



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INFORMÁTICOS

UNIVERSIDAD POLITÉCNICA DE MADRID

**Búsqueda tabú para la
optimización de un proceso
industrial de inyección de piezas
de aluminio**

TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN INTELIGENCIA ARTIFICIAL

AUTOR: Javier Montero Isusi
TUTOR: Antonio Jiménez Martín

2018

Agradecimientos

Expreso el más sincero de mis agradecimientos a:

Antonio Jiménez Martín, Subdirector del Departamento de Inteligencia Artificial y Coordinador del Máster en Inteligencia Artificial de la Universidad Politécnica de Madrid.

A mi mujer, mis padres y mi hermano por haberme brindado su apoyo y aliento durante el duro camino que he recorrido durante la realización del Máster y su comprensión mientras también he tratado de compatibilizarlo con mi empleo.

Resumen

En este caso, el objetivo es la optimización de los costes de fabricación, teniendo en cuenta los factores tradicionales de programación como pueden ser la demanda de piezas a modelar, la fecha de entrega de las mismas, los tiempos de desarrollo de éstas, el stock máximo que puede desarrollarse y el consumo de electricidad y gas en la producción de dichas piezas.

Un proceso tan costoso, puede aprovecharse de la aplicación de una metaheurística para obtener una solución cercana a lo óptimo, dentro de un tiempo lógico. Se trata por tanto de buscar una reducción del número de piezas demandadas y que no han sido producidas, así como de reducir al máximo el consumo de electricidad y gas, lo que significaría un aumento de los beneficios, atendiendo todas las solicitudes recibidas, y una minimización de los costes, por medio de la disminución del consumo de energía.

Con el fin de cumplir con nuestro objetivo, se ha propuesto un proceso de trabajo en dos fases. En la primera de ellas, se obtendrá una solución inicial factible basada en los datos de entrada, es decir, se buscará el recurso más óptimo o más cercano para mejorar el beneficio de la empresa y la reducción de los costes teniendo siempre en cuenta la viabilidad de la solución. En la segunda parte, se aplicará una *búsqueda tabú* centrándonos inicialmente en la exploración, con el fin de examinar todo el espacio de soluciones para posteriormente intensificar la búsqueda y obtener la opción más cercana a lo óptimo, que minimice la función objetivo.

Finalmente se ha realizado un ajuste paramétrico con el fin de conocer para ajustar los pesos de cada objetivo, es decir, dentro de las opciones que cumplen con nuestros propósitos, como son la reducción de los costes y el aumento de los beneficios, se dan soluciones que generan todas las piezas demandadas en la producción de las piezas con un coste de producción bajo, lo que se traduce en un mayor beneficio para la empresa.

La implementación realizada en este trabajo, ha permitido obtener resultados válidos que posibilitan mejorar la solución inicial obtenida. Esto supone la satisfacción de las demandas y por tanto un aumento de los beneficios de las empresas, que les permiten competir dentro del sector frente a otras empresas.

Abstract

In this end-of-master project, the application of a metaheuristic will be studied, such as the taboo search, in order to perfect the programming of the production in an area of aluminum foundry. A demand that has been increased in recent years by large companies in order to improve their competitiveness compared to other companies, as well as their valuation.

In this case, the objective is the optimization of manufacturing costs, taking into account the traditional factors of programming such as the demand for parts to be modeled, the date of delivery of the same, the development times of these, the stock maximum that can be developed and the consumption of electricity and gas in the production of said pieces.

Such a costly process can take advantage of the application of a metaheuristic to obtain a solution close to the optimum, within a logical time. It is therefore a question of looking for a reduction in the number of pieces demanded and that have not been produced, as well as reducing the consumption of electricity and gas as much as possible, which would mean an increase in profits, meeting all the requests received, and a minimization of costs, through the reduction of energy consumption.

In order to fulfill our objective, a work process has been proposed in two phases. In the first one, an initial feasible solution based on the input data will be obtained, that is, the most optimal or closest resource will be searched to improve the company's profit and the reduction of costs always taking into account the viability of the solution. In the second part, a taboo search will be applied focusing initially on the exploration, in order to examine the entire space of solutions to subsequently intensify the search and obtain the option closest to the optimum, which minimizes the objective function.

Finally a parametric adjustment has been made in order to know to adjust the weights of each objective, that is, within the options that meet our purposes, such as reducing costs and increasing profits, solutions are given that generate all the parts demanded in the production of the pieces with a low production cost, which translates into a greater benefit for the company.

The implementation carried out in this work, has allowed to obtain valid results that make it possible to improve the initial solution obtained. This supposes the satisfaction of the demands and therefore an increase of the profits of the companies, that allow them to compete within the sector in front of other companies.

Índice

1.	Introducción	3
2.	Objetivos	7
3.	Descripción del problema	11
3.1.	Datos de entrada	16
4.	Problemas de optimización	19
4.1.	Metaheurísticas	22
4.2.	Búsqueda Tabú	25
5.	Método de solución	31
5.1.	Modelización de la solución	31
5.2.	Pasos de la solución	31
5.2.1.	Solución inicial	31
5.2.2.	Búsqueda Tabú	33
5.2.3.	Entorno de una solución	33
5.2.4.	Función fitness	36
5.2.5.	Representación de un vecino de la solución actual	37
5.2.6.	Análisis de la factibilidad	37
6.	Resultados y ajuste paramétrico	39
6.1.	Instancia	39
6.1.1.	Información de moldes	39
6.1.2.	Demandas diarias	43
6.1.3.	Información de máquinas	47
6.1.4.	Información de piezas	48
6.2.	Solución inicial	50
6.3.	Solución factible	55
6.4.	Solución infactible	56
6.5.	Ajuste de parámetros	58
6.6.	Mejor resultado	59
6.7.	Función fitness agregado	64
7.	Conclusiones y trabajo futuro	65

Índice de figuras

1.	Parque de máquinas del problema	11
2.	Consumo de gas	16
3.	Matriz de moldes y Matriz de piezas de la solución	17
4.	Matriz de mantenimiento y Matriz de demandas	17
5.	Matriz de asignación de moldes	18
6.	Clasificación de programación matemática	19
7.	Programación lineal entera	20
8.	Clasificación de metaheurísticas	24
9.	Solución inicial	31
10.	Eliminación completa de una inyección	35
11.	Ejemplo de inyección	35
12.	Función fitness del problema	36
13.	Matriz de solución: Solución inicial, día 1	50
14.	Matriz de solución: Solución inicial, día 2	51
15.	Matriz de solución: Solución inicial, día 3	51
16.	Matriz de solución: Solución inicial, día 4	51
17.	Matriz de solución: Solución inicial, día 5	52
18.	Matriz de solución: Solución inicial, día 6	52
19.	Matriz de solución: Solución inicial, día 7	52
20.	Matriz de solución: Solución inicial, día 8	53
21.	Matriz de solución: Solución inicial, día 9	53
22.	Matriz de solución: Solución inicial, día 10	53
23.	Matriz de asignación: Solución inicial, día 11	54
24.	Matriz de solución: Solución inicial, día 12	54
25.	Matriz de solución: Solución inicial, día 13	54
26.	Matriz de asignación: Solución inicial, día 14	55
27.	Solución con cambios de molde inválidos	56
28.	Moldes no liberado antes de realizar una nueva inyección	57
29.	Cambios de molde en un cambio de turno	57
30.	Inyección parada	57
31.	Matriz de solución: Mejor resultado, día 1	59
32.	Matriz de solución: Mejor resultado, día 2	60
33.	Matriz de solución: Mejor resultado, día 3	60
34.	Matriz de solución: Mejor resultado, día 4	60
35.	Matriz de solución: Mejor resultado, día 5	60
36.	Matriz de solución: Mejor resultado, día 6	61
37.	Matriz de solución: Mejor resultado, día 7	61
38.	Matriz de solución: Mejor resultado, día 8	61
39.	Matriz de solución: Mejor resultado, día 9	61
40.	Matriz de solución: Mejor resultado, día 10	62
41.	Matriz de solución: Mejor resultado, día 11	62
42.	Matriz de solución: Mejor resultado, día 12	62

43.	Matriz de solución: Mejor resultado, día 13	62
44.	Matriz de solución: Mejor resultado, día 14	63
45.	Función fitness agregada	64

Índice de cuadros

1.	Información de moldes	43
2.	Demandas diarias	46
3.	Horas de reserva que requiere cada máquina	47
4.	Matriz compatibilidad máquinas - moldes 1	47
5.	Matriz compatibilidad máquinas - moldes 2	47
6.	Matriz compatibilidad máquinas - moldes 3	48
7.	Matriz compatibilidad máquinas - moldes 4	48
8.	Matriz compatibilidad máquinas - moldes 5	48
9.	Stocks máximos de piezas	50
10.	Ejemplo de entorno de una solución	58

1. Introducción

La creación de una conveniente planificación así como de una buena administración de la producción y de los suministros resulta clave en la gestión de una organización, debido a que afecta a los demás procesos de la empresa (procesos de compra, producción, mercado, etc.).

Las instalaciones de fabricación son sistemas complejos, dinámicos y estocásticos. Desde el comienzo de la fabricación organizada, los empleados de forma vertical han tenido que desarrollar muchos métodos que, de alguna forma, pudieran controlar las actividades de producción.

Los cronogramas de producción son planes que establecen cuándo ciertas actividades controlables deberían tener lugar o cómo se van a aprovechar los recursos de los que dispone la empresa para su actividad. Los horarios de producción coordinan las actividades y los recursos para aumentar la productividad y minimizar los costes operativos. Un cronograma de producción puede identificar conflictos de recursos, controlar el lanzamiento de trabajos a la empresa, garantizar que las materias primas requeridas se ordenan a tiempo, determina si se pueden cumplir las promesas de entrega e identifica los periodos de tiempo disponibles para el mantenimiento preventivo.

En la actualidad, la creación de un cronograma que modele la planificación de los moldes en las máquinas, es una solución inefectiva e ineficiente si es elaborado por un humano[15], ya que éste no tiene una visión global del sistema de producción. La optimización de parámetros requiere de muchos casos experimentales de prueba y su cálculo puede llevar días, por lo que es necesario hacer uso de metaheurísticas que mejoren de forma sustancial la reducción de los tiempos y conseguir una solución cercana al óptimo en base a la función fitness.

Nuestro proceso se basa en la inyección de piezas de aluminio. Si se decidiese planificarlo de forma manual, tendríamos que tener en cuenta que para la generación de una pieza podemos utilizar varios moldes que generan una o más piezas. Esto nos puede desembocar en un problema del inventariado en el que debemos prevenirnos de superar el stock máximo, porque ello demandaría un mayor número de mano de obra y limitaría el espacio de almacén y la flexibilidad de la empresa para cambios sensibles que, en un futuro, podría sufrir la empresa en las demandas de piezas.

La mayoría de procesos de manufacturación industrial requieren una parametrización cuidadosa para conseguir alcanzar el número óptimo de operaciones en temas de costes, calidad y tiempo. La programación de una producción, también denominada *scheduling*, es fundamental para optimizar la producción de un bien, en este caso, o de un servicio en otras áreas. Es uno de los elementos centrales de la gestión adecuada de la industria (en este caso, de un área de fundición de aluminio), además

de la planificación y del control de producción. Por medio de la misma podemos optimizar los recursos que una empresa posee.

La complejidad con la que se elaboran productos de alta demanda reside en la existencia de un conjunto grande de posibles escenarios al tener que generar un grupo de piezas basándonos en una demanda que hay que cumplir, así como por las limitaciones en el número de máquinas. Además, la empresa tiene que poder ser flexible y permitir que una producción pueda ser modificada al tener otra lista de demandas como entrada.

Esto ha ocasionado un avance en las técnicas y nuevas prácticas de negocio utilizadas para la programación de la producción, que han incrementado la demanda por parte de las empresas de una solución informática que resuelva el problema de producción para su caso concreto, consiguiendo que tome una gran importancia el desarrollo de nuevas técnicas de programación y desarrollos informáticos que crean valor para la empresa y para la satisfacción del cliente, tras haber prevenido que haya incumplimientos en las entregas pactadas con éste, haciendo de paso que las empresas adquieran una ventaja competitiva respecto a su competencia directa.

Por otra parte, para estos escenarios, hay que contemplar los horarios de trabajo, la compatibilidad entre las máquinas y los moldes y que seamos capaces de satisfacer la demanda semanal.

La complejidad de este problema ha conseguido extenderlo y que hayan surgido numerosas alternativas para atajarlo. En el estado del arte actual existen muchas técnicas para los problemas de programación de producción[14]. También disponemos de soluciones realizadas por medio de modelos matemáticos que permiten resolver de forma óptima este problema[16, 28].

Para resolver este tipo de cuestiones se ha hecho uso de programación dinámica, dividiendo de forma sucesiva el espacio de búsqueda con el objetivo de resolver muchos subproblemas que sean mucho más sencillos. Esta solución inicialmente tiene como inconveniente que su coste computacional es exponencial, haciendo necesario adaptarlo a un método híbrido que permita incorporar una heurística, a cambio de sacrificar la solución óptima, de modo que podamos encontrar soluciones muy cercanas al óptimo en un tiempo computacional razonable.

En nuestro caso, haciendo uso de la *búsqueda tabú*, se buscará una solución inicial factible y a continuación nos moveremos por soluciones vecinas factibles, primero centrándonos en la exploración y finalmente con una etapa de intensificación. Dado que la *búsqueda tabú* es una metaheurística, tendremos un apartado en el que se investiga sobre ellas y su aplicabilidad para este tipo de problemas, y una explicación más profunda de la *búsqueda tabú*.

2. Objetivos

El objetivo principal de esta tesis de fin de máster (TFM) es el estudio de la metaheurística *búsqueda tabú*, presentada por Glover, F [11, 13], aplicándola sobre el objetivo del problema, que es la planificación de una programación de la producción de un área de fundición de aluminio. Tiene que ser capaz de optimizar los costes de producción, teniendo en cuenta, además de los factores tradicionales de programación (demanda, fecha de entrega, tiempos de ciclo, stock, etc.), el consumo y coste de la energía en un tiempo razonable, que nos permita explorar miles de posibles escenarios de planificación en la producción de los moldes.

En un primer momento buscaremos entender el problema, contemplando cada una de las restricciones que se nos presentan para este tipo de problemas. Seguidamente tendremos que comprender el algoritmo y modelizar la solución inicial, los movimientos de entorno, la modelización de las restricciones, la forma en la que se calcula el fitness y las condiciones que hacen que una solución sea deseable.

Posteriormente, se realizará una investigación de este tipo de problemas. Comenzando desde un punto de vista teórico, estudiando los problemas de optimización, pasando por un estudio de las metaheurísticas y su aplicabilidad en este tipo de problemas y finalizando con un estudio de la *búsqueda tabú* para adaptarla al problema a resolver: cómo funciona la metaheurística, la modelización de la función fitness, en función de los objetivos que tendremos.

A continuación, en una tercera fase, se define la forma en la que vamos a proceder a resolver el problema, implementando el algoritmo de generación de la solución inicial cercana al óptimo. Asimismo se explicará cómo es el entorno de una solución. Después, se demostrará la forma en la que se ha aplicado la *búsqueda tabú multiobjetivo* para generar un conjunto de soluciones a partir de los posibles movimientos que podamos realizar en el entorno de una solución y se buscará obtener la mejor de forma voraz durante la ejecución que nos dé el mejor resultado posible para la función fitness. Adicionalmente, tendremos una serie de objetivos sobre la implementación y las soluciones generadas.

1. Introducción al problema.

Para poder enfrentarnos al problema inicial, es necesario conocer en profundidad el mismo, por ello debemos captar los requisitos a los que nos enfrentamos y explicar las diferentes variables y restricciones del mismo. Asimismo, deberemos estudiar los datos de entrada del problema y la salida esperada, así como buscar las diferentes soluciones que a día de hoy se han aplicado en casos similares y que pueden servirnos de ejemplo.

