

Capítulo 8

Algoritmo de murciélago



8.1 Introducción

El algoritmo de murciélago es una técnica innovadora o basada en la población que pertenece a la inteligencia del enjambre. También se lo conoce como un algoritmo metaheurístico desarrollado por Yang [1]. El algoritmo bat, como algoritmo único, proporciona una técnica de solución adecuada que los numerosos y predominantes algoritmos clásicos y heurísticos. El algoritmo se utiliza para una rápida toma de decisiones y para resolver problemas complejos en diversos campos de operaciones que van desde la ingeniería, los negocios, el transporte y otros campos de la actividad humana. Es importante comprender el patrón de comunicación y navegación de los murciélagos mientras se define el algoritmo, ya que el algoritmo se basa en la ecolocalización de micro murciélagos (EL) [1]. EL es un encantador y cautivador sonar producido por la propensión a los murciélagos. La atractiva ola de sonido que hacen los murciélagos es una gran fuerza que a menudo exhiben mientras buscan presas. La ola de sonido es una fórmula que adoptan no solo en la búsqueda de alimentos, sino que sirve para otros fines [2]. Por ejemplo, mientras busca comida en un ambiente completamente oscuro, puede haber obstáculos o peligros en su camino hacia la fuente de comida. Esto se puede detectar fácilmente de alguna manera mágica, ya que pueden sentir y discernir un posible peligro en su camino hacia la fuente de alimento, como se muestra en la Fig.8.1 [3]. La estructura y la posición de vuelo de un murciélago se muestran en la Fig.8.1. Por lo tanto, BA es un algoritmo muy poderoso, ya que utiliza la metodología de sintonización de frecuencia para la intensificación del rango de las soluciones en la población. Al mismo tiempo, el sistema BA implementa el procedimiento instintivo de disparar o hacer zoom para mantener la estabilidad durante el proceso de búsqueda o exploración de alimentos. Al desarrollar el BA, el patrón de búsqueda de dirección, manipulación y exploración durante la caza de presas se tiene en cuenta al imitar las disparidades en términos de tasas de emisión de pulso e intensidad del sonido liberado por los murciélagos mientras cazan o buscan presas. El algoritmo demuestra un alto nivel de eficiencia con un proceso de inicio rápido distintivo (Fig.8.2).



Figura 8.1 Estructura y posición de vuelo de un murciélago.

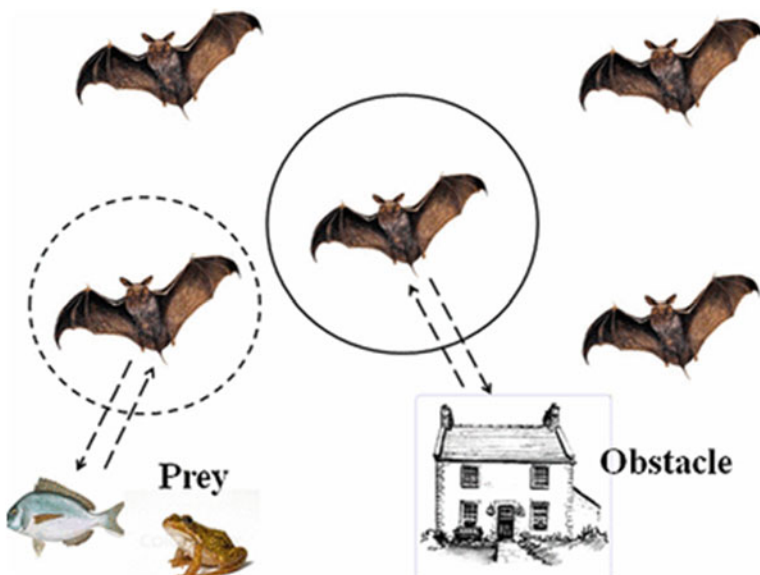


Figura 8.2 Patrón de comportamiento y navegación del murciélago

8.2 Reglas idealizadas del algoritmo de murciélago

Es muy importante tener en cuenta las reglas asociadas con los murciélagos. Dado que los murciélagos emiten fuertes ondas de sonido ultrasónicas y escuchan el eco que se refleja en los objetos circundantes, el algoritmo de murciélagos utiliza algunas reglas idealizadas para la sencillez. La siguiente sección señala algunas reglas comunes del algoritmo bat [1]:

- Todos los murciélagos emplean el patrón de ecolocalización para percibir, detectar o discernir presas, barreras, distancias, depredadores o cualquier forma de obstáculo durante la búsqueda de dirección, la ruta o la navegación por senderos. Este patrón de navegación se realiza de una forma muy mágica, extraña al hombre.
- Los murciélagos, en el curso de búsqueda de dirección o navegación de ruta, vuelan al azar con una supuesta velocidad, v_i manteniendo la posición, x_i . Mantienen una frecuencia fija 'f' con diferentes longitudes de onda y volumen, A_i para alcanzar su presa. Pueden ajustar la frecuencia de emisión de pulsos, r_i .
- A medida que se acercan a la presa, el pulso aumenta y el volumen disminuye.

