## Reto algoritmos enjambre

Ejercicio voluntario (tiempo objetivo 3 semanas)

Objetivo del ejercicio: aplicar distintos algoritmos enjambre para el análisis de distribución de rendimientos de inversiones aleatorias en 24.000 fondos de inversión.

Se concederá hasta un máximo de 3 puntos extra en la práctica de algoritmos de inversión

El ejercicio deberá realizarse en python. Tendréis que documentar con suficiente detalle cómo planteáis la aplicación de algoritmos enjambre en el análisis de distribución de rendimientos aleatorios. No vale solo una propuesta, debe entregarse código ejecutable. Tanto la propuesta como el código estará en el script.

**Se valorará**

1. Diseño y evolución del entorno: deberás construir un entorno financiero coherente para la exploración por parte de los enjambres. Este punto es muy importante, por lo que deberás justificar el proceso de construcción con detalle. Tanto el entorno, como los enjambres, deben poder visualizarse durante su ejecución.
2. Adaptación de cada uno de los algoritmos enjambre al problema financiero: No vas a poder aplicar directamente ninguno de los algoritmos, por lo que tendrás que adaptarlos para poder resolver el problema.
3. La consecución del objetivo solicitado: debéis mostrar dos tablas con la distribución de las rentabilidades obtenidas por cada tipología de enjambre y proceso de optimización (como las tablas de abajo). Sin la tabla no se valorará el ejercicio.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Monos | Global | Hormigas | Abejas | Partículas | Enfriamiento | | 0% | 0,13933684 | 0,96231998 | 0,61213551 | 0,2696555 | 0,12828039 | | 10% | 0,24400691 | 0,0536884 | 0,0048859 | 0,0506488 | 0,20549354 | | 20% | 0,40198583 | 0,11695722 | 0,06776552 | 0,11544442 | 0,35699873 | | 30% | 0,54152434 | 0,18722177 | 0,06928791 | 0,16005605 | 0,38239474 | | 40% | 0,54494871 | 0,19779197 | 0,40333086 | 0,17217564 | 0,47581741 | | 50% | 0,5760797 | 0,23987933 | 0,41063137 | 0,21211029 | 0,64923014 | | 60% | 0,63604852 | 0,42709617 | 0,44972604 | 0,32431681 | 0,65312552 | | 70% | 0,65587064 | 0,55570666 | 0,45025704 | 0,36681138 | 0,696021 | | 80% | 0,783724 | 0,66962364 | 0,45128863 | 0,43919074 | 0,79031067 | | 90% | 0,84318171 | 0,67241269 | 0,4981956 | 0,45283626 | 0,81652288 | | 95% | 0,85432879 | 0,72884727 | 0,54217769 | 0,54316786 | 0,87040301 | | 99% | 0,86555517 | 0,7524975 | 0,63998639 | 0,57744331 | 0,90712736 | | 100% | 0,87395172 | 0,86433734 | 0,84471223 | 0,99807287 | 0,92603968 | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Monos | Global | Max rent | Min pdas | Max Sharpe | Max Alpha | | 0% | 0,13933684 | 0,010860797 | 0,216853707 | 0,19027236 | 0,052124541 | | 10% | 0,24400691 | 0,015015675 | 0,291663965 | 0,247124039 | 0,510174316 | | 20% | 0,40198583 | 0,088127094 | 0,365233872 | 0,297758004 | 0,537554243 | | 30% | 0,54152434 | 0,104331866 | 0,431380187 | 0,413590428 | 0,558589153 | | 40% | 0,54494871 | 0,17855059 | 0,545659806 | 0,480028057 | 0,624969229 | | 50% | 0,5760797 | 0,337803479 | 0,577321447 | 0,587608846 | 0,655324895 | | 60% | 0,63604852 | 0,386595496 | 0,578746875 | 0,642337852 | 0,673454096 | | 70% | 0,65587064 | 0,432952599 | 0,58808505 | 0,660607468 | 0,685947049 | | 80% | 0,783724 | 0,501875514 | 0,626205109 | 0,711930644 | 0,791135059 | | 90% | 0,84318171 | 0,66953266 | 0,667840761 | 0,774913937 | 0,840077637 | | 95% | 0,85432879 | 0,727852926 | 0,691315644 | 0,829128061 | 0,861666527 | | 99% | 0,86555517 | 0,747507568 | 0,759559502 | 0,83199145 | 0,923517593 | | 100% | 0,87395172 | 0,831033737 | 0,903369578 | 0,861438084 | 0,999948214 | |