2. Investigación sobre las soluciones a los problemas de optimización.

Una vez hemos realizado una investigación previa sobre las diferentes soluciones posibles a los problemas de optimización, es el momento de exponer las distintas técnicas de resolución para este tipo de procesos. Hablaremos de las metodologías empleadas en función de la problemática, así como sobre la necesidad de utilizar metaheurísticas en los problemas de optimización combinatoria y las diferentes propiedades de éstas. De esta manera elegiremos una metaheurística que nos permita resolver el problema en un tiempo razonable.

3. Investigación sobre la *búsqueda tabú*

En nuestro caso, la metaheurística seleccionada como método a aplicar ante este problema es la *búsqueda tabú*. Para emplearla de forma correcta, es necesario previamente una investigación sobre su funcionamiento que se basará principalmente en cuatro aspectos: conocer sus estructuras de memoria, el formato que debe tener la *lista tabú*, los criterios de aspiración y los criterios de selección.

4. Objetivos a la implementación.

- Realizar una implementación del algoritmo que genere una solución inicial válida que satisfaga las demandas o que esté muy cerca de esta.
- Realizar una implementación de la *búsqueda tabú* que genere soluciones aleatorias en la que se busque un hueco en la planificación en el que la máquina esté parada y podamos realizar un cambio de molde inyectando un nuevo molde para satisfacer la demanda de esa pieza. Las soluciones se seleccionarán y se mejorarán en base a la función objetivo tratando de minimizar la demanda pendiente de las piezas y los costes de suministros (luz y electricidad).
- Objetivos a la validación de las soluciones.

Las soluciones que generemos deben de ser válidas, que respeten el número de cambios de molde diarios máximos, que no superen el stock máximo semanal y que produzcan las piezas demandadas.

5. Ajuste paramétrico.

Una vez generadas soluciones válidas al problema y haber comprobado cómo funciona la implementación, realizaremos un ajuste paramétrico con el objetivo de optimizar la implementación desarrollada.

3. Descripción del problema

Se dispone de 6 máquinas inyectoras (alimentadas por 6 hornos eléctricos de mantenimiento que a su vez son alimentados por 4 hornos fusores de gas propano) que vierten el aluminio en unos moldes para los distintos tipos de piezas.

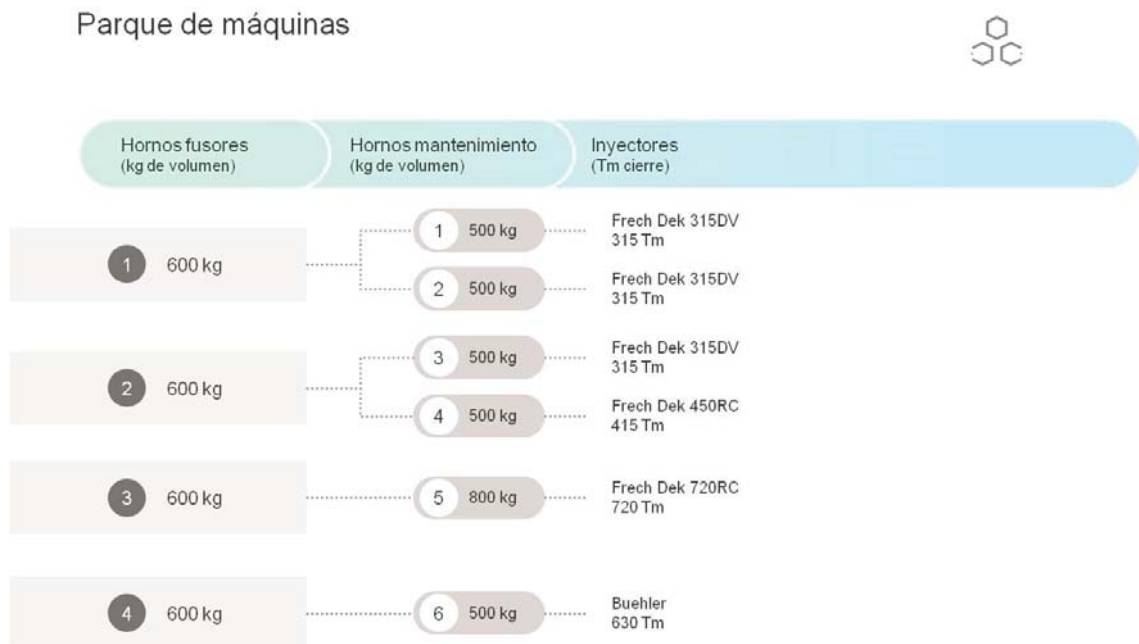


Fig. 1: Parque de máquinas del problema

En un molde podemos inyectar distintas piezas. En concreto, hay moldes que tienen una única pieza, la mayoría con dos piezas (50 % de cada tipo) y hay un molde con tres piezas con una unidad de cada una de ellas. Se proporcionará una tabla donde se especifiquen qué moldes hay y las piezas que incluyen. Podemos tener varios moldes iguales.

Se dispone de una **matriz de asignación** (binaria) que nos indica para cada máquina los moldes que puede utilizar para inyectar piezas y los que no. El tiempo de inyección de cada molde depende del molde concreto, pero no de la máquina en que se procesa. Los tiempos de inyección son del orden de segundos, pero en nuestro problema consideraremos producciones mínimas de una hora en las que se realizarán varias inyecciones del molde consecutivas. Se proporcionará un listado con el número de piezas que se inyectan por hora en cada uno de los moldes.

Al haber varios ejemplares de algunos moldes y moldes que producen varias piezas, se puede dar la circunstancia de que estemos inyectando en varias máquinas simultáneamente la misma pieza usando moldes distintos.

Los tiempos de **cambio de molde** están entre 1 y 2 horas, pero se redondean todos a 2 horas y no dependen de la máquina ni del molde en concreto. Asumimos que la primera hora del cambio es para quitar el molde y la segunda para poner el nuevo molde. Se pueden realizar un máximo de 3 cambios de molde al día entre todas las máquinas.

Si se decide lanzar dos inyecciones consecutivas del mismo molde en la misma máquina, entonces no hay que considerar ningún tiempo de cambio de molde.

Un cambio de molde entre dos días de 11 de la noche un día y la 1 de la madrugada del siguiente, se contabilizará al día siguiente.

Los cambios de molde siempre se hacen dentro de un turno de trabajo, no suelen empezarse dentro de un turno y acabar en otro. Por tanto, en la última hora de cada turno, no se inicia un cambio de molde. Los turnos son: de 06:00-14:00, de 14:00-22:00 y de 22:00-06:00. Por lo tanto, no están permitidos los cambios de 5:00 a 7:00, de 13:00 a 15:00 o de 21:00 a 23:00 horas.

Suponemos que los moldes que se estaban usando al final de la planificación en las distintas máquinas se quedan puestos para la siguiente planificación, cosa que hay que tener en cuenta (como entrada al modelo) para la siguiente planificación. Esta es la única información que se arrastra de una planificación a otra, junto con la demanda no satisfecha en el primer periodo de planificación, que se incorpora a la planificación de la segunda (es de suponer que a los primeros días para que se minimice el retraso en su producción).

Las máquinas trabajan 24 horas al día. Los fines de semana y festivos tenemos la opción de un turno extra de 24h cada día (en 2 turnos de 12h: de 6:00 a 18:00 y de 18:00 a 6:00 del día siguiente). Por tanto, este turno opcional es necesario cuando hay que cubrir las demandas y, cuando es de fin de semana, se trabajan los dos días (no tiene sentido trabajar solo uno). En caso de usar el fin de semana, debemos concentrar la producción en un solo día si es posible y en el menor número de máquinas también. No interesa una solución donde se trabaje unas horas un martes festivo y unas horas el fin de semana, se intentará que todas las horas estén en el festivo o en el fin de semana.

Las máquinas requieren un **mantenimiento**. Los mantenimientos en los hornos fusores duran 3 días, mientras que en los hornos de mantenimiento duran 1 día. Durante el mantenimiento de los hornos fusores, las máquinas asociadas a los mismos siguen trabajando, pero solo al 40% de su capacidad. En cambio, durante el mantenimiento de los hornos de mantenimiento no se puede producir (inyectar) en la máquina asociada. En el primer caso, con una matriz de mantenimiento (binaria) que nos indique qué días de mantenimiento tienen asignadas las distintas máquinas en las dos semanas consideradas es suficiente. En el segundo caso, hay que ver si la

franja horaria de mantenimiento es fija o varía y tratar la entrada de datos de la forma que corresponda. No se puede realizar el mantenimiento durante los días de no producción.

Cuando se empieza un mantenimiento, no se quita el molde, siempre y cuando sea necesario continuar con la producción. Al llegar un festivo o las horas de reserva hemos supuesto que el molde si se deja puesto.

Hay un **stock máximo** de almacenamiento de piezas al final de la semana que será distinto para las distintas piezas fabricadas. El stock inicial de la primera semana no se considera porque se utiliza para calcular las demandas en la misma. Por ello, el stock al final de la primera semana se obtiene del exceso de producción de la pieza con respecto a la demanda total de la misma esa semana restando las piezas fabricadas que son defectuosas (se conoce la proporción de piezas defectuosas para cada tipo de pieza), mientras que en la segunda semana se calcula como el stock inicial de la pieza correspondiente (es decir el stock al final de la primera semana de esa pieza) más el número de piezas que se han fabricado durante la semana, menos las piezas fabricadas que son defectuosas.

El periodo de planificación será de dos semanas, aunque los datos de demandas de piezas sean diarios y los stocks máximos sean semanales. Por simplificación las demandas de las piezas serán agrupadas de forma semanal.

Se pueden producir **averías de las máquinas**. Para afrontar este problema se propone reservar al final de la semana un porcentaje de las horas semanales disponibles de producción como reserva para hacer frente a dichas averías. Una vez resuelto el problema, si se produce una avería cuando se está produciendo una pieza, esa producción se realiza en las horas que habíamos reservado. Las horas de reserva se ponen al final del periodo de planificación. Si no se han producido averías se podrían utilizar para producir piezas adicionales. El porcentaje de horas de reserva no tiene por qué ser el mismo para todas las máquinas, aunque inicialmente todas serán de un 6%.

La cantidad disponible de kilogramos de aluminio en las máquinas no supone ninguna restricción porque hay una persona que se encarga de recargarlas y de que nunca estén vacías.

Todos los **tiempos** considerados en el modelo son valores enteros, por lo que el tiempo está discretizado en intervalos de una hora.

Se van a considerar simultáneamente dos **funciones objetivo**, una relacionada con el coste de producción y la otra con la violación de las demandas establecidas para los distintos tipos de piezas. El vector de pesos que representa su importancia relativa será una entrada al modelo.

Con respecto a la **satisfacción de la demanda**. Debemos tener en cuenta los retrasos que se produzcan en relación con las demandas diarias y, por otro lado, las demandas que no se llegan a satisfacer en el periodo planificado (dos semanas). Inicialmente, no se considerará ninguna prioridad entre las piezas demandadas y los retrasos. Por tanto, se minimizará la suma total de las demandas no satisfechas y la suma de los retrasos. Sin embargo, realizar un cambio en un siguiente paso para añadir prioridades no supondrá un gran esfuerzo debido a que sería una adición de pesos en la función objetivo.

Con respecto al **coste de producción**, asumimos que es el coste de consumo eléctrico en el proceso de producción y que dicho consumo solo se produce cuando las máquinas están procesando moldes, pero no cuando hay cambios de molde, cuando se están manteniendo o simplemente cuando están paradas (si es que lo están en algún momento). A esto hay que añadir el consumo de gas asociado a los hornos fusores.

La **tarificación eléctrica** se compone de un término de facturación de potencia y un término de facturación de energía. Cobran la potencia total del año prorrateada por mes (este coste no lo consideramos) le suman la energía activa consumida (este coste sí lo consideramos) y cobran un plus por picos de potencia superiores a 2450 KW (este coste tampoco lo consideramos).

Para el término de **facturación de energía** se dispone de una tabla donde se indica el precio por KW consumido en cada una de las 24 horas del día en los distintos meses del año y diferenciando si el día es de lunes a viernes o sábado-domingo-festivo. El consumo en KW/h de las 6 máquinas se calcula de la forma siguiente:

Máquina 1 (igual para las máquinas 2 y 3):

$$\begin{aligned} 2 + 4,94 * Q_{1,2,3} & \quad si \ Q_{1,2,3} \leq 12,5 \\ 60 + 0,1(50 - Q_{1,2,3}) & \quad si \leq 12,5 \ Q_{1,2,3} \leq 50 \end{aligned} \quad (1)$$

Máquina 4:

$$\begin{aligned} 2 + 5,74 * Q_4 & \quad si \ Q_4 \leq 12,5 \\ 70 + 0,1(50 - Q_4) & \quad si \leq 12,5 \ Q_4 \leq 50 \end{aligned} \quad (2)$$

Máquina 5:

$$\begin{aligned} 2 + 6,54 * Q_5 & \quad si \ Q_5 \leq 12,5 \\ 80 + 0,1(50 - Q_5) & \quad si \leq 12,5 \ Q_5 \leq 50 \end{aligned} \quad (3)$$

Máquina 6:

$$\begin{aligned} 2 + 9,34 * Q_6 & \quad \text{si } Q_6 \leq 12,5 \\ 115 + 0,1(50 - Q_6) & \quad \text{si } 12,5 Q_6 \leq 50 \end{aligned} \quad (4)$$

dónde Q_i es el número de kilogramos de aluminio inyectados por hora en el molde que se está usando en la máquina i .

Con respecto al consumo de gas en los hornos fusores (4 hornos), tenemos:

Horno Fusor 1:

$$\begin{aligned} 50 + 7,5 * (Q_1 + Q_2) & \quad \text{si } (Q_1 + Q_2) \leq 50 \\ 200 + 1,5 * (200 - (Q_1 + Q_2)) & \quad \text{si } 50 \leq (Q_1 + Q_2) \leq 200 \end{aligned} \quad (5)$$

Horno Fusor 2:

$$\begin{aligned} 50 + 7,5 * (Q_3 + Q_4) & \quad \text{si } (Q_3 + Q_4) \leq 50 \\ 200 + 1,5 * (200 - (Q_3 + Q_4)) & \quad \text{si } 50 \leq (Q_3 + Q_4) \leq 200 \end{aligned} \quad (6)$$

Horno Fusor 3:

$$\begin{aligned} 25 + 7,5 * Q_5 & \quad \text{si } (Q_5) \leq 25 \\ 100 + 1,5 * (100 - (Q_5)) & \quad \text{si } 25 \leq (Q_5) \leq 100 \end{aligned} \quad (7)$$

Horno Fusor 4:

$$\begin{aligned} 25 + 7,5 * Q_6 & \quad \text{si } (Q_6) \leq 25 \\ 100 + 1,5 * (100 - (Q_6)) & \quad \text{si } 25 \leq (Q_6) \leq 100 \end{aligned} \quad (8)$$

Como se refleja en la figura, hay en cada uno de ellos un coste fijo y otro coste que depende de si se están utilizando las máquinas que alimentan estos hornos fusores.

El coste del gas por KWh es de 0.13 euros para cualquier hora o día, independientemente de que sea festivo o no.

