

Práctica Final: Biogeography-based optimization (BBO)

Juan Antonio Martínez Sánchez

Curso 2021/22

Metaheurísticas

?abstractname?

En este documento comentaremos la adaptación del algoritmo Biogeography-based optimization (BBO) al problema de la Mínima Dispersión Diferencial (MDD) visto en prácticas.

?contentsname?

1	Introducción	3
2	Biogeography-based optimization (BBO)	4
2.1	Introducción al BBO	4
2.2	Exploración y Exploración del Ecosistema	5
2.3	Resumen la implementación	5
2.4	Hibridación del BBO con BL	5
3	Comparativa de resultados	6

1 Introducción

El objetivo de esta práctica es estudiar una metaheurística no vista durante este curso con la cual la adaptaremos al Problema de la Mínima Dispersión Diferencial (MDD) de las prácticas anteriores.

Se utilizarán 50 casos seleccionados de varios de los conjuntos de instancias todas disponibles en la [MDPLIB](#), pertenecientes al grupo GDK-b con distancias aleatorias reales con, n entre $\{25, 50, 75, 100, 125, 150\}$, y m entre 2 y 45 (GDK-bGKD-b_1_n25_m2.txt a GKD-b_50_n150_m45.txt)

El Biogeography-based optimization (BBO) será nuestra metaheurística a estudiar. Veremos sus componentes y su funcionamiento, y una vez entendamos su filosofía de optimización explicaremos la adaptación de este a nuestro problema.

2 Biogeography-based optimization (BBO)

2.1 Introducción al BBO

El Biogeography-based optimization (BBO) es un algoritmo evolutivo que optimiza una función mejorando estocásticamente e iterativamente las soluciones candidatas con respecto a una determinada medida de calidad o función de aptitud.

BBO se usa típicamente para optimizar funciones multidimensionales de valor real, pero no usa el gradiente de la función, lo que significa que no requiere que la función sea diferenciable como lo requieren los métodos de optimización clásicos como el descenso de gradiente y los métodos cuasi-newton. Por lo tanto, BBO se puede utilizar en funciones discontinuas.

BBO optimiza un problema manteniendo una población de soluciones candidatas y creando nuevas soluciones candidatas combinando las existentes de acuerdo con una fórmula simple. De esta forma, la función objetivo se trata como una caja negra que simplemente proporciona una medida de calidad dada una solución candidata y no se necesita el gradiente de la función.

El BBO fue motivado por la biogeografía, que es el estudio de la distribución de especies biológicas a través del tiempo y el espacio. BBO fue presentado originalmente por Dan Simon en 2008.

A continuación, se hace una revisión de las componentes del modelo de biogeografía y su correspondencia con el algoritmo BBO desde el punto de vista de la optimización:

- **Hábitat:** Representa una solución del problema compuesta por una cadena de 0 y 1 de tamaño n y con exactamente m 1 que nos indicará los puntos seleccionados de la solución que vendrán dados por la posición que ocupan de la cadena.
- **Índice de Idoneidad del Habitat (HSI):** En un modelo biogeográfico, tener un alto HSI indica que son áreas geográficas bien acondicionadas, en este caso práctico significará que una solución con más HSI será una solución que produzca una menor dispersión.
- **Ecosistema:** Es un conjunto de n habitats. Desde el punto de vista de la optimización basada en poblaciones, se corresponde a la población de soluciones, este tamaño será constante.
- **Índice de inmigración:** Será el parametro λ que controlará la inmigración de un hábitat.
- **Índice de migración:** Será el parametro μ que controlará la emigración de un hábitat.
- **Modelo Migratorio:** Cada habitat del ecosistema tendrá asociados los índices de inmigración y de emigración con los cuales se evaluará si los componentes seleccionados en el habitat como solución son más óptimos para emigrar a soluciones peores (μ alto), o si la solución es adecuada para que imigren componentes de habitats mejores (λ alto).
- **Elitismo:** En cada generación del ecosistema los mejores habitats que mantendrán en el ecosistema por los peores habitats al finalizar la generación actual.

Una vez visto las componentes principales podemos pasar a comentar la exploración y explotación de soluciones en el ecosistema.

2.2 Exploración y Exploración del Ecosistema

El BBO a diferencia de los algoritmos evolutivos vistos en la práctica 2, carece de operadores de cruce y es aquí donde podemos diferenciarlo del resto. Hasta ahora hemos visto algoritmos que exploran un vecindario de soluciones y se encuentran una mejor la actualizaban y la explotaban para mejorarla. Ahora la exploración y la explotación se basan en los índices de inmigración y emigración de los habitats del ecosistema en el cual las mejores soluciones serán más propensas a migrar componentes a peores soluciones.

