas se emplean en problemas de combinatoria para conseguir una solución factible evitando así tener que valorar todas las combinaciones posibles.

Existen varios tipos de metaheurísticas, pero para el tipo de problema del ELSP las más empleadas son el Recocido Simulado y sobre todo los Algoritmos Genéticos.

Recocido Simulado

La idea del Recocido Simulado surge a partir del procedimiento físico conocido como recocido, en el que se eleva la temperatura de un sólido que posteriormente se va reduciendo lentamente hasta conseguir una estructura cristalina.

El recocido, suele ser un proceso físico relacionado con la ciencia de materiales y metalurgia. En este proceso al elevar la temperatura del metal, se permite que cualquier átomo que esté mal colocado dentro de la retícula de la estructura fluya libremente a lo largo del material. Durante el proceso de enfriamiento, se espera que cada uno de los átomos busque una configuración de mínima energía.

Si se enfría rápidamente el material cabe la posibilidad de atrapar algún átomo en una situación en la que no esté bien colocado, de modo que, al enfriar el sólido de forma paulatina, se permite que cada uno de los átomos vaya encontrando su ubicación dentro de la retícula, de tal forma que al final se consiga un material con una estructura cristalina.

El recocido simulado va a ser entonces un algoritmo de optimización metaheurística que va a utilizar como metáfora los procesos termodinámicos de recocido, se aceptarán variaciones agresivas de la solución en etapas iniciales mientras que en etapas finales del algoritmo los cambios en la solución serán menores. Este algoritmo fue propuesto en un artículo publicado en Science en el año 1983 por Kirkpatrick, Gelatt y Vecci.

Ilustración 1 Secuencia Recocido Simulado

```
introducción de datos(
    temperatura= T,
    temperatura final= Tf,
    a #función de mecanismo de descenso de temperatura
    L #tiempo que se va a permanecer en una temperatura dada
    )

    x=solución
    while T<=Tf:
        for cont in range(1,L):
           crea xc # nueva solución candidata
           evaluación x vs xc #evaluación de ambas solucines en la función objetivo
           if (evaluación x) > (evaluación xc):
               Criterio de Metropolis() #calcula la probabilidad de aceptar una solución peor.
               if xc=Aceptada:
                   x=xc
            else:
                   x=xc
            T=T*a
```

El algoritmo de Recocido simulado se inicia con una solución aleatoria *x,* a continuación, se selecciona una solución vecina[1] *x'* y se evalúan ambas.

Si la solución vecina mejora el valor de la función objetivo se acepta con probabilidad 1, y en caso contrario se calcula la probabilidad de aceptación mediante el criterio de Metrópolis dado por (14). Una vez calculada la probabilidad de aceptación, se lanza un numero aleatorio y en caso de que éste sea menor a la probabilidad de aceptación, se acepta como válida la solución candidata *x'*,

[1] Una solución vecina parte de una solución actual, a la cual se le aplica algún cambio en alguno de sus parámetros con el objetivo de encontrar una solución óptima para el problema de optimización. El cambio que se aplica al vector solución de un individuo depende del algoritmo utilizado.

$$\text{probabilidad [aceptar x']} \begin{cases} 1 & x' < x \\ \exp\left(-\frac{x' - x}{T}\right) & x' \geq x \end{cases} \quad (14)$$

Se aceptarán soluciones peores para escapar de óptimos locales[2], pero claro, no se aceptarán soluciones peores de forma incontrolada, de ser así, se estaría vagando por el espacio de soluciones acercándose a una búsqueda aleatoria que no es una búsqueda eficiente. Por lo tanto, se define un parámetro llamado temperatura que va a controlar todo el proceso de aceptación de soluciones. Inicialmente el parámetro temperatura será alto permitiendo que las opciones vaguen por el espacio de soluciones.

Existen diferentes mecanismos de descenso del parámetro temperatura, el más habitual es el descenso exponencial. A medida que vaya descendiendo la temperatura la solución cada vez será más robusta, siempre y cuando los parámetros del algoritmo se adecuen correctamente al problema a resolver.

Algoritmo genético

Como se ha comentado anteriormente el recocido simulado emplea técnicas termodinámicas de recocido como metáfora para formar el procedimiento de desarrollo de la metaheurística.

Los algoritmos genéticos se inspiran en la teoría de la evolución o teoría Darwiniana, donde básicamente se define que las especies van cambiando y evolucionando con el paso del tiempo y donde los individuos mejor adaptados son los que más tiempo suelen prevalecer.

Basándose en la teoría de la evolución de Charles R. Darwin y los conocimientos sobre la transferencia evolutiva, en 1970 John Holland definió un método de cálculo de búsqueda de soluciones basadas en las normas de selección natural. A partir de la implementación de estos métodos y estrategias nacen los Algoritmos genéticos.

Ilustración 2 Secuencia Algoritmo Genético

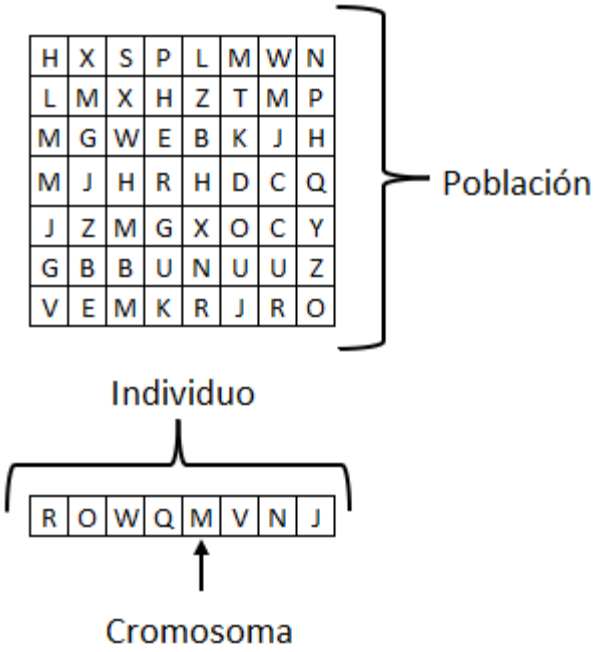
```
def run_alg(self):

        introducción de datos(
            Nº artículos=N,
            poblacion=P,
            generaciones=G,)

        crea_individuos() #cantidad=P
        Evaluación de individuos () #función fitness
        crear población() #cantidad=P
        for i in range(0,G):
            Evaluación de individuos()
            reproduccion y mutación()
            actualizar población() #salida peores individuos,
                                   #entrada nuevos individuos
            return población
```

[2] un óptimo local de un problema de optimización es una solución que es óptima (ya sea máxima o mínima) dentro de un conjunto vecino de soluciones candidatas. Esto contrasta con un óptimo global , que es la solución óptima entre todas las soluciones posibles , no solo aquellas en una vecindad particular de valores.