Este problema ha sido solucionado aplicando el recocido simulado, aunque la forma de trabajar las demandas ha sido distinta puesto que en el caso del otro proyecto se ha buscado optimizar las demandas a diario en vez de semanalmente, esto

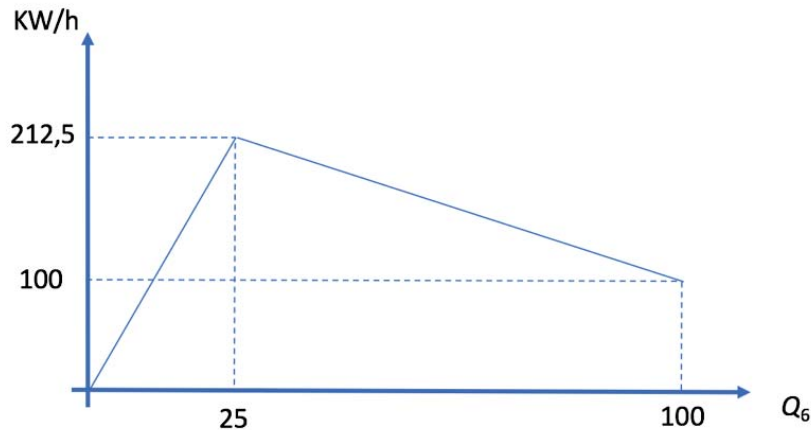


Fig. 2: Consumo de gas

hace que las demandas estén mucho más apretadas y que el margen de mejora que exige la solución sea mayor puesto que a nivel semanal las piezas que se tienen que generar un lunes pueden haberse producido un domingo y serían válidas.

3.1. Datos de entrada

Para poder empezar a trabajar en la obtención de una solución inicial tenemos que tener en cuenta los múltiples datos de entrada, así como las restricciones.

En este apartado se van a explicar los diferentes datos de entrada de la aplicación. En morado se indican los datos que son fijos, independientemente de la planificación que estemos resolviendo y en verde los datos de entrada para la planificación concreta que se vaya a resolver. Hay 153 piezas distintas y 85 moldes.

Para cada una de las piezas existirá un stock máximo al final de cada semana. Esto nos limitará en la producción de otras piezas que son generadas por ese mismo molde cuando las inyecciones de moldes generen piezas que ya están muy cercano a su stock máximo y nos impida producir la pieza que nosotros deseamos producir debido a que superaríamos el stock máximo de la otra pieza. Interpretamos que todas las piezas producidas por un molde son similares a pesar del porcentaje de piezas erróneas que serán descartadas.

Damos por hecho que las máquinas son todas similares, y lo único que podemos tener en cuenta, es que tengan distintas horas de reserva mostradas como un valor porcentual sobre el total de horas de las dos semanas.

En los moldes tenemos muchas características, su referencia, la cantidad de piezas que realiza ese molde, el porcentaje de piezas que realiza de media defectuosas,

el número de piezas que produce y el consumo en kilos de aluminio (un valor que buscamos minimizar).

Pieza	Referencia	Stock máximo final de semana				Máquinas	% horas de reserva		
1	-					Máquina 1	6		
2	-					Máquina 2	6		
3	-					Máquina 3	6		
4	-					Máquina 4	6		
...	-					Máquina 5	6		
153	-					Máquina 6	6		

Molde	Referencia	Nº Piezas distintas en el molde	Referencia pieza 1	Referencia pieza 2	Referencia pieza 3	% piezas defectuosas	Nº piezas inyectadas por hora	Kg de aluminio inyectados por hora
1	-	-	-	-	-	-		
2	-	-	-	-	-	-		
3	-	-	-	-	-	-		
4	-	-	-	-	-	-		
...	-	-	-	-	-	-		
85	-	-	-	-	-	-		

Fig. 3: Matriz de moldes y Matriz de piezas de la solución

En la matriz de mantenimiento se indicará qué días cada una de las máquinas pasará a una situación de estar parcial o totalmente parada. Es un proceso fundamental en la empresa para prevenir averías que a largo plazo podrían incrementar los costes de mantenimiento de la máquina, aumentar el tiempo de amortización, una reducción en la producción y una mayor insatisfacción del cliente.

La matriz de demandas diarias nos indica la demanda diaria que tienen las diferentes piezas en la implementación realizada. Estas demandas se han agrupado de forma semanal.

Matriz de mantenimiento	Semana 1							Semana 2							Molde cargado
	Date 1	Date 2	Date 3	Date 4	Date 5	Date 6	Date 7	Date 8	Date 9	Date 10	Date 11	Date 12	Date 13	Date 14	
Máquina 1															
Máquina 2															
Máquina 3															
Máquina 4															
Máquina 5															
Máquina 6															
Festivos															

Demandas	Semana 1							Semana 2						
	Date 1	Date 2	Date 3	Date 4	Date 5	Date 6	Date 7	Date 8	Date 9	Date 10	Date 11	Date 12	Date 13	Date 14
1														
2														
3														
4														
...														
p														

Fig. 4: Matriz de mantenimiento y Matriz de demandas

En la siguiente matriz se representa la relación entre máquinas y moldes, ya que no todos los moldes son compatibles con todas las máquinas:

Matriz de asignación	1	2	3	...	85
Máquina 1	-	-	-	-	-
Máquina 2	-	-	-	-	-
Máquina 3	-	-	-	-	-
Máquina 4	-	-	-	-	-
Máquina 5	-	-	-	-	-
Máquina 6	-	-	-	-	-

Fig. 5: Matriz de asignación de moldes

4. Problemas de optimización

La optimización matemática es una disciplina de gran importancia en muchos campos de la ciencia. Cuando hablamos de un proceso de optimización nos referimos al proceso que busca la mejor solución para un problema, en la mayoría de los casos con un tiempo limitado.

Se denomina solución óptima a aquella que maximice o minimice una función lineal, llamada función objetivo, y que mida la calidad de una solución teniendo en cuenta las restricciones que se le aplican, expresadas mediante un sistema de ecuaciones o inecuaciones.

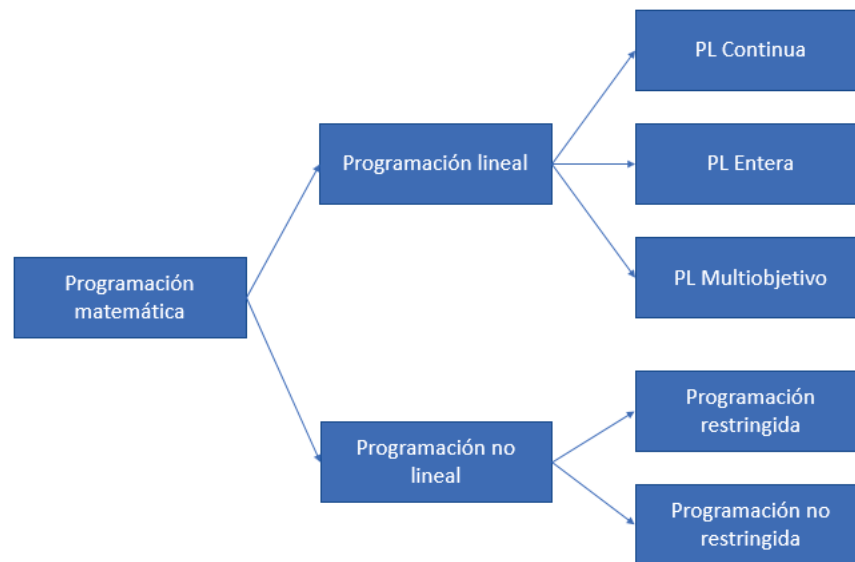


Fig. 6: Clasificación de programación matemática

Dentro de la programación matemática tenemos dos clasificaciones. La programación lineal es el campo de la optimización operativa cuyo objetivo es maximizar o minimizar una función objetivo que se encuentra sujeta a determinadas limitaciones. La programación no lineal tenemos que la función objetivo y/o sus restricciones son funciones no lineales de las variables de decisión.

El método más popular para resolver los problemas de programación lineal es el *método del simplex*.

Problemas de programación lineal

El *método simplex* o *algoritmo simplex*, creado por George Dantzig en 1947[7, 23], es un conjunto de métodos que nos permiten resolver problemas de programación lineal que maximizan o minimizan la función lineal sobre un conjunto de variables,

teniendo en cuenta el conjunto de inecuaciones lineales (que sean todas lineales es una condición expresa). Ha tenido tanto éxito a día de hoy se sigue empleando.

Su procedimiento nos permite movernos y mejorar la solución, o al menos no empeorarla, reconociendo la optimalidad que un punto extremo tiene sin necesidad de tener en cuenta todos los puntos extremos. También es capaz de detectar si la región factible está vacía o si la solución óptima no está acotada.

Problemas de programación entera

Son aquellos problemas, clasificados en función de si J es un subconjunto de $1, \dots, n$, todas las variables de decisión tienen la condición de tener un valor entero, este tipo de problema se denominará *problema de programación lineal entera pura*. Mientras que si J es un subconjunto propio de $1, \dots, n$, solo algunas de las variables de decisión deben tomar valores enteros, denominándose este problema como *problema de programación lineal entera mixta*. [30]

$$\begin{aligned}
 \text{máx o min } Z &= c_1x_1 + c_2x_2 + \dots + c_nx_n \\
 \text{s. a: } &a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq (\geq)(=)b_1 \\
 &a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq (\geq)(=)b_2 \\
 &\dots \\
 &a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq (\geq)(=)b_m \\
 &x_j \text{ enteras, } j \in J \subseteq \{1, \dots, n\}
 \end{aligned}$$

Fig. 7: Programación lineal entera

El método de resolución más popular para los problemas de programación lineal entera, es el algoritmo de *ramificación y acotación*, en el que se soluciona un problema de programación lineal entera, resolviendo una secuencia ordenada de problemas de programación lineales obtenidos a partir de una relajación en las restricciones de integralidad y de restricciones adicionales. El número de restricciones se incrementará a medida que el algoritmo avanza. Estas restricciones separan la región factible en subregiones complementarias.

La ramificación consiste en dividir el problema o subproblema en dos nuevos subproblemas mediante la imposición de restricciones excluyentes que eliminan la posibilidad de que haya soluciones fraccionarias para el problema entero. [22]

En la acotación, para una minimización:

Si la solución del problema actual satisface las condiciones de integralidad y el

valor óptimo de su función objetivo es menor que la cota superior actual, dicha cota se actualiza al valor óptimo de la función objetivo del problema resuelto, y el minimizador actual se almacena como el mejor candidato del problema original.

En el caso de que la solución obtenida no satisfaga las restricciones de integralidad y el valor de la correspondiente función objetivo esté entre las cotas inferior y superior, se actualiza el valor de la cota inferior al valor de la función objetivo del problema resuelto y se procede a bifurcar de nuevo.

Se podan aquellas ramas donde el problema considerado es infactible, la solución obtenida satisface las condiciones de integralidad y la solución del problema relajado es mayor que la cota superior disponible.

Su principal ventaja es que nos van a permitir modelar muchas más situaciones que la programación lineal, aunque tendrá como inconveniente que la resolución de los problemas será mucho más costosa, ya que su coste computacional es mucho más elevado[19], puesto que perdemos la referencia de que haya, al menos, una solución óptima del problema en un punto extremo.

Variables de decisión

Las variables de decisión suelen estar escritas de la siguiente forma:

$$x_j, \quad j = 1, 2, \dots, n. \quad (9)$$

El objetivo en la programación lineal es la maximización o la minimización de las funciones lineales de estas variables de decisión:

$$\zeta = c_1x_1 + c_2x_2 + \dots + c_nx_n. \quad (10)$$

Función de objetivo

Esta función ζ será la **función objetivo**. En la mayoría de los casos suele centrarse en su minimización, en la que buscamos reducir costes, reducir tiempos, reducir los retrasos en la demanda o por caso contrario aumentar los beneficios y aumentar la producción (maximizar sería ζ mientras que minimizar sería $-\zeta$)[8].

Conforme los problemas a resolver han sido más complejos, han sido necesarias nuevas metodologías que han surgido de forma natural en muchas áreas, para que se enfrenten a problemas de decisión en las que el éxito para alcanzar una buena

solución no estaba centrado en un único objetivo, sino que se tienen en cuenta de forma simultánea múltiples objetivos que suelen ser conflictivos en el sentido de que la mejora en uno o varios conlleva un deterioro en otros. Su resolución, de manera eficiente, suele constituir un desafío.

4.1. Metaheurísticas

Los problemas de optimización combinatoria son aquellos en los que se busca optimizar uno o varios objetivos sobre un gran número de soluciones que crecerá de forma exponencial dependiendo del tamaño del problema. Se caracterizan frecuentemente por sus tamaños a gran escala y la necesidad de obtener soluciones de alta calidad en tiempos de computación cortos.

Para resolver un problema de esta magnitud está establecido como una opción recomendable el uso de una metaheurística, que son estrategias generales que sirven para guiar a las heurísticas. Es una metodología que busca buenas soluciones, cercanas a la óptima, pero que cuentan con un coste computacional razonable, aunque sin ser capaces de garantizar la optimalidad de ésta, ya que no se centran en buscar la solución óptima sino una que esté muy próxima a la misma siempre garantizando que sea una solución factible. La generación de todo el conjunto eficiente para una posible elección posterior, resulta en general muy complejo y puede conllevar un esfuerzo computacional muy grande.

La aplicabilidad de las heurísticas se ha centrado en problemas específicos, de modo que un método que funcione para un problema no se puede utilizar para resolver otro problema distinto. La desventaja principal que presentan los métodos heurísticos es su dificultad o incapacidad para escapar de óptimos locales debido a su limitación en ese sentido. Para solventar este problema, se desarrollaron los algoritmos de búsqueda inteligentes, denominados metaheurísticas, que subsanan dentro de sus capacidades este problema. Se benefician de diferentes paradigmas de búsqueda aleatoria y del uso de la paralelización, aunque a menudo estas técnicas dan por hecho que las entradas de los problemas, la función objetivo y el conjunto de restricciones de optimización del problema son deterministas. Sin embargo, esto no representa de forma fidedigna la incertidumbre que hay en los entornos y por ello lo que se suele realizar es diluir, simplificar la versión del problema en la vida real en un modelo determinista.

Las metaheurísticas son potentes técnicas, generalmente aplicables a un gran número de problemas. Son una clase de métodos aproximados de optimización combinatoria que proporcionan un marco general para crear algoritmos híbridos combinando diferentes conceptos derivados de la Inteligencia Artificial, la evolución biológica, los mecanismos biológicos (colonia de hormigas) y los mecanismos estadísticos.

Por ello, las estrategias de solución para la optimización que se han desarrollado van dirigidas en su totalidad hacia la obtención directa de una solución eficiente y que además sea satisfactoria. Estos procedimientos conforman los denominados métodos aproximados o de aproximación, que como en el caso uniobjetivo, constituyen una alternativa razonable a los métodos exactos.

Para que una metaheurística funcione de forma eficaz y eficiente tiene que disponer de tres propiedades:

- Diversificación o Exploración: Se deben de facilitar los movimientos entre áreas separadas del espacio de búsqueda.
- Intensificación: Dentro de una área pequeña del espacio de búsqueda debemos de obtener la mejor solución.
- Comunicación: Desde cualquier solución poder alcanzar una solución final realizando los movimientos necesarios, pasando a soluciones vecinas idealmente, en pocos pasos.

La intensificación y la diversificación son propiedades opuestas en muchos casos pero ambas son necesarias para que el algoritmo sea bueno y no se quede en óptimos locales, priorizando inicialmente en la diversificación, y finalmente pueda obtener en la intensificación para obtener una buena solución.

Un tipo de algoritmo, de la familia de los algoritmos evolutivos (en los que se maneja un conjunto de soluciones en cada iteración) aplicables a este problema son los algoritmos genéticos. Se basan en los siguientes pasos:

1. Generar la población inicial.
2. Evaluar la población generada.
3. Seleccionar unos padres para el cruce, basándose en la información del paso anterior.
4. Emplear los operadores de cruce para obtener una descendencia.
5. Emplear un operador de mutación en los hijos.
6. Seleccionar padres e hijos que constituyan la nueva población.
7. En el caso de que se satisfaga la condición de parada el algoritmo se detiene, sino, se vuelve al Paso 2 en el que se volvería a reevaluar la población generada.