El diagrama de flujo que describe el algoritmo bat se muestra en la Fig. 8.3.

8.3 Ejemplo de implementación del algoritmo BAT

Consideremos 3 murciélagos, la población de murciélagos = 3 ($i = 1, 2, 3$), X_1 , X_2 y X_3 representan respectivamente la posición de los tres murciélagos.

Pasos de la implementación del algoritmo BAT.

- Inicialización de la frecuencia (F_i), la población de murciélagos (X_i), el volumen (A_i), el pulso (r_i) y la velocidad (v_i) con ($i = 1, 2, 3, \dots$.norte).
Inicialmente ($t = 0$), asignaremos valores aleatorios a la frecuencia (F_i), posición (X_i), sonoridad (A_i), Pulso (r_i) y velocidad (v_i) de la siguiente manera (Tabla 8.1):
- Mientras $t < \text{Número máximo de iteraciones}$ ($t = 1, 2, 3, 4, 5, \dots$), se generarán nuevas soluciones ajustando frecuencias, velocidades y ubicaciones.

MTD 1: Suponga que consideramos 20 iteraciones para este ejemplo. Por tanto, para la primera iteración ($t = 1$), la condición es verdadera. Por lo tanto, se generarán nuevas soluciones ajustando la frecuencia y actualizando la velocidad y la posición. La siguiente ecuación se utilizará para ajustar la frecuencia (NOTA: BAT puede ajustar automáticamente la frecuencia).

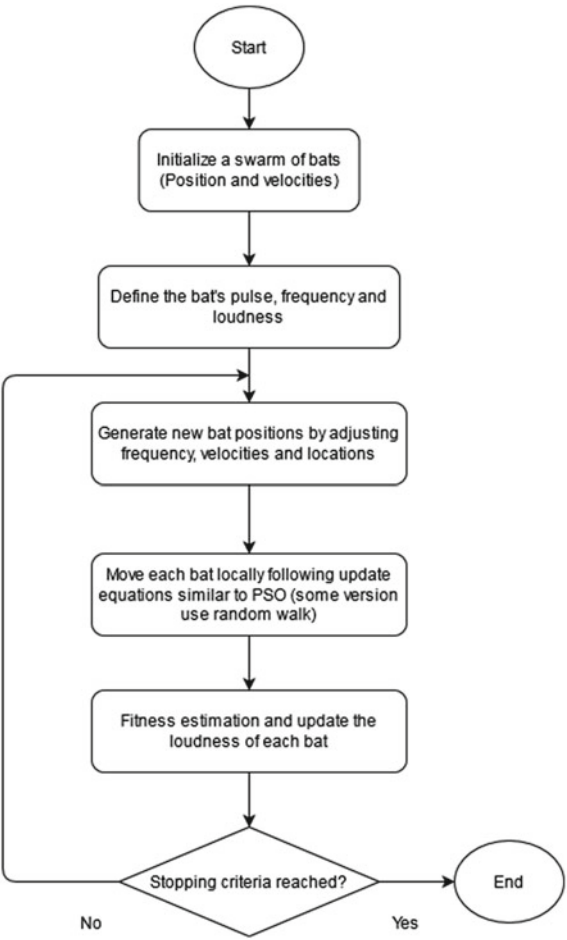
$$F_i = f_{\min} + \beta(F_{\max} - f_{\min}) \quad (8,1)$$

con el tamaño del dominio del problema $[f_{\min}, f_{\max}] = [0, 10]$ y $\beta \in [0, 1]$

$$v_i = v_i \cdot \left(\frac{I - X_t}{I - X_{t_{\text{gbest}}}} \right) + F_i \quad (8,2)$$

donde $x_{t_{\text{gbest}}}$ es la mejor posición actual (más cercana).

Figura 8.3 Diagrama de flujo del algoritmo bat [1]



Cuadro 8.1 Parámetros iniciales de BAT

| BAT 1 | BAT 2 | BAT 3 |
|-----------|-------------|-------------|
| $F_0 = 2$ | $F_0 = 6$ | $F_0 = 3$ |
| $X_0 = 4$ | $X_0 = 8$ | $X_0 = 5$ |
| $v_0 = 3$ | $v_0 = 7$ | $v_0 = 4$ |
| $A_0 = 1$ | $A_0 = 0.8$ | $A_0 = 0.9$ |
| $r_0 = 0$ | $r_0 = 0.2$ | $r_0 = 0.1$ |

$$X_t = x_{t-1} + v_t$$

(8,3)

- Si el valor aleatorio $\text{rand} > r_t$, se puede seleccionar una solución entre las mejores. Se puede generar una solución local en torno a la mejor solución.

