

3.1. Algoritmos heurísticos

Existen multitud de métodos heurísticos diseñados de forma específica para resolver distintos problemas de optimización, como puede ser el “Método Vogel” para el problema del transporte o el “Vecino más cercano” para el problema del viajante de comercio, en inglés Taveling Salesman Problem (TSP) por citar algunos ejemplos conocidos.

Además, más recientemente se han desarrollado distintos tipos de técnicas metaheurísticas, de propósito en general, que pueden ser aplicadas para resolver problemas muy diferentes de optimización combinatoria.

A continuación, se muestran varios de los métodos metaheurísticos más comunes. Osman y Kelly en 1996 aportan la siguiente definición: “Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son ni efectivos ni eficientes. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y la mecánica estadística.”

Los algoritmos metaheurísticos son algoritmos aproximados de optimización y búsqueda de propósito en general. Son procedimientos iterativos que guían una heurística subordinada combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda. Suelen tener un gran éxito en la práctica siendo, fácilmente implementables y paralelizables, en cambio, son algoritmos aproximados y no aseguran una solución exacta, además detrás de ellos no existe como tal siempre una base teórica establecida.

Cualquier algoritmo de búsqueda, para obtener soluciones de calidad, debe establecer un adecuado balance entre dos características del proceso, la intensificación y la diversificación. La intensificación se puede definir como la cantidad de esfuerzo empleado en la búsqueda en la región actual, es decir, la explotación del espacio, mientras que la diversificación es aquella cantidad de esfuerzo empleado en la búsqueda en regiones distantes del espacio, la exploración como tal. El equilibrio entre estas dos características, es necesario para identificar regiones en el espacio con soluciones buenas rápidamente, y a su vez, no consumir excesivo tiempo en regiones del espacio no prometedoras o ya exploradas.

Existen distintas metaheurísticas en función de conceptos como:

- Seguimiento de trayectoria considerado
- Uso de poblaciones de soluciones
- Uso de memoria
- Fuente de inspiración

Los métodos metaheurísticos, pueden ser clasificados en cuatro grandes grupos, que son los siguientes:

- Basados en métodos constructivos

- Basados en trayectorias
- Basados en poblaciones
- Algoritmos bioinspirados

Los basados en métodos constructivos son métodos que parten de una solución inicial vacía, y van añadiéndose componentes hasta construir una solución. Un buen ejemplo, es el problema de optimización de colonia de hormigas.

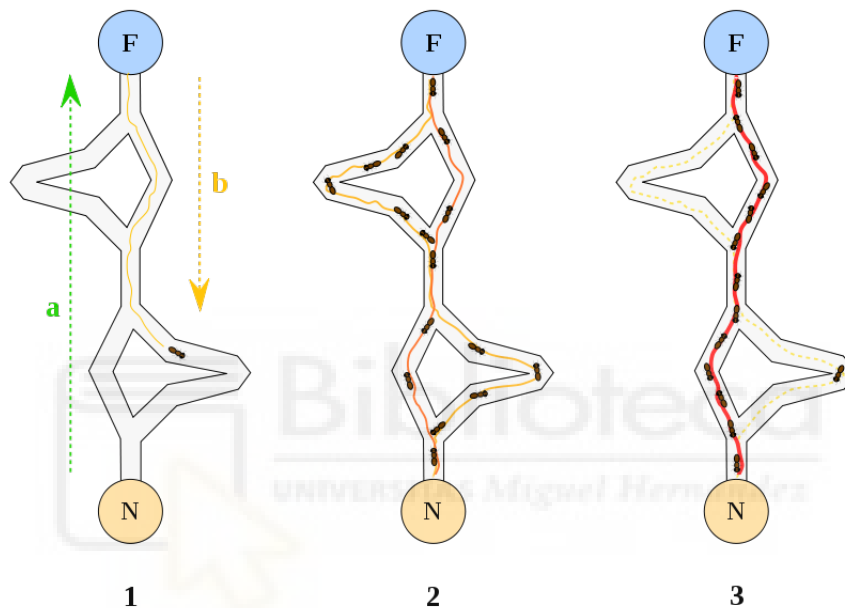


Figura 7: Ejemplo de un diagrama del algoritmo colonia de hormigas. Fuente: Wikipedia

En la figura 7 se muestra la evolución de este problema. Como se puede observar, existen dos nodos y un camino que puede originar distintas rutas, el nodo F representa la fuente de alimento y el nodo N la colonia de hormigas.

En primer lugar, una hormiga va en busca de alimento siguiendo cualquier ruta y vuelve a la colonia de hormigas. En ese momento, el resto de hormigas acuden también en busca de la fuente de alimentos tomando cualquier ruta de manera indiscriminada como se muestra en el paso 2. Por último, tras fortalecerse el camino la ruta más corta se convierte en la más atractiva y el resto empiezan a perder su rastro. De esta manera, a largo plazo todas las hormigas utilizan la ruta más corta.

Este experimento resuelto biológicamente, demostró que si a un conjunto de hormigas se le da la posibilidad de escoger dos rutas de distancias diferentes, escogen la ruta más corta. Esto sucede porque las hormigas dejan tras de sí un rastro de feromonas que la convierten en más atractiva, de esta manera al existir dos rutas, en una misma cantidad de tiempo, la ruta más corta será recorrida por más hormigas que la ruta más larga, aumentando así, la cantidad

de feromonas en la ruta más corta y siendo cada vez más atractiva. Finalmente, la ruta más larga irá desapareciendo, porque las feromonas son volátiles y todas las hormigas escogerán el camino más corto.

Así pues, tras partir de una solución vacía, de una manera continuada se va añadiendo información al problema, hasta poder construir una solución de calidad.

Los métodos basados en trayectorias, utilizan la heurística subordinada, que son algoritmos de búsqueda local que sigue una trayectoria en el espacio de búsqueda. Parten de una solución inicial e iterativamente se trata de reemplazar esa solución inicial por otra de mejor calidad. Existen ejemplos como pueden ser la búsqueda local, el recocido simulado o enfriamiento simulado (simulated annealing), o la búsqueda tabú (tabu search).

Fred Glover, como precursor de la búsqueda tabú, la definió como: “La búsqueda tabú guía un procedimiento de búsqueda local para el explorar el espacio de soluciones más allá del óptimo local.” La búsqueda tabú es un intento de aportar inteligencia a los algoritmos de búsqueda local. A través de la inteligencia artificial se utiliza el concepto de memoria y lo implementa mediante estructuras simples con el objetivo de dirigir la búsqueda teniendo en cuenta los antecedentes de ésta, es decir, se trata de extraer información de lo sucedido anteriormente, actuando así en consecuencia, existiendo un cierto aprendizaje y conllevando una búsqueda inteligente. Este método permite moverse a una solución aunque no sea tan buena como la actual, para no quedar así atrapado en óptimos locales y construir de una manera estratégica la búsqueda de soluciones aún mejores.

La búsqueda tabú en términos generales, es un algoritmo de búsqueda local basado en el uso de la memoria, existiendo dos tipos de memoria, a largo plazo y a corto plazo o memoria reciente. La memoria tiene la función de guardar atributos de las soluciones y es utilizada para no repetir la trayectoria de búsqueda, además, como se ha comentado anteriormente, es posible aceptar soluciones que empeoren la situación actual, con el objetivo de mejorar a largo plazo.

Para entender mejor este método, vamos a exponer un breve ejemplo. Disponemos de siete materiales y el objetivo es maximizar el poder aislante de esos materiales en función del orden en el que se encuentren. La función objetivo actúa como caja negra, es decir, no es posible obtener el funcionamiento de esta y el algoritmo avanza intercambiando las posiciones por pares. La iteración 0, obtiene la siguiente secuencia, $M = (2,5,7,3,4,6,1)$ con un poder aislante de 10. La búsqueda tabú nos indica que el mejor intercambio es el 4-5, ya que aumenta el poder aislante en 3 unidades. La iteración 1, obtiene la siguiente secuencia, $M = (2,4,7,3,5,6,1)$ con un poder aislante de 13, y la particularidad de que el intercambio 4-5 ya no se puede realizar más, hasta que pasen el número de iteraciones que se le indique al algoritmo, es decir, en cada iteración el intercambio realizado anteriormente queda tabú durante un número de iteraciones. El algoritmo funciona así sucesivamente. En ocasiones, se utilizará un intercambio que disminuirá el poder aislante, significando así que nos encontramos en un óptimo local y debemos avanzar en búsqueda de una solución mejor.

Existen diversos trabajos que han aplicado procedimientos heurísticos de búsqueda tabú para el problema de localización de instalaciones, bien con restricciones de capacidad o sin ellas

de la siguiente manera. El procedimiento heurístico, utiliza memorias a corto y largo plazo para realizar el proceso de búsqueda principal, así como las funciones de diversificación e intensificación. Las soluciones visitadas son almacenadas en un árbol primogénito que actúa como memoria a largo plazo. Además, la iteración reciente en la que una instalación cambia de estado se almacena para cada ubicación como memoria a corto plazo. Los límites inferiores en la disminución del coste total sirven como medidores del atractivo de cambiar de estado en cada instalación, y se usa para usar un movimiento en el proceso de búsqueda principal.