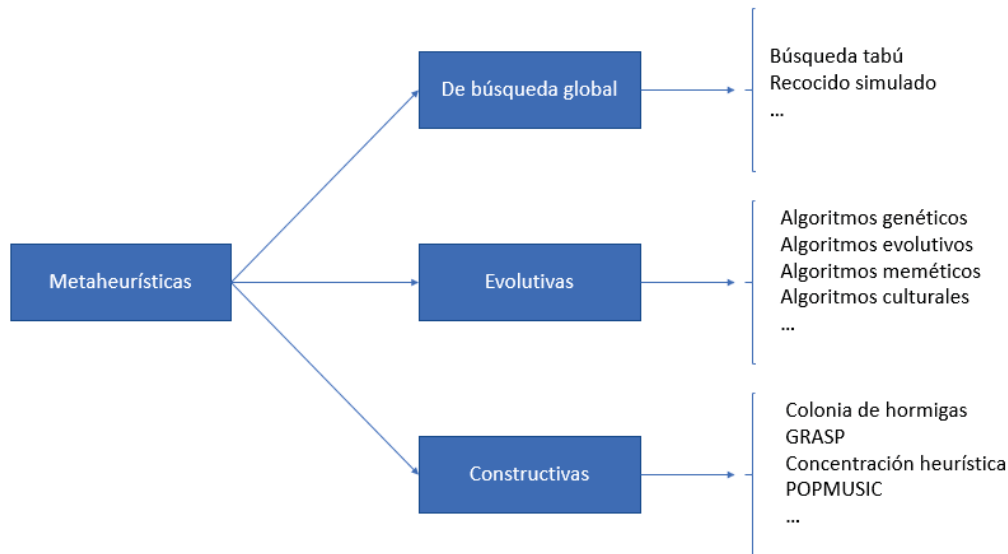


Fig. 8: Clasificación de metaheurísticas

Existen aplicaciones para este tipo de problema[9, 31]. Según esta metaheurística ha ido avanzando, se han aplicado distintas técnicas. en el estado del arte actual, se aplican principalmente distintas variaciones de un algoritmo genético híbrido[1, 5, 17, 21].

Los algoritmos evolutivos tienen su origen en los algoritmos genéticos. Emulan la forma de procreación y adaptación de las especies en el campo de la genética. Su aspecto más destacable es que los miembros de la población interactúan entre ellos frente a las heurísticas de búsqueda, centradas en soluciones individuales.

Se basan en el pensamiento evolutivo, el cual establece que toda vida que existe en el planeta puede ser explicada a través de cuatro procesos:

1. Reproducción.
2. Mutación.
3. Competencia.
4. Selección.

Su algoritmo se reduce a :

1. Generar (aleatoriamente) una población inicial.
2. Calcular aptitud de **cada** individuo.

3. Seleccionar (probabilísticamente) con base en aptitud.
4. Aplicar operadores genéticos (cruce y mutación) para generar la siguiente población.
5. Repetir hasta que las condiciones de parada se satisfagan.

Se han desarrollado aplicaciones recientes para una planificación en la producción de una factoría de acero empleando un algoritmo evolutivo diferencial [29]. En Zhao et al. [33] emplean una mejora sobre un algoritmo evolutivo denominado *shuffled complex evolution*. Este algoritmo mezcla una combinación de enfoques probabilísticos y deterministas, con una evolución competitiva que dispone de un mecanismo de mapeo secuencial para resolver este problema.

En la familia de metaheurísticas constructivas, aquellas que parten de una solución inicial vacía y van añadiéndole componentes hasta construir una solución, tenemos otra metaheurística disponible que son los algoritmos de optimización basados en colonias de hormigas, Colorni et al.[4]. Están inspirados por el comportamiento que siguen las hormigas en las rutas cuando buscan comida o explorar el terreno.

Estas se mueven aleatoriamente inicialmente y van dejando un rastro de feromonas. Otras hormigas que receptan esas feromonas siguen esta ruta, reforzando el camino, y si consiguen llegar a la comida, vuelven a casa dejando de nuevo un rastro encima del camino empleado.

Hay que tener en cuenta que las feromonas de las hormigas se evaporan ligeramente y el camino va perdiendo interés para la siguiente hormiga. El objetivo de esto consiste en encontrar el camino más corto, funciona muy bien en el *problema del viajante*.

En su aplicación se implementa cómo una colonia de hormigas resuelve este problema, simulando una colonia finita de hormigas artificiales que buscan soluciones de buena calidad. A día de hoy se sigue empleando. A. Rossi [27] hace uso de feromonas reforzadas entre las relaciones y Zhang et al.[32] hace uso de este algoritmo para resolver problemas de asignación de trabajos y el problema de programación de herramientas.

4.2. Búsqueda Tabú

La *búsqueda tabú* surge como un “intento de dotar de inteligencia a los algoritmos de búsqueda locales”[3]. Un proceso heurístico que mezcla una búsqueda local con una heurística para no caer en mínimos locales, haciendo posible movimientos hacia soluciones que no mejoran la actual. De esta forma se evita entrar en ciclos en el

proceso de búsqueda (mediante el uso de una memoria temporal que se denomina **lista tabú**) y nos permite continuar estratégicamente la búsqueda de soluciones aún mejores[10].

Es una de las metaheurísticas más empleadas para los problemas de optimización combinatorios. Anteriormente expusimos brevemente el *método del simplex*, que resulta muy útil para los problemas de programación lineal. Hay métodos que se inspiran en éste para hacer una heurística de búsqueda para el diseño de red capacitado[6].

La *lista tabú* es aquello que caracteriza a la *búsqueda tabú* como metaheurística, ya que almacena los últimos movimientos realizados y se emplea para recordar aquellos movimientos que nos hacen caer de nuevo en soluciones ya exploradas. Esto nos permitirá prevenir ciclos e inducirá a la exploración de nuevas regiones.

El procedimiento de cómo funciona la *búsqueda tabú* es el siguiente:

1. Paso 1 (Inicialización)
 - a) Selección de una solución inicial $x_{Actual} \in X$.
 - b) Iniciaremos la *lista tabú* en vacío.
 - c) Almacenamos la mejor solución actual haciendo que sea $x_{mejorCoste}$.
2. Paso 2 (Elección y finalización)
 - a) Calcular el vecindario de la solución inicial.
 - b) Seleccionar una nueva solución de acuerdo al criterio de selección. Escogiendo $x_{Siguiente} \in N(x_{Actual})$.
3. El algoritmo se detiene cuando la solución no puede ser mejorada o cuando se cumple el criterio de parada.

Las estructuras de memoria de la *búsqueda tabú* almacenan estas cuatro dimensiones fundamentales, incluyendo que un movimiento haya sido reciente.[2]

1. Tiempo desde que se ha procesado el movimiento (memoria a corto plazo).

Si solo empleamos este tipo de memoria podríamos permitir que una solución sea visitada más de una vez, aunque posiblemente el entorno se reduzca en cada exploración. Entremezclada con la memoria a largo plazo, reducimos en gran medida la probabilidad de explorar opciones que ya habíamos contemplado previamente.

La memoria a corto plazo puede ser de soluciones o de atributos. Cuando la memoria es de atributos, tiene como objetivo almacenar los atributos más comunes de un subconjunto de soluciones seleccionadas durante un cierto período de búsqueda, que con más probabilidad lleven hacia mejores zonas para explorar.

2. Frecuencia del movimiento (memoria a largo plazo)

A pesar de que muchas veces con una memoria a corto plazo somos capaces de generar una solución de alta calidad, el uso de una memoria a largo plazo refuerza la *búsqueda tabú*. En los entornos originales de las soluciones actuales puede haber soluciones que no sean las óptimas y por eso debamos escapar inicialmente del entorno de búsqueda inicial. Normalmente almacena soluciones muy buenas encontradas durante el proceso de búsqueda.

Su objetivo es la diversificación de la búsqueda sobre regiones poco exploradas y/o intensificar la búsqueda privilegiando los atributos que están presentes en las mejores soluciones.

3. Calidad del movimiento: La calidad se explica como la habilidad para encontrar lo buenos que son los movimientos empleados que generan nuevas soluciones a explorar en el proceso de búsqueda, con ello obtenemos una mejor identificación de los movimientos de *buena calidad*.
4. Influencia de un movimiento: Tiene en cuenta el impacto de las decisiones tomadas durante el proceso de búsqueda, tanto para la calidad de las soluciones como para las estructuras de las soluciones.

Una memoria que almacene tanto la frecuencia como que un movimiento sea reciente forma un híbrido y encontrar un equilibrio entre la exploración como de la intensificación.

La *lista tabú* es una lista (*linked list*) en el contexto computacional [24], donde se registran aquellas soluciones o atributos de soluciones que no deben ser elegidas. Una forma sencilla de construir una *lista tabú* consiste en que cada vez que se realiza un movimiento, se introduce su inverso (si se pasó de x_0 a x_1 , el inverso es x_1 a x_0) en una lista circular, de forma que los elementos de dicha lista están penalizados durante un cierto tiempo. Por tanto, si un movimiento está en la *lista tabú*, no será aceptado aunque aparentemente sea mejor solución que la solución actual.

La *lista tabú* puede contener múltiples valores, adaptándola en función del problema a tratar:

1. Soluciones visitadas recientemente.
2. Movimientos realizados recientemente.
3. Atributos o características que tenían las soluciones visitadas.

El tamaño de la *lista tabú* representará el número de iteraciones que un elemento (soluciones, movimientos o atributos) permanece en la misma. Si el tamaño máximo de la lista es de 10 elementos, tardará 10 iteraciones en ser una operación de nuevo válida. Éste será un parámetro que tendremos que ajustar ya que nos puede interesar algunos movimientos que intensifiquen la solución.

Otra propiedad fundamental de la *búsqueda tabú* es el **criterio de aspiración**, por el que cuando un movimiento tabú proporciona una solución mejor que cualquiera previamente encontrada, su clasificación tabú puede ser eliminada y reemplazaría a la solución hasta el momento mejor. Esta condición es lo que se denomina criterio de aspiración, con esto logramos un refuerzo positivo en la combinación de movimientos y características que nos han llevado a encontrar buenas soluciones.

Tendremos varios tipos de criterios de aspiración[12]:

1. Aspiración por Default

Si todos los movimientos disponibles están clasificados como tabú, y no son admisibles por otro criterio de aspiración, entonces el movimiento menos tabú (el menos reciente o el más ideal en otro criterio de aspiración) es seleccionado.

2. Aspiración por objetivo

- a) Forma global: Es el más empleado, permitimos a x_{prueba} en el caso de que su $c(x_{prueba}) < x_{mejorCoste}$ (en un problema de minimización).
- b) Forma local: Subdividiendo el espacio de búsqueda en regiones $R \in \mathfrak{R}$ identificados por los valores límite de las funciones. Teniendo $mejor_{coste}(\mathfrak{R})$ que denota el mínimo $c(x_{prueba})$ para x en \mathfrak{R} . Para $x_{prueba} \in \mathfrak{R}$.

3. Aspiración por Dirección de Búsqueda

Un atributo de aspiración para la solución x se satisface cuando la dirección de x proporciona una mejora y el actual movimiento es un movimiento de mejora. Entonces x se considera un movimiento de mejora si $c(x_{prueba}) < c(x_{actual})$ (en un problema de minimización).

El mecanismo de control del algoritmo de la *búsqueda tabú* está basado en la combinación de la *lista tabú* y del criterio de aspiración ya que uno contiene las condiciones que restringen y el otro libera el proceso de búsqueda en el espacio de soluciones.

El espacio de búsqueda está formado por todas las posibles soluciones y para estos problemas es muy amplio, por lo que no somos capaces de explorar todas las posibles soluciones si queremos tener un tiempo computacional admisible. Este espacio de búsqueda está formado tanto por las soluciones factibles como por las infactibles que en el caso de la *búsqueda tabú* a veces son interesantes que las recorramos para movernos a otras factibles que de otro modo no hubiéramos accedido en un primer momento.

Para este problema el espacio de búsqueda será el conjunto de soluciones factibles, es decir, todas las posibles inyecciones de moldes que respeten las restricciones.

El entorno o vecindario de una solución, es lo que se llama a las sencillas transformaciones locales sobre la solución actual, que definen un conjunto de soluciones vecinas o cercanas en el espacio de búsqueda.

Siendo x el conjunto de posibles soluciones $N(x)$ denota al entorno y c para la función objetivo.

La solución inicial depende del algoritmo que la genere, se puede usar una heurística para proporcionar una buena solución inicial o se puede comenzar de una solución obtenida aleatoriamente. Siempre que podamos es recomendable aplicar el conocimiento del que disponemos para generar una buena solución, pero, en el caso de que no se tuviera ninguna información podríamos partir de cualquier valor y mejorarlo en el proceso de solución. Al fin y al cabo, si somos capaces de generar una buena solución y ésta vira a una región con soluciones desfavorables, nos repercutiría en un gran coste computacional.

En los criterios de selección de la *búsqueda tabú* podemos hacer uso del criterio más simple que es el voraz, aquel que busca tomar la solución que tenga el mayor valor en la función objetivo, siempre que sea mejor que la actual. En el contexto de los óptimos locales R Marti dice: “El óptimo local alcanzado no puede mejorarse mediante el movimiento definido. Sin embargo, el método empleado no permite garantizar, de ningún modo, que sea el óptimo global del problema. Más aún, dada la “miopía” de la búsqueda local, es de esperar que, en problemas de cierta dificultad, en general no lo sea.[18].

Que el algoritmo se detenga porque no encuentra una mejor solución implica que hemos llegado a un óptimo, el peligro está en que sea local por no haber hecho suficiente exploración del espacio de soluciones.

5. Método de solución

En esta sección se explican las fases que seguimos hasta obtener una solución final: la modelización de la solución, cómo obtenemos la solución inicial, la adaptación de la *búsqueda tabú*: cómo ha sido aplicada la metaheurística al problema, la exploración del espacio de soluciones, la obtención de la función fitness, el análisis de factibilidad de las soluciones, la representación del entorno de una solución y la forma en la que se ha almacenado la memoria a corto y a largo plazo por separado.

5.1. Modelización de la solución

La solución inicial que generaremos será una matriz que almacenaremos en un fichero con formato csv que posteriormente podremos convertir a un fichero con formato xlsx para aumentar su legibilidad. Contendrá las dos semanas generadas con todas las horas de estas en las columnas y en las filas tendremos las máquinas.

Los datos contenidos serán los moldes empleados para las inyecciones, un valor que sea superior a cero representa al identificador del molde que se esté empleando en ese instante por esa máquina. Un valor negativo corresponde a situaciones planificadas: mantenimiento y horas de reserva. Un valor igual a cero significa que esa máquina no está produciendo piezas ya que no tiene un molde inyectado.

	Lunes																							Martes																									
Hora	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	0
Maquina 0	-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Maquina 1	-5	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	7	7	
Maquina 2	-5	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	
Maquina 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74

Fig. 9: Solución inicial

5.2. Pasos de la solución

En este apartado se describe la manera en la que se obtiene una solución inicial. A partir de ésta generaremos nuevas soluciones, primero intentando diversificar para no caer en un óptimo local y finalmente intensificaremos haciendo una búsqueda vertical, explorando el espacio de búsqueda de una buena solución para obtener la mejor del entorno disponible.