1. El número de monos que utilizarás en cada uno de los algoritmos enjambre para realizar la optimización: En cada algoritmo enjambre hay un rango óptimo: justifica el tamaño.
2. El tiempo de ejecución: debéis conocer el tiempo de ejecución (es decir, debe ser ejecutable, aunque dure varias horas). Más de 24h no es admisible.
3. Sistema de asignación de habilidades a los monos y balanceo de clases: cada mono tendrá asignado una tipología de enjambre, un proceso de optimización de carteras y una asignación a un entorno (ver más abajo). ¿Cómo realizas este proceso?, ¿cómo balanceas la asignación para que los resultados sean coherentes? Detalla el proceso de asignación.

**Preguntas que debes hacerte, resolver y argumentar con detalle al inicio de tu script:**

* Los activos no tienen los mismos días cotizados, ¿cómo los homogeneizamos?, ¿con cuántos activos nos quedamos y por qué?
* Tenemos muchas preguntas que hacernos respecto al entorno. De su correcta construcción dependerá que los enjambres puedan explorarlo y obtengamos resultados coherentes o no.
  + Tenemos 24.000 fondos de inversión. Ya no tenemos un laberinto, con sus pasillos, paredes, hormiguero y comida.

|  |  |
| --- | --- |
| ¿Cómo vas a traducir los 24.000 fondos es un entorno explorable?, ¿cómo vas a colocar los fondos? |  |
| Podríamos enfocar el problema colocando los fondos en un entorno 3D. Pero tendríamos que seguir resolviendo el problema de la colocación, de cara a optimizar la exploración del entorno. |  |

Orientación: para optimizar el proceso de exploración podríamos utilizar más de un entorno y más de un algoritmo de exploración / enjambre (ver más adelante). Podríamos generar N entornos, ordenados con distintos criterios, para garantizar el proceso de exploración. Lo que nos lleva a la siguiente pregunta.

* + Generación de entornos y colocación de activos: ¿Cuántos entornos vamos a generar?, ¿cómo colocaremos los activos?, ¿qué información pondremos en cada celda? Podríamos generar entornos ordenados por:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| * + - Moneda / país     - Categoría del fondo     - Tamaño     - Gestión (activa, pasiva)     - Correlación     - Ratings     - Composición     - Volatilidad | * + - Máximo drawdown     - Tiempo de recuperación     - Tracking error     - Omega     - Calmar     - Sortino     - Information ratio     - Beta |  |  |  |  |
|  |  |  |  |

Por desgracia no contarás con toda esta información. Deberás argumentar cómo resolver este problema, cuantos entornos generas y porqué. Cómo colocas los activos en cada entorno, y porqué. Quizás en Morningstar podrías obtener más información.

<https://www.morningstar.es/es/screener/fund.aspx#?filtersSelectedValue=%7B%7D&page=1&sortField=legalName&sortOrder=asc>