El enfriamiento o recocido simulado es un algoritmo de búsqueda por entornos con un criterio probabilístico de aceptación de soluciones basado en termodinámica. Como se ha comentado anteriormente, un modo de evitar que la búsqueda local finalice en óptimos locales, hecho que puede ocurrir con los métodos tradicionales de búsqueda local, es permitir que se produzcan movimientos hacia soluciones peores. Este proceso de escapar de óptimos locales, debe realizarse siempre de un modo controlado.

En el caso del enfriamiento simulado, esto se realiza controlando la frecuencia de los movimientos de escape mediante una función de probabilidad por la cual se disminuye la probabilidad de estos movimientos hacia soluciones peores conforme avanza la búsqueda, estando previsiblemente más cerca del óptimo local, explicando así la habitual filosofía de diversificar el principio e intensificar el final.

El fundamento de este control se basa en el trabajo de Metrópolis (1953) en el campo de la termodinámica estadística. Este autor, modeló el proceso de enfriamiento simulando los cambios energéticos de partículas conforme decrece la temperatura, hasta que converge a un estado estable (congelado). En el modelo de Metrópolis, se genera una perturbación aleatoria en el sistema y se calculan los cambios de energía resultantes, si existe una caída energética, el cambio se acepta automáticamente, en cambio, si se produce un incremento energético, el cambio será aceptado con una probabilidad determinada.

El algoritmo de enfriamiento simulado es un método de búsqueda por entornos que se caracteriza por un criterio de aceptación de soluciones vecinas que se adapta de manera continua a lo largo de su ejecución. Hace uso de una variable llamada T (temperatura), cuyo valor determina en qué medida pueden ser aceptadas soluciones vecinas peores que la actual. Esta variable se inicia a un valor alto como temperatura inicial, y se reduce a cada iteración mediante un mecanismo de enfriamiento de la temperatura, hasta alcanzar una temperatura final. Una de las diferencias, que expresa este método en comparación con la búsqueda tabú, es que el elemento memoria no es utilizado en ningún momento.

En cada iteración se genera un número concreto de vecinos, que puede depender de la iteración concreta o ser fijo para toda la ejecución. Cada vez que se genera un vecino, se aplica el criterio de aceptación para ver si sustituye a la solución actual, si la solución vecina es mejor que la actual, se acepta automáticamente, tal como se haría en la búsqueda local clásica. Sin embargo, si es peor, sigue existiendo la probabilidad de que el vecino sustituya a la solución actual, permitiendo así al algoritmo abandonar óptimos locales para no quedar atrapado. A mayor temperatura, mayor probabilidad de aceptación de soluciones peores. Así, el algoritmo actúa aceptando soluciones peores que al comienzo de la ejecución, en lo que se denomina la fase de exploración. Una vez finalizada la iteración, generando soluciones vecinas, se enfría la

temperatura y se pasa a la siguiente.

Este método ha sido utilizado con éxito, por ejemplo, en el problema del viajante, donde dada una lista de ciudades y de distancias entre cada par de ellas, el objetivo es encontrar aquella ruta más corta por la cual el viajero visite cada ciudad una única vez y vuelva al punto de origen. Supongamos que en el problema a resolver existen siete ciudades y se desea encontrar aquella ruta que las visita todas y vuelve al origen. Partiendo de una permutación aleatoria, en cada iteración se establece el conjunto de vecinos y de éstos se elige uno, que será la solución de la siguiente iteración. Cabe recordar, que en este método es necesario establecer adecuadamente el conjunto de vecinos, ya que no todos los nodos nos pueden transportar a todos los nodos, por tanto, el intercambio siempre será controlado para realizarse entre nodos vecinos.

Fijando el nodo 1 como punto de partida, que deberá ser siempre el primer elemento y el último de la secuencia, en la iteración 0 la solución podría ser esta, $C=(1,2,3,4,5,6,7,1)$. A partir de esta solución se establece un conjunto de vecinos, cada uno de los cuales podría obtenerse intercambiando dos posiciones consecutivas. De todos los vecinos se elige el mejor, que será la solución de la siguiente iteración si la probabilidad dada por el parámetro de temperatura así lo determina. Dicho valor podría ser inicialmente de 0.8 e ir decrementándolo en 0.05 unidades en cada iteración.

Este método también ha sido utilizado para la resolución de problemas del tipo de localización de plantas, con la utilización de la función Boltzmann, que toma parte cuando el siguiente valor es peor que la situación actual. A través de esta función existe una probabilidad de que el nuevo valor sustituya al actual, con el objetivo de no restringir la búsqueda únicamente a aquellas soluciones que aporten un impacto positivo a la función objetivo, sino también permitir movimientos que tengan un impacto negativo. Evitando con este mecanismo el procedimiento de quedar atrapado de una manera prematura en un óptimo local.

En los métodos basados en poblaciones el proceso considera múltiples puntos de búsqueda en el espacio que evolucionan en paralelo. Entre sus ejemplos más destacados se encuentran, los algoritmos genéticos, la búsqueda dispersa (scatter search) o los algoritmos meméticos.

La búsqueda dispersa, es un procedimiento metaheurístico introducido en la década de los setenta basado en formulaciones y estrategias. Los conceptos y principios fundamentales del método, fueron propuestos al comienzo de la década, basados en las estrategias para combinar reglas de decisión, especialmente en problemas de secuenciación o en la combinación de restricciones. Este método opera sobre un conjunto de soluciones, denominado conjunto de referencia, combinando éstas para crear nuevas soluciones de modo que mejoren a sus predecesoras.

En este sentido, es clasificado como un método evolutivo. Sin embargo, a diferencia de otros métodos evolutivos, como los algoritmos genéticos, la búsqueda dispersa no está fundamentada en la aleatorización sobre un conjunto relativamente grande de soluciones sino en elecciones sistemáticas y estratégicas sobre un conjunto pequeño. Una breve ilustración es, que mientras los algoritmos genéticos suelen trabajar con una población de 100 soluciones, la búsqueda dispersa, únicamente trabaja con un conjunto habitual de 10 soluciones.

La primera descripción de este método fue propuesta por Fred Glover en 1977 donde establece los principios de la búsqueda dispersa y la describe como una exploración sistemática sobre una serie de buenas soluciones llamadas conjunto de referencia. La búsqueda dispersa se basa en que la información sobre la calidad o atractivo de un conjunto de restricciones, reglas o soluciones puede ser utilizado mediante la combinación de éstas en vez de una manera aislada. Además, se basa en integrar la combinación de soluciones con la búsqueda local. Sin embargo, en diseños más avanzados también se puede hacer uso de la memoria, pero se asemeja más a una búsqueda tabú, y en la mayoría de casos nos limitamos a una búsqueda local convencional.

La búsqueda dispersa, trata de combinar las soluciones del conjunto de referencia, este conjunto almacena las buenas soluciones encontradas durante el proceso de búsqueda. Es importante hacer énfasis en el que la definición de buena solución no se restringe a la calidad de la solución sino también considerando la diversidad que aporta al conjunto de referencia. A grandes rasgos, el método funciona de la siguiente manera.

En primer lugar, a través de un generador de soluciones diversas, se genera un conjunto de P soluciones diversas (aproximadamente 100), de las cuales extraeremos un subconjunto (de alrededor 10) que se denomina conjunto de referencia. A continuación, se aplica un método de mejora con el objetivo de perfeccionar las soluciones a través de una búsqueda local.

En tercer lugar, se establece un método para crear y actualizar el conjunto de referencia. Dicho conjunto ha sido extraído según los criterios de calidad y diversidad, es decir, contener soluciones buenas y diferentes entre sí.

En último lugar, la búsqueda dispersa, se basa en combinar todas las soluciones del conjunto de referencia a través de un determinado método de combinación. Las soluciones que se obtienen de esta combinación pueden ser introducidas de manera instantánea o almacenadas temporalmente hasta realizar la totalidad de combinaciones y después observar que soluciones entran en el conjunto.

Un ejemplo de éxito de aplicación de este tipo de métodos es el problema de localización de plantas. Las soluciones del conjunto de referencia se combinan mediante un procedimiento que consta de dos fases: la fase de inicialización y la fase de mejora. Durante la primera fase, cada cliente es asignado a una instalación abierta para obtener una solución que posteriormente será mejorada en la fase dos.

En último lugar, se encuentran los algoritmos bioinspirados, que son metaheurísticas basadas en la naturaleza empleando analogías con sistemas naturales o sociales para la resolución de problemas. Los algoritmos bioinspirados simulan el comportamiento de sistemas naturales para el diseño de métodos heurísticos de búsqueda, aprendizaje o comportamiento. Actualmente, se consideran uno de los campos de investigación más prometedores en el diseño de algoritmos.

Entre las características que suelen presentar, se puede destacar que son métodos no determinísticos, en ocasiones presentan de manera implícita una estructura paralela y son adaptativos, utilizando una retroalimentación con el entorno para modificar el modelo y los parámetros. En resumen, actúan como una metáfora biológica, modelando de forma aproximada un fenómeno

existente en la naturaleza.

Entre algunos ejemplos se encuentran, las redes neuronales, que se basan en la simulación del comportamiento del sistema nervioso aplicando un paradigma de aprendizaje automático, o la optimización basada en la colonia de hormigas, que se basa en la simulación del comportamiento de las colonias de hormigas cuando recogen comida.

La inteligencia de enjambre (swarm intelligence), “son algoritmos o mecanismos distribuidos de problemas inspirados en el comportamiento colectivo de colonias de insectos sociales u otras sociedades de animales”, como definieron Bonabeau, Dorigo y Thereaulaz en 1999.