5.2.1. Solución inicial

En un primer momento tendremos en cuenta los datos de entrada. Analizando qué días de la semana se va a trabajar, omitiendo aquellos que sean festivo, qué

máquinas quedan reservadas en casos de averías y conocer en la planificación a qué horas las máquinas realizarán su mantenimiento para poder planificar la asignación de moldes adecuadamente.

Una máquina que esté en mantenimiento un día no producirá durante todo el día mientras que una máquina que esté en mantenimiento durante tres días producirá un 40 %.

Actualizaremos la solución, representada como una matriz de máquinas por horas, con las circunstancias especiales. Una vez que hayamos hecho esto, tendremos que ir rellenando máquina a máquina, qué molde será inyectado en función de la demanda de esa semana.

La demanda de piezas puede ser satisfecha por uno o varios moldes y se da el caso de la existencia de moldes que pueden producir más de una pieza. Para seleccionar uno de estos lo que haremos será ordenarlos en función de su tasa de producción y en caso de que hubiese empates, por su consumo de aluminio. Hay que tener en cuenta que los moldes no son universales y no están preparados para ser utilizados por cualquiera de las máquinas, por lo que se pueden dar casos en los que haya máquinas que no puedan producir un tipo de pieza porque no hay un molde compatible con ésta.

Otro de los problemas que nos podemos encontrar a la hora de seleccionar un molde para producir una pieza, es que su uso pueda suponer el stock máximo de almacenamiento para esa semana de las piezas incluidas en el molde tras realizar la inyección candidata, y que al desear producir una pieza X superemos el stock máximo de la pieza Y que también produce. Cuando un molde produce más de una pieza, la producción de una pieza por un molde con esta cualidad se entiende como *produccion/n*, siendo n el número de piezas.

Una vez que tengamos un molde válido para esa máquina y sabiendo que no se va a realizar el cambio de molde en las horas en que se realiza un cambio de turno, podremos realizar un cambio de molde. Como se ha indicado, solo podemos tener tres cambios de molde en un día por lo que tendríamos que retrasar la producción a las 0:00 del día siguiente para poder realizar el cambio de molde, descartando todas las horas intermedias.

En caso de que no se encontrase molde para la pieza más demandada en esa semana se exploraría la opción de tomar uno de los moldes de la siguiente pieza más demandada y así de forma sucesiva. Si en ningún caso encontramos un molde válido continuaremos con la siguiente hora. Una vez que finalicemos con las inyecciones de una máquina pasaremos a la siguiente.

El proceso de inyecciones de moldes para la segunda semana se realiza de forma

similar a la de la primera semana, su única diferencia son los datos de demanda y que tenemos que tener en cuenta la producción de la segunda semana. Por tanto debemos realizar un reajuste de las demandas para la segunda semana. Inicialmente conocemos las demandas para la segunda semana, pero esta sufre cambios por variables tales como piezas que puedan no haberse realizado durante la primera semana o piezas que se han producido en una cantidad superior a la cantidad demandada con lo que puede darse que la demanda de la segunda semana quedase parcial o totalmente cubierta con la producción realizada durante la primera semana.

Indicar que la solución inicial ya de por sí es una solución válida que salva todas las restricciones y que no incumple ninguna condición de trabajo (cambios de molde diarios y cambios de turno diarios).

5.2.2. Búsqueda Tabú

En este apartado se explica cómo hemos aplicado funcionalmente la *búsqueda tabú* a nuestro problema.

5.2.3. Entorno de una solución

En el entorno de una solución podemos encontrarnos con muchísimas nuevas soluciones alternativas:

- Eliminar cada inyección: La eliminación de una inyección aleatoria es un proceso que creará diversidad de posibles soluciones. Debido a que no tiene restricciones, podemos eliminar cualquier molde y seguiría siendo una solución válida, solo que menos eficiente al aumentar el número de demandas insatisfechas.
- Inyecta un molde: Este caso es más incierto y no podemos estimar a ciencia cierta cuántos moldes podemos inyectar en una solución inicialmente hasta haber tenido en cuenta todas las restricciones. Teniendo en cuenta la restricción de cambios de molde diarios, esta nos limita en gran cantidad los posibles espacios en la planificación dónde podemos realizar una inyección. Otras restricciones limitan severamente los posibles escenarios en los que podemos inyectar un molde. La compatibilidad de moldes y máquinas nos impide emplear un molde en una máquina al no estar soportado. La nueva inyección tampoco debe de superar el stock máximo de piezas semanal.

Proponemos hacer uso de movimientos híbridos, ya que un espacio de búsqueda en el que hagamos movimientos básicos sería muy caótico y desperdiciaríamos mu-

cho tiempo computacional y espacio de memoria.

Adaptación de la búsqueda para este problema

Los posibles movimientos básicos que podemos realizar en el espacio de búsqueda son:

1. Eliminación de una inyección.

Este paso es tan sencillo como fingir que una de las inyecciones que hay en la solución actual no ha sucedido. El procedimiento es el siguiente:

- Seleccionamos una máquina de forma aleatoria.
- Seleccionamos una hora aleatoria entre las dos semanas.
- Una vez que hemos seleccionado una hora válida (que no sea una hora que sabemos que es un cambio de turno), comprobamos que en la matriz de solución tenemos almacenado un valor positivo, que indique que en ese instante hay una inyección, si es $valor \leq 0$ seleccionamos otra hora.
- Si hemos hallado un molde, buscaremos el origen de la inyección recorriendo las fechas hacia la izquierda. Tendremos especial cuidado cuando nos encontremos las horas de mantenimiento para que el algoritmo de búsqueda del punto inicial no se pare en este punto y nos pararemos en los cambios de molde.
- Una vez hallado el punto de partida eliminaremos cualquier referencia a posteriori que encontremos, cuidándonos de las horas de mantenimiento que podamos encontrar. También es fundamental eliminar las horas empleadas al cambio de molde, tanto para su colocación como para su retirada.
- Restamos en uno el número de cambios de origen donde se produjo el cambio de molde.

2. Inserción de una inyección.

Esta inserción es muy similar al de la eliminación de una inyección, aunque debe de respetar todas las restricciones que se han tenido en cuenta hasta ahora en la generación inicial.

		13	-5	0	0	0	0	0	0	0	0	0	0	0	0	-5	2	2	2	
1	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	-5	-5	2	2	2
2	13	-5	-5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
3	13	13	13	13	-5	0	0	0	0	0	0	0	0	0	0	-5	2	2	2	
4	13	-5	-5	2	2	2	2	-5	0	0	-5	2	2	2	2	2	2	2	2	
5	13	-5	-5	4	4	4	4	4	4	4	4	4	4	4	-5	-5	2	2	2	
6	13	-5	-5	4	4	4	-5	0	0	0	0	0	0	0	-5	-5	2	2	2	

Fig. 10: Eliminación completa de una inyección

- Seleccionamos una máquina de forma aleatoria.
- Seleccionamos una hora aleatoria entre las dos semanas.
- Si el valor en la matriz es distinto de 0 implica que alguna acción se está realizando con esa máquina por lo que no debemos de insertar un molde.
- Una vez que tenemos una hora válida, realizaremos una inyección en ese punto, primero indicaremos un cambio de molde asignando el valor **-5** hasta que llegue a satisfacer la demanda. La hora siguiente se empleará en la liberación del molde en algunos de los posibles casos que nos podemos encontrar.

		13	-5	0	0	0	0	0	0	0	0	0	0	0	0	-5	2	2	2	
1	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	-5	-5	2	2	2
2	13	-5	-5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
3	13	13	13	13	-5	0	0	0	0	0	0	0	0	0	0	-5	2	2	2	
4	13	-5	-5	2	2	2	2	-5	0	0	-5	2	2	2	2	2	2	2	2	
5	13	-5	-5	4	4	4	4	4	4	4	4	4	4	4	-5	-5	2	2	2	
6	13	-5	-5	4	4	4	-5	0	0	0	0	0	0	0	-5	-5	2	2	2	

Fig. 11: Ejemplo de inyección

En las inyecciones se nos presentan 6 posibles escenarios:

1. Producir la inyección que tenemos antes del hueco. Tendríamos que eliminar el cambio de molde entre la inyección anterior y la actual. Será necesario añadir la retirada del molde al finalizar la inyección.
2. Producir la inyección que tenemos a posteriori. Tendríamos que eliminar el cambio de molde entre la nueva inyección y la que está a continuación. Será necesario añadir la retirada del molde al inicio de la inyección.
3. Producir la inyección anterior hasta satisfacer la demanda y que finalice antes de llegar a la siguiente inyección, dejando un hueco en el que la máquina está parada.

4. Producir la inyección posteriori hasta satisfacer la demanda y que finalice antes de llegar a la siguiente inyección, dejando un hueco en el que la máquina está parada.
5. Producir una nueva inyección hasta el final de la nueva inyección. Será necesario añadir el cambio de molde al inicio y al final de la inyección.
6. Producir una nueva inyección hasta satisfacer la demanda. Será necesario añadir un cambio de molde al inicio y en la hora posterior a la satisfacción de la demanda.

5.2.4. Función fitness

Tendremos en la función fitness la suma de los diferentes objetivos normalizados. La emplearemos para que nuestro algoritmo sea guiado a soluciones mejores.

- F_1 : Minimización de la demanda no satisfecha en las dos semanas de las piezas demandadas.
- F_2 : Minimización del consumo energético de la solución (consumo de electricidad y gas).

$$\min F = w_1 F_1 + w_2 F_2$$

Fig. 12: Función fitness del problema

La normalización de la función fitness se hará de la siguiente forma.

- F_1 : Minimización de la demanda no satisfecha en las dos semanas de las piezas demandadas.

Siendo $F_{1min} = 0$ $F_{1max} = \text{demanda total}$.

- F_2 : Minimización del consumo energético de la solución (consumo de electricidad y gas).

Siendo $F_{2min} = 0$ $F_{2max} = \text{produccion con mayor consumo de aluminio}$.

Función fitness normalizada.

$$NF_i = \frac{F_i - F_{imin}}{F_{imax} - F_{imin}}$$

5.2.5. Representación de un vecino de la solución actual

Durante la etapa de la búsqueda tabú, tendremos que hacer una exploración del entorno de una solución. Cada vecino contendrá los siguientes datos:

- Consumo eléctrico.
- Consumo de gas.
- Número de piezas demandadas no satisfechas.
- Movimiento realizado para llegar a esa solución vecina.

Será seleccionada una solución que maximice la función fitness aplicando. En el caso de que este movimiento minimice los costos y mejore el número de piezas no demandadas, es permitida la aspiración de ese movimiento.

5.2.6. Análisis de la factibilidad

En la implementación realizada no hemos contemplado escenarios que no pudieran llegar a darse porque esto implicaría que realizásemos cambios en la forma de trabajar de la empresa.

De las restricciones que tenemos en el problema, debido a la forma en la que realizamos los movimientos en el entorno, las podemos formular de la siguiente forma.

1. *En un molde podemos inyectar distintas piezas. En concreto, hay moldes que tienen una única pieza, la mayoría con dos piezas (50% de cada tipo) y hay un molde con tres piezas con una unidad de cada una de ellas. Se proporcionará una tabla donde se especifiquen qué moldes hay y las piezas que incluyen. Podemos tener varios moldes iguales..*

2. *Se dispone de una matriz de asignación (binaria) que nos indica para cada máquina los moldes que puede utilizar para inyectar piezas y los que no.* Hay condiciones que ya de por sí no podemos cambiarlas como las restricciones en las inyecciones en las que una máquina no es compatible con un molde, esta situación no

va a cambiar de ninguna forma a menos que cambien la maquinaria, algo impensable.

No podremos permitir que la demanda y la producción superen al stock máximo. El motivo es que es un valor fijo que la empresa tiene estudiado según la demanda de las piezas, y a falta de que la empresa cambiase de nave o adquiriese otro almacenaje, no podríamos generar este tipo de soluciones que harían que un exceso de producción saturase el almacén de piezas, con los costes que repercute de almacenamiento y una mala gestión de la materia prima en la que no priorizaríamos sobre la producción de otras piezas que tienen una demanda, independientemente de que sea mayor o menor.

3. Los cambios de molde siempre se hacen dentro de un turno de trabajo, no suelen empezarse dentro de un turno y acabar en otro. Por tanto, en la última hora de cada turno, no se inicia un cambio de molde. Los turnos son: de 06:00-14:00, de 14:00-22:00 y de 22:00-06:00. Por lo tanto, no están permitidos los cambios de 5:00 a 7:00, de 13:00 a 15:00 o de 21:00 a 23:00 horas.

Los horarios de los trabajadores es otra restricción, según estos, tenemos cambios de turno en los que no se pueden realizar cambios de molde, hay casos en las planificaciones de la producción generadas que se pierden horas de trabajo en las máquinas por este motivo, pero es algo que no podemos evitar porque en alguna hora los trabajadores deben de hacer el cambio de turno. A menos que la empresa modificase su política de turnos es irrealizable y por tanto no nos cabe esta posibilidad.

Una posibilidad que también podríamos valorar sería generar soluciones en las que el número de cambios de molde superase el número de cambios de molde permitidos por día, esta propuesta es muy interesante ya que nos daría una mayor flexibilidad a la hora de inyectar moldes durante la *búsqueda tabú* y nos permitiría desarrollar posiblemente mejores soluciones.

*4. Hay un **stock máximo** de almacenamiento de piezas al final de la semana que será distinto para las distintas piezas fabricadas*

No podemos superar la cantidad de piezas a alojar en el almacén sumando la demanda de esa semana por lo que no podemos superar este valor ya que si no limitaríamos la flexibilidad de la empresa a la hora de enfrentarse a nuevas demandas.

6. Resultados y ajuste paramétrico

En este apartado mostraremos los datos de entrada empleados y la salida que ha tenido la implementación para estos datos concretos. En un primer momento generamos una solución que no es capaz de suplir toda la demanda necesaria, esto se debe a la compatibilidad de las máquinas. Aquellas que pueden producir esas piezas, porque sus moldes son compatibles, están ocupadas con piezas más demandadas.

6.1. Instancia

La instancia se compone de varios ficheros de datos de entrada necesarios para poder desarrollar una solución.

6.1.1. Información de moldes

Los datos de moldes son datos estáticos que nos proveen información clave. Qué moldes producen qué piezas, en qué cantidad y que porcentaje de estas son defectuosas y descartadas.