* + Evolución del entorno: El laberinto no era un entorno cambiante. Pero la información de los fondos sí que lo será: navs, rendimiento, volatilidad, correlación… ¿Cómo vas a hacer que la información sea dinámica durante la exploración? Orientación:
    - Cada X movimientos
    - Cada X tiempo: rebalanceando, por ejemplo, cada 6 meses.
* Respecto a la construcción de los monos, también tenemos preguntas a resolver.
  + Número de monos y puesta en producción: durante la optimización que realizamos en el reto de “dotar de habilidades a los monos” trabajamos con 100 millones de monos. ¿Con cuántos monos deberíamos trabajar en nuestro sistema multi-enjambre, con N entornos a explorar? Si el número de monos fuera muy elevado, ¿cómo lo pondrías en producción?
  + ¿Cómo asignarás cada mono a cada entorno?, ¿y cómo determinarás su posición inicial?
  + Movimiento: ¿Cómo se moverá el mono a lo largo del entorno?, ¿qué algoritmo utilizarás? Ahora no tenemos un laberinto que resolver, una comida que encontrar, ni un hormiguero al que volver. Orientación: Podría ser interesante implementar varios algoritmos enjambre y asignar a cada mono, al nacer, uno de estos algoritmos.
    - Algoritmo de la colonia de hormigas (ver anexo)
    - Algoritmo de la colonia de abejas (ver anexo)
    - Algoritmo del enfriamiento del metal (ver anexo)
    - Algoritmo del enjambre de partículas (ver anexo)
  + Número de activos: ¿Con cuántos activos trabajará cada mono?, ¿será un número constante durante toda su vida? Argumenta la respuesta.
  + Criterio de selección: Cada mono va a moverse por su entorno, encontrando constantemente distintos fondos de inversión. Ya no hay una única “comida” que encontrar. ¿Cuál va a ser el criterio de selección de cada mono? Orientación: cada mono deberá tener un criterio de selección / optimización de cartera, que le será asignado al nacer.
    - Criterios de cartera
      * Maximizar la rentabilidad de capital
      * Minimizar las pérdidas que podamos tener
      * Maximizar un estadístico rentabilidad/riesgo
      * Replicar una referencia
      * Batir una referencia
      * Distribución periódica de rentas
      * Obtener rentabilidad positiva en cualquier entorno de mercado
    - Criterios de activo (complementan a los criterios de cartera, cada mono deberá tener un % de esta tipología en su cartera, aleatorio, entre el 5% y el 20%)
      * Preferencia por algunos activos exóticos: descorrelacionados (por encima de un umbral), o de una temática en concreto.
      * Activos populares (elegidos por un número relevante de otros monos)
      * Minimicen las comisiones globales de la cartera
    - Ventana: Cada mono tendrá una ventana distinta con la que realizará los cálculos para optimizar el criterio de selección. La ventana se asignará al mono al nacer y será aleatoria entre: 30, 60 y 90 días.
  + Criterio de asignación: Una vez has seleccionado el activo, o los activos de la cartera, ¿cómo vas a realizar la asignación de recursos (capital disponible)? Orientación: optimiza (maximiza o minimiza, depende del caso) el criterio de selección de cada mono.
  + Criterio de rebalanceo: Desde el instante inicial (en tiempo y posición), el mono explorará el entorno, generando una cartera óptima. ¿Cuál va a ser el criterio de rebalanceo de cada mono? Orientación: Podrías emplear cualquiera de los siguientes criterios (argumenta tu elección).
    - Cada X tiempo de ejecución.
    - Cada vez que encuentre una cartera mejor (con un umbral de mejora).
    - Con cada evolución del entorno.
* Respecto a los algoritmos enjambre, tenemos una cuestión muy importante que resolver: ¿Cómo vas a adaptar cada algoritmo?
  + Algoritmo de la colonia de hormigas: el algoritmo se basaba en encontrar la comida y volver al hormiguero dejando feromonas. Ahora no hay “una comida”, ni un hormiguero al que volver. La feromona no podrá guiar de igual manera al resto de hormigas. Tendrás que adaptar el algoritmo.
  + Algoritmo de la colonia de abejas: probablemente es el algoritmo más fácil de adaptar al problema. Cuentas con abejas obreras y exploradoras. Sin embargo, deberás definir correctamente la “zona de baile” y cómo las abejas obreras deciden a qué zona acudir para sacar el máximo rendimiento al algoritmo.
  + Algoritmo del enfriamiento del metal: Este algoritmo no está pensado para este tipo de optimización financiera, por lo que tendrás que adaptarlo. Definir bien el sistema de exploración de vecinos, así como la definición de vecino, será crucial.
  + Algoritmo del enjambre de partículas: ¿Qué es un mono?, ¿una partícula, o un enjambre de partículas? Hay muchos parámetros que optimizar y adaptar en este algoritmo, especialmente para la exploración. El cómo hayas colocado los activos en el entorno será clave.