Una de las sociedades de insectos más comunes es la de las abejas, que actúan bajo una cooperación total dentro de la colmena, dividiendo la labor a través de la especialización de cada una y con una comunicación entre las fuentes de comida y su distancia. Un enjambre se encuentra compuesto de agentes simples existiendo una auto-organización descentralizada al no existir un único supervisor. Además de ser un conjunto robusto, ya que las actuaciones se completan aún con la deficiencia de algún individuo, y flexible, pudiendo responder a cambios extremos al disponer de una percepción del entorno. En resumen, la complejidad de la auto-organización se lleva a cabo sin un líder de la sociedad y la modelización de los insectos sociales puede ser de ayuda para el diseño de modelos artificiales distribuidos de resolución de problemas. El objetivo principal es coordinar el esfuerzo individual para alcanzar un fin común. Este tipo de algoritmos se ha utilizado con éxito en problemas de localización de plantas.

De entre los algoritmos inspirados en la naturaleza, sin duda, los que más se han utilizado con éxito en muy diversos problemas de optimización combinatoria son los algoritmos genéticos.

3.2. Algoritmos genéticos

Los algoritmos genéticos reciben su nombre al estar inspirados en los procesos de evolución natural y evolución genética. Surgieron en la década de los 70 de la mano de John Henry Holland y siguen los principios postulados por Darwin de que las poblaciones evolucionan en la naturaleza a través de la selección natural y la supervivencia de aquellos mejor adaptados.

En la naturaleza los individuos de una población deben competir entre sí en la búsqueda de recursos que aseguren su supervivencia. En ocasiones, incluso los miembros de una misma especie pelean entre sí por la búsqueda de recursos. Aquellos individuos que tengan éxito en la supervivencia y en la posibilidad de atraer parejas para la reproducción, serán aquellos que más probabilidad de descendencia tendrán. De esta manera, los genes de los mejores individuos y los más adaptados se propagarán hacia las siguientes generaciones. A lo largo del tiempo, las especies evolucionan logrando unas características cada vez más adaptadas al entorno en el que se encuentran.

Los algoritmos genéticos actúan de forma análoga a la evolución natural. Trabajan con una población de individuos, donde cada uno de ellos representa una posible solución al problema dado. A cada individuo se le asigna un valor de adecuación, que determina la bondad de ajuste de cada solución. En el mundo real sería equivalente a la probabilidad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adecuación mayor será la probabilidad de supervivencia y reproducción de ese individuo, cruzando su material genético con una pareja y dejando descendencia. De otra manera, cuando menor sea la adecuación del individuo al entorno, menor será la probabilidad de ser seleccionado para la reproducción y que su material genético sea propagado.

De esta forma se genera una nueva población que reemplaza a la anterior y contiene una mayor proporción de características mejores en comparación anterior. Sucesivamente a lo largo de las distintas generaciones las buenas características se van distribuyendo a lo largo de la población, explorando las áreas más interesantes del espacio de búsqueda y con la seguridad de que si el algoritmo ha sido diseñado de manera correcta, la población proporcionará una buena solución del problema.

Una de las características que ha hecho tan utilizada a este tipo de técnicas, es la robustez. Estos algoritmos pueden tratar con multitud de problemas provenientes de diferentes áreas. Los algoritmos genéticos no garantizan el obtener la solución óptima del problema, pero existe evidencia empírica de que se encuentran soluciones de un gran nivel, en un tiempo competitivo si es comparado con el resto de métodos de optimización combinatoria.

A grandes rasgos, un algoritmo genético sigue la siguiente estructura. En primer lugar, se genera una población inicial de individuos, para posteriormente evaluar cada individuo de dicha población, asignando a cada individuo un valor. Posteriormente, se produce el proceso de selección, por el cual los individuos mejores dotados dispondrán mayor probabilidad de ser seleccionados. Después, se aplica el proceso de cruce emparejando de manera aleatoria aquellos individuos que mejor valor de adecuación representan sin alterar el tamaño de la población. Existen diversos métodos de selección y de cruce. En último lugar, se produce el proceso de mutación, que únicamente tiene el objetivo de introducir variabilidad en la población.

Este proceso es repetido hasta que la condición de finalización sea cumplida, esta determina la parada del algoritmo que puede ser establecida por tiempo de computación o por número de generaciones que se desean estudiar en el algoritmo por citar solo las más comunes. Otro aspecto importante de este método, es la necesidad de definir de una manera adecuada como representar cada una de las posibles soluciones al problema, en lo que es conocido, como la codificación de las soluciones. El comportamiento del algoritmo y los resultados que proporcionan dependen en gran medida de este aspecto.

Codificación de las soluciones

Como se ha comentado anteriormente, un aspecto importante es la codificación de las soluciones, ya que es de vital importancia definir de una manera adecuada cada una de las posibles soluciones. El algoritmo genético no opera de manera directa sobre las propias soluciones, sino sobre una codificación representada de estas, que se asemeja al material genético de un individuo. Se han propuesto multitud de formas de como representar estas soluciones, Holland propuso el Teorema de los Esquemas por el cual era aconsejable representar las soluciones mediante un alfabeto binario, pero en la actualidad se ha demostrado que en la mayoría de problemas de optimización combinatoria, esta representación no es adecuada.

Durante estos años, se han propuesto otras formas de codificación como cadenas de enteros, de números reales o en ocasiones cadenas donde siquiera existen números, es decir, en la actualidad se puede trabajar con cadenas que representan cualquier tipo de alfabeto. En el caso de nuestro problema, quizás sería una buena opción crear una cadena a la que podemos llamar cromosoma del tamaño de la totalidad de plantas del problema, y cada posición de este cromosoma a la que podemos llamar gen, representa el vértice a donde cada planta envía su producto.

Población inicial

Otro aspecto importante, será la creación de la población inicial, este conjunto de individuos tiene la posibilidad de ser creado de una manera aleatoria o con la aplicación de un algoritmo heurístico.

Tras la creación de esta población inicial, cada individuo de dicha población ha de ser evaluado. Esto se consigue con la función de evaluación, que asigna a cada individuo un valor de adecuación e indica la competencia de cada individuo en comparación al resto que forman parte del conjunto de la población.

La mayoría de los modelos propuestos en el capítulo anterior, describen problemas donde las funciones objetivos tratan de minimizar o maximizar distancias, esto aporta una facilidad a la hora de determinar que individuos representan mejores valores en referencia a la función objetivo, puesto que si el objetivo es minimizar costes o distancias, aquellos individuos que representen menos valor en comparación con el resto serán más idóneos para nuestro modelo, mientras que en el caso de maximizar ocurre lo contrario.

Proceso de selección

El proceso de selección trata de simular la selección natural de las especies propuesta por Darwin para explicar la supervivencia de los mejores individuos. El proceso de selección asegura que aquellos individuos mejor dotados tendrán mayor presencia en las generaciones futuras, es decir, tendrán mayor posibilidad de sobrevivir. Como se ha comentado anteriormente, el valor de adecuación representa la probabilidad que tienen los individuos de salir victoriosos al competir por la supervivencia, existirán individuos que por sus buenas características reciben uno o varias copias en la siguiente población, mientras que otros pueden no recibir ninguna y llegar a desaparecer.

Se crea entonces una población de segunda generación que contendrá el mismo número de individuos que la población anterior y sustituye a ésta, produciéndose el cambio generacional. Es necesario tener en cuenta los aspectos de diversidad y tensión del proceso. Si el proceso de selección es demasiado agresivo solo aquellos individuos superdotados, serán aquellos que traspasen sus genes a la siguiente población, perdiendo la diversidad de las soluciones y con la posibilidad de que el algoritmo converja de manera temprana. En cambio, si la tensión del proceso es frágil, es decir, que la totalidad de los individuos tendrían la posibilidad de pasar a la siguiente generación, se pierde el objetivo primordial del proceso de selección, que es que aquellos individuos más preparados sean los que vayan estableciendo su material genético en las siguientes generaciones. Por tanto, será necesario realizar un equilibrio correcto entre ambos conceptos para que el proceso de selección sea eficiente y cumpla las expectativas.

Existen diversos mecanismos de selección de los individuos, desde la selección simple hasta los métodos más sofisticados como pueden ser los procesos determinísticos o estocásticos, pero siempre con la premisa de que aquellos individuos mejores y mejor adaptados deberán tener mayor posibilidad de ser copiados en las siguientes poblaciones. Para el problema presentado en este estudio se ha optado por un mecanismo de selección basado en el torneo y se explicará detalladamente en la sección 4.

Proceso de cruce

Tras haber generado la segunda población que sustituye a la original, se aplica el proceso de cruce. De nuevo se empareja de manera aleatoria a la totalidad de los individuos de esta segunda población, simulando el proceso de generación de hijos por parte de padres y madres. Es necesario añadir, que no es estrictamente imprescindible la necesidad de que todos los padres y madres crucen su material genético, de hecho existe una probabilidad de cruce, de manera que existirán emparejamientos que no llevarán a cabo el cruce, siendo ellos mismos los que pasen a la siguiente generación.