ID	Referencia	Nº piezas	Referencia pieza 1	Referencia pieza 2	Referencia pieza 3	% piezas defectuosas	N piezas inyectadas por hora
1	850077	1	850077-1			3	947
2	850089	1	850089-1			6	758
3	856033	1	856033-1			11	170
4	856034	1	856034-1			25	170
5	111192632	1	111192632-1			7	167
6	213850123	1	213850123-1			3	379
7	111192767-1	1	111192767-1			6	360
8	113103065-1	1	113103065-1			3	211
9	113103171-1..2	1	113103171-1			3	228
10	212831667-68	2	212831667	212831668		6	1308
11	213850149-50	2	213850149-1	213850150-1		6	240
12	213850161-62	2	213850161-1	213850162-1		5	342

13	213850163-64	2	213850164-1	213850163-1		5	342
14	213850165-66	2	213850165-1	213850166-1		5	684
15	213850167-68	2	213850167-1	213850168-1		5	376
16	213850169-70	2	213850169-1	213850170-1		5	610
17	213850171-72	2	213850171-1	213850172-1		4	342
18	213850175-76	2	213850175-1	213850176-1		15	256
19	213850177-78	2	213850177-1	213850178-1		15	256
20	213850179-80	2	213850179-1	213850180-1		5	256
21	213850181-82	2	213850181-1	213850182-1		11	256
22	213850239-40	2	213850239-1	213850240-1		11	456
23	213850239-40..2	2	213850239-1	213850240-1		3	456
24	213850241-42	2	213850241-1	213850242-1		5	342
25	213850245-46	2	213850245-1	213850246-1		5	446
26	213850245-46..2	2	213850245-1	213850246-1		6	446
27	213850291-1	1	213850291-1			6	947
28	213856071-72	2	213856072-1	213856071-1		7	300
29	213856073-74	2	213856074-1	213856073-1		6	368
30	213856087-88	2	213856087-1	213856088-1		7	288
31	213856089-90	2	213856089-1	213856090-1		8	288
32	213856091-92	2	213856091-1	213856092-1		5	288
33	213856093-94	2	213856094-1	213856093-1		5	256

34	213856105-06	2	213856105	213856106		4	300
35	213856109-10	2	213856109-1	213856110-1		16	428
36	213856109-10..2	2	213856109-1	213856110-1		16	428
37	213856111-12	2	213856112-1	213856111-1		6	312
38	213856119-20	2	213856119-1	213856120-1		5	292
39	213856127-28	2	213856127-1	213856128-1		5	542
40	213856129-30	2	213856129-1	213856130-1		6	312
41	213856143-44	2	213856144-1	213856143-1		6	256
42	213856159-60	2	213856160-1	213856159-1		6	534
43	213856161-62	2	213856161-1	213856162-1		6	300
44	213856165-66	2	213856166-1	213856165-1		5	300
45	213856167-68	2	213856168-1	213856167-1		5	342
46	213856169-70	2	213856170-1	213856169-1		4	342
47	213856171-72	2	213856172-1	213856171-1		5	342
48	213856203-04	2	213856203-1	213856204-1		6	276
49	213856205-06	2	213856205-1	213856206-1		6	276
50	213856225-26	2	213856225	213856226		11	384
51	213856227-28	2	213856227-1	213856228-1		8	276
52	213856237-38	2	213856237	213856238		5	526
53	213856239-40	2	213856240	213856239		5	312
54	213856253-54	2	213856254-1	213856253-1		5	464

55	213856255-56	2	213856256-1	213856255-1		15	288
56	213856489-90	2	213856490	213856489		5	378
57	213856491-92	2	213856492	213856491		5	378
58	213856493-94	2	213856494-1	213856493-1		9	342
59	213856495-96	2	213856495-1	213856496-1		5	262
60	213856497-98	2	213856497-1	213856498-1		8	342
61	213856499-500	2	213856499-1	213856500-1		8	342
62	233650039-40	2	233650040-1	233650039-1		6	270
63	233656011-12-13	3	233656011-1	233656013	233656012-1	6	1002
64	233656051-52	2	233656051-1	233656052-1		5	240
65	233656053-54	2	233656053-1	233656054-1		5	260
66	253356001-02	2	253356001-1	253356002-1		3	378
67	253356003-04	2	253356004-1	253356003-1		3	378
68	262236055-6	2	262236055	262236056		3	378
69	262236257-8	2	262236058	262236057		7	378
70	850077..2	1	850077-1			7	947
71	856027-28	2	856027-1	856028-1		4	334
72	856029-30	2	856030-1	856029-1		4	334
73	856035-36	2	856035-1	856036-1		4	340
74	SP0000449-434	2	SP0000434	SP0000449		6	276
75	SP0000450-442	2	SP0000442	SP0000450		6	276
76	SP0000451-445	2	SP0000451	SP0000445		5	276
77	SP0000452-447	2	SP0000452	SP0000447		5	276

78	SP0000750-51	2	SP0000751	SP0000750		6	276
79	SP0000752-53	2	SP0000752	SP0000753		6	276
80	SP0000772-73	2	SP0000772-1	SP0000773-1		6	342
81	SP0000774-75	2	SP0000775-1	SP0000774-1		6	276
82	SP0001038-39	2	SP0001038-1	SP0001039-1		16	296
83	SP0001040-41	2	SP0001040-1	SP0001041-1		10	294
84	SP0001042-43	2	SP0001043-1	SP0001042-1		6	342
85	SP0001044-45	2	SP0001045-1	SP0001044-1		6	266

Tab. 1: Información de moldes

6.1.2. Demandas diarias

La planificación de la programación que vamos a realizar debe de satisfacer las demandas de las dos semanas expuestas en la siguiente tabla. Dentro de la implementación, almacenamos los valores de demanda por pieza en dos valores independientes, al igual que haremos con los datos de producción. El objetivo de esta práctica es la satisfacción cada semana de sus demandas, priorizando en cada semana su demanda. Las máquinas seleccionan la pieza más demandada para la que disponen de un molde compatible, que no esté siendo empleado por otra máquina.

Pieza	Demanda	Día
SP0000434	804.0	20/11/2017
SP0000449	804.0	20/11/2017
SP0001040-1	1235.0	20/11/2017
SP0001041-1	1235.0	20/11/2017
113103171-1	300.0	20/11/2017
213850239-1	587.0	20/11/2017
213850240-1	587.0	20/11/2017
212831667-68	555.0	20/11/2017
SP0001040-1	1235.0	21/11/2017
SP0001041-1	1235.0	21/11/2017

111192767-1	6048.0	21/11/2017
113103171-1	1089.0	21/11/2017
213850239-1	949.0	21/11/2017
213850240-1	949.0	21/11/2017
213856237	1114.0	21/11/2017
213856238	1114.0	21/11/2017
SP0001040-1	339.0	22/11/2017
SP0001041-1	339.0	22/11/2017
111192767-1	3952.0	22/11/2017
113103171-1	571.0	22/11/2017
213856127-1	2276.0	22/11/2017
213856128-1	2276.0	22/11/2017
213856165-1	1260.0	22/11/2017
213856166-1	1260.0	22/11/2017
213856237	2209.0	22/11/2017
213856238	2209.0	22/11/2017
213850245-1	1045.0	23/11/2017
213850246-1	1045.0	23/11/2017
213856127-1	1223.0	23/11/2017
213856128-1	1223.0	23/11/2017
213856165-1	360.0	23/11/2017
213856166-1	360.0	23/11/2017
213856237	2209.0	23/11/2017
213856238	2209.0	23/11/2017
850077-1	15910.0	23/11/2017
213850169-1	2562.0	24/11/2017
213850170-1	2562.0	24/11/2017
213850245-1	1873.0	24/11/2017
213850246-1	1873.0	24/11/2017
213856237	2209.0	24/11/2017
213856238	2209.0	24/11/2017
233656051-1	1008.0	24/11/2017
233656052-1	1008.0	24/11/2017
850077-1	15910.0	24/11/2017
SP0001038-1	545.0	25/11/2017
SP0001039-1	545.0	25/11/2017
213850169-1	1438.0	25/11/2017
213850170-1	1438.0	25/11/2017
213850245-1	1873.0	25/11/2017
213850246-1	1873.0	25/11/2017
213856237	2209.0	25/11/2017
213856238	2209.0	25/11/2017

233656051-1	1008.0	25/11/2017
233656052-1	1008.0	25/11/2017
850077-1	15910.0	25/11/2017
SP0001038-1	1243.0	26/11/2017
SP0001039-1	1243.0	26/11/2017
213850245-1	708.0	26/11/2017
213850246-1	708.0	26/11/2017
213856237	2209.0	26/11/2017
213856238	2209.0	26/11/2017
233656051-1	1008.0	26/11/2017
233656052-1	1008.0	26/11/2017
850077-1	15910.0	26/11/2017
SP0001038-1	1243.0	27/11/2017
SP0001039-1	1243.0	27/11/2017
213850239-1	1007.0	27/11/2017
213850240-1	1007.0	27/11/2017
213856237	1047.0	27/11/2017
213856238	1047.0	27/11/2017
233656051-1	1008.0	27/11/2017
233656052-1	1008.0	27/11/2017
850077-1	15910.0	27/11/2017
SP0001038-1	1243.0	28/11/2017
SP0001039-1	1243.0	28/11/2017
213850239-1	528.0	28/11/2017
213850240-1	528.0	28/11/2017
213856127-1	2276.0	28/11/2017
213856128-1	2276.0	28/11/2017
233650039-1	821.0	28/11/2017
233650040-1	821.0	28/11/2017
233656051-1	1008.0	28/11/2017
233656052-1	1008.0	28/11/2017
850077-1	452.0	28/11/2017
SP0001038-1	1243.0	29/11/2017
SP0001039-1	1243.0	29/11/2017
213856127-1	1223.0	29/11/2017
213856128-1	1223.0	29/11/2017
213856255-1	559.0	29/11/2017
213856256-1	559.0	29/11/2017
233650039-1	1134.0	29/11/2017
233650040-1	1134.0	29/11/2017
233656051-1	1008.0	29/11/2017
233656052-1	1008.0	29/11/2017

SP0000442	720.0	30/11/2017
SP0000450	720.0	30/11/2017
SP0000774-1	331.0	30/11/2017
SP0000775-1	331.0	30/11/2017
SP0001038-1	882.0	30/11/2017
SP0001039-1	882.0	30/11/2017
113103171-1	1140.0	30/11/2017
213856255-1	422.0	30/11/2017
213856256-1	422.0	30/11/2017
213856495-1	319.0	30/11/2017
213856496-1	319.0	30/11/2017
233650039-1	1134.0	30/11/2017
233650040-1	1134.0	30/11/2017
233656051-1	720.0	30/11/2017
233656052-1	720.0	30/11/2017
SP0000774-1	1159.0	01/12/2017
SP0000775-1	1159.0	01/12/2017
213856495-1	1100.0	01/12/2017
213856496-1	1100.0	01/12/2017
233650039-1	1134.0	01/12/2017
233650040-1	1134.0	01/12/2017
SP0000774-1	1159.0	02/12/2017
SP0000775-1	1159.0	02/12/2017
213850239-1	1447.0	02/12/2017
213850240-1	1447.0	02/12/2017
213856495-1	1100.0	02/12/2017
213856496-1	1100.0	02/12/2017
233650039-1	277.0	02/12/2017
233650040-1	277.0	02/12/2017
SP0000774-1	1159.0	03/12/2017
SP0000775-1	1159.0	03/12/2017
113103171-1	330.0	03/12/2017
213850239-1	88.0	03/12/2017
213850240-1	88.0	03/12/2017
213856495-1	479.0	03/12/2017
213856496-1	479.0	03/12/2017

Tab. 2: Demandas diarias

6.1.3. Información de máquinas

Los datos de información de las máquinas nos indican cuántas horas debemos de reservar al final de la producción:

Número de máquina	Horas de reserva
27012001	6
27012002	6
27012003	6
27012004	6
27012005	6
27012006	6

Tab. 3: Horas de reserva que requiere cada máquina

También disponemos de una tabla que nos permite conocer la compatibilidad entre moldes y máquinas en el problema:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
27012001	1	1	0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	0
27012002	1	1	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0
27012003	1	1	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0
27012004	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27012005	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1
27012006	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tab. 4: Matriz compatibilidad máquinas - moldes 1

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
27012001	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0
27012002	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0
27012003	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0
27012004	0	0	0	1	0	0	1	0	0	0	1	1	1	1	1	0	1	1	1
27012005	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
27012006	0	0	0	1	0	0	1	0	0	0	1	1	1	1	1	0	1	1	1

Tab. 5: Matriz compatibilidad máquinas - moldes 2

	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
27012001	0	1	1	0	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0
27012002	0	1	1	0	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0
27012003	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0
27012004	1	0	1	1	1	0	0	0	0	1	1	0	1	1	0	0	1	1	1
27012005	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
27012006	1	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0	1	1	1

Tab. 6: Matriz compatibilidad máquinas - moldes 3

	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76
27012001	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0
27012002	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0
27012003	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0
27012004	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
27012005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27012006	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1

Tab. 7: Matriz compatibilidad máquinas - moldes 4

	77	78	79	80	81	82	83	84	85
27012001	0	0	0	1	1	1	1	1	1
27012002	0	0	0	1	1	1	1	1	1
27012003	0	0	0	1	1	1	1	1	1
27012004	1	1	1	1	1	0	0	0	0
27012005	0	0	0	0	0	0	0	0	0
27012006	1	1	1	0	1	0	0	0	0

Tab. 8: Matriz compatibilidad máquinas - moldes 5

6.1.4. Información de piezas

Como hemos indicado anteriormente, tenemos que tener en cuenta que para cada semana no podremos producir un valor de piezas superior a su stock máxima sumada a su demanda semanal. En estos datos de entrada conocemos los stocks máximos de cada pieza.

Nº	Pieza	Stock	Nº	Pieza	Stock	Nº	Pieza	Stock
1	212831667	9000	52	213850246-1	6000	103	213856496-1	3000
2	212831668	9000	53	213850291-1	5425	104	213856497-1	5425
3	213856105	5425	54	213856071-1	1200	105	213856498-1	5425
4	213856106	5425	55	213856072-1	1200	106	213856499-1	5425
5	213856225	5425	56	213856073-1	600	107	213856500-1	5425
6	213856226	5425	57	213856074-1	600	108	233650039-1	8000
7	213856237	5425	58	213856087-1	1200	109	233650040-1	8000
8	213856238	5425	59	213856088-1	1200	110	233656011-1	5425
9	213856239	5425	60	213856089-1	600	111	233656012-1	5425

10	213856240	5425	61	213856090-1	600	112	233656051-1	6000
11	213856489	5425	62	213856091-1	5425	113	233656052-1	6000
12	213856490	5425	63	213856092-1	5425	114	233656053-1	3000
13	213856491	5425	64	213856093-1	5425	115	233656054-1	3000
14	213856492	5425	65	213856094-1	5425	116	253356001-1	5425
15	233656013	5425	66	213856109-1	8000	117	253356002-1	5425
16	262236055	5425	67	213856110-1	8000	118	253356003-1	5425
17	262236056	5425	68	213856111-1	3000	119	253356004-1	5425
18	262236057	5425	69	213856112-1	3000	120	850077-1	150000
19	262236058	5425	70	213856119-1	10000	121	850089-1	4000
20	111192632-1	5425	71	213856120-1	10000	122	856027-1	1200
21	111192767-1	5425	72	213856127-1	8000	123	856028-1	1200
22	113103065-1	5425	73	213856128-1	8000	124	856029-1	600
23	113103171-1	5425	74	213856129-1	3000	125	856030-1	600
24	213850123-1	4000	75	213856130-1	3000	126	856033-1	5425
25	213850149-1	5425	76	213856143-1	10000	127	856034-1	5425
26	213850150-1	5425	77	213856144-1	10000	128	856035-1	5425
27	213850161-1	1000	78	213856159-1	5425	129	856036-1	5425
28	213850162-1	1000	79	213856160-1	5425	130	SP0000434	3000
29	213850163-1	1000	80	213856161-1	5425	131	SP0000442	2000
30	213850164-1	1000	81	213856162-1	5425	132	SP0000445	500
31	213850165-1	1000	82	213856165-1	3000	133	SP0000447	500
32	213850166-1	1000	83	213856166-1	3000	134	SP0000449	3000
33	213850167-1	1000	84	213856167-1	5425	135	SP0000450	2000
34	213850168-1	1000	85	213856168-1	5425	136	SP0000451	500
35	213850169-1	6000	86	213856169-1	5425	137	SP0000452	5425
36	213850170-1	6000	87	213856170-1	5425	138	SP0000750	2000
37	213850171-1	2000	88	213856171-1	5425	139	SP0000751	2000
38	213850172-1	2000	89	213856172-1	5425	140	SP0000752	2000
39	213850175-1	5425	90	213856203-1	5425	141	SP0000753	2000
40	213850176-1	5425	91	213856204-1	5425	142	SP0000772-1	2000
41	213850177-1	5425	92	213856205-1	5425	143	SP0000773-1	2000
42	213850178-1	5425	93	213856206-1	5425	144	SP0000774-1	6000
43	213850179-1	5425	94	213856227-1	5425	145	SP0000775-1	6000
44	213850180-1	5425	95	213856228-1	5425	146	SP0001038-1	4000
45	213850181-1	5425	96	213856253-1	5425	147	SP0001039-1	4000

46	213850182-1	5425	97	213856254-1	5425	148	SP0001040-1	4000
47	213850239-1	5425	98	213856255-1	5425	149	SP0001041-1	4000
48	213850240-1	5425	99	213856256-1	5425	150	SP0001042-1	2000
49	213850241-1	5425	100	213856493-1	5425	151	SP0001043-1	2000
50	213850242-1	5425	101	213856494-1	5425	152	SP0001044-1	2000
51	213850245-1	6000	102	213856495-1	3000	153	SP0001045-1	2000

Tab. 9: Stocks máximos de piezas

6.2. Solución inicial

La solución inicial tiene una demanda no satisfecha de piezas con el nombre 212831667-68. Esto demuestra que la heurística implementada no es lo suficientemente potente y demuestra por qué debemos aplicar una metaheurística para obtener una solución que cubra toda la demanda con un coste energético mínimo.