## Anexo

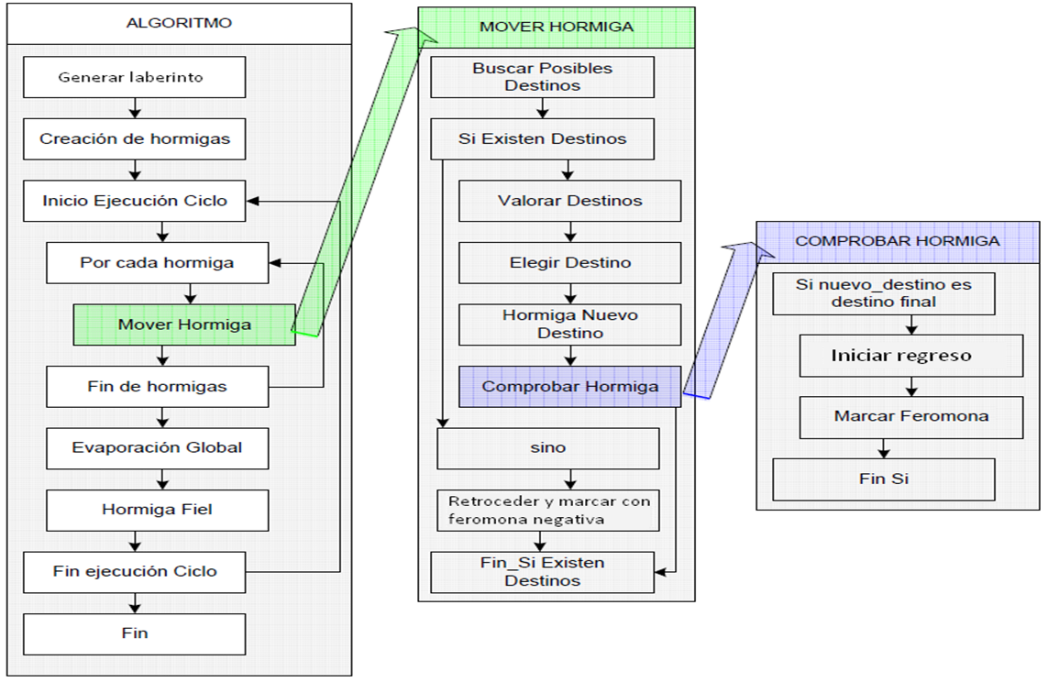
**Algoritmo de la colonia de hormigas (ACO, visto en clase)**

Las hormigas se comunican mediante feromonas depositadas en el suelo. Las exploradoras se desplazan aleatoriamente por el entorno, cuando encuentran comida, vuelven al hormiguero por el mismo camino dejando un rastro de feromonas (las hormigas recuerdan perfectamente cada paso que han realizado).

Las demás exploradoras que encuentran la feromona tienen tendencia a seguirla (la probabilidad depende de la cantidad de feromona depositada).

Las feromonas se evaporan de manera natural.

Siguiendo este sistema, las hormigas son capaces de determinar el camino más corto entre la colonia y el alimento.



**Algoritmo de la colonia de abejas (ABC)**

Las abejas en busca de alimento tratan de localizar la región del espacio con mayor densidad de flores, ya que es allí donde presumiblemente existe más cantidad de polen. Cada abeja vuela de modo errático por el espacio, recordando en todo momento cuál es la región donde ha visto más flores. A su vez, el enjambre sabe colectivamente cuál es la región del espacio, de entre todas las exploradas, donde se han encontrado más flores. Cada abeja variará individualmente su movimiento con arreglo a estas dos direcciones, volando hacia algún lugar intermedio (vecino). Es posible que la abeja durante ese sobrevuelo encuentre una región con mayor densidad de flores que la conocida hasta entonces (óptimo local). Pasado un tiempo, si se descubre otra región con mayor densidad floral, el enjambre reorientará nuevamente la búsqueda hacia allí, y así sucesivamente.