Los algoritmos genéticos emplean lo que se conoce conceptualmente como operadores genéticos: Selección, reproducción, y mutación.

Inicialmente, el algoritmo genético requiere de un conjunto de soluciones. Cada una de estas soluciones recibe el nombre de individuo y el conjunto de soluciones se define como población.

Ilustración 3 ejemplo de población AG



Para inicializar la metaheurística, será necesario definir adecuadamente para cada problema el tamaño de la población y el número de generaciones o ciclos que realizará el algoritmo.

El tamaño de la población que se defina inicialmente tendrá una repercusión directamente proporcional con el tiempo de ejecución del algoritmo, por otro lado, cabe destacar que a mayor número de individuos registrados en la población mayores probabilidades de conseguir mejores resultados en la función objetivo del problema.

Conocidos los parámetros de entrada del algoritmo genético se define como concepto de selección a la generación aleatoria de los individuos o soluciones que formarán la población inicial.

Cada uno de los individuos de la población es evaluado por una función de aptitud o fitness, que entregará una calificación sobre la bondad de cada una de las soluciones creada de forma aleatoria. Por supuesto, gran parte de las soluciones creadas inicialmente no funcionarán en absoluto y en posteriores generaciones serán eliminadas. Sin embargo, por puro azar, algunos individuos pueden contener parte de la solución, aunque esta todavía sea débil e imperfecta. Por lo tanto, en este momento se puede definir que el algoritmo genético es un método de búsqueda dirigida basado en probabilidades.

Una vez evaluadas cada una de las soluciones que forman la población, se definirá la probabilidad de reproducción. Los individuos mejor adaptados tendrán una mayor probabilidad de reproducirse. Para ello se realiza un reparto porcentual proporcional en caso de que la función objetivo sea de maximización y un reparto porcentual inversamente proporcional en caso de que la función objetivo del algoritmo sea de minimización.

En II Ejemplo reparto porcentual inversamente proporcional se muestran valores fitness de distintos individuos, y en este caso en concreto al ser de minimización la función objetivo, los individuos con menor fitness son los mejor valorados.

II Ejemplo reparto porcentual inversamente proporcional

Fitness	Reparto inversamente proporcional	Reparto inversamente proporcional acumulado
159	9,042%	9,04%
172	8,359%	17,40%
172	8,359%	25,76%
200	7,188%	32,95%
200	7,188%	40,14%
208	6,912%	47,05%
222	6,476%	53,52%
231	6,224%	59,75%
231	6,224%	65,97%
245	5,868%	71,84%
245	5,868%	77,71%
255	5,638%	83,35%
257	5,594%	88,94%
259	5,551%	94,49%
261	5,508%	100,00%

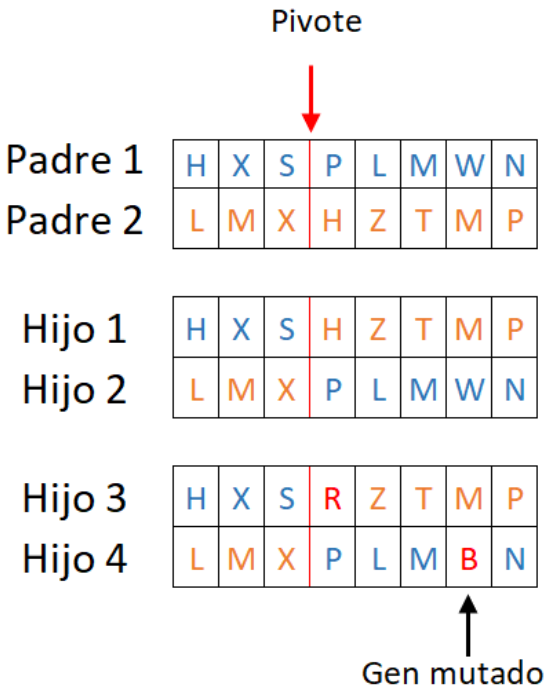
Para finalizar la etapa dentro del concepto de selección, una vez conocida la probabilidad de reproducción de cada uno de los individuos, se procede al proceso de selección, donde se requiere lanzar un numero aleatorio que seleccionará a un individuo como padre para la fase de reproducción. El candidato de la elección será aquel individuo con el valor superior más cercano al número aleatorio lanzado. Este proceso se repetirá una vez más para obtener dos individuos candidatos que recibirán el nombre de padre.

La siguiente etapa que se aborda en el algoritmo genético es la etapa de reproducción o cruce en la que cada uno de los padres candidatos se combinan para formar un nuevo individuo que contendrá información genética de cada uno de los padres.

Existen diferentes técnicas de cruce, pero la más común y utilizada es el cruce de 1 punto o pivote. Una vez seleccionados los padres, los cromosomas de cada individuo se separan en un punto seleccionado aleatoriamente para generar dos segmentos diferenciados: la cabeza y la cola. Al intercambiarse las colas de los padres se generan dos nuevos descendentes. De esta manera cada uno de los descendentes u hijos obtienen información genética de cada uno de los padres.

La etapa de mutación se completa realizando una modificación en un cromosoma elegido de forma aleatoria en cada uno de los nuevos individuos. Lo más habitual es que en esta fase la mutación tenga presencia con una probabilidad inferior al 1% en cada una de las generaciones, y que, al mutar los nuevos individuos, estos se sustituyan por los generados en la etapa de reproducción o cruce. Esto se debe sobre todo a que los individuos suelen tener un ajuste menor después de ser mutados.

Ilustración 4 Ejemplo reproducción y mutación AG



Sin embargo, en el algoritmo genético que se presentará posteriormente, se ha preferido modificar la etapa de mutación permitiendo que esta aparezca en cada una de las generaciones. En este caso se crean cuatro nuevos individuos, los dos resultantes de la combinación de cada uno de los padres y dos nuevos individuos que difieren de estos por haber sido mutados al modificarse de forma aleatoria uno de sus cromosomas.

La mutación, permite evitar que los individuos de la población queden atrapados en óptimos locales.

Para finalizar con el ciclo del algoritmo genético, se deberán eliminar de la población aquellos individuos peor valorados para ser sustituidos por las nuevas soluciones.