En la aplicación de la *búsqueda tabú*, obtendremos nuevas soluciones al explorar el vecindario de una solución, eliminando inyecciones, añadiendo inyecciones o haciendo un movimiento mixto que nos elimine una inyección y nos añada otra inmediatamente después. Nos quedaremos con la que más mejore la función fitness, aplicando un criterio de aspiración por objetivo en el que permitimos una aspiración de movimiento si minimiza el mejor costo (al ser un problema de minimización).

A continuación se va a explicar la solución inicial y qué ha sucedido cada día de ésta:

El primer día se ha inyectado tres moldes en las tres primeras máquinas, produciendo las piezas correspondientes a los moldes 1, 7 y 83.

	LUNES - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	-5	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Maquina 2	-5	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
Maquina 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 13: Matriz de solución: Solución inicial, día 1

El segundo día tenemos la inyección de tres nuevos moldes en las tres últimas máquinas, también tenemos un día de mantenimiento. Empleamos los moldes 1, 9, 52, 74 y 83 para la producción.

		MARTES - SEMANA 1																							
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1		-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
Maquina 2		83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
Maquina 3		-5	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52
Maquina 4		-5	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Maquina 5		-5	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74

Fig. 14: Matriz de solución: Solución inicial, día 2

El tercer día finalizamos la producción de algunos moldes y los reemplazamos por algunos nuevos. Se producen las piezas generadas por los moldes 1, 9, 11, 22, 52, 74, 82 y 83. Se producen tres cambios de molde.

		MIÉRCOLES - SEMANA 1																							
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1		7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	-5	-5
Maquina 2		83	83	83	83	-5	-5	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82
Maquina 3		52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52
Maquina 4		9	9	9	9	9	9	9	9	9	9	-5	-5	22	22	22	22	22	22	22	22	22	22	22	22
Maquina 5		74	74	74	74	74	74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 15: Matriz de solución: Solución inicial, día 3

El cuarto día liberamos el molde 74, que había satisfecho la demanda pero al llegar a tres los cambios de molde, no se pudo retirar este. A partir de este punto veremos que las máquinas están ociosas porque al llegar a la programación de esta máquina se cubre la demanda, que podemos cubrir con las máquinas ociosas, para las dos semanas. En este punto las máquinas paradas son la 4 y la 5.

		JUEVES - SEMANA 1																							
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Maquina 2		82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82
Maquina 3		52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	-5
Maquina 4		22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	-5
Maquina 5		-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 16: Matriz de solución: Solución inicial, día 4

En el quinto día tenemos también tres cambios de molde y dos de las cuatro máquinas están paradas. Tendremos un cambio de molde entre este día y el anterior para colocar el molde 26.

	VIERNES - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	16	16	16	16	16	16	16	16	16	16	-5	-5	64	64	64	64	64	64	64	64	64	64	64	64
Maquina 2	82	82	82	82	-5	-5	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
Maquina 3	-5	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 17: Matriz de solución: Solución inicial, día 5

En el sexto día también se libera la tercera máquina.

	SÁBADO - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
Maquina 2	44	44	44	44	44	44	44	44	44	44	44	44	44	44	-5	0	0	0	0	0	0	0	0	0
Maquina 3	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 18: Matriz de solución: Solución inicial, día 6

En el séptimo día tendremos tres de las máquinas paradas completamente y un cambio de molde exclusivamente en la máquina 3.

	DOMINGO - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 3	26	26	26	26	26	26	26	26	-5	-5	39	39	39	39	39	39	39	39	39	39	39	39	39	39
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 19: Matriz de solución: Solución inicial, día 7

En el octavo día tendremos la máquina 4 realizando mantenimiento y dos cambios de molde para las máquinas más activas, Máquina 0 y Máquina 1

	LUNES - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	-5	-5	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
Maquina 1	-5	-5	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 3	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39
Maquina 4	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 20: Matriz de solución: Solución inicial, día 8

En el noveno día tendremos un cambio de molde en la máquina 3

	MARTES - SEMANA 2																								
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
Maquina 0	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	
Maquina 1	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Maquina 3	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	-5	-5	55	55
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig. 21: Matriz de solución: Solución inicial, día 9

En el décimo día satisfacemos la demanda de las piezas producidas por el molde 59 y procedemos a emplear el 82.

	MIÉRCOLES - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
Maquina 1	59	59	-5	-5	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 3	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 22: Matriz de solución: Solución inicial, día 10

En el undécimo día tendremos 4 de las 6 máquinas liberadas.

	JUEVES - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
Maquina 1	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	-5	0	0	0	0	0	0
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 3	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 23: Matriz de asignación: Solución inicial, día 11

En el duodécimo día seguimos produciendo con las máquinas 0 y 3

	VIERNES - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	62	62	62	62	62	0	-5	-5	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	
Maquina 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Maquina 3	55	55	-5	-5	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig. 24: Matriz de solución: Solución inicial, día 12

En el decimotercero día se libera la máquina 3

	SABADO - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	
Maquina 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Maquina 3	75	-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig. 25: Matriz de solución: Solución inicial, día 13

En el decimocuarto y último día la máquina 0 continuará con la inyección del molde 81 que queda montado en la máquina 0 para la próxima planificación.

	DOMINGO - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	81	81	81	81	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 1	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 2	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 3	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 4	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 5	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4

Fig. 26: Matriz de asignación: Solución inicial, día 14

En la producción creada por el algoritmo inicial no es capaz de cubrir la demanda de una pieza porque la máquina 1, que es la única que puede emplear el molde 10, está ocupada realizando otras piezas con una mayor demanda, por lo que se queda sin horas de trabajo posibles para realizar estas piezas demandadas en menor medida, la implementación busca corregir este tipo de fallos que en el algoritmo no se tiene en cuenta.

6.3. Solución factible

Una solución factible es aquella que consigue cumplir las siguientes condiciones:

1. Emplea moldes habiendo realizado un cambio de molde previamente.
2. Retira los moldes una vez ha finalizado con ellos.
3. Respeta los cambios de turno de los trabajadores y no se realizan cambios de molde durante esas franjas.
4. No supere los stocks máximos semanales de cada pieza.
5. Es coherente con los datos de entrada:
 - Mantiene las horas de reserva al final de la semana.
 - Las máquinas solo pueden emplear aquellos moldes compatibles.

6.4. Solución infactible

Una solución infactible puede ser definida así por las siguientes circunstancias:

1. Número de cambios de molde en un día superior a 3.

	Lunes																								
Hora	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	0
Maquina 0	-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	-5	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	-3
Maquina 2	-5	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
Maquina 3	-5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5

Fig. 27: Solución con cambios de molde inválidos

2. Una máquina emplee un molde que no puede soportar.

En los datos de entrada dispondremos de la relación que informa en una tabla binaria la compatibilidad entre cada máquina y cada molde. Dentro de nuestra implementación nos hemos asegurado que se respeten las compatibilidades.

3. Horas de reserva reemplazadas por producciones.

Cada máquina tiene unas horas de reserva que están especificadas en los datos de entrada en una tabla que indica el porcentaje de horas semanal que esa máquina requiere para tareas de reserva. En estas pruebas el porcentaje de horas de reserva es del 6

4. Moldes no liberados antes de colocar un nuevo molde.

Nuestra implementación debe asegurarse que los moldes quedan liberados tras la satisfacción de la demanda de las piezas, o porque vaya a ser empleada por otra máquina (escenario que pueda darse en un movimiento de inyección de molde en la *lista tabú*).

5. Cambios de molde en un cambio de turno.

Los cambios de molde siempre se hacen dentro de un turno de trabajo, no suelen empezarse dentro de un turno y acabar en otro. Por tanto, en la última hora de cada turno, no se inicia un cambio de molde. Los turnos son: de

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
83	83	83	83	83	83	83	83	83	83	83	2	2	2	2	2	2	2	2	2	2	2	2	2
-5	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52
-5	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
-5	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74

Fig. 28: Moldes no liberado antes de realizar una nueva inyección

06:00-14:00, de 14:00-22:00 y de 22:00-06:00. Por lo tanto, no están permitidos los cambios de 5:00 a 7:00, de 13:00 a 15:00 o de 21:00 a 23:00 horas.

Martes																							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
62	62	62	62	62	-5	-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	-5	-5	6
85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85
39	39	39	39	39	39	39	39	39	39	39	39	39	-5	-5	2	2	2	2	2	2	2	2	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 29: Cambios de molde en un cambio de turno

- 6. Horas en las que una máquina, durante una producción se pare, a menos que haya satisfecho la demanda.

Nuestra implementación debe de asegurarse que la inyección de un molde continúe produciendo piezas hasta que satisfaga la demanda o se tope con otro molde. Este caso puede darse durante la aplicación de la *búsqueda tabú* y hayamos realizado una inyección.

Hora	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	0
Maquina 0	-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	-5	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	-3
Maquina 2	-5	83	83	83	83	83	83	83	83	83	0	0	0	83	83	83	83	83	83	83	83	83	83	83	83
Maquina 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5

Fig. 30: Inyección parada

6.5. Ajuste de parámetros

En el ajuste de parámetros hemos buscado el equilibrio necesario para que una solución con una demanda no satisfecha no muy inferior a otra pero con un coste eléctrico y de gas sea sensiblemente inferior. Para ello lo que hemos realizado ha sido probar con diferentes pesos para ver qué tal se comportaba la *búsqueda tabú*.

Con el mismo peso entre las demandas no satisfechas y los costes de electricidad y gas, en la mayoría de ocasiones seleccionaba una solución vecina con una demanda no satisfecha mayor que la óptima de su vecindario, esto se debe a que una producción que no satisfaga todas las piezas, consumirá menos suministros al estar las máquinas ociosas, que siguen consumiendo pero no es comparativo al coste que tiene una solución con toda la demanda cubierta, todo esto depende de qué máquinas sean las que más trabajen, en el apartado de la descripción del problema se ve que las máquinas más eficientes son las tres primeras.

Como ejemplo se va a mostrar el funcionamiento de la selección de una solución en un vecindario con unos pesos idénticos entre las funciones objetivos.

Id	Consumo eléctrico	Consumo de gas	Demanda no satisfechas	Tipo de movimiento
1	3096,10€	20210,93€	27017	Eliminar molde
2	2943,61€	20753,05€	24811	Eliminar molde
3	3157,74€	20238,44€	0	Añadir molde
4	2395,40€	19993,47€	80607	Eliminar molde

Tab. 10: Ejemplo de entorno de una solución

Uno podría esperar que la solución escogida fuera la tercera, sin embargo, la seleccionada sería la primera. Esto se debe a un valor máximo de piezas no demandadas muy altas, 80607, que relativiza la demanda no satisfecha y hace parecer que no hay tanta diferencia entre una demanda no satisfecha de 0 piezas y 27000 piezas.

Como se ve, la balanza de la función objetivo cae a favor de los consumos, por eso debemos de dar un peso alto a las demandas no satisfechas para que la ejecución de la *búsqueda tabú* no acabe en soluciones vacías con una demanda no satisfecha cercana a la totalidad y unos consumos de suministros muy bajos. De poco nos sirve una solución que no cumple la demanda, por lo que esta priorizará sobre el objetivo de reducción de costes.

La *lista tabú* ha almacenado los últimos 10 movimientos realizados, con ello hemos evitado repetir aquellos movimientos que nos hacen caer de nuevo en soluciones ya exploradas. De esta forma hemos prevenido ciclos y hemos conseguido llevar la exploración a nuevas regiones. Haciendo que la lista tuviera un tamaño inferior hacía que repitiésemos movimientos y fuese un poco redundante la exploración por lo que definimos que para nuestro problema 10 era un número adecuado, aumentarlo hu-

quiera limitado en gran manera la capacidad de explorar el espacio de soluciones de la *búsqueda tabú* ya que no hay muchas inyecciones realizadas en estas dos semanas.

6.6. Mejor resultado

La mejor solución obtenida ha sido aquella que cubre las demandas de las piezas demandadas con el consumo mínimo posible. Esta solución se ha repetido en el tiempo sucesivas veces. Hemos permitido que el algoritmo siga explorando el espacio de soluciones con el objetivo de buscar una solución mejor. Múltiples de las inyecciones han sufrido cambios, aunque es necesario recordar que estos están limitado por la compatibilidad máquina-molde.

Hay bastantes cambios respecto a la solución inicial, por destacar uno crítico es la inyección del molde 10 en la máquina 0, que nos permitirá cubrir la demanda que la solución inicial no satisfacía. Que ahora se puede se debe a que en algún momento el molde 1 fue retirado de la inyección y el 10 fuera insertado, posteriormente la producción de la máquina quedaría supeditada también a ese molde.

Este resultado obtenido, por medio de la comunicación, puede ser obtenido desde cualquier solución inicial mediante la eliminación e inyección de elecciones. Habrá soluciones con un fitness muy próximo ya que al cubrir las demandas el otro factor es la optimización económica de los recursos de la empresa.

	LUNES - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 2	-5	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
Maquina 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 4	0	0	0	0	0	0	-5	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 31: Matriz de solución: Mejor resultado, día 1

	MARTES - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
Maquina 2	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
Maquina 3	-5	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52
Maquina 4	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	-5	-5	22	22	22	22	22	22
Maquina 5	-5	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74	74

Fig. 32: Matriz de solución: Mejor resultado, día 2

	MIERCOLES - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5
Maquina 2	83	83	83	83	-5	-5	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82
Maquina 3	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52
Maquina 4	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
Maquina 5	74	74	74	74	74	74	-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 33: Matriz de solución: Mejor resultado, día 3

	JUEVES - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Maquina 2	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82
Maquina 3	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	-5
Maquina 4	22	22	22	22	-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 34: Matriz de solución: Mejor resultado, día 4

	VIERNES - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	16	16	16	16	16	16	16	16	16	16	-5	-5	64	64	64	64	64	64	64	64	64	64	64	64
Maquina 2	82	82	82	82	-5	-5	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
Maquina 3	-5	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 35: Matriz de solución: Mejor resultado, día 5

	SÁBADO - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Maquina 1	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
Maquina 2	44	44	44	44	44	44	44	44	44	44	44	44	44	44	-5	-5	7	7	7	7	7	7	7	7
Maquina 3	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 36: Matriz de solución: Mejor resultado, día 6

	DOMINGO - SEMANA 1																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	1	1	1	1	1	1	1	1	1	1	1	-5	-5	10	10	10	10	10	10	10	10	10	10	10
Maquina 1	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
Maquina 2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Maquina 3	26	26	26	26	26	26	26	26	-5	-5	39	39	39	39	39	39	39	39	39	39	39	39	39	39
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 37: Matriz de solución: Mejor resultado, día 7

	LUNES - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	-5	-5	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
Maquina 1	-5	-5	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59
Maquina 2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	0	0	0	0	0	0
Maquina 3	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39
Maquina 4	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
Maquina 5	-5	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	0

Fig. 38: Matriz de solución: Mejor resultado, día 8

	MARTES - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
Maquina 1	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59	59
Maquina 2	-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 3	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	0	0	-5	-5
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 39: Matriz de solución: Mejor resultado, día 9

	MIERCOLES - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
Maquina 1	59	59	-5	-5	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 3	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 40: Matriz de solución: Mejor resultado, día 10

	JUEVES - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
Maquina 1	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	-5	0	0	0	0	0	0
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 3	55	55	55	55	55	55	55	55	55	55	55	55	-5	0	0	0	0	0	0	0	0	0	0	0
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 41: Matriz de solución: Mejor resultado, día 11

	VIERNES - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	62	62	62	62	62	0	-5	-5	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81
Maquina 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 42: Matriz de solución: Mejor resultado, día 12

	SABADO - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81
Maquina 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 2	0	0	0	0	0	-5	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	-5
Maquina 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maquina 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 43: Matriz de solución: Mejor resultado, día 13

	DOMINGO - SEMANA 2																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Maquina 0	81	81	81	81	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 1	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 2	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 3	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 4	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
Maquina 5	0	0	0	0	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4

Fig. 44: Matriz de solución: Mejor resultado, día 14

6.7. Función fitness agregado

En el siguiente gráfico se ve el resultado agregado de los dos objetivos que buscábamos minimizar: el número de demanda no satisfecha y los costes de producción de las producciones. En un primer momento experimentaremos mucha diversidad de resultados porque los pesos de la función fitness están orientados hacia una mayor exploración, puntualmente habrá valores buenos y malos, aunque no son capaces de mejorar la solución inicial hasta llegado un punto en el que comienza a mejorar la solución.