La segunda población tras haberle aplicado el proceso de cruce, tendrá un porcentaje de individuos que habrán sido inalterados, por lo que si se implementa una probabilidad de cruce del 80 %, de un tamaño poblacional de 1000 individuos, 800 de ellos serán resultado de la combinación de sus madres y padres siendo considerados hijos, mientras que 200 serán padres y madres que se han traspasado de manera inalterada, conviviendo así individuos de distintas generaciones en la misma población.

En este proceso, aunque los emparejamientos se realicen de manera aleatoria, en esta ocasión los mejores individuos sí que van a ser seleccionados más veces, puesto que tendrán más copias en la población que los individuos peor adaptados, participando así con mayor presencia en los emparejamientos y aumentando las posibilidades de pasar su material genético.

De nuevo existen multitud de métodos que pueden ser empleados, pero es necesario que estos sean aplicados de una manera adecuado respetando la combinación de material genético de padres y madres, es decir, el proceso de cruce debe ser diseñado de manera que no sólo se haga una combinación del material genético de los padres, sino que esa combinación sea beneficiosa y capaz de generar mejores individuos.

Además, para respetar el tamaño de la población, es necesario que cada proceso de cruce genere dos hijos, de forma que si el primer hijo, al que llamaremos hijo A, representa una serie de características, el segundo hijo, al que denominamos hijo B, representará las contrarias. Por ejemplo, si se determina que el hijo A recibe en las posiciones pares los índices de la madre, y en las posiciones impares las del padre, el hijo B lo recibirá de manera contraria, en las posiciones pares los índices del padre y en las inferiores los de la madre, para que no se generen dos hijos exactamente iguales.

Como en los procesos anteriores, existen diversos métodos de cruce que pueden ser aplicados. En primer lugar, el cruce 1-punto marca una posición k que determina en qué punto se pasa de obtener genes del padre al de la madre. Por ejemplo, supongamos que $k = 3$ y disponemos de una cadena de 5 elementos, entonces el hijo A recibirá las tres primeras posiciones del padre y las dos restantes de la madre, mientras que el hijo B todo lo contrario. De igual manera, funciona el cruce 2-puntos donde ahora existen las posiciones k_1 y k_2 , de la posición 1 a k_1 y de la posición k_2+1 hasta el final hereda los genes del padre, y entre k_1+1 hasta k_2 de la madre, es decir, funciona de manera análoga pero con dos pivotes.

Además existen otros métodos que actúan de una manera muy distinta como puede ser el cruce por orden, en el que es importante el orden en el que aparecen los genes para calcular su valor de adecuación. Sin embargo, se ha visto conveniente en nuestro caso utilizar el cruce uniforme, este método genera una máscara de cruce que está compuesta por una cadena que toman los valores cero y uno de manera aleatoria. Esta máscara se genera en cada emparejamiento y se determina que, si en la posición i aparece un 1 se intercambian los índices entre los padres, mientras que si aparece un 0 los índices se muestran inalterados.

En modelos donde se requería una especial precaución porque no se deben producir individuos que repitan índices en una misma cadena, o que en todas las posiciones de esa cadena no puedan aparecer todos los índices sino solo un conjunto de ellos, es necesario dirigir al proceso de cruce a través de distintas restricciones para que no se produzca esta incidencia de creación de soluciones no factibles para el modelo.

Proceso de mutación

En último lugar, aparece el proceso de mutación, que tiene el objetivo de añadir variabilidad al proceso tal y como ocurre en la evolución natural. Tras la finalización del proceso de cruce, este paso se asegura de que exista al menos una probabilidad de examinar cada punto

distinto del espacio de búsqueda con la posibilidad de introducir características nuevas en los individuos.

Es necesario determinar una probabilidad de mutación, que debe afectar a cada uno de los genes de cada uno de los individuos. Esta probabilidad debe ser pequeña en equiparación con la del proceso de cruce, para no trastocar en gran medida el proceso, de esta manera, todos los genes de todos los individuos tendrán la pequeña posibilidad de mutar, y así cambiar de manera aleatoria su valor.

El proceso actúa de forma simple recorriendo cada gen de cada individuo y con una especie de dado aleatorio se van obteniendo valores en cada posición, si el número obtenido cumple con las condiciones de la mutación, es decir, es menor o igual que a la probabilidad de mutación entonces el valor de dicho gen es modificado aleatoriamente seleccionando un elemento del resto de la población.

Como ocurría en el proceso anterior, existirán modelos que exijan la no repetición de índices en la misma cadena de un individuo o en el cual existan posiciones de esa cadena que no puedan utilizar la totalidad de los valores del alfabeto, por tanto, de nuevo se debería realizar un proceso de mutación de manera guiada impidiendo que ocurran estos inconvenientes durante la realización del algoritmo.

Cabe destacar que, tanto para los procesos de cruce y de selección, en este estudio se ha optado por la utilización de ambos métodos por una parte, con la posibilidad de que puedan repetirse índices dentro de un mismo individuo y por otra parte, con el impedimento de que esto pueda producirse.

Con el objetivo de intentar realizar una comparación entre ambas elecciones para observar que característica produce mejores soluciones.

4. Un algoritmo genético para MLP

En este capítulo se va a describir de forma detallada el diseño del algoritmo genético que posteriormente se ha implementado para resolver el MLP.

4.1. Desarrollo del algoritmo

El procedimiento del algoritmo es similar al que se explicó en el capítulo anterior, y sigue todos los pasos de un algoritmo genético. Además, en este apartado se va a profundizar en los distintos apartados que componen la creación de este método.

Codificación de la soluciones

Es el primer paso del algoritmo, la decisión de como van a ser representadas las distintas soluciones, es de gran importancia definir estas de una manera adecuada. Como fue comentado en el capítulo anterior, se decantó por escoger la opción de codificar las soluciones a través de una cadena de genes a la que se le denomina cromosoma y que cuenta con la misma longitud que el número de instalaciones existentes en el problema.

Cada una de las posiciones del cromosoma, indica el índice al cual la planta posicionada en ese lugar envía el material, es decir, si en la posición 2 de la cadena, se encuentra el índice 18, este hecho nos indica que la planta 2 envía la producción al a ese vértice en el cual se encontrará un cliente o un intermediario, pero no otra instalación.

Población Inicial

Es uno de los pasos fundamentales del algoritmo, la creación de una población inicial de manera aleatoria. Supongamos que se dispone de 10 ubicaciones, de las cuales tres de estas actúan como instalaciones y decidir donde deben dirigir el producto es la pieza fundamental del problema, ya que el objetivo principal es minimizar el coste encontrando aquella combinación de plantas y clientes más eficiente.

Cada individuo, va a ser representado por un cromosoma de tres posiciones, donde cada posición va a representar el vértice donde esa planta va a transportar su producto, es decir, la posición uno del individuo muestra donde la planta 1 conduce su material, como es sabido, con anterior este material puede ser conducido a la misma planta, un cliente o un intermediario, y así sucesivamente con el resto de las plantas.

Una de las características del modelo, es que no todos los índices pueden ser implementados, en todas las posiciones de un individuo, esto conlleva la creación de distintos conjuntos, en concreto de tres, donde cada cual representa a una única planta, por tanto, cuando se requiera introducir un índice en una posición determinada de un individuo, con hacer referencia al conjunto adecuado, se evitan problemas de índices en posiciones donde no pueden existir por la definición del problema.

Aquí se muestran los conjuntos utilizados:

Conjunto 1: $\langle 1,4,5,6,7,\dots,10 \rangle$

Conjunto 2: $\langle 2,4,5,6,7,\dots,10 \rangle$

Conjunto 3: $\langle 3,4,5,6,7,\dots,10 \rangle$

Sin embargo, como será comentado más adelante, el problema cuenta con la implementación de dos vertientes diferentes, implicando así la necesidad de usar dos codificaciones diferentes, una en la cual se permitirá la repetición de índices en cualquier posición de la cadena y otra en la cual este hecho no será posible, y por ello, es necesario conocer el engranaje que debe ser utilizado y este es mostrado a continuación.

Como se puede observar, el procedimiento es similar en todas las instalaciones, la planta 1 puede albergar todas las ubicaciones, excepto las otras dos plantas, la planta 2, de igual manera todas las ubicaciones, incluida la suya, exceptuando las dos plantas restantes, y así sucesivamente.

Si no existieran restricciones de repetición, la creación de la población inicial sería un proceso muy simple, se escogería aleatoriamente una muestra de cada uno de los conjuntos y eso comprendería un individuo, sin embargo, la utilización de los conjuntos facilita este proceso, ya que población inicial se genera de la siguiente manera.

Se escoge un elemento aleatorio del conjunto 1 y se evalúa ese índice, si el elemento está comprendido en el resto de los conjuntos como podría ser los índices de 4 a 10, se elimina ese índice de la totalidad de conjuntos restantes, mientras que si el índice seleccionado es el 1, como únicamente se muestra en ese conjunto no es necesario realizar ninguna modificación en el resto de conjuntos.

Así pues, se escoge un elemento, si se encuentra en otro conjunto se elimina del resto, si no existe se mantienen inalterables los conjuntos, y tras cada individuo de la población formado, los conjuntos vuelven a tomar su valor original. De esta manera, aseguramos que el índice del elemento 2 no coincida con el del elemento 1, al no encontrarse en el conjunto y no poder ser seleccionado de manera aleatoria, y así con el resto de elementos que comprenden un individuo.