En este momento finaliza la primera generación del algoritmo genético. La secuencia de evaluación, reproducción y mutación se repite cíclicamente hasta alcanzar un número determinado de generaciones o una valoración suficientemente buena de la solución obtenida.

Modelado

Definidas las estrategias que aportan solución al problema del ELSP, en este apartado se presenta el modelo general con la notación empleada, así como la aplicación de la estrategia de la Solución Independiente y a la solución de Ciclo Común. A continuación, se presentará un algoritmo basado en la técnica de Ciclo Común para el análisis del ELSP con condiciones de demanda aleatoria y conocida. Posteriormente, se presenta un algoritmo genético que pretende determinar que secuencia de productos se debe de seguir en cada uno de los ciclos determinados por la. En el modelo que se presenta, se permite deshabilitar la ejecución del algoritmo genético debido a que el tiempo de calculo que necesita es superior a la heurística, agravándose el tiempo de ejecución de forma exponencial al aumentar el número de artículos.

En la III Notación modelo general ELSP se define la notación que se empleará en el desarrollo del modelo general del problema.

III Notación modelo general ELSP

Símbolo	Definición
i	*Índice de artículo, i=1..n*
D_i	*Demanda del artículo i (unidades por unidad de tiempo)*
r_i	*Ratio de producción del artículo i (unidades por unidad de tiempo)*
k_i	*Costo de preparación de la máquina para iniciar la producción del artículo i (unidades de tiempo)*
h_i	*Costo de almacenamiento del artículo i (unidad por unidad de tiempo)*
C_i	*Tiempo de cambio de partida del artículo i*
T_i	*Tiempo de ciclo para el producto i*
T^{cc}	*Tiempo de ciclo (modelo de Ciclo Común)*
CT	*Costo total (unidades monetarias)*

Modelo general

Como se ha comentado con anterioridad, el modelo general del problema parte del planteamiento del famoso problema de lote económico o EOQ. El problema de secuenciación del lote económico o ELSP aparece en el momento en que distintos artículos compiten por un mismo recurso productivo. En este momento surge la necesidad de acomodar patrones cíclicos de producción que consiguen evitar periodos de saturación en los recursos productivos como pueden ser instalaciones o máquinas.

Como se ha visto en anterioridad, existen distintas variantes que aportan solución al problema, no obstante, para definir el modelo general se deben de asumir las siguientes restricciones:

- La máquina produce un único producto simultáneamente
- La capacidad de producción es limitada pero suficiente para satisfacer a la demanda
- Existe un tiempo y costo de *setup* constante para cada lanzamiento de lote de producto.
- Los costes de tiempo y setup son independientes de la secuencia
- La demanda de cada producto es conocida y contante a lo largo del horizonte de planificación
- El ratio de producción de cada producto es conocido y contante
- Los costes de inventario son proporcionales a los niveles de stock.
- No se contemplan ni roturas de stock ni limitaciones en el tamaño del inventario de cada artículo.
- No se contempla que exista escasez de materia prima para la producción de alguno de los productos.

Como ya se ha visto en este documento, las estrategias que consiguen entregar una solución factible del ELSP pretenden definir una secuencia de productos que se repetirá cíclicamente simplificando en gran medida la problemática de secuenciación de productos. Al mismo tiempo al emplear esta estrategia se evita la aparición del fenómeno de interferencia en el que dos o más artículos compiten por un mismo recurso productivo.

En cada uno de los ciclos, se produce un lote de cantidad q_i^* de cada uno de los artículos que permite satisfacer la demanda $Q_i = D_i T_i$ durante el tiempo del periodo de ciclo T_i.

El tiempo que se requiere para fabricar un lote con una cantidad q_i^* viene definido por (15):

$$L_i = \frac{D_i T_i}{r_i} \tag{15}$$

Cabe tener en cuenta que la ocupación de la máquina para la fabricación de un producto se compone de: la suma del tiempo de fabricación L_i y el tiempo de preparación de máquina para dicho artículo c_i

Al igual que pasaba con el modelo de tasa continua del EOQ, la fabricación de los productos no es instantánea, de modo que, durante el tiempo de producción la demanda no es interrumpida de tal forma que, el nivel máximo del inventario se puede calcular según (16) quedando el nivel de inventario medio como(17):

$$I_{max_i} = D_i T_i \left(1 - \frac{D_i}{r_i}\right) \tag{16}$$

$$\bar{I}_i = \frac{D_i T_i}{2}\left(1 - \frac{D_i}{r_i}\right) \tag{17}$$

Aplicación de la Solución Independiente

A continuación, se expone un pequeño ejemplo extraído de la tesis publicada por el Doctor Raúl Cortés Fibla.

En este pequeño ejemplo se pretenden mostrar los inconvenientes que presenta la resolución de la Solución Independiente, aunque solamente intervengan 3 artículos en el ejemplo.

Como ya se ha visto anteriormente, la aproximación de la Solución Independiente varía directamente del modelo EOQ y en caso de obtener una solución factible, esta será considerada como cota inferior de la función objetivo del problema.

IV Datos ejemplo SI

Datos						Resultado		
Producto (i)	r_i (uds/día)	Costo Setup	Tiempo de Setup(día)	H_i	D_i (Uds/día)	T^{SI}	L_i	$L_i + Ci$
1	2000	15	1	0.1	200	20	2	3
2	1000	30	2	0.064	100	50	5	7
3	2400	20	2	0.09	125	30	1	3

En las siguientes figuras, cada uno de los colores representa un artículo. Se puede observar el nivel de inventario de cada uno de los ítems representado por la línea continua. En el fondo del gráfico se colorea del mismo color que el producto i, para representar el tiempo de ocupación de máquina.