Este proceso dará un equilibrio al ecosistema donde las mejores soluciones ayudarán a las peores a mejorar como ellas para así tener cada vez mejores habitats. Sin embargo, añadiremos un operador de mutación con una probabilidad baja, para que no nos estancemos en las mismas soluciones, ya que introduciendo una mutación aleatoria al habitat podemos dar lugar a una solución mejor por esto es un factor importante. Un inconveniente que podemos destacar que aunque una componente de un habitat prometedor migre a otro habitat menos prometedor no nos asegura que este mejore y por tanto se puede dar que el habitat empeore.

2.3 Resumen la implementación

Inicialmente generaremos un ecosistema inicial de tamaño 50 compuesto de habitats de cadenas de booleanos de tamaño n y con exactamente m 1. Los índices de inmigración (λ) y de emigración (μ) se calcularán en base al tamaño del ecosistema donde λ será $1 - \mu$ para cada habitat con la relación de que siempre $\lambda + \mu = 1$. Tendremos un 5% de habitats elites, que serán el 5% de las mejores soluciones para mantener elitismo en cada generación.

Antes de iniciar la primera generación del ecosistema, lo ordenaremos de manera que las primeras soluciones serán las mejores (μ más alto) y las últimas las peores (λ más alto).

En cada generación recorreremos todo el ecosistema y aleatoriamente migraremos o no componentes al habitat actual. Esto vendrá dado por un número aleatorio entre 0 y 1 y se tendrá que cumplir que el λ del habitat sea mayor o igual a este número y así contra mayor índice de inmigración, menor será la probabilidad de empeorar las mejores soluciones. Una vez el habitat sea aceptado aleatoriamente para que migren hacia él, cogeremos aleatoriamente por el método Roulette Wheel (RW), un habitat del cual emigrarán componentes prometedoras. El RW en función de μ cogerá aleatoriamente una solución prometedora. Después de esto actualizaremos el habitat en el ecosistema y su dispersión. A la par migramos también aleatoriamente mutaremos soluciones o no. La mutación consiste en coger una solución dada con unos puntos seleccionados, intercambiar uno de estos por otro punto que no este seleccionado y actualizar el habitat.

Finalmente, ordenaremos el ecosistema y las peores soluciones las cambiaremos por las soluciones elites escogidas al comienzo de la generación y ya manteniendo el elitismo pasaremos de generación.

2.4 Hibridación del BBO con BL

Como se pedía en la práctica he introducido un algoritmo memético que hibridiza el BBO con la Búsqueda Local vista en la práctica 1. Este algoritmo memético está basado en el AM-(10, 1.0) visto en la práctica 2 y que he adaptado para esta práctica. Esta hibridación consiste en realizar 20 generaciones de ecosistemas donde cada generación es pasada al algoritmo BBO y cada 5 generaciones al BL.

3 Comparativa de resultados

Tras adaptar e implementar toda la idea del BBO nos queda aplicarle los mismos 50 casos que aplicamos a los algoritmo de las prácticas anteriores y así poder compararlos.

Algoritmo	Desv	Tiempo
Gready	66,537926258761	0,002293429016
BL	56,710723675577	0,019550993792
AGG-uniforme	48,255379081004	1,84162086
AGG-posicion	55,644894909961	1,64951558
AGE-uniforme	59,074721715184	1,43318966
AGE-posicion	67,793194192169	1,23076892
AM-(10, 1.0)	48,475101230593	0,32171192
AM-(10, 0.1)	54,621593826528	0,63516988
AM-(10, 0.1mej)	52,105787374991	0,64134536
ES	40,33419921395	0,075851557
BMB	36,835742803665	0,17037027956
ILS	39,155412831162	0,15700306784
ILS-ES	40,266445389518	0,3026681576
BBO	51,025299074043	0,98738792
AM-(BBO+BL)	48,508468511468	1,71769808

?figurename? 1: Resultados

Como podemos ver el BBO obtiene mejores resultados que BL y mejora a la mayoría de algoritmos genéticos. No obtiene mejores resultados que el Enfriamiento Simulado ya que el BBO es más probable de que caiga en óptimos locales y el ES no. Si nos fijamos en la hibridación mejoramos los resultados obtenidos del BBO individuales.

Podemos hablar del tiempo de ejecución medio del algoritmo que no llega a ser muy alto pero es algo que controlé en la implementación porque obtenía mejores resultados que los mostrados pero el coste de tiempo era muy alto y preferí tiempos pequeños y algo de mejora. Para controlar el tiempo disminuía el número de generaciones a explorar, el tamaño del ecosistema, las probabilidades de migración y de mutación y el porcentaje de soluciones elites para el elitismo. Puedo mencionar un ejemplo con el caso 50, donde aumenté el número de generaciones y cambié algunos parametros más de los mencionados, conseguí aproximadamente 200 menos de dispersión pero con un coste de 30 segundos a como lo tengo en la implementación que tarda 1,5 segundos.