Cuadro 8.2 Parámetros BAT actualizados

| BAT 1 | MTD2 | MTD3 |
|---------------------------|---------------------------|---------------------------|
| $F_1 = 2 + 1(8 - 2) = 8$ | $F_2 = 3 + 1(7 - 3) = 7$ | $F_3 = 1 + (5 - 1) = 5$ |
| $X_1 = 4 + 35 = 39$ | $X_2 = 8 + 63 = 71$ | $X_3 = 5 + 29 = 34$ |
| $V_1 = 3 + (4 - 0)8 = 35$ | $V_2 = 7 + (8 - 0)7 = 63$ | $V_3 = 4 + (5 - 0)5 = 29$ |
| $A_1 = 1$ | $A_2 = 0.8$ | $A_3 = 0.9$ |
| $r_1 = 0$ | $r_2 = 0.2$ | $r_3 = 0.1$ |

Para los tres murciélagos, r_1 , r_2 y r_3 son respectivamente iguales a 0, 0,2 y 0,1. Elegiremos el valor aleatorio entre [0, 1], digamos 0,5. Para BAT 1, $0.5 > 0$: VERDADERO, para BAT 2, $0.5 > 0.2$: VERDADERO y para BAT 3, $0.5 > 0.1$: VERDADERO. Si la condición es verdadera, seleccionaremos una solución entre las mejores.

¿Cómo seleccionar la mejor solución? Según la frecuencia: Mejor solución = si la presa está en el alcance del aumento de frecuencia. La frecuencia de las ondas aumenta si el BAT encuentra la presa (es decir, la mejor solución). Refiriéndose a la tabla 8.2, la frecuencia más alta corresponde a BAT 1, que es la mejor solución en este caso.

Podemos generar una solución local en torno a la mejor solución. Actualizaremos la posición del primer BAT utilizando la siguiente ecuación:

$$X_{\text{nuevo}} = X_{\text{viejo}} + \varepsilon A_t \quad (8,4)$$

con $\varepsilon \in [-1, 1]$

$$X_{\text{nuevo}} = 35 + 1.1 = 36$$

- Si $\text{rand} < A_t \text{ y } f(X_t) < F(X_*)$ aceptar la solución y aumentar r_t y reducir A_t (a medida que el BAT se acerca a la presa, el volumen disminuye y la frecuencia del pulso aumenta). $F(X_t) \text{ y } f(X_*)$ dependen de la nueva ubicación de las MTD y de la mejor solución según la función. Si la solución es VERDADERA, aumentaremos r_t y reducir A_t usando las siguientes ecuaciones:

$$r_{t+1} = r_t \left[1 - e^{-\gamma t} \right] \quad (8,5)$$

$$A_{t+1} = \alpha A_t \quad (8,6)$$

donde los dos constantes α y γ están en el intervalo [0, 1].

Asumamos $\alpha = \gamma = 0.9$, los valores actualizados son:

| BAT 1 | MTD2 | MTD3 |
|----------------------|----------------------|----------------------|
| F ₁ = 8 | F ₂ = 7 | F ₃ = 5 |
| X ₁ = 36 | X ₂ = 71 | X ₃ = 34 |
| v ₁ = 39 | v ₂ = 63 | v ₃ = 29 |
| A ₁ = 1 | A ₂ = 0.8 | A ₃ = 0.9 |
| r ₁ = 0.4 | r ₂ = 0.2 | r ₃ = 0.1 |

- Ahora podemos clasificar el BAT y encontrar el mejor actual. Primero comprobaremos la sonoridad y la frecuencia. Basado en eso, el primer BAT tiene el mayor volumen y frecuencia. Es la mejor solución actual.

8.4 Aplicación del algoritmo de optimización BAT con un ejemplo numérico

El código AMATLAB se proporciona en el Apéndice para ilustrar la implementación del algoritmo BAT para un problema de optimización típico considerando una ecuación numérica $F(x) = (x_1 + x_2 - 17)^2 + (2x_1 + x_2 - 4)^2$ describiendo el Wayburen función.