Realizando la simulación de cada artículo por separado y según el criterio de la Solución Independiente el resultado se puede observar en Gráfico 1 SI artículo 1, Gráfico 2 SI artículo 2, Gráfico 3 SI artículo 3

En el Gráfico 4 SI todos los artículos, se muestra el resultado de la simulación de los tres artículos

Gráfico 1 SI artículo 1

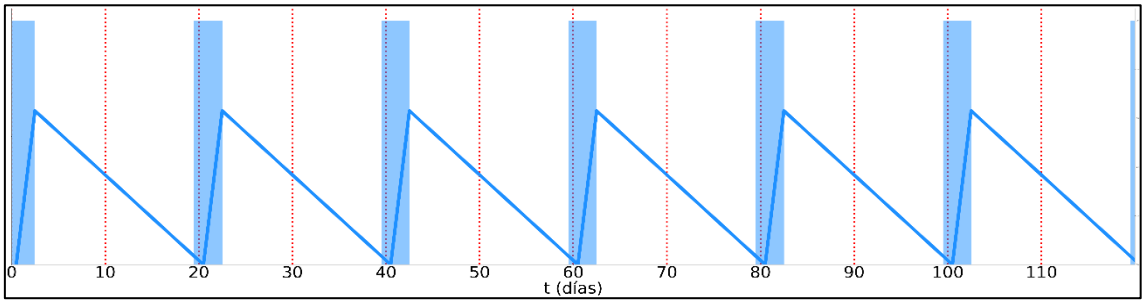


Gráfico 2 SI artículo 2

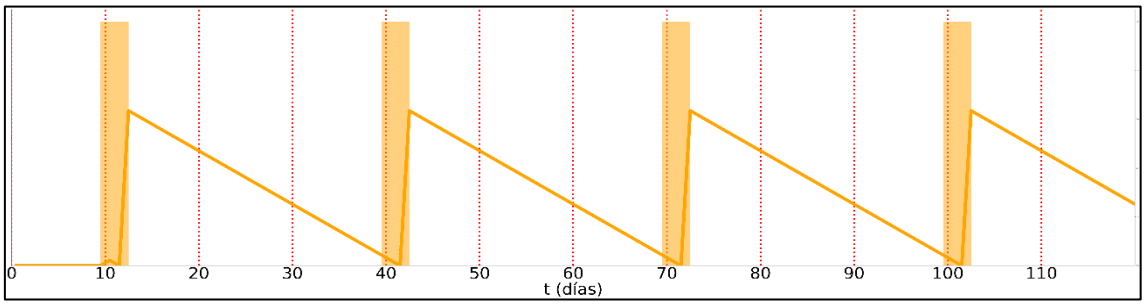


Gráfico 3 SI artículo 3

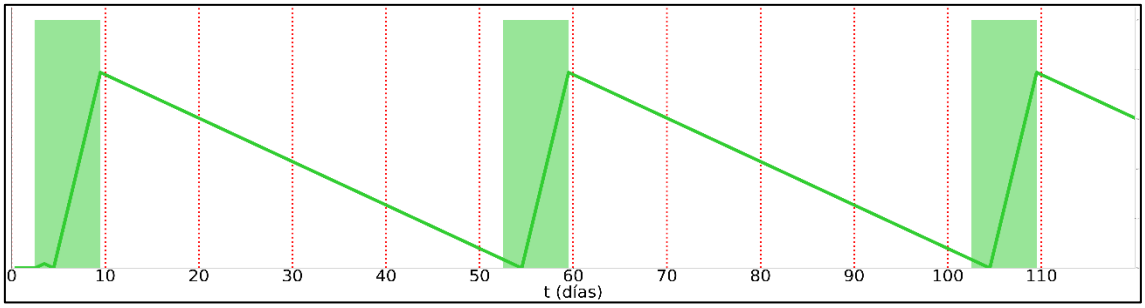
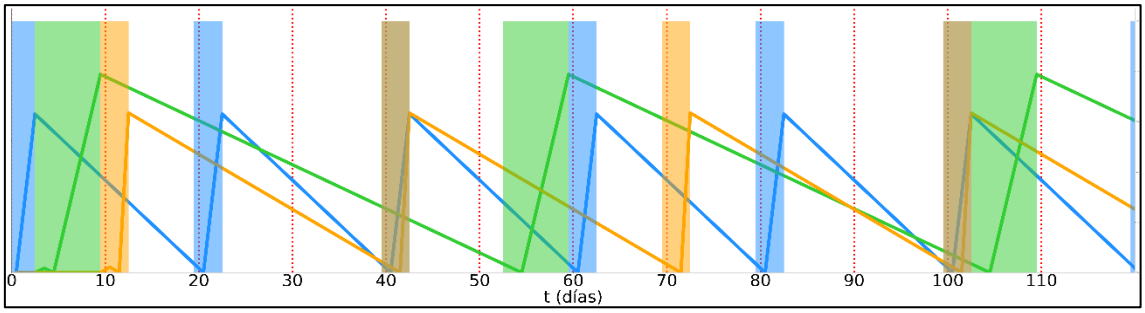


Gráfico 4 SI todos los artículos



En el caso que se muestra en el ejemplo, se puede observar que en el día 40 tanto el artículo 1 como el 3, demandan el mismo recurso productivo para satisfacer la demanda. La máquina que se emplea para la fabricación de productos permanece ociosa en unos periodos de tiempo

mientras que en otros satura. Es en este momento donde se debe prestar gran atención a la programación de cada uno de los productos para que no se presente el fenómeno de interferencia.

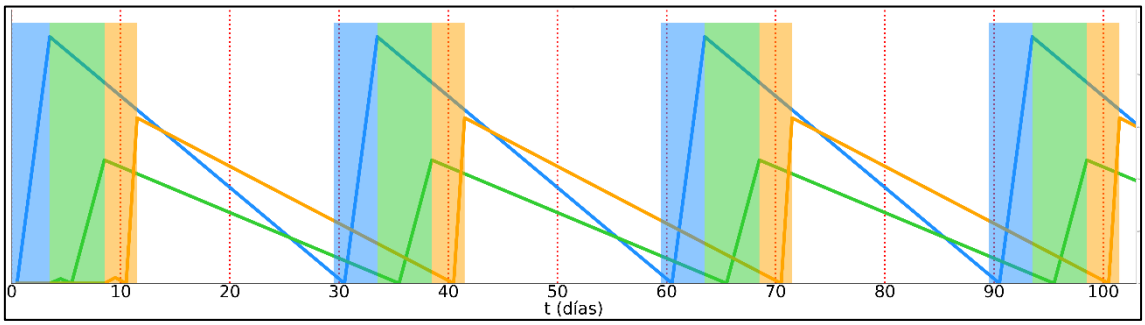
Aplicación del Ciclo Común

El método de ciclo común es un modelo sencillo y factible en el que cada uno de los n artículos comparten el mismo tiempo de ciclo T. Esta generalidad común genera que los valores obtenidos tanto de T_i como de q_i^* y L_i difieran de los óptimos calculados en la metodología de la SI. Estos valores generan un peor resultado en la función objetivo del problema relacionándose directamente con un aumento en los costes de inventario y considerando que se consigue una cota superior al aplicar esta resolución.