Nos situamos en el ejemplo de que disponemos de un conjunto de 100 vértices, de los cuales 10 actúan como plantas y los 90 restantes como clientes. Cada solución se representaría a través de una cadena de 10 posiciones, donde la posición uno representa donde sirve la planta 1 su producto, la posición dos donde realiza este proceso la planta 2, y así sucesivamente. Una posible solución sería, el siguiente cromosoma: (25,33,45,4,12,89,74,68,93,10). El índice 25 nos indica que la planta 1 envía su producto al cliente 25 o el índice 4 nos indica que la planta 4 no realiza ningún envío y sirve su producto en la propia planta.

Función Objetivo

Tras la creación de la población inicial, el siguiente paso es el de evaluar cada individuo

de dicha población, para conocer el nivel de adecuación al problema y determinar como de probable es que sea seleccionado en el proceso de selección.

Como el objetivo del problema es el de minimizar costes, aquellos individuos que representen menor valor de función objetivo, serán aquellos con mayores posibilidades de ser escogidos en el proceso de selección. La función objetivo contiene dos partes bien diferenciadas, en la primera se refleja el coste de las plantas y en la segunda el de los clientes.

Dado un individuo de la población, se ha de calcular el coste que representa cada una de las plantas y cada uno de los clientes en función de los índices del individuo. Se observa de una manera a través de un ejemplo. Se imagina que disponemos del siguiente individuo (19,15,25,14,16). La primera parte consiste en calcular los costes derivados de las plantas, por lo que en el siguiente caso, se suma el coste de transportar material del nodo 1 al 19 por el producto del coste de abrir la planta 1, más el coste de transportar del nodo 2 al 15 por el producto de abrir la planta 2.

Para calcular el coste de los clientes se calcula el mínimo coste de los nodos 19,15,25,14 y 16 al nodo 6 mas el producto del coste de operar con el cliente 6, para el cliente 7 de nuevo se calcula el mínimo coste entre los cinco nodos que componen el individuo y se multiplica por el coste de operar con dicho cliente, y así sucesivamente.

Es importante destacar que, cuando una planta no transporta su producto y lo sirve en la propia planta no incurre en ningún coste de transporte pues el coste de transportar nodo 3 al nodo 3 será de cero, ya que no se modifica la ubicación, sin embargo si los clientes se encuentran excesivamente lejos el coste de transporte de estos será mucho mayor que si la planta hubiese decidido transportar el producto. De esta manera, encontraremos entre las soluciones más eficientes individuos que representen instalaciones que han optado por el autoservicio, al no conllevar costes de transporte.

Tras la adjudicación del valor de adecuación a cada individuo de la población, ya se puede proceder al siguiente paso que corresponde el proceso de selección, en código el cálculo del valor de cada individuo no ha tenido gran complejidad, pues con la utilización de grandes matrices y los parámetros adecuados, así como varias sentencias for de una manera rápida y eficiente se le puede atribuir un valor a cada individuo.

Proceso de Selección

El proceso de selección tiene el objetivo de escoger un conjunto de padres y madres que van a formar la población de segunda generación. Los tamaños de estos conjuntos serán la mitad del tamaño de la población, ya que en el posterior proceso de cruce se selecciona un individuo que actúa como madre y otro como padre, surgiendo de él dos individuos hijos.

Se aplica un método de torneo, por el cual se realizarán tantos emparejamientos como individuos existen en la población inicial, y durante estos duelos se medirá el valor de adecuación de cada individuo, de manera que aquellos que representen un valor más competente para nuestro modelo serán seleccionados por delante de su oponente, en nuestro caso tenemos preferencia por aquellos individuos con menor valor de adecuación ya que es necesario minimizar

distancias.

Los individuos que van a participar en los emparejamientos serán seleccionados de forma aleatoria, de manera que cualquier individuo de la población tendrá la posibilidad de competir por su supervivencia y por dejar descendencia en las futuras generaciones. Al realizarse de manera aleatoria los emparejamientos, aquellos individuos con un valor idóneo tendrán la posibilidad de traspasar sus genes en distintos individuos, mientras que aquellos más débiles, si disponen de suerte en el emparejamiento podrán permanecer una generación más.

Nos situamos en un ejemplo de 10 plantas y 90 clientes, y generamos una población inicial de 1000 individuos, tras evaluar cada uno de ellos por la función de evaluación se realizan de manera aleatoria 1000 emparejamientos en el que participarán 2000 individuos, por lo que se asegura con una probabilidad muy alta que todos los individuos participen al menos una vez en un desafío, y además, tengan la posibilidad de participar en multitud de ocasiones. De estos 1000 emparejamientos obtendremos un vencedor por emparejamiento en función de la eficiencia que aportan al modelo cada uno de ellos, de esta forma el tamaño poblacional queda inalterado y obtenemos una población de segunda generación.

De esta manera, se van rellenando los conjuntos de padres y madres, como se ha comentado estos deberán comprender un tamaño que se adecue a la mitad del tamaño poblacional, si de cada enfrentamiento surge únicamente un vencedor, y se necesitan tantos padres y madres que sumen en conjunto el total de individuos, se deberán realizar tantos enfrentamientos como individuos existen en la población actual.

Con este proceso se consigue que casi la totalidad de los individuos participen en al menos un enfrentamiento del torneo o en multitud de ocasiones, asegurando así que los mejores individuos tengan mayor probabilidad de ser escogidos para formar la siguiente generación, además aquellos individuos mejor capacitados tendrán la posibilidad de registrar varias copias en los distintos conjuntos de padres y madres.

Sin embargo, como ocurre en los procesos de selección natural, aquellos individuos menos adecuados a las condiciones del medio tendrán dificultades para pasar la criba del torneo y en ocasiones poder llegar a desaparecer al no conseguir registrar ninguna copia en alguno de los conjuntos.

El objetivo principal de este proceso es conformar ambos grupos a través de una correcta selección de los individuos, que posteriormente ambos conjuntos de individuos son tratados en los siguientes procesos de cruce y selección, produciendo así el cambio generacional sin sufrir alteraciones en los tamaños de la población.

Proceso de Cruce

Tras la selección de los individuos que van actuar como padres y madres, es el turno del proceso de cruce, este simula la combinación de características en la creación de individuos. Como se ha comentado anteriormente, este modelo presenta dos variaciones en las cuales intervienen los procesos de cruce y mutación al permitir la repetición o no.

El objetivo es que la población de segunda generación, al menos un 20 % de los individuos se muestren inalterados, mientras que el 80 % restante sean formados a través de la combinación de los conjuntos de padres y madres seleccionados. Por tanto, se ha de implementar un probabilidad de mutación de 0.8, es decir, cada combinación de padres y madres se producirá 8 de cada 10 veces, mientras que en las 2 restantes ambos individuos de primera generación, serán copiados de manera similar en la población de segunda generación.

En la formación de los individuos, la selección de las partes que el primer hijo recibe de la madre y de las partes que el primer hijo hereda del padre no ha seguido un patrón fijo y ha sido decidido de manera aleatoria a través de una máscara de cruce, así como para el segundo hijo. Esta máscara de cruce cuenta con un tamaño similar al de los individuos, es decir, cinco posiciones, y comprende los valores binarios 1 y 0.

Si la posición refleja el valor 1, quiere decir que el primer hijo hereda la posición de la madre y el segundo hijo el del padre, de esta forma, si la máscara muestra en la primera posición un 0, el índice 1 del padre será transferido al primer hijo y el índice 1 de la madre será copiado en el segundo hijo.

Como la máscara de cruce muestra un tamaño de cinco índices, de los cuales cada uno de ellos solo puede tomar dos valores, al ser de carácter binario, la máscara puede mostrar 32 combinaciones diferentes (dos elevado a cinco) de como un individuo puede ser formado a través de sus creadores. Para entender mejor este proceso, es mejor apoyarse en un ejemplo.

Padre: (15,21,12,4,19)
Madre: (18,20,11,9,5)
Máscara: (1,0,0,1,0)

Ahora existen dos posibilidades, se lanza un dado que contiene diez posibles resultados, de 1 a 10, si el dado muestra el valor 9 o 10, los individuos padre y madre serán copiados en la población de segunda generación de manera idéntica olvidando la máscara de cruce, si el dado muestra alguno de los valores de 1 a 8, los individuos pasarán por la máscara de cruce.

El primer hijo al que se puede nombrar como hijo A, hereda las posiciones 1 y 4 de la madre, y las posiciones 2,3 y 5 del padre, el segundo hijo, al que se puede denominar como hijo B, transfiere las posiciones 1 y 4 del padre, y las posiciones 2,3 y 5 de la madre, es decir, de manera contraria siendo el resultado final el siguiente.

Hijo A: (18,21,12,8,19)
Hijo B: (15,20,11,4,5)

En los modelos donde se implemente el cruce con repetición, el proceso finaliza aquí, ya que no es necesario evaluar los individuos por si muestran alguna malformación en la repetición de índices en distintas posiciones. Sin embargo, en el proceso de cruce sin repetición, sí que es obligatoria evaluar estos individuos para asegurar la correcta formación de estos.

Esta evaluación se hace comparando posición a posición en la combinación de los individuos de tal manera que en ocasiones no se seguirán las instrucciones de la máscara de cruce

para poder mantener la restricción de no repetición. Por tanto, se evalúa la posición 2 con la 1, si no existe problema de repetición se sigue la indicación de la máscara de cruce, si existe problema el índice del individuo elige que valor escoger. De nuevo, es necesario la utilización de un ejemplo para entender mejor el procedimiento.