El algoritmo define una colmena artificial formada por una zona de comunicación (o zona de baile) y dos tipos de abejas

Exploradoras: Marcan zonas con abundancia de flores

Obreras: Deciden ir a una zona marcada por una exploradora para buscar alimento.

Las abejas obreras acuden a zonas con abundancia de alimento y bailan en esa área para marcarla. Las abejas obreras deciden a qué fuente de alimento acudir en función de los bailes de las exploradoras (recordando los sitios marcados anteriormente). Cuando el alimento se agota en la zona, la abeja obrera baila para marcarlo como agotado.

Los pasos principales del algoritmo se exponen a continuación

|  |
| --- |
| Se inicializa la población de abejas  Se inicializan las fuentes de alimento  REPETIR  Las abejas exploradoras se desplazan buscando una fuente de alimento  Evalúa su cantidad de néctar y bailan para trasladar la información a la colmena  Cada abeja obrera analiza la danza, elige una de las fuentes y se dirige hacia allí.  Elige una flor de la zona (vecino), evalúa su cantidad de néctar, y decide si llevarlo a la colmena o pasar a otro vecino.  Se determina si el alimento de la zona está agotado.  Se registran / recuerdan las mejores fuentes de alimento encontradas hasta el momento.  Iteración = iteración + 1  Se cumple criterio de parada |

**Algoritmo de enfriamiento simulado (Simulated annealing o “recocido” simulado)**

Para evitar óptimos locales, este algoritmo permite cambiar a soluciones peores, en función de una probabilidad T (temperatura), que disminuye con el tiempo. La T0 se inicia con un valor alto y se enfría en cada iteración en función de α hasta alcanzar Tf.

Cada iteración evalúa un nº de vecinos L(T), el cual puede ser fijo para toda la ejecución o depender de la iteración (no analiza todos los vecinos).

Si el mejor vecino es mejor que la solución actual se adopta, en caso contrario, se podría cambiar en función de la diferencia entre la solución actual y la del vecino, y de la probabilidad asociada a la temperatura. Finalizada la iteración, se enfría la temperatura.

Prob cambiar = e^(- dif entre solución actual y vecina / temperatura). EJ 2,71^(-1/10) da una prob de 90,5% de cambiar, (-10/10) un 36,9%, (-10/1) 0,04%.

|  |  |
| --- | --- |
| Inicio  Se inicia la temperatura, genera la solución inicial.  Repetir  Repetir evaluar vecinos (la definición de vecino depende de ti)  Si la solución del vecino es mejor que la actual se adopta y se guarda como la mejor.  Si la solución es peor, se calcula la probabilidad de saltar.  Si aleatorio es > prob saltar, Se salta al mejor vecino  T = α \* T (se enfría la temperatura).  Hasta (la temperatura baje de un umbral mínimo y no haya ningún vecino mejor).  Devolver (mejor solución).  Fin | Valor inicial de la temperatura T0  Mecanismos de enfriamiento (alternativas)  Descenso cte  Descenso geométrico α entre 0,8 y 0,99  Condición de parada: Cuando T baja de un umbral prefijado o un nº fijo de iteraciones |

**Algoritmo de crecimiento bacteriano**

La evolución del estado de las células está determinada por el estado inicial y no necesita ninguna entrada de datos posterior. El "tablero de juego" es una malla (de dos, o más dimensiones) formada por células que se extiende en todas las direcciones.

En una malla de dos dimensiones, cada célula tiene 8 células vecinas, que son las que están próximas a ella, incluidas las diagonales. Las células tienen dos estados: están "vivas" o "muertas”. El estado de las células evoluciona a lo largo de unidades de tiempo discretas (se podría decir que por turnos). El estado de todas las células se tiene en cuenta para calcular el estado de las mismas al turno siguiente. Todas las células se actualizan simultáneamente en cada turno, siguiendo estas reglas:

* Una célula muerta con exactamente 3 células vecinas vivas "nace" (es decir, al turno siguiente estará viva).
* Una célula viva con 2 o 3 células vecinas vivas sigue viva
* En caso contrario muere: ya sea por "soledad" o por "superpoblación".