El modelo considera 10 murciélagos y el número máximo de iteraciones fue 1000. La mejor solución obtenida por BAT es: [1.3138 1.8528 0.261 0.83905 0.34859 1.1436 0.73859 1.7623 0.16537 0.28717]. El mejor valor óptimo de la función objetivo encontrado por BAT es: 0.23604. El espacio de parámetros y el espacio objetivo se muestran respectivamente en las Figs.8.4 y 8.5. El código MATLAB se proporciona en el Apéndice F. Se proporciona una descripción detallada del CÓDIGO en la Ref. [1].

Figura 8.4 Espacio de parámetros

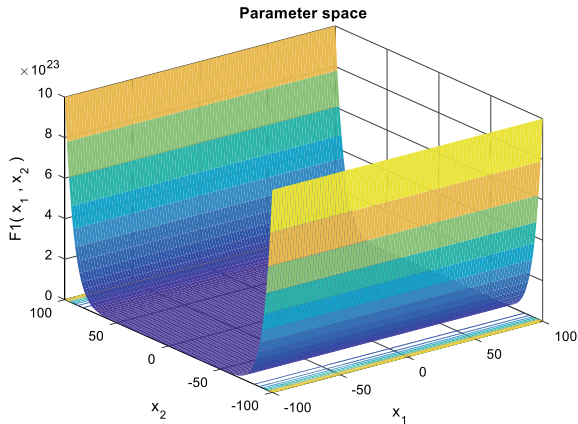
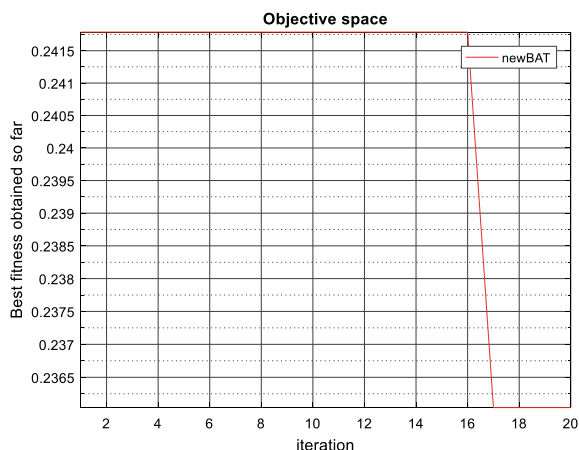


Figura 8.5 Espacio objetivo

8.5 Aplicación del algoritmo de optimización BAT para problemas de congestión de tráfico

De los casos

El tráfico congestionado puede causar varios problemas. La congestión se debe principalmente al uso intensivo de vehículos. La congestión del tráfico en el estado de Delta es un gran desafío, especialmente en las horas de la mañana y la noche. Las personas que se desplazan al lugar de trabajo en las horas de la mañana (8 a. M.) Se encuentran con esta congestión de tráfico recurrente, lo que genera retrasos, un alto consumo de combustible, contaminación ambiental y otros costos operativos. Además, regresar a casa después del trabajo (4 pm) también podría ser un desafío. Este problema es común en ciertas ubicaciones del estado (unión PP en DSC y unión Mofor). Otro lugar con un alto nivel de congestión de tráfico es la carretera Udu al cruce de Enerhen. Desde la observación, las carreteras son muy estrechas para permitir un flujo de tráfico ininterrumpido.

Observación

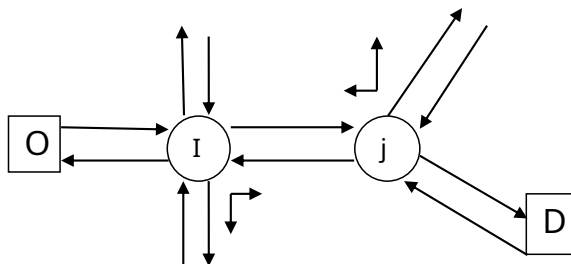
Se observó que el sistema de transporte en el estado estaba mal planificado, ya que no se trazaron rutas alternativas para facilitar el acceso y para congestionar aún más la carretera, los comercializadores y vendedores de carreteras ocupan la gran extensión de la carretera. Todo esto condujo a la congestión del tráfico en una sección del estado en circunstancias no señalizadas.

Red de arquitectura

Para abordar el problema, la red de carreteras se puede tomar como un gráfico dirigido $G = (N, a)$, donde 'N' es el conjunto de nodos; es decir, los cruces de carreteras 'a' son los enlaces que conectan los cruces como se muestra en la Fig.8.6.

Para cada par de origen (O) y destino (D), (OD) significa una demanda de viaje no negativa, d_{rs} . La red de carreteras puede representar un gráfico altamente conectado, donde cada nodo "i" es accesible por otro nodo "j" siguiendo la ruta dirigida de la red.