Padre: (15,18,12,4,19)

Madre: (18,15,11,9,5)

Máscara: (1,0,0,1,0)

El resultado debería ser el siguiente: hijo A (18,18,12,9,19) hijo B (15,15,11,4,5). Por lo tanto, cuando se haga la comprobación entre cada posición, en el turno de la segunda posición comprobará que es similar a la primera por tanto aunque el hijo A en la segunda posición deba mostrar el valor 18, porque la máscara de cruce nos lo dicta así, el proceso está realizando de tal forma que en estas situaciones excepcionales se escoja entre el valor 2 del padre o de la madre según mejor convenga en cada situación.

El resultado final sería el siguiente: hijo A (18,15,12,9,19) hijo B (15,18,11,4,5). En este proceso no se encuentran problemas de índices en posiciones donde no corresponden, ya que se modifican los índices entre posición y por tanto, si el algoritmo se inicia correctamente en la creación de la población inicial, nunca va a existir un índice 1 que no sea en otra posición que la 1 y así con los cinco primeros índices.

En este proceso se ha encontrado un inconveniente que ocurre en un porcentaje muy pequeño de veces y con mayor frecuencia cuando se utilizan tamaños poblacionales muy elevados. Se supone un gen padre (23,10,12,16,11) y un gen madre (11,15,22,4,12), con una máscara de cruce (1,0,0,1,0), obteniendo los siguientes correspondientes hijos: hijo A (11,10,12,4,11) e hijo B (23,15,22,16,12). Nos encontramos con la mala fortuna de que la última posición del hijo A es indiferente el índice que escoja de los padres (11 o 12) ya que ambos ya han sido implementados en el individuo en las posiciones 1 y 3.

Cuando ocurre esta incidencia no se modifica la malformación de un hijo ya que eso pertenece al proceso de mutación, simplemente se deja en la población con la seguridad de que en generaciones futuras acabará desapareciendo al no ser una solución factible del modelo.

Proceso de Mutación

En último lugar se produce el proceso de mutación con el objetivo de añadir variabilidad al proceso. Cualquier gen de cualquier individuo contará con una probabilidad de mutación, esta es notablemente inferior a la probabilidad de cruce siendo únicamente del 0.1.

Los conjuntos utilizados para la creación de la población inicial, también son utilizados en este proceso para prevenir errores de índices en posiciones que no correspondan. Así, el proceso de mutación actúa en todo momento con el lanzamiento constante de dados por cada gen de cada individuo, 1 de cada 10 ocasiones el dado indicará que es momento de producir una mutación.

En ese instante, se obtiene que posición es la que ha sido seleccionada para producirse en

ella la mutación, entonces se selecciona el conjunto de esa posición y se extrae de manera aleatoria una muestra de ese conjunto.

Como con el proceso de cruce, el proceso de mutación, también forma parte de una de las dos variantes de los modelos a implementar, la variante de la población sin importar la repetición, el proceso finaliza aquí, pues los individuos no son evaluados y simplemente se cambia una posición de cada diez por una muestra aleatoria.

Sin embargo, para el modelo que incluye la restricción de no repetición cada posición que ha sido mutada, deberá posteriormente ser evaluada para observar si coincide o no con cualquier otro índice del individuo, en caso de que no coincida el individuo habrá sido mutado de forma satisfactoria.

En cambio, si el índice coincide con cualquier otra posición del individuo, la muestra obtenida no será válida y se deberá escoger otra, para de nuevo volver a ser evaluada y comprobar que el individuo muestra un aspecto adecuado a las características del modelo, de esta forma hasta que un individuo no muestre repetición entre alguno de sus índices el proceso de mutación no habrá finalizado.

Todos estos pasos que componen el algoritmo son implementados en única iteración de este. El algoritmo ha sido probado con distintos tamaños de la población, durante distintas iteraciones y en diferentes ocasiones. Las explicaciones del porqué se seleccionan el número de iteraciones a implementar o el tamaño de la población, serán explicados en la siguiente sección.

La sección de resultados mostrará los tiempos de computación de cada modelo, el número de iteraciones necesarias para encontrar el óptimo o la comparativa entre los distintos tamaños poblacionales, así como una gran serie de conclusiones que se han podido extraer de la realización de este estudio que serán mostradas en el último capítulo.

4.2. Estudio computacional

En este apartado se muestran los datos que se han utilizado en el algoritmo genético, así como, las distintas variantes del algoritmo implementado o los diferentes tamaños de la población implementados.

Los datos han sido obtenidos de una librería creada por J.E. Beasley, dicha librería contiene una gran colección de conjuntos de datos que pueden ser utilizados en pruebas para la resolución de problemas de investigación operativa.

El conjunto de datos seleccionado dispone de 25 nodos que actúan como ubicaciones, esos nodos van actuar en distinta medida entre los tres tipos de formas que pueden adoptar. Los cinco primeros nodos van a ser marcados como plantas, los dieciséis restantes como clientes, y los cuatro últimos como restos.

La base de datos únicamente muestra los costes entre los distintos nodos, es decir, el coste de transportar unidades del nodo i al nodo j , las coordenadas de cada ubicación son desconocidas, por lo que no se dispone de las ubicaciones exactas de cada punto, impidiendo dar un enfoque visual al problema.

El objetivo del modelo es disponiendo solamente del coste de transportar unidades entre los distintos nodos, determinar aquella combinación que deben formar las cinco plantas sobre a quién servir su producto, para conseguir así un coste total mínimo y un nivel de actuación eficiente.

El algoritmo tiene un tamaño de unas 700 líneas de código que se mostrarán en su totalidad en la sección denominada anexo. La población inicial introducida será creada de manera aleatoria y comprenderá distintos tamaños poblacionales, para sacar conclusiones sobre si influye el tamaño de la población para encontrar una solución de manera más rápida.

El método va a ser estudiado en dos supuestos diferentes, tras la creación de una población inicial aleatoria, la cual no va a permitir la repetición de ninguno de los índices de cada individuo, posteriormente sí que se va producir la posibilidad de que ocurra la repetición o no en los procesos de cruce y selección.

De esta manera, van a coexistir dos métodos de resolución totalmente diferentes uno donde el conjunto final de soluciones esté comprendido por individuos que no representan genes repetidos, mientras que el otro mostrará un pequeño porcentaje de individuos con las mismas características, ya que al ser un proceso aleatorio si no se restringe la repetición es muy probable que ocurran casos de similitud.

Esta acción se realiza con el objetivo de estudiar si el hecho de no permitir la repetición dentro de un mismo individuo influye de manera eficiente en el algoritmo, o si simplemente aunque esta restricción no sea utilizada, las soluciones que el método proporcionan ya descartan de por sí los individuos con genes repetidos al no ser eficientes.

Entre las distintas variables que se han utilizado, se dispone de costes tanto para abrir una

planta como para mantener operativo un cliente, así como una matriz de costes que relaciona los 25 nodos del problema. A continuación, se muestra cada uno de ellos:

Costes por Planta					
Planta	1	2	3	4	5
Coste	24	25	24	23	24

El coste por planta ha sido obtenido de manera aleatoria entre los parámetros 22 y 25 unidades de coste, una vez se establecen, son inalterados a lo largo del estudio.

Costes por Cliente																
Cliente	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Coste	14	14	14	15	13	12	14	13	13	15	12	12	13	14	14	13

El coste por cliente ha sido obtenido de manera aleatoria entre los parámetros 12 y 15 unidades de coste, una vez se establecen, son inalterados a lo largo del estudio.

La matriz de costes de tamaño 25x25 ha sido obtenida de la base de datos de la librería OR-Library creada por J.E. Beasley, específicamente del apartado de Location: p-hub, en su cuarto archivo de datos phub4.txt. A continuación, se muestra un extracto de este, de las cinco primeras filas y las quince primeras columnas:

Costes de transporte entre Clientes y Plantas															
Nodo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.000	0.576	0.946	0.597	0.373	0.559	0.709	1.208	0.603	0.695	0.680	1.936	0.332	0.592	0.908
2	0.576	0.000	0.369	0.613	0.429	0.312	1.196	1.502	0.405	1.241	0.960	2.318	0.786	0.949	0.938
3	0.946	0.369	0.000	0.858	0.749	0.556	1.541	1.764	0.621	1.603	1.250	2.600	1.137	1.266	1.124
4	0.597	0.613	0.858	0.000	0.255	0.311	0.790	0.907	0.237	0.932	0.406	1.741	0.485	1.186	0.345
5	0.373	0.429	0.741	0.255	0.000	0.225	0.794	1.080	0.238	0.879	0.533	1.889	0.402	0.947	0.598

El modelo ha sido estudiado con cuatro tamaños poblacionales (50, 100, 150, 200) y durante 50 iteraciones, además es estudiado el tiempo computacional de cada variante y se realizan tres pruebas distintas entre cada una de ellas, ya que los algoritmos cuentan con una gran parte de aleatoriedad y así se evita que quepa la posibilidad de que la mejor versión dé la peor solución.

Existen dos versiones del algoritmo bien diferenciadas que han sido comentadas en la sección anterior y hacen referencia a la posibilidad de permitir o no la repetición de índices en posiciones diferentes dentro del mismo individuo.