**Algoritmo de manada**

Partiendo de algunas reglas sencillas, es posible simular los comportamientos de manadas o de grupos.

Cada individuo tiene tres comportamientos:

* Un individuo muy cercano se separará para evitar invadir o chocar con otro individuo.
* Un individuo próximo modifica la dirección de movimiento, existiendo una tendencia a alinearse en la dirección del vecino.
* Un individuo a una distancia media provoca un acercamiento (como cuando un animal ve a otro de su misma especie).

Se busca, en primer lugar, evitar un muro (límite del entorno), a continuación, un obstáculo y por último otro individuo. Si no hay nada que evitar se pasa al comportamiento de alineamiento.

Es posible agregar un ángulo muerto para hacerlo lo más realista posible.

En función de las distancias configuradas, podemos observar individuos que se desplazan en manadas comportándose como un banco de peces o una bandada de pájaros, cuyas trayectorias parecen aleatorias, evitando chocarse entre sí y evitando a los depredadores.

**Algoritmo del enjambre de partículas (PSO)**

PSO permite optimizar un problema a partir de una población de soluciones candidatas, denotadas como "partículas", moviendo éstas por todo el espacio de búsqueda según reglas matemáticas que tienen en cuenta la posición y la velocidad de las partículas. El movimiento de cada partícula se ve influido por su mejor posición encontrada hasta el momento, así como por la mejor posición global encontrada por el enjambre.

Cada partícula consta de 4 elementos: una posición que representa una determinada combinación de valores de las variables, el valor de la función objetivo en la posición donde se encuentra la partícula, una velocidad que indica cómo y hacia donde se desplaza la partícula, y un registro de la mejor posición en la que ha estado la partícula hasta el momento.

|  |
| --- |
| Se inicializan el número de partículas con su posición y velocidad  Mientras no se cumpla el criterio de parada (maximización o minimización de la función)  Evaluar cada partícula con la función objetivo  Actualizar la posición y velocidad de cada partícula  Guardar mejor posición conocida de la partícula  Guardar mejor posición conocida global  Iteración = iteración + 1  Se cumple criterio de parada: devolver mejor posición global |

Crear partícula

Cada partícula está definida por una posición, velocidad y valor, los cuales varían a medida que la partícula se mueve. La partícula almacena la mejor posición en la que ha estado hasta el momento. Cuando se crea una nueva partícula, únicamente se dispone de información sobre su posición (aleatoria) y velocidad (normalmente iniciada como cero), el resto de valores no se conocen hasta que la partícula es evaluada.

Crear enjambre

La creación del enjambre consiste en la inicializar n nuevas partículas.

Evaluar partícula

Evaluar una partícula consiste en calcular el valor de la función objetivo en la posición que ocupa la partícula es ese momento. Cada partícula almacena también la posición con mejor valor en la que ha estado hasta el momento. Para poder identificar si una nueva posición es mejor que las anteriores, es necesario conocer si se trata de un problema de minimización o maximización.

Devuelve una nueva partícula en la que los campos valor, mejor\_valor y mejor\_posicion han sido actualizados.

Evaluar enjambre

Evaluar un enjambre consiste en calcular el valor de la función objetivo en la posición de cada una de las partículas que lo forman.

Mover partícula

Mover una partícula implica actualizar su velocidad y posición. Este paso es el más importante ya que otorga al algoritmo la capacidad de optimizar. La velocidad de cada partícula del enjambre se actualiza empleando la siguiente ecuación:



donde:

vi(t+1): velocidad de la partícula i en el momento t+1, es decir, la nueva velocidad.

vi(t): velocidad de la partícula i en el momento t, es decir, la velocidad actual.

w: coeficiente de inercia, reduce o aumenta a la velocidad de la partícula.

c1: coeficiente cognitivo responsable de que la partícula tienda a moverse hacia la mejor posición hasta el momento

r1: vector de valores aleatorios entre 0 y 1 (un valor por cada dimensión) que aporta comportamiento estocástico.

x^i(t): mejor posición en la que ha estado la partícula i hasta el momento.

xi(t): posición de la partícula i en el momento t.

c2: coeficiente social responsable de que la partícula tienda a moverse hacia la mejor posición global conocida.

r2: vector de valores aleatorios entre 0 y 1 (un valor por cada dimensión) que aporta comportamiento estocástico.

g(t): posición del mejor valor global en el momento t.