Según vamos avanzando progresivamente el porcentaje posible de exploración se reduce y vamos siendo más selectivos con las soluciones vecinas, esto hace que el rango se vaya reduciendo progresivamente según avancemos en la ejecución de la implementación. No hemos conseguido minimizar estas variaciones más allá puesto que puede darse que una solución esté rodeada por varias soluciones que no la mejoren y queramos aún así explorar estas soluciones vecinas por escapar de los óptimos locales (motivo por el cual hemos aplicado una metaheurística). No hemos percibido que haya un comportamiento destacable con los movimientos híbridos empleados que incorporaban la retirada de un molde para la inyección de otro, ya que según íbamos avanzando estos nos alejan del óptimo muchas veces al hacer eliminaciones de inyecciones de forma innecesaria.

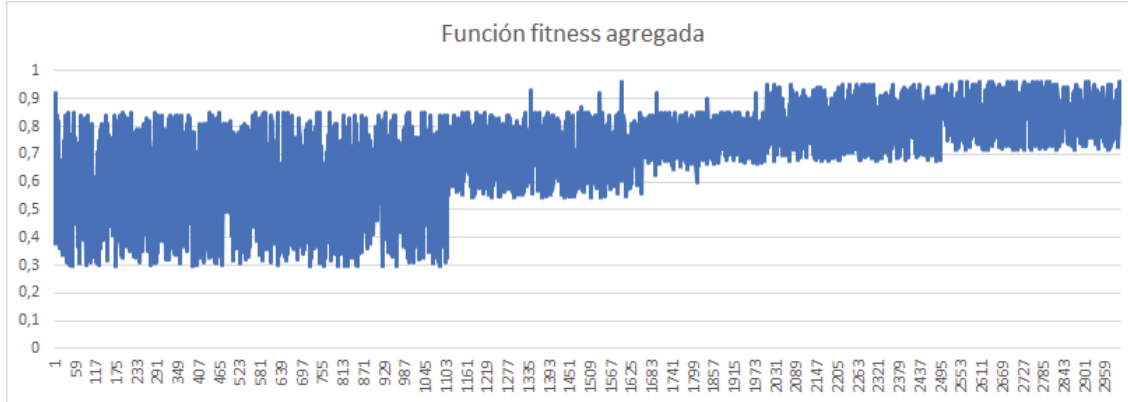


Fig. 45: Función fitness agregada

7. Conclusiones y trabajo futuro

En un primer momento hemos realizado una introducción a esta familia de problemas y se ha hablado de cómo se han enfrentado las empresas a los problemas de programación, concretamente para este problema hemos abordado un problema en la producción de un área de fundición de aluminio. Posteriormente, hemos descrito el problema, explicando su complejidad y a que se debe que sea necesario la aplicación de una técnica que nos permita realizar su planificación de forma automática para prevenir fallos humanos y conseguir que esta se genere más rápido que de forma manual.

A continuación, hemos ido explicando los diferentes datos de entrada que tiene el problema. Todos los datos de entrada con los que contamos para este problema: compatibilidad entre las máquinas y los moldes, los días festivos, las horas de reserva de las máquinas, las horas de mantenimiento de las máquinas, la información de las piezas en la que conoceremos sus demandas, y moldes que nos permiten generar las piezas.

Posteriormente, hemos hablado de los problemas de optimización matemática, de las diversas técnicas que nos permiten resolver esta familia de problemas, comenzando por los Problemas de programación lineal, explicando el *método del símplex*, a continuación los problemas de programación entera dónde narramos brevemente el funcionamiento del *algoritmo de ramificación y acotación*. Hemos explicado la motivación por la que necesitamos la aplicación de una metaheurística para los problemas de optimización combinatoria y seguidamente hemos explicado las diversas metaheurísticas que se han aplicado a esta familia de problemas y que pueden ser interesantes por los trabajos relacionados.

En la siguiente fase explica la metodología que hemos aplicado para resolver esta problemática, inicialmente modelamos la solución esperada en este problema. Después hemos expuesto los pasos realizados, en los que partiendo de la generación de una solución inicial por una heurística aplicaremos la *búsqueda tabú* para conseguir refinar esa solución y obtener una solución que mejore la función fitness. La búsqueda tabú incorpora diversos movimientos para tratar de mejorar la exploración inicialmente y finalmente tratar de posibilitar una mejora en la intensificación.

Hemos modelizado el problema, con una implementación que genera una solución inicial válida que es cercana a la satisfacción de todas las demandas. Posteriormente, hemos implementado la *búsqueda tabú* que genera soluciones aleatorias en la que se busca un hueco en la planificación en el que la máquina esté parada, para realizar un cambio de molde. Inyectamos un molde hasta satisfacer la demanda de esa pieza. Además de esta posibilidad, el movimiento de eliminación de inyecciones es lo que nos permite ampliar la diversificación en la búsqueda de soluciones porque solo con inyecciones seguramente hubiéramos intensificado un óptimo local. Las soluciones

se seleccionan y se mejoran en base a la función objetivo tratando de minimizar la demanda pendiente de las piezas y los costes de suministros (luz y electricidad) ajustándola en un software propio. Inicialmente buscamos ser más exploratorios y diversificarnos en el espacio de soluciones. Finalmente hemos tratado de intensificar para obtener la mejor solución posibles. Todas las generaciones han sido válidas y han respetado las restricciones del problema.

Finalmente hemos ejecutado el programa generado para comprobar para unos datos de entrada que representaban la demanda de dos semanas. Estos datos contienen la misma producción que los mismos que la empresa está empleando a día de hoy.

Una vez generadas soluciones válidas al problema y haber comprobado cómo funciona la implementación, hemos realizado un ajuste paramétrico con el objetivo de optimizar la implementación desarrollada, ajustando los pesos de los distintos objetivos y el tamaño de la *lista tabú*.

Con esto hemos conseguido demostrar la aplicabilidad de la *búsqueda tabú* para este problema concreto consiguiendo satisfacer la demanda minimizando los consumos de electricidad y gas.

Trabajo futuro

Una vez que se ha hecho uso de esta metaheurística para la resolución de este problema sería interesante realizar un estudio comparativo entre los resultados que se extraen en este trabajo y con los que podrían ofrecer otras alternativas, además dentro de la *búsqueda tabú* tiene multitud de variaciones.

Haciendo uso de otras metaheurísticas podríamos conseguir unos mejores resultados, porque se obtenga una solución superior en la la función fitness a la obtenida con esta metaheurística o porque se obtenga una similar en un tiempo inferior.

Mejorar la implementación realizada para que se aproveche de las ventajas de la programación multihilo, reduciendo su tiempo de ejecución.

Modificar la forma en la que se obtiene la solución inicial, podemos aleatorizar el orden en que se van completando las máquinas/filas de la tabla.

Incorporar nuevos movimientos que se pueden realizar, la mutación de una inyección, eliminación de una inyección e inyección de un molde parciales.

Minimizar las simplificaciones que se han realizado en la descripción del problema.

1. Tener en cuenta que los tiempos de inyección están en orden de segundos y

por tanto podríamos optimizar las producciones aumentando la calidad de solución media.

2. Las demandas en nuestra solución se agrupan de forma semanal, si se hiciese de forma diaria podríamos ajustarnos más a las demandas reales que tienen los clientes.
3. Los cambios de molde están redondeados a dos horas, lo cual si hacemos una estimación más exhaustiva de los tiempos en los cambios de molde podríamos conseguir minimizar los tiempos en los que la máquina no está funcionando. Esto también afecta a los cambios de molde diarios que hay entre varios días ya que cuando se hace un cambio de molde a medianoche lo contabilizaremos como hecho el segundo día.
4. Modificar la función fitness para que tenga en cuenta más factores a parte de la minimización de la demanda no satisfecha y la minimización del consumo energético. Podríamos minimizar el número total de cambios de molde realizados y minimizar el retraso en las partes producidas respecto al día de la entrega, teniendo en cuenta la demanda de forma diaria.

Referencias

- [1] Asadzadeh, L. (2015). A local search genetic algorithm for the job shop scheduling problem with intelligent agents. *Computers & Industrial Engineering*, **85**, 376-383.
- [2] Batista, B. M., & Glover, F. (2006). *Introducción a la búsqueda tabú*. Universidad de La Laguna, Santa Cruz de Tenerife.
- [3] Cañari, A. C. R. (2005). *Capítulo 3:” Búsqueda Tabú*. Universidad Nacional Mayor de San Marcos. Facultad Ciencias Matemáticas (Investigación Operativa). Lima.
- [4] Colorni, A., Dorigo, M., Maniezzo, V., & Trubian, M. (1994). Ant system for job-shop scheduling. *Belgian Journal of Operations Research, Statistics and Computer Science*, **34(1)**, 39-53.
- [5] Chang, H. C., Chen, Y. P., Liu, T. K., & Chou, J. H. (2015). Solving the flexible job shop scheduling problem with makespan optimization by using a hybrid Taguchi-genetic algorithm. *IEEE Access*, **3**, 1740-1754.
- [6] Crainic, T. G., Gendreau, M., & Farvolden, J. M. (2000). A simplex-based tabu search method for capacitated network design. *INFORMS journal on Computing*, **12(3)**, 223-236.
- [7] Dantzig, G. B. (1990). *Origins of the simplex method* (pp. 141-151). ACM, Nueva York.
- [8] Dantzig, G. B. (2016). *Linear programming and extensions*. Princeton university press, Princeton.
- [9] Davis, L. (1985, July). *Job shop scheduling with genetic algorithms*. In *Proceedings of an international conference on genetic algorithms and their applications* (Vol. **140**). Carnegie-Mellon University Pittsburgh, Pennsylvania.
- [10] Díaz, A., Glover, F., Ghaziri, H. M., González, J. L., Laguna, M., Moscato, P., & Tseng, F. T. (2000). *Optimización heurística y redes neuronales*, Paraninfo, Madrid.
- [11] Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing*, **1(3)**, 190-206.
- [12] Glover, F., & Laguna, M. (1998). *Tabu search*. In *Handbook of combinatorial optimization*. Springer, Boston.
- [13] Glover, F., & Melián, B. (2003). Búsqueda tabú. Inteligencia Artificial. *Revista Iberoamericana de Inteligencia Artificial*, **7(19)**.

- [14] Herrera, M. (2011). Programación de la producción: una perspectiva de productividad y competitividad. *Revista Virtual Pro*, **42(11)**.
- [15] Herrmann, J. W. (2007). The legacy of Taylor, Gantt, and Johnson: how to improve production scheduling. *ISR TECHNICAL REPORT 2007*, **26**.
- [16] Jain, A. S., Meeran, S. (1998). *A state-of-the-art review of job-shop scheduling techniques* (pp. 1-48). Technical report, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland.
- [17] Kundakcı, N., & Kulak, O. (2016). Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem. *Computers & Industrial Engineering*, **96**, 31-51.
- [18] Marti, R. (2003). *Procedimientos metaheurísticos en optimización combinatoria*. Universidad de Valencia, Valencia.
- [19] Mateo, P., & Lahoz, D. (2009). *Programación lineal entera*, Universidad de Zaragoza, Zaragoza.
- [20] Mateos Caballero, A., Ríos Insua, S., Jiménez Martín, A. y Fernández Martín, A. (2006), *Investigación Operativa: Ejercicios y Aplicaciones*, Fundación General de la Universidad Politécnica de Madrid, ISBN 84-96244-63-6, Madrid.
- [21] May, G., Stahl, B., Taisch, M., & Prabhu, V. (2015). Multi-objective genetic algorithm for energy-efficient job shop scheduling. *International Journal of Production Research*, **53(23)**, 7071-7089.
- [22] Mocholí, M., & Sala, R. (1984). *Programación lineal. Ejercicios y Aplicaciones*. Ed. Tebar Flores, Madrid.
- [23] Nash, J. C. (2000). The (Dantzig) simplex method for linear programming. *Computing in Science & Engineering*, **2(1)**, 29-31.
- [24] Ortiz-Trian, V. K., & Caicedo-Rolón, Á. J. (2014). Programación óptima de la producción en una pequeña empresa de calzado-en Colombia. *Ingeniería Industrial*, **35(2)**, 114-130.
- [25] Ríos Insua, S., Ríos Insua, D., Mateos Caballero, A., Martín Jiménez, J., & Jiménez Martín, A. (2006), *Problemas de Investigación Operativa: Programación Lineal y Extensiones*, RA-MA, ISBN 8478977309, Madrid.
- [26] Ríos Insua, S., Mateos Caballero, A., Bielza, C. y Jiménez Martín, A. (2004). *Investigación Operativa: Modelos Determinísticos y Estocásticos*, Editorial Centro de Estudios Ramón Areces S.A., ISBN 84-8004-666-X, Madrid.
- [27] Rossi, A. (2014). Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships. *International Journal of Production Economics*, **153**, 253-267.

- [28] Sharma, P., & Jain, A. (2016). A review on job shop scheduling with setup times. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, **230(3)**, 517-533.
- [29] Tang, L., Zhao, Y., & Liu, J. (2014). An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production. *IEEE Transactions on Evolutionary Computation*, **18(2)**, 209-225.
- [30] Tomás, D., & Rosa, M. (2015). *Programación lineal entera: un problema de planificación de plantilla*, Universidad de Valladolid, Valladolid.
- [31] Ying, W., & Bin, L. (1996). Job-shop scheduling using genetic algorithm. *In Systems, Man, and Cybernetics, 1996., IEEE International Conference on* (Vol. **3**, pp. 1994-1999). IEEE.
- [32] Zhang, X., Wang, S., Yi, L., Xue, H., Yang, S., & Xiong, X. (2018). An integrated ant colony optimization algorithm to solve job allocating and tool scheduling problem. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, **232(1)**, 172-182.
- [33] Zhao, F., Zhang, J., Zhang, C., & Wang, J. (2015). An improved shuffled complex evolution algorithm with sequence mapping mechanism for job shop scheduling problems. *Expert Systems with Applications*, **42(8)**, 3953-3966.