Todas las variables expuestas son aplicadas a cada modelo en cada versión y en los cuatro tamaños de la población distintos, por lo que se puede entender, que la totalidad de los modelos utilizan de los 25 nodos disponibles, 5 como plantas, 16 como clientes y 4 como ubicaciones restantes.

4.3. Análisis de resultados

En este apartado se muestran los resultados obtenidos con el algoritmo genético desarrollado en este estudio, así como las deducciones que se pueden obtener al realizar un análisis de estos.

A continuación, se muestran los distintos resultados:

Resultados del estudio					
Tamaño	Repetición	Prueba	Valor	Iteración	Tiempo
50	Sí	1	108.1264	33	9.22
50	Sí	2	115.6047	4	9.11
50	Sí	3	125.1214	26	8.94
50	No	1	108.2384	34	12.55
50	No	2	109.6229	3	12.40
50	No	3	113.4109	44	13.23
100	Sí	1	113.7693	50	13.20
100	Sí	2	112.3553	4	10.23
100	Sí	3	110.0429	39	12.11
100	No	1	112.1225	19	11.15
100	No	2	114.2527	37	15.05
100	No	3	108.6116	38	11.07
150	Sí	1	109.5050	5	9.67
150	Sí	2	111.3846	7	9.86
150	Sí	3	113.8128	3	11.75
150	No	1	115.4097	13	11.53
150	No	2	112.8156	37	11.56
150	No	3	114.2597	4	11.00
200	Sí	1	109.9185	13	13.33
200	Sí	2	112.8317	38	14.83
200	Sí	3	111.1129	40	14.64
200	No	1	109.3456	27	15.23
200	No	2	112.1994	14	16.10
200	No	3	107.1431	12	13.55

Se han utilizado cuatro tamaños de la población distintos, cada uno en su variación de permitir o no la repetición de índices por parte de un individuo, obteniendo ocho modelos distintos, cada uno de ellos ejecutado tres veces, obteniendo 24 ejecuciones cuyos resultados vamos a analizar en esta sección.

Las dos primeras columnas muestran simplemente el número de individuos evaluado durante cada iteración del algoritmo y si se trata de la versión con repetición o sin ella, la tercera columna simplemente enumera el número del experimento de cada modelo distinto.

La columna que representa el valor de la función objetivo, muestra la del mejor individuo encontrado en cada ejecución del correspondiente algoritmo y en la columna siguiente se indica el número de iteración en la que se ha encontrado este individuo. La última columna indica el tiempo total de ejecución hasta completar las 50 iteraciones. Por ejemplo, la primera

línea indica que en la primera ejecución del algoritmo que sí permite repeticiones, usando un tamaño de la población de 50 individuos, el mejor, que proporciona un valor en la función objetivo de 108.12, ha sido encontrado en la iteración 33 y el tiempo total de ejecución hasta completar las 50 iteraciones ha sido de 9.22 segundos.

Resultados medios del estudio			
Tamaño	Repetición	Valor	Tiempo
50	Sí	116.2841	9.09
50	No	110.4240	12.72
100	Sí	112.0558	11.84
100	No	111.6622	12.42
150	Sí	111.5674	10.42
150	No	114.1616	11.36
200	Sí	111.2872	14.26
200	No	109.5627	14.96

Otro factor importante que servirá de ayuda a la hora de extraer conclusiones sobre el funcionamiento del modelo, es la obtención de los valores medios tanto del coste mínimo, como del tiempo de computación de cada uno de los ocho modelos distintos.

Mejores resultados del estudio			
Tamaño	Repetición	Valor	Tiempo
50	Sí	108.1264	8.94
50	No	108.2384	12.40
100	Sí	110.0429	10.23
100	No	108.6116	11.07
150	Sí	109.5050	9.67
150	No	112.8156	11.00
200	Sí	109.5155	13.33
200	No	107.1431	13.55

Así como los mejores resultados de cada modelo, estas dos tablas se utilizan para eliminar la parte de aleatoriedad de los algoritmos, ya que si se ejecuta únicamente una vez cada modelo, cabría la posibilidad de que la mejor versión otorgue una solución peor o al contrario.

Resultados del total de soluciones por modelo		
Tamaño	Valor	Tiempo
2500	112.3553	4.97
5000	111.7150	5.31
7500	109.8018	5.86
10000	108.7675	6.77

Además, las distintas versiones del algoritmo se van a comparar con búsquedas aleatorias puras. Así pues, la versión que utiliza un tamaño de población de 50 individuos, con o sin repetición de índices, que es ejecutado durante 50 iteraciones se va a comparar con el resultado de generar aleatoriamente un total de 2500 soluciones y quedarse con la mejor. En la tabla se muestra el resultado de las 4 búsquedas aleatorias llevadas a cabo.

Este proyecto tenía la intención de mostrar que un algoritmo genético aplicado a un problema de localización de plantas con ubicaciones desconocidas contaba con la posibilidad de señalar buenos resultados y funcionar correctamente, habiendo conseguido el objetivo del trabajo.

En primer lugar, una de las premisas principales del proyecto, era mostrar si era estrictamente necesario restringir la repetición o no de índices dentro de cada individuo de una población.

Observando las tablas de resultados medios y de mejores resultados, en referencia a los tiempos de computación de cada variación distinta, se observa una gran diferencia entre estos. La totalidad de las variaciones donde se obliga a restringir la repetición y esta no es permitida, tienen un tiempo de computación mayor, tanto en tiempos medios como en tiempos mejores.

Esto es debido a que al existir aproximadamente un 25 % de individuos que tienen repeticiones, los métodos de la no repetición tienen que corregir esta incidencia y supone un gasto más en tiempo en comparación con los modelos de repetición.

Además, el código donde se permite la repetición es unas 100 líneas más corto por iteración, hecho que también se traduce en un gasto menor del tiempo de computación en comparación con los modelos donde no se permite la repetición.

En cuanto a los valores obtenidos de coste mínimo, excepto en el caso del tamaño de la población de 150, tanto en las tablas de resultados medios como en las de resultados mejores, los modelos que no permiten la repetición, obtienen mejores resultados que los que sí que la permiten.

Esto se debe, a que normalmente aquellos individuos que no presentan genes repetidos en su cromosoma, suelen mostrar valores de adecuación mejores que a los que sí que les ocurre esta incidencia. Por tanto, si un 25 % de la población presenta peores resultados en los modelos de repetición, existen mayores probabilidades de obtener mejores resultados en aquellos modelos que restrinjan este hecho.

Otro de los hechos relevantes a destacar, es el de la aleatoriedad, como se muestra en la excepción del caso del tamaño de 150 de la población, al final un algoritmo cuenta con un componente aleatorio grande, y en ocasiones el azar da resultados poco esperados como que una población que evalúa 2500 o 5000 individuos más que otra con el mismo modelo, como es el caso de la variación 150 con repetición, no pueda obtener un valor de adecuación mejor que las otras, como es observable en la tabla de mejores resultados del estudio.

En cuanto a qué tamaño de la población sería el idóneo para trabajar con este algoritmo, es necesario observar las tablas de resultados medios y de mejores resultados del estudio, donde se muestran que tamaños de la población de 50 o 100 individuos señalan valores casi similares al caso de 200 individuos con tiempos de computación notablemente inferiores.

Además, en la tabla de resultados medios se observa como únicamente el caso de no repetición con un tamaño de la población de 200 individuos muestra un valor inferior a 110

unidades de coste, pero a su vez, es el único modelo que roza los 15 segundos en tiempo de computación. Mientras que en la tabla de mejores resultados del estudio, este caso también es el que mejor valor representa de las 24 pruebas realizadas con un valor cercano a 107 unidades de coste. Por tanto no existe un tamaño óptimo y depende de la persona decisora decidir que características prefiere que sean más eficientes si el tiempo o los resultados.

En último lugar, la última tabla sirve de ayuda para demostrar como el algoritmo funciona correctamente y la importancia que este representa. El procedimiento ha sido el de generar tantos individuos de manera aleatoria como han sido evaluados en cada variación del tamaño poblacional y sin aplicarle el algoritmo genético y sus correspondientes procesos de cruce y/o mutación determinar el mejor.

Se observa como cuanto mayor es el número de individuos evaluados, mejor son los resultados obtenidos en términos de valor de coste y a su vez, mayores son los tiempos de computación. Además, en comparación con la tabla de mejores resultados del estudio, estos no son comparables, obteniendo resultados notablemente mejores con la utilización del algoritmo genético, demostrando la eficiencia del algoritmo desarrollado.