V Datos ejemplo CC

Datos						Resultado		
Producto (i)	r_i (uds/día)	Costo Setup	Tiempo de Setup(día)	H_i	D_i (Uds/día)	T^{CC}	L_i	$L_i + Ci$
1	2000	15	1	0.1	200	30	3	4
2	1000	30	2	0.064	100	30	3	5
3	2400	20	2	0.09	125	30	1	3

Gráfico 5 solución CC



Como se puede observar en Gráfico 5 solución CC, esta solución acomoda patrones de producción que se van repitiendo cíclicamente. Además, los niveles de inventario se van reduciendo de forma constante por las restricciones de continuidad de la demanda.

Desarrollo del algoritmo propio

Heurística

Como se ha comentado con anterioridad, la heurística que desarrolla el autor está basada en la premisa de la aproximación del Ciclo Común y pretende aportar una solución cuando la demanda del problema es conocida y aleatoria.

Observando los resultados del Gráfico 5 solución CC, se intuye que, a mayor número de artículos y menor cantidad de tiempo ocioso en máquina, mayor es la probabilidad de que se presente el fenómeno de interferencia, dificultando así la planificación de los productos.

Para el desarrollo del algoritmo, se ha creado una función que genera valores de demanda de forma aleatoria para cada uno de los artículos.

La dificultad de trabajar con datos sintéticos radica en que no siempre se generan problemas resolubles, es decir, en este caso en concreto cabe la posibilidad de que los valores de demanda sean superiores a la capacidad productiva de la instalación. En caso de que este fenómeno ocurra en periodos avanzados de la simulación, el algoritmo programará la producción de dichos artículos a periodos de tiempo anteriores, siempre y cuando existan holguras en el recurso productivo. Si en el primer periodo de tiempo permanece el problema por no disponer de suficiente capacidad productiva, el algoritmo requerirá disponer de un inventario inicial suficiente para cumplir con las exigencias de demanda del cliente.

En la tabla VI Simbología algoritmo propio se muestra la nomenclatura utilizada.

VI Simbología algoritmo propio

Símbolo	Definición
n	*Número de artículos a fabricar en la instalación.*
i	*Índice de artículo, i=1,…,n*
$T_{long.}$	*Longitud del periodo de simulación. (unidades de tiempo).*
j	*Índice de la unidad de tiempo, j=1,…,$T_{long.}$*
d_{max_i}	*Valor máximo que puede alcanzar el valor de demanda del producto i en una unidad de tiempo. (unidades)*
d_{min_i}	*Valor mínimo de demanda del producto i en una unidad de tiempo. (unidades)*
r_i	*Ratio de producción del artículo i (unidades por unidad de tiempo).*
k_i	*Costo de preparación de la máquina para iniciar la producción del artículo i (unidades de tiempo).*
h_i	*Costo de almacenamiento del artículo i (unidad por unidad de tiempo)*
c_i	*Tiempo de cambio de partida del artículo i (unidades de tiempo).*
L_i	*Tiempo de ocupación de máquina por la producción del artículo i (unidades de tiempo).*
T^{cc}	*Tiempo de ciclo (modelo de Ciclo Común)*
m	*Índice de ciclos T^{cc}*
$dT^{cc}{}_{im}$	*Sumatorio de demanda del producto i en el periodo $T^{cc}{}_m$*
CT	*Costo total (unidades monetarias)*

Posteriormente, se muestra un listado con los pasos que se han seguido para proceder al desarrollo de la heurística.

Ilustración 5 Secuencia heurística

Introducción de datos de entrada (n, $T_{long.}$, d_{max_i}, d_{min_i}, r_i, k_i)

Generar matriz de valores aleatorios de d_i

Cálculo $\bar{d}_i$

Calcular T^{cc}, q_i^*, L_i tomando como referencia $\bar{d}_i$

Obtención de la matriz de demanda agrupada

$$D = \left(\sum_{j=0}^{T_1^{cc}} d_{ij}, \dots, \sum_{j=T_{m-1}^{cc}}^{T_m^{cc}} d_{ij}\right) \forall i$$

Se toma D como referencia inicial para la planificación de la producción.

Calcular tiempos de producción necesarios en cada T^{cc} según D

Comprobación de posibles holguras y saturaciones de máquina en cada T^{cc}

Mientras existan periodos de saturación:

-Observar durante los periodos de saturación T_m^{cc} que artículo i tiene una demanda superior a q_i^*

- Programar fabricación de unidades del producto i que excedan de q_i^* a periodos de fabricación anteriores.

- En caso de no disponer de capacidad productiva en el periodo inicial, asignar un inventario inicial de las unidades necesarias para satisfacer demanda.

Definición de la secuencia de fabricación.

Ejecutar la función del Algoritmo Genético para mejorar el costo en cambios de partida en cada tiempo de ciclo.

Los pasos anteriormente listados, se definen a continuación de forma más detalla y con la ayuda de un ejemplo.

El primer paso para iniciar la ejecución del algoritmo es proceder a la introducción de los datos de entrada como pueden ser:

```
long_secuencia=240,
min_demanda=[200,100,367],
max_demanda=[801,601,668],
vel_prod=[2000,1000,3300], #cantidad de ud. producidas al dia
costo_almacen=[0.1,0.064,0.002],#emplear 3 decimales
costo_preparacion=[15,30,11],
tim_preparacion=[1,2,1]
```

Ya introducidos los datos de entrada, el siguiente paso es poder definir la demanda de cada producto i en cada periodo de tiempo j. Para ello se emplean funciones de aleatoriedad que definirán para cada periodo de tiempo j un valor de demanda comprendido entre los valores de

$d_{max_i} d_{min_i}$. A continuación, se muestran los primeros 15 registros de demanda aleatoria de cada uno de los 3 artículos que se presentan en la simulación.

[[656 584 356 399 285 421 217 223 616 518 343 396 239 219 357]
[175 392 103 115 454 448 255 437 232 455 139 191 293 387 222]
[566 541 403 650 651 428 616 388 538 400 616 477 432 396 570]]

Como se ha visto con anterioridad, para aplicar correctamente el modelo del ELSP basado en la solución de Ciclo Común, previamente se contempla que la demanda es continua y determinista. Antes de proceder a la aplicación de esta metodología es necesario conocer un valor de demanda estable durante cada periodo de tiempo, de modo que la solución más sencilla es poder calcular el promedio de la demanda de cada uno de los artículos $\overline{d_i}$ durante el horizonte de la simulación T_{long}. Llegados a este punto, ahora sí, se procede con el cálculo de las variables T^{cc}, q_i^*, L_i según la estrategia del CC. En el ejemplo mostrado se obtienen los siguientes valores:

Tcc 22
Qcc: [1848 965 3762]
L: [6, 8, 3]
Tiempo de uso de máquina = L_i+c_i :[7, 10, 4]

La variable c_i muestra los periodos de tiempo que permanece la instalación ocupada para fabricar q_i^* en cada T^{cc} que, tomando en cuenta los valores del ejemplo T^{cc} presenta una duración de 22 unidades de tiempo.