Figura 8.6 Una intersección de la red que muestra un par OD conectado por una unión de 3 vías y una de 4 vías



'NORTE'. Se supone que los enlaces que conectan los nodos tienen una función de tiempo de viaje t_a , para la tasa de flujo asignada x_a .

Este estudio tiene como objetivo imitar el algoritmo bat para resolver problemas de congestión de tráfico en el estado.

Los principales objetivos incluyen:

- (i) elegir la ruta más corta posible para viajar desde el origen al destino mientras se mantiene un flujo de tráfico ininterrumpido;
- (ii) Reducir el retraso del tráfico en cada cruce en circunstancias no señalizadas.

El modelo de diseño de red continua se elige con restricciones presupuestarias para la expansión de la capacidad del enlace. Los objetivos son interdependientes y pueden formularse como un problema de dos niveles. El nivel superior es responsable de reducir el tiempo de viaje de los usuarios o viajeros de la vía asignados. El nivel inferior es el modelo de asignación de tráfico que estima el flujo de viajeros. El modelo se presenta claramente en la Fig.8.7.

Este modelo se puede formular matemáticamente como se muestra a continuación para ambas capas.

Función de nivel superior

$$\begin{aligned} \text{menta}(y) = & \sum_{a \in U} \sum_{a \in A(y)} X_a t_a(X_a) + D_a y_a \\ \text{Tal que, } & \sum C_a y_a \leq B \end{aligned} \quad (8,7)$$

donde A es el conjunto de todos los enlaces 'a' en la red 'N'.

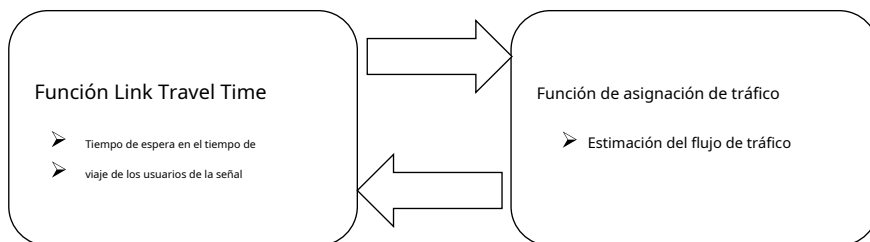


Figura 8.7 Modelo de dos niveles para el problema de determinación de la red de tráfico

X(a) da al usuario el flujo de equilibrio, que se estima a partir del nivel inferior del modelo para el valor asignado de la capacidad del enlace 'y'. C_a es el costo de construcción del enlace 'a' y B es el presupuesto.

Función de nivel inferior

$$\min \sum_{a \in A} X_a \int_0^{t_a(t_u)} t_a(t_u) du \quad (8,8)$$

Tal que, $\sum F_k = d_{ij} \quad F_k \geq 0$

$$\sum_{rs} F_k \delta_{ij} = x_a \quad \forall \delta_{ij} = \begin{cases} 1, & \text{if } a \in k \\ 0, & \text{otherwise} \end{cases}$$

donde f es el flujo en la ruta k y rs son los nodos en la ruta que conecta el par OD.

El diagrama de flujo para la optimización de la señal de tráfico desde una perspectiva BA se muestra en la Fig. 8.8.

Procedimiento de solución

Usando DeltaState, Udu road como un caso de estudio desde el Enerhen Junction hasta el Orhuwhorun Junction. Los nodos y enlaces se presentan en la Fig.8.9.

Notación y formulaciones de modelos

- No de enlaces, $a = 2$
- No de nodos / uniones, $N = 3$
- El número posible de redes en este caso se da como 2_{norte}
 $2_3 = 8$ número posible de redes
- Restricción presupuestaria, $B = 500.000$ NGN
- Se considera la función de nivel superior
- Primero, de acuerdo con el diagrama de flujo, se consideran los parámetros necesarios
- La distancia en coche es de 22 km.
- El tiempo que se tarda en navegar es 33 min.
- Asumiendo lo siguiente:

- $X_a = 0,67$ km / min
- $T_a = 33$ min
- Y_a (capacidad de enlace) = 50
- D_a (demanda) = 300
- $C_a = 50.000$ NGN por enlace

Darse cuenta de:

$$\sum C_a y_a \leq B$$

Figura 8.8 Diagrama de flujo para el problema de optimización de la señal de tráfico usando el algoritmo bat