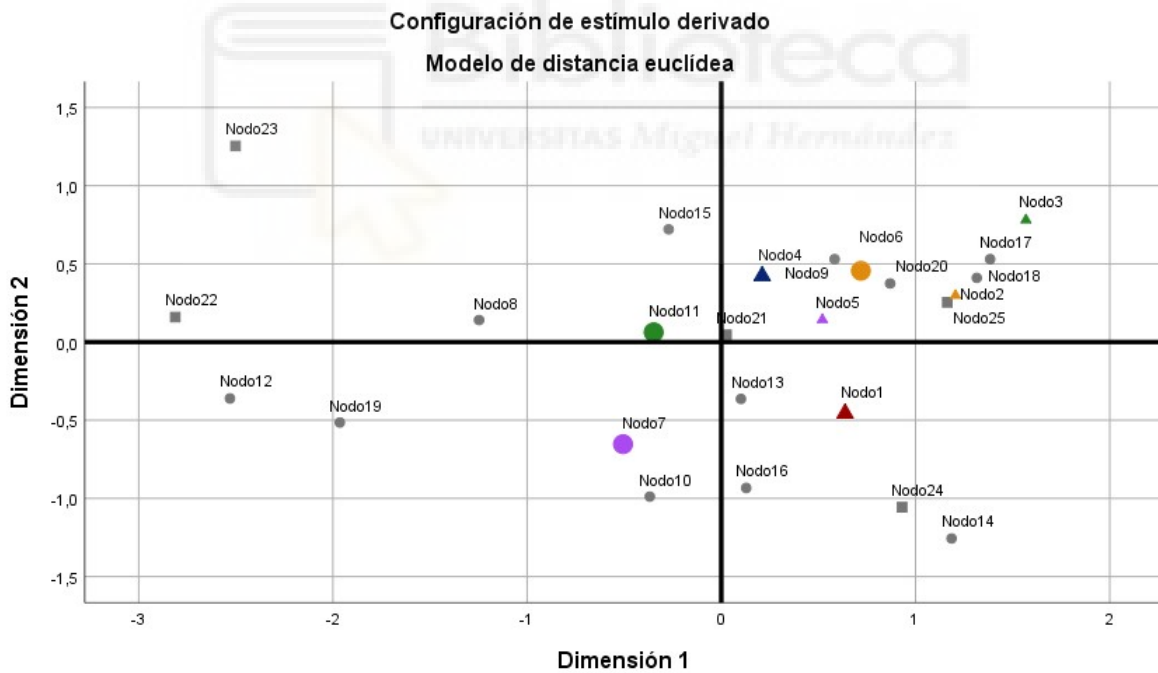


Figura 8: Disposición de los nodos correspondiente a la solución de la prueba 1 del modelo con repetición con un tamaño poblacional de 50 individuos. Fuente: Elaboración propia

Con la ayuda del programa estadístico SPSS de IBM, se ha conseguido situar a la totalidad de los nodos que participan en el modelo a través del escalamiento multidimensional, una técnica de análisis de variables cualitativas que analiza matrices cuadradas de similitudes o

disimilitudes. En este proyecto, se trabaja con una matriz de costes que de igual manera está relacionada con la cercanía entre los distintos nodos, por lo que se puede utilizar como una matriz de distancias, es decir, una matriz de disimilitudes.

De esta forma, se consigue conocer la ubicación exacta de cada una de las ubicaciones que participan en el modelo y se dispone de un enfoque visual que sirve de ayuda para entender porque una solución representada por un individuo es mejor que otras.

La Figura 8 representa la disposición de los nodos de la primera línea de la tabla de resultados del estudio que representa un valor en la función objetivo de 108.12. Las cinco ubicaciones que aumentan de tamaño en comparación con el resto, son aquellas por las cuales las cinco plantas distribuyen su producto. Si una de las plantas es la que está aumentada de tamaño como ocurre con el nodo 1 o el nodo 4, es porque sirven su producto en su misma ubicación. Además, los colores ayudan a identificar que nodo activado corresponde a que planta, sobretodo en los casos en los que una planta envía su producto a un cliente o intermediario.

En el caso del nodo 2, representado con un tono amarillento, se muestra claramente como se relaciona con el cliente 20, el nodo 3, con un tono verdoso, con el cliente 11 y el nodo 5, con un tono violeta, con el cliente 7. En último lugar, como se ha hecho referencia en la totalidad del trabajo las ubicaciones representadas con triángulos hacen referencia a plantas, las representadas con círculos a los clientes y las dibujadas mediante cuadrados a lugares que pueden participar como intermediarios.

Esta solución es representada a través de la siguiente cadena (1,20,11,4,7) y es una de las mejores soluciones que se han obtenido a lo largo de la obtención de resultados por parte del algoritmo. Una de las razones es, porque muestra un ecosistema muy equilibrado al repartir de una manera muy simétrica cada uno de los 16 clientes. Cabe recordar, que cada cliente acude a aquella ubicación activada en función de la cercanía, por lo que el cliente 7 debería recibir a los clientes que se encuentran en el sector inferior izquierdo, el nodo 1 los del sector inferior derecho, y el resto de nodos se encuentran con el grueso mayor de clientes y cada uno se ocupa de un número muy similar al del resto.

5. Conclusiones

En este capítulo, se realiza un breve resumen de las aportaciones del trabajo, haciendo énfasis en aquellos aspectos que han resultado más interesantes en el estudio realizado. El objetivo de este proyecto, era el de mostrar como un algoritmo genético podía ser capaz de al implementarse en un problema de localización de plantas móviles proporcionar buenos resultados y funcionar correctamente.

Una de las condiciones principales del modelo que debían ser probadas, era la de mostrar si al restringir la repetición o no de índices dentro de la cadena de cada individuo, el método arrojaría mejores resultados y como influiría esto en el tiempo de computación. Esta demostración fue probada, ya que aquellas pruebas realizadas con modelos donde la repetición no era posible de realizar obtenía unos tiempos de computación mayores a aquellos modelos en los cuales este hecho se erradicaba.

Esto se debe a que aquellos modelos que prohíben la repetición de índices cuentan con un tamaño de código mayor al tener que evaluar que este hecho no ocurra en ningún individuo y corregirlo así es. En cuanto a los resultados obtenidos de valores de la función objetivo en cada prueba, los modelos que no permiten la repetición, sí que cuentan con mejores resultados que los que sí la permiten.

El suceso de que los individuos sin repetición muestren mejores valores que los individuos con repetición, se debe a que cuantas más vías de acceso a las instalaciones existan, más facilidades se les das a los clientes para encontrar una ruta más eficiente para ellos, en cambio, si solo se permitiera acudir a 16 clientes a dos o tres plantas es seguro que el valor de adecuación sería mayor en términos de coste.

Otro de los hechos relevantes a destacar es el del gran componente aleatorio que conlleva un algoritmo de este tipo, lo que lleva a que a veces se obtengan resultados no esperados como que el evaluar más individuos de golpe proporcione resultados peores en cuanto a costes mínimo, por ello la necesidad de realizar tres pruebas por modelo ha ayudado a evitar este problema.

El número de iteraciones es otro de los temas a analizar, este tipo de algoritmos suelen utilizar 50 o 100 iteraciones, sin embargo no ha sido necesaria la implementación de tantas porque el algoritmo se encuentra en dificultades de poder mejorar su valor cuando ya han corrido varias iteraciones, ocurriendo en ocasiones que el método no puede mejorar después de las primeras.

Por ello, se ha optado por la vía de utilización de únicamente 50 iteraciones, ya que en escasas ocasiones se encuentran mejores resultados en las últimas iteraciones, y con este número de evaluaciones se consigue un tiempo de computación constante durante los 24 experimentos donde se mantiene un tiempo de computación constante.

En referencia al tamaño de la población, no existe un tamaño óptimo, sí que es cierto que entre la restricción de la repetición o no, se ha de implementar la repetición, porque maximiza las posibilidades de obtener mejores resultados que es el objetivo final del proyecto. Sin embar-

go, en el caso del tamaño de la población a determinar, existen variaciones muy pequeñas en cuantos a los valores de coste y variaciones más grande en cuanto a los tiempo de computación.

Por tanto, recae sobre el decisor hacer uso de su poder y determinar que es preferible en su caso, si obtener resultados levemente mejores, tiempos notablemente inferiores, o lo que suele ser más habitual, realizar una combinación de ambos balanceando de manera correcta los parámetros.



Referencias

- [1] J.PUERTO y M.MUÑOZ, E.CONDE, E.CARRIZOSA, *Lectura en Teorías de Localización*, Universidad de Sevilla, 1996.
- [2] M.J. CANÓS y M. MOCHOLÍ, *Un algoritmo para el cálculo del conjunto dominante finito del problema generalizado de la p -centdiana*, Universidad de Valencia, 2005.
- [3] M. TAGHAVI y H. SAHVANDI, *El problema p -centro bajo incertidumbre*, Universidad Tecnológica de Irán, 2012.
- [4] M.C. MOLERO y R. BLANQUERO, E. CARRIZOSA, *Problemas de Localización Competitiva. El modelo de Huff*, Universidad de Sevilla, 2016.
- [5] R. HALPER, S. RAGHAVAN y M. SAHIN, *Búsqueda local de heurísticos para el problema de localización de plantas móvil*, Elsevier, 2014.
- [6] S. RAGHAVAN, S. SALMAN y M. SAHIN, *El problema de localización de plantas móvil con capacidades*, Elsevier, 2018.
- [7] A. VIDAL y M.L. CARPENTE , B. CASAS, *Algoritmos heurísticos en optimización*, Universidad de Santiago de Compostela, 2013.
- [8] A.C. RIOJAS *Conceptos, algoritmo y aplicación al problema de las N -reinas*, Universidad Nacional Mayor de San Marcos, 2005.
- [9] F. HERRERA *Introducción a los Algoritmos Metaheurísticos*, Universidad de Granada, 2009.
- [10] F. HERRERA *Algoritmos Bioinspirados*, Universidad de Granada, 2013.
- [11] R. MARTÍ Y M. LAGUNA *Scatter search: Diseño básico y estrategias avanzadas*, Universidad de Valencia, 2003.
- [12] J.E. BEASLEY *OR-Library: Location p -hub*, Operational Research Society, 2018.