En este momento, es necesario comprobar que en cada uno de los periodos T^{cc} ,no se presente el fenómeno de interferencia en el que dos o mas artículos compiten por la disponibilidad del mismo recurso productivo. Para realizar esta comprobación se procede a crear una nueva matriz D (18) de dimensiones $n \times \frac{T_{Long}}{T^{cc}}$ en las que cada ciclo T^{cc} se realiza el sumatorio de la demanda de cada uno de los artículos. Esta matriz se tomará como referencia para planificar las cantidades de producción en cada uno de los periodos T^{cc}.

$$D = \left(\sum_{j=0}^{T_1^{cc}} d_{ij} , \dots , \sum_{j=T_{m-1}^{cc}}^{T_m^{cc}} d_{ij} \right); \forall i \qquad (18)$$

Los resultados de este apartado en el ejemplo analizado son:

Esta es la matriz D (en cada ciclo se realiza un sumatorio de la demanda)
[[8552 11866 11432 11206 11552 11833 10996 11153 10908 11327 10074]
[6223 7962 7262 7990 8105 9543 7881 7344 7424 7924 6682]
[11465 11584 10681 10915 12118 10889 10753 11297 11326 11746 10261]]

Conocida la demanda de cada uno de los artículos en cada uno de los periodos T^{cc} el próximo paso es comprobar la factibilidad de que no se manifieste el fenómeno de interferencia. Para ello se calcularan las unidades de tiempo de ocupación de máquina de cada uno de los productos en cada T^{cc}. La nueva matriz de tiempos de uso de máquina vendrá dada por $T_{uso} = \frac{D_{mi}}{q_i^*} c_i$.

Destacar que, para simplificar la programación del modelo (que está basado en matrices) se necesitan conseguir valores enteros de tiempo de uso de máquina. Es por ello que las cantidades de tiempo de uso se redondean al entero más próximo. Los resultados de la simulación en esta fase son:

Esta es T_{uso}
[[6 7 7 7 7 7 7 7 7 6]
[8 10 10 10 10 12 10 10 10 10 9]
[4 4 4 4 4 4 4 4 4 4 4]]

A continuación se debe comprobar que no existe saturación en el recurso productivo, de modo que se realiza el sumatorio de las unidades de tiempo en cada uno de los *m* ciclos, y se comparan los resultados con $Li + c_i$,quedando como resultado:

esta es suma_tiempos:
[18 21 21 21 21 23 21 21 21 21 19]

holguras: #un valor negativo indica que existen holguras, mientras que un valor positivo indica que se excede el tiempo de fabricación con respecto al teórico
[-4 -1 -1 -1 -1 1 -1 -1 -1 -1 -3]

Como se ha calculado con anterioridad, el tiempo de ciclo máximo T^{cc} es de 22 unidades de tiempo, y existe un registro que supera esta restricción. En esta situación en la que se está presentando el fenómeno de interferencia, se llega a la conclusión de que aparece un periodo de tiempo en el que la demanda es superior a la capacidad productiva de la planta, de modo que será necesario anticipar la fabricación de cierta cantidad de artículos para satisfacer las necesidades del cliente.

Para conocer qué artículo es el que sobrepasa en unidades temporales la ocupación del recurso productivo, se consulta en la matriz T_{uso}y se observa que el producto 2 es el causante de la saturación en máquina. Se calculan las unidades que provocan el rebose de L_i y en la matriz D se restan dichas unidades del periodo saturado y se introducen en el periodo anterior.

Los anteriores pasos de comprobación de saturación de máquina se van repitiendo cíclicamente hasta no detectar saturaciones en el recurso productivo, o en su defecto, hasta trasladar la saturación de la instalación al ciclo inicial, en el que el algoritmo definirá la necesidad de disponer de un inventario inicial para cumplir con el objetivo de entrega de producto.

En el ejemplo analizado, se requieren 5 repeticiones en la comprobación de saturación de máquina para por fin encontrar una solución factible al problema, resultando:

Esta es demanda_uso:
[[7 7 7 7 7 7 7 7 7 6]
[11 10 10 10 10 10 10 10 10 10 9]
[4 4 4 4 4 4 4 4 4 4 4]]
esta es suma_tiempos:
[22 21 21 21 21 21 21 21 21 21 19]
holguras:
[0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -3]

Esta es la matriz D con las cantidades rectificadas.
[[11001 11088 11088 11088 11088 11088 10996 11153 10908 11327 10074]

[8485 7720 7720 7720 7720 7720 7881 7344 7424 7924 6682]

[11763 11286 11142 11286 11286 10889 10753 11297 11326 11746 10261]]

Finalmente, solo quedaría mostrar la planificación de la producción en cada unidad temporal, de modo que se crea una nueva matriz en de dimensiones $m \times \frac{T_{Long}}{T^{cc}}$, en la que en las filas se

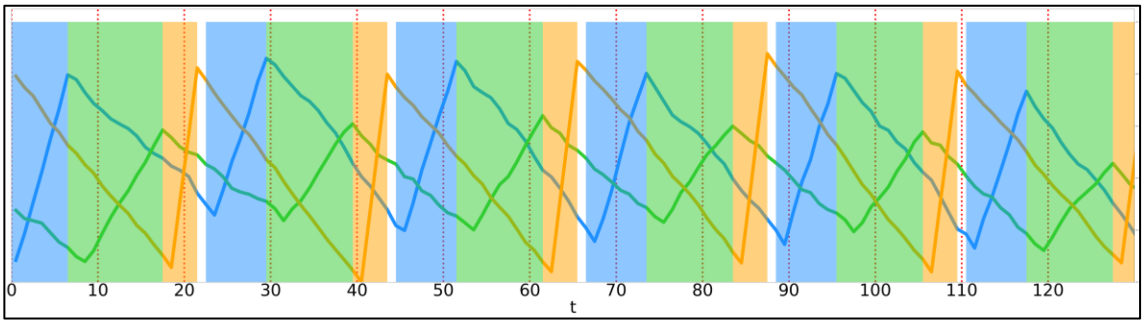
muestra que artículo se está fabricando en cada unidad temporal de cada ciclo y en las columnas se encuentran el número de ciclos que forman el periodo de simulación.