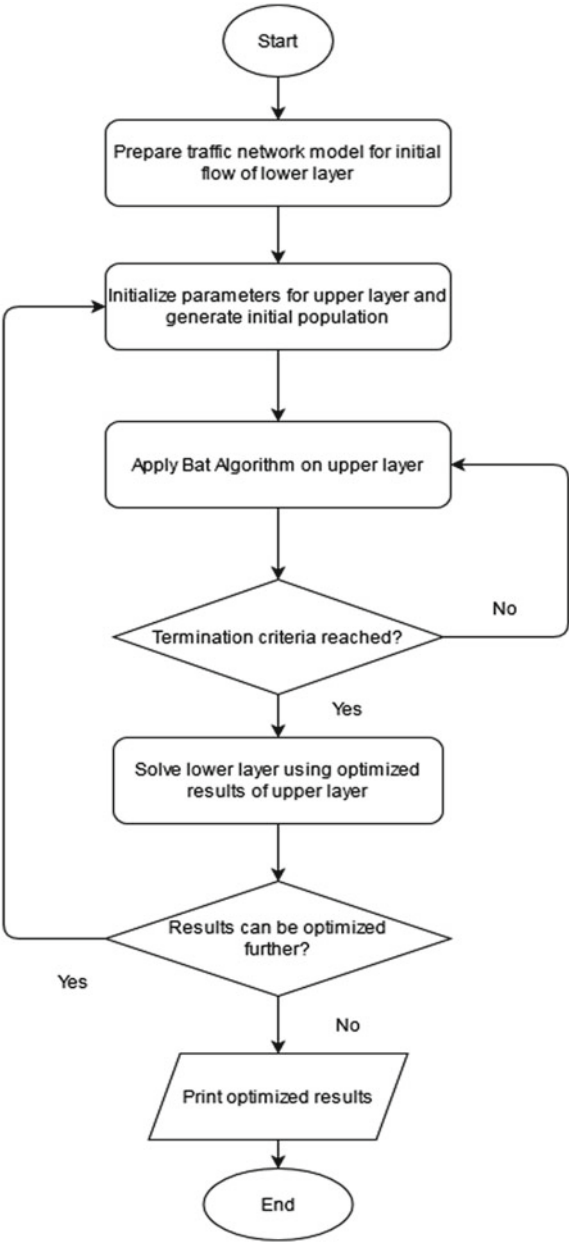
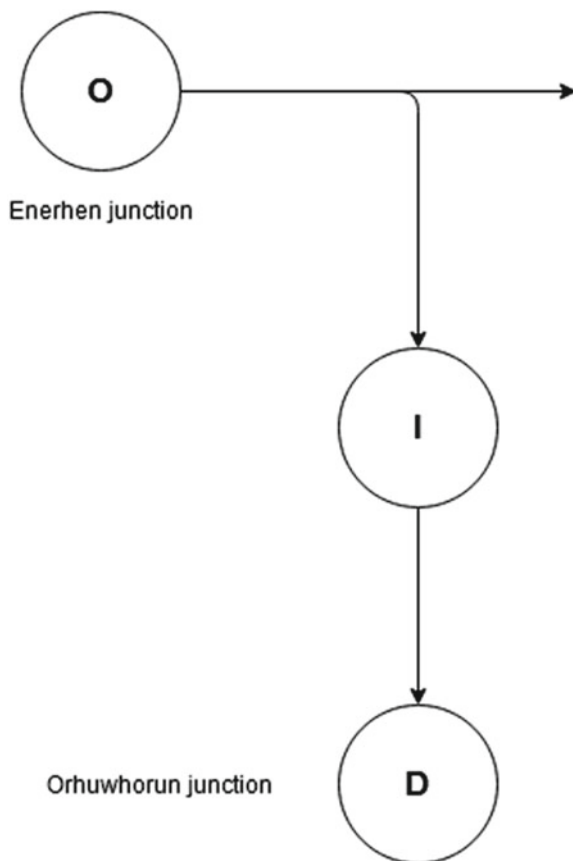


Figura 8.9 Red de carreteras para tres nodos y dos enlaces



$$\sum \text{N}50000 \times 50 = \text{N}2500000 \leq \text{N}4000000$$

Dado que se cumple la condición anterior, procedemos;

$$\begin{aligned} \text{menta}(y) &= \sum_{a \in U} \sum_{a \in A(y)} X_{aT} (X_a) + D_{aT} y \\ &\sum_{a \in U} \sum_{a \in A(y)} 0.67 \times 33 \times 0.67 + (300 \times 50) \end{aligned}$$

Esto da como mínimo $T(y) = 15.014,8137$.

El algoritmo de murciélago se aplica en la función de nivel superior. Definición de la función objetivo $f(x)$ y asignación de los parámetros.

- Población de murciélagos (demanda), $x_1 = 300$

- Velocidad, $v_i = 0,67 \text{ km / min}$
- Sonoridad, $u_i = 50.000$ (costo de construcción del enlace)
- Frecuencia de pulso, $r_i = 50$ (capacidad de enlace)
- Frecuencia, $f_i = 0,030$.