Para comprender como se relaciona esta ecuación con el movimiento de la partícula la diferenciamos en tres partes:

wvi(t) es la componente de inercia, responsable de mantener a la partícula moviéndose en la dirección en la que lo ha estado haciendo hasta el momento. El valor recomendado del coeficiente de inercia w suele ser entre 0.8 y 1.2. Si w<1, la partícula se va desacelerando a medida que avanzan las iteraciones, esto se traduce en menor exploración, pero una convergencia hacia el óptimo más rápida. Si w>1, la partícula se va acelerando, lo que permite explorar más zonas del espacio de la función, pero dificulta la convergencia.

c1r1[x^i(t)−xi(t)] es la componente cognitiva, responsable de que la partícula tienda a moverse hacia la posición donde ha obtenido mejores resultados hasta el momento. El coeficiente cognitivo c1 suele estar acotado en el rango [0, 2], siendo 2 el valor recomendado. r1 es un vector de valores aleatorios entre 0 y 1 (un valor por cada dimensión) que aporta cierto comportamiento estocástico al movimiento de las partículas, mejorando así la capacidad de escapar de mínimos locales.

c2r2[g(t)−xi(t)] es la componente social, responsable de que la partícula tienda a moverse hacia la mejor posición encontrada por el enjambre hasta el momento. Puede interpretarse como el “conocimiento colectivo”. El valor del coeficiente social c2 suele estar acotado en el rango [0, 2], siendo 2 el valor recomendado. r2 es un vector de valores aleatorios entre 0 y 1 (un valor por cada dimensión) que aporta cierto comportamiento estocástico al movimiento de las partículas, mejorando así la capacidad de escapar de mínimos locales.

La magnitud relativa entre la componente cognitiva y la componente social permite regular el comportamiento exploratorio del algoritmo. Cuanto mayor es el valor de c1 respecto a c2, mayor independencia de movimiento tiene cada partícula, lo que permite mayor exploración, pero mayor lentitud en la convergencia. Por el contrario, cuanto mayor es el valor de c2 respecto a c1, más obligadas están las partículas a moverse hacia la mejor región encontrada hasta el momento, lo que reduce la exploración, pero acelera la convergencia.

En algunas versiones del algoritmo, r1 y r2 son escalares en lugar de vectores. Multiplicar cada componente de la velocidad por un valor aleatorio distinto añade mayores fluctuaciones al movimiento de las partículas, lo que, aun a riesgo de retrasar la convergencia, suele generar mejores resultados.

Una vez calculada la nueva velocidad, se puede actualizar la posición de la partícula con la ecuación: xi(t+1)=xi(t)+vi(t+1)

Uno de los principales problemas del algoritmo PSO es que las partículas suelen adquirir velocidades excesivamente altas, lo que las lleva a salirse de los límites del espacio de búsqueda, o a que sean incapaces de converger en la región óptima. Algunas de las soluciones son:

Limitar la velocidad máxima que puede alcanzar una partícula. Siendo [xmin, xmax] los límites inferior y superior del espacio de búsqueda de cada variable, la velocidad máxima que puede alcanzar la partícula en esa dirección es vmax=k(xmax−xmin)/2, donde k suele ser un valor entre 0.1 y 1.

Si el valor de alguna de las variables excede los límites impuestos, se sobrescribe con el valor del límite correspondiente y se reinicia su velocidad a cero.

Reducción lineal del coeficiente de inercia w. Esta estrategia consiste en ir reduciendo el coeficiente de inercia a medida que avanzan las iteraciones. En las primeras iteraciones, las partículas tienen mucha capacidad de exploración y, a medida que avanza el proceso, va reduciéndose su velocidad favoreciendo la convergencia. Puede conseguirse este efecto con la ecuación: wt=(wmax−wmin)\*((tmax−t)/tmax)+wmin

donde:

wt: coeficiente de inercia en la iteración t.

wmax: coeficiente de inercia máximo. Valor con el que se inicia el algoritmo. Valor recomendado de 0.9.

wmin: coeficiente de inercia mínimo. Valor que se alcanza en la última iteración. Valor recomendado de 0.4.

tmax: número máximo de iteraciones.