A continuación, se muestran los resultados de la matriz de planificación de la producción. En el Gráfico 6 Solución CC con demanda aleatoria que permite visualizar las variaciones de inventario y ocupación de máquina durante el periodo de simulación.

En la matriz de planificación, se muestra en cada unidad temporal el artículo destinado a ocupar la instalación en dicho instante. El artículo con numeración cero, hará referencia a periodos de no fabricación, es decir, unidades temporales de holguras en máquina.

```
Esta es la matriz de planificación:
[[1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3]
[0 0 0 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3]]
```

Gráfico 6 Solución CC con demanda aleatoria



Metaheurística (Algoritmo Genético)

Llegados a este punto del problema, se ha desarrollado un algoritmo genético que permite eliminar la siguiente restricción:

- los costos de setup de cada artículo permanecen contantes sin importar la secuencia del sistema.

Claro está que desarrollar un AG es absurdo para analizar solamente la secuenciación de tres artículos, pero bajo la incertidumbre de desconocer los parámetros de las posteriores simulaciones se ha decidido desarrollar una metaheurística que aporte una solución a este problema.

Inicialmente, para poder evaluar cada una de las secuencias, es necesario disponer de una matriz que defina el coste de los cambios de referencia. De igual manera que se está tratando con datos sintéticos durante el desarrollo del trabajo, este apartado no iba a ser menos, de modo que se crea una matriz de cambios de referencia con valores aleatorios. Hay que destacar que en esta matriz se toma la posición de las filas como articulo actual de fabricación y en posición de las

columnas, siguiente artículo a fabricar. EL costo de un cambio de un artículo A, a un artículo B, no necesariamente debe de coincidir el costo de un cambio de un artículo B, a un artículo A.

La matriz creada para este ejemplo resulta ser:

[[0 74 56]

[95 0 66]

[59 86 0]]

A partir de estos resultados, le mejor secuencia en cada uno de los ciclos es: [2 3 1]

De modo que la matriz de planificación resultante quedaría:

```
[[2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1]
 [0 0 0 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 1 1 1 1]]
```

El resultado del coste de inventarios con la solución aportada por el algoritmo es:

CT= 2937 unidades monetarias.

Test

Se muestran en este apartado los resultados que entrega el algoritmo bajo distintos datos de entrada.

Los resultados de los experimentos se muestran en dos tablas, separadas por el número de artículos analizados.

En cada una de las tablas, en la columna “Capacidad”, se marca con un “NO” aquellos experimentos que requieren de un inventario inicial para poder satisfacer la demanda prevista. En caso de que la demanda en periodos tardíos supere a la capacidad productiva, el algoritmo asigna la fabricación de dichos artículos a periodos anteriores en caso de que existan holguras. Si no existen suficientes holguras en periodos anteriores para acomodar la demanda de periodos posteriores, se marcará como “NO” la columna “Capacidad”.

EL Ratio de ocupación, se define como (19) y indica el porcentaje de ocupación de la máquina.

$$\rho = \sum_{i=1}^{n} \frac{D_i}{r_i} \qquad (19)$$

El tiempo de ejecución del algoritmo se dividido en dos partes. Por un lado, se evalúa la heurística, y por el otro se evalúa la metaheurística. Se ha procedido de esta manera debido a

que se deben definir los parámetros del algoritmo genético y dependiendo de las exigencias del usuario por encontrar una solución mejor, el tiempo de ejecución varía en gran medida.

La columna CT de cada una de las tablas, indica los costos totales de inventario y cambios de partida realizados durante la simulación del modelo.

Los datos empleados en cada uno de los experimentos se muestran en

VII Datos test 4 artículos

Datos test 4 artículos						
Producto (i)	r_i (uds/día)	Costo Setup	Tiempo de Setup(día)	H_i	D_{min_i} (Uds/día)	D_{max_i} (Uds/día)
1	2000	15	2	0.1	200	1000
2	2500	30	1	0.064	100	600
3	3300	11	2	0.002	367	867
4	2500	20	1	0.05	250	900

VIII Resultados test 4 artículos

Resultados 4 artículos					
Experimento	Falta de capacidad	ratio ocupación	tiempo de ejecución heurística	CT	tiempo AG
1	NO	15%	0.0060	53436	0.0050
2	NO	15%	0.0050	48123	0.0050
3	NO	15%	0.0060	53083	0.0050
4	NO	15%	0.0060	50242	0.0050
5	NO	15%	0.0060	46974	0.0050
6	NO	15%	0.0050	47708	0.0050
7	NO	15%	0.0060	46217	0.0050
8	NO	15%	0.0060	50249	0.0050
9	NO	15%	0.0050	49797	0.0050
10	NO	15%	0.0050	46059	0.0050
11	SI	23%	0.0140	186204	0.0050
12	SI	23%	0.0140	199271	0.0050
13	SI	24%	0.0140	182081	0.0050
14	SI	24%	0.0140	200938	0.0050
15	SI	23%	0.0130	194094	0.0050
16	SI	23%	0.0140	170858	0.0050
17	SI	23%	0.0140	173537	0.0050
18	SI	23%	0.0140	190929	0.0050
19	SI	24%	0.0150	195913	0.0050
20	SI	23%	0.0140	195212	0.0050

IX Datos test 10 artículos

Datos test 10 artículos						
Producto (i)	r_i (uds/día)	Costo Setup	Tiempo de Setup(día)	H_i	D_{min_i} (Uds/día)	D_{max_i} (Uds/día)
1	3000	15	1	0.0006	50	100
2	8000	20	1	0.0177	50	100
3	9500	30	2	0.0127	150	200
4	7500	10	1	0.01	400	400
5	2000	110	4	0.2785	15	20
6	6000	50	2	0.0267	10	20
7	2400	310	8	0.15	4	5
8	1300	130	4	0.59	30	85
9	2000	200	6	0.09	30	85
10	1500	5	1	0.004	25	100