Método de solución

- Deje que los murciélagos estacionados para encontrar direcciones representen los vehículos disponibles en el espacio de búsqueda o la red de carreteras. La lectura de mapas o el trazado del curso se simplifican.
- Dado que se ha estimado que el número de vehículos que pasan por la carretera es de aproximadamente 300 periódicamente y la capacidad de enlace es de 50 por enlace, entonces la capacidad total para ambos enlaces es de $50 \times 2 = 100$.
- Las carreteras mapeadas para el estudio son tales que se pueden permitir 100 vehículos para una fácil navegación.
- 300 vehículos utilizan la carretera con frecuencia, se asigna un valor aleatorio a la frecuencia del pulso o la capacidad del enlace.
- Para la primera iteración, se supone que la capacidad del enlace o la frecuencia de pulsos es de 70 por enlace. Esto hace un total de 140 vehículos para ambos enlaces.
- Según la condición inicialmente definida, $\text{Rand} > r_i$; $70 > 50$ satisface la condición.
- Es importante generar una nueva solución en torno a la condición definida.
- Para lograr esto, suponga que la capacidad del enlace (frecuencia de pulso) es 75, lo que hace que la demanda de ambos enlaces sea igual a 150 vehículos.
- Dado que se ha aumentado la capacidad del enlace, se reducirá la tasa de flujo de tiempo. Para que t_a igual a 31 min; F_i igual a 0.032 y velocidad, v_i igual a 0,71 km / min. El costo de construcción (sonoridad) aumenta a 90,000 NGN.
- Para la segunda iteración, suponga que la capacidad del enlace es 100, por lo tanto, la demanda de ambos enlaces es igual a 200 vehículos.
- Recuerde la condición establecida, $\text{Rand} > r_i$; $100 > 50$ satisface la condición, es importante almacenar y generar una nueva solución alrededor de la condición indicada.
- Suponga que la capacidad del enlace es 105, haciendo que la demanda de ambos enlaces sea igual a 210.
- Una vez más, la tasa de flujo de tiempo aumentará automáticamente a medida que se reduzca la capacidad del enlace.
- Entonces, t_a es igual a 29 min; F_i es igual a 0.034 y v_i es igual a 0,76 km / min.
- El costo de construcción (sonoridad) aumenta a 100,000 NGN.
- Para la tercera iteración, deje que Rand sea igual a 140, haciendo que la demanda de ambos enlaces sea igual a 280 vehículos.
- Siguiendo la condición establecida, $\text{Rand} > r_i$; $140 > 50$ satisface la condición indicada. Es importante almacenar y generar una nueva solución en torno a la condición indicada.
- Teniendo en cuenta la capacidad del enlace de 145. La demanda de ambos enlaces se convierte en 290. Con el aumento de la capacidad del enlace, la tasa de flujo de tiempo aumentará en $t_a = 27 \text{ min}$; $F_i = 0.037$ y $v_i = 0,81 \text{ km / min}$. El costo de construcción (sonoridad) es igual a 150.000 NGN.

- Finalmente, para la última iteración, Rand es igual a 155 y la demanda de ambos enlaces es igual a 310. Dado que $Rand > r_i$; $155 > 50$. Dado que se cumple la condición, se genera una nueva solución.
- Con una frecuencia de pulso de 160, la demanda de ambos enlaces se convierte en 320 vehículos. El tiempo de flujo aumenta a 25 min, en f_i de 0,040, y v_i igual a 0,88 km / min. El costo de construcción (sonoridad) aumenta a 200,000 NGN.

Calculando la aptitud de la población generada, la tabla se presenta así:

| Población / demanda | Enlaces de capacidad |
|---------------------|----------------------|
| 100 | 50 |
| 150 | 75 |
| 210 | 105 |
| 290 | 145 |
| 320 | 155 |

Comparando la información obtenida utilizando la condición establecida así:

- $Rand < a_i$ y $f(x_i) < f(x)$
- $a_i = 50.000$ y $f(x_i) = 0,030$.

Para la primera iteración, la sonoridad aleatoria fue ₦90.000 que no satisface la condición ya que es mayor ₦50.000. Pero, $f(x_i)$, 0.030 es menor que $f(x)$ que son 0.032, 0.034, 0.037 y 0.040, por lo que los resultados se muestran y clasifican en la siguiente tabla.

| Población / demanda | Capacidad por enlace | Velocidad (v_i) (km / min) | Tiempo (t_a) (min) | Frecuencia (f_i) |
|---------------------|----------------------|-----------------------------------|------------------------|----------------------|
| 100 | 50 | 0,67 | 33 | 0,030 |
| 150 | 75 | 0,71 | 31 | 0.032 |
| 210 | 105 | 0,76 | 29 | 0.034 |
| 290 | 145 | 0,81 | 27 | 0,037 |
| 320 | 155 | 0,88 | 25 | 0.040 |

En esta etapa, se alcanzan los criterios de terminación. El resultado óptimo se genera al crear una vía con la capacidad de enlace de 155 vehículos para que un total de 320 vehículos puedan transitar libremente, desde el origen hasta el destino a una velocidad más rápida, 0,88 km / min y por un tiempo menor, 25 min. La construcción de esta carretera solo costará 200.000 NGN por enlace, sumando una cantidad total de ₦400.000, lo cual está claramente dentro del presupuesto asignado.

Entonces, habiendo resuelto el problema del tiempo, la función de nivel inferior se puede obtener utilizando resultados optimizados

$$\sum_{a \in A} x_a \int_0^{t_a(t_u)} t_a(t_u) du$$

donde $T_a = 25$ min, $x_a = 0,88$ km / min.

Sustituyendo en la ecuación anterior,

$$\sum_0^{0.88} 25u \cdot du$$

Esto da como resultado un valor total de 9,68 que representa el flujo del viajero. El resultado no se puede optimizar más para que no se exceda el presupuesto de construcción.

Al imitar el algoritmo de los murciélagos y seguir el patrón de ecolocalización de los murciélagos para localizar fácilmente a las presas y evitar obstáculos, se ha resuelto un problema de la red de transporte en Delta State, Udu road. Se han considerado los siguientes parámetros: población de murciélagos (de los vehículos que pasan por las carreteras), velocidad, sonoridad (costo de construcción del enlace) y frecuencia de pulso (capacidad del enlace), correspondientes al comportamiento del murciélago. El problema del tiempo de viaje se ha optimizado para permitir un flujo de tráfico ininterrumpido en una situación sin señalización en la carretera Udu. Después de cuatro iteraciones, se obtuvo la mejor solución con una capacidad máxima de enlace de 160 para dar cabida a los vehículos que viajan por esa ruta a diario.

Referencias

1. Yang, XS 2010. Un nuevo algoritmo inspirado en murciélagos metaheurísticos. En *Estrategias cooperativas inspiradas en la naturaleza para la optimización (NICSO 2010)* 65–74. Berlín, Heidelberg: Springer.
2. Yang, XS 2011. Algoritmo Bat para optimización multiobjetivo. *Revista internacional de computación bioinspirada* 3 (5): 267–274.
3. Rizk-Allah, RM y A.E. Hassanien. 2018. Nuevo algoritmo de murciélago binario para resolver un problema de mochila 0-1. *Sistemas complejos e inteligentes* 4 (1): 31–53.

Capítulo 9

Algoritmo de optimización Ant Lion



9.1 Introducción

La optimización de hormigas león (ALO) es una técnica de optimización innovadora inspirada en la naturaleza y fue propuesta por primera vez por [1]. El algoritmo ALO como biomimetismo replica el patrón de caza de las hormigas leones en un escenario de la vida real. Seguir la naturaleza cazadora de una hormiga león para desarrollar el algoritmo es un proceso de cinco pasos. Ant lion tiene cinco estrategias principales para cazar presas. El primer proceso que suele implementar una hormiga león es la técnica de caminar al azar de las hormigas. El segundo tiene que ver con el proceso de construcción de trampas para presas. El tercer proceso es la técnica de picar y atrapar hormigas en trampas. El cuarto proceso tiene que ver con sujetar o atrapar presas y el último es la reconstrucción de trampas para cazar presas. Se han desarrollado ecuaciones matemáticas sólidas y sólidas del ALO imitando el comportamiento del ALO en la vida real. Esto es muy útil para resolver problemas en un área diversa de la vida humana. El ciclo de vida de la hormiga león se muestra en la Fig.9.1. Ant Lion Optimizer (ALO) se puede utilizar para resolver problemas NP-hard, problemas de optimización combinatoria como problemas de programación, problemas de transporte y transbordo y otros problemas de planificación de rutas. El algoritmo Ant lion es muy útil en operaciones combinatorias y estocásticas. Especialmente para: programación de máquinas paralelas, mantenimiento, programación de mantenimiento integrado rentable en sistemas de energía, dimensionamiento y ubicación óptimos de generación distribuida, problema de diseño de tren de engranajes, diseño de hélice de barco, problema de diseño de viga en voladizo, programación óptima de unidades térmicas para electricidad a pequeña escala problema de despacho de energía económica y muchas más áreas de operaciones.