X Resultados test 10 artículos

10 artículos					
Experimento	Falta de capacidad	ratio ocupación	tiempo de ejecución heurística	CT	tiempo AG
1	SI	11%	0.0140	4761	0.0050
2	SI	10%	0.0120	4260	0.0050
3	SI	10%	0.0140	4375	0.0050
4	SI	11%	0.0130	5727	0.0050
5	SI	11%	0.0140	4455	0.0050
6	SI	10%	0.0120	4718	0.0050
7	SI	10%	0.0120	4413	0.0050
8	SI	10%	0.0120	4585	0.0050
9	SI	10%	0.0120	4417	0.0040
10	SI	11%	0.0130	5017	0.0040
11	NO	4%	0.0100	2887	0.0050
12	NO	4%	0.0100	3039	0.0050
13	NO	4%	0.0110	2961	0.0050
14	NO	4%	0.0120	2887	0.0050
15	NO	4%	0.0110	2684	0.0050
16	NO	4%	0.0110	3135	0.0050
17	NO	4%	0.0110	2888	0.0040
18	NO	4%	0.0110	2893	0.0040
19	NO	4%	0.0120	3045	0.0040
20	NO	4%	0.0100	2803	0.0040

A la vista de los resultados mostrados en las tablas anteriores, se puede observar que el tiempo de ejecución es mayor cuando el algoritmo debe de modificar la programación de la

producción con respecto a la demanda, es decir, cuando debe de programar fabricación de artículos en periodos anteriores.

Con los resultados obtenidos de forma experimental, se corrobora que el análisis de un mayor número de artículos dificulta la planificación. En los distintos experimentos que se realizan con 4 artículos y disponen de suficiente capacidad productiva para satisfacer la demanda, la ocupación de la instalación ronda el 15%, mientras que en las pruebas realizadas con 10 artículos aquellos experimentos que encuentran dificultades para cumplir con la demanda establecida ocupan entre el 10% y 11% de la capacidad productiva de la máquina.

Conclusiones

Como se ha comentado durante la redacción del trabajo, el algoritmo diseñado se basa en el modelo más básico que aparece en la literatura, la solución de Ciclo Común. Esta solución proporciona una cota superior del ELSP, es decir, el modelo es capaz de encontrar una solución factible, pero en un gran número de problemas, esta está lejos de ser optima.

No obstante, el objetivo del trabajo no era crear un algoritmo infalible, sino familiarizarse con un lenguaje de programación, y sobre todo inicializarse en el desarrollo de herramientas capaces de aportar una solución a un problema. Para aumentar la visión de las estrategias que emplean los investigadores en problemas de secuenciación, se decidió emplear técnicas metaheurísticas en la última etapa del algoritmo.

El desarrollo del modelo ha llevado al autor a conocer de primera mano las posibles problemáticas o dificultades se presentan durante la elaboración del algoritmo. Una de las problemáticas mas incomodas encontradas, ha sido que al trabajar con matrices y asignar cada posición de la matriz a una unidad temporal, obliga al software a que las variables temporales sean de tipo entero. En problemas modelo con datos seleccionados se puede evitar esta problemática, sin embargo, en datos reales se deben redondear los valores temporales al entero más próximo. Esta problemática apareció cuando se decidió eliminar la restricción de demanda constante. Una propuesta para solucionar este problema sería asignar varias posiciones de la matriz a una única unidad temporal. Por ejemplo, si se desea analizar un problema en el que los datos vienen en días y se trabaja en la instalación durante 8h diarias, se reservarían 8 posiciones de la matriz para una unidad temporal que en este caso es un día.

La producción de un modelo propio relacionado con la planificación de la producción y la gestión de inventarios aporta una visión global entre departamentos en plantas de fabricación, en las que la mayoría de las personas, solamente se centran en incrementar rendimientos o beneficios en el departamento en la que están destinados sin valorar los efectos de sus acciones en otras secciones.

Referencias

Pilar I. Vidal-Carreras. Enfoques para la Resolución del Problema *ELSP* Working Papers on Operations Management Vol 1, Nº 2 (31-43) ISSN: 1989-9068.

Pilar I. Vidal-Carreras José Pedro García Sabater. El Problema de la Programación del Lote Económico (ELSP): Una revisión de la literatura. X Congreso de Ingeniería de Organización. Valencia 7 y 8 de septiembre de 2006.

Robert C. Leachmand and André Gascon. A heuristic scheduling policy for multi-item, single-machine production systems with tme-varyng, stochastic demands. Management science vol.34, No 3, March 1988.

Pilar I. Vidal-Carreras, Jose P. Garcia-Sabater, Julien Maheut, Julio J.Garcia-Sabater. Heurísticas Dinámicas para el DCC-ELSP. 5th International Conference on Industrial Engineering and Industrial Management. XV Congreso de Ingeniería de Organización. Cartagena, 7 a 9 de Septiembre de 2011

Jose Omar Hernandez, Salvador Hernandez e Idalia Flores. Simulated annealing algorithm to solve the economic lot scheduling problem and the basic cycle approach. Ingeniare. Revista chilena de ingenieria, vol. 19 No 3, 2011, pp. 473-485

Heechul Bae, Ilkyeong Moon andWonyoung Yun. Economic lot and supply scheduling problem: a time-varying lot sizes approach. International Journal of Production Research, 2014 Vol. 52, No. 8, 2422–2435,

Raúl Cortés-Fibla, Pilar I. Vidal-Carreras, José P. García-Sabater. Influencia del número de artículos y nivel de utilización en el rendimiento de estrategias de resolución para el problema de secuenciación del lote económico. www.revistadyo.com Dirección y Organización 56 (2015) 10-17

Raúl Cortés-Fibla, Pilar I. Vidal-Carreras y Jose P. García-Sabater. Simulación del comportamiento de estrategias de resolución para el ELSP en diferentes escenarios de complejidad en función del número de artículos y el nivel de utilización. Working Papers on Operations Management. Vol. 5, Nº2 (15-23) ISSN: 1989-9068

Bret J. Wagner , Darwin J. Davis. A search heuristic for the sequence-dependent economic lot scheduling problem. European Journal of Operational Research 141 (2002) 133–146

Horacio Ocampo Azocar, Eduardo Salazar Hornig. DIMENSIONAMIENTO DE LOTES Y PROGRAMACIÓN DE UNA MÁQUINA PARA ÚLTIPLES PRODUCTOS CON SETUP Y ESCASEZ. revista Ingeniería Industrial-Año. 13 Nº2: 49-62, 2014

T.C. Edwin Cheng, Valery S. Gordon, Mikhail Y. Kovalyov. Single machine scheduling with batch deliveries. European Journal of Operational Research 94 (1996) 277-283