Mover enjambre

Mover el enjambre consiste en actualizar la posición de cada una de las partículas que lo forman.

**Definiciones de ratios propuestos durante el enunciado**

* Máximo drawdown: Mayor caída que ha tenido el fondo en el plazo de estudio. Se expresa en porcentaje.
* Tiempo de recuperación: Mide el tiempo que ha tardado el fondo en recuperar el valor respecto al máximo drawdown.
* Tracking error: Mide lo que se desvía un fondo respecto del índice de referencia. Es importante para ver si el fondo es de gestión activa o pasiva. Cuanto más cercano a 0, más pasivo.
* Omega: Mide la rentabilidad por cada unidad de riesgo (volatilidad) que se asume. Un ratio cercano a 1, o superior, es muy bueno.
* Calmar: Variación del ratio de Sharpe. Utiliza el máximo drawdown en vez de la volatilidad como medida de riesgo. Mida la rentabilidad obtenida por cada unidad de riesgo máximo asumido. Cuando mayor, mejor.
* Sortino: Variación del ratio de Sharpe. Usa como medida de riesgo únicamente las desviaciones negativas en vez de la volatilidad. Mide la rentabilidad conseguida, por cada unidad de riesgo asumido, en las caídas.
* Information ratio: Mide si las desviaciones respecto al índice que conforman el tracking error aporta o destruyen valor. Cuanto mayor, mejor.
* Beta: Mide como es la intensidad de los movimientos del fondo respecto al índice. Por encima de 1 los amplifica. Por debajo de 1, los mitiga.

**Ayuda para la codificación del entorno y mejora de la comprensión de las variables**

Las categorias asset\_type no corresponden a una estructura jerarquica sobre 'asset' asi que hay que hacer un mapping directo sobre este campo

asset\_class = {

'ABR': 'alternative', 'ACF': 'alternative', 'ACU': 'alternative', 'AED': 'alternative', 'AGM': 'alternative', 'ALS': 'alternative',

'ALT': 'alternative', 'AMF': 'alternative', 'AMS': 'alternative', 'ARV': 'alternative', 'BAL': 'mixed', 'BIN': 'equity', 'BIO': 'equity',

'COM': 'other', 'CON': 'fixed\_income', 'COR': 'fixed\_income', 'COS': 'equity', 'DCO': 'equity', 'DEF': 'mixed', 'DRS': 'alternative',

'DYN': 'mixed', 'EM': 'fixed\_income', 'FIN': 'equity', 'FLE': 'mixed', 'GAR': 'fixed\_income', 'GOL': 'equity', 'GOV': 'fixed\_income',

'HDG': 'alternative', 'HEA': 'equity', 'HY': 'fixed\_income', 'IDT': 'equity', 'INF': 'fixed\_income', 'MX': 'equity', 'MMD': 'money',

'MMK': 'money', 'NEU': 'alternative', 'NAR': 'equity', 'RFG': 'fixed\_income', 'RFL': 'fixed\_income', 'RFM': 'fixed\_income', 'RFS': 'fixed\_income',

'RST': 'equity', 'RVA': 'equity', 'RVG': 'equity', 'SMD': 'equity', 'TEC': 'equity', 'TEL': 'equity', 'TMA': 'mixed', 'UTI': 'equity', 'OTH': 'mixed', 'OTS': 'mixed',

}

family = {

'ABR': 'retorno absoluto', 'ACF': 'renta variable ocde', 'ACU': 'renta variable ocde', 'AED': 'renta variable ocde', 'AGM': 'renta variable ocde',

'ALS': 'renta variable ocde', 'ALT': 'renta variable ocde', 'AMF': 'renta variable ocde', 'AMS': 'renta variable ocde', 'ARV': 'renta variable ocde',

'BAL': 'deuda corporativa grado de inversion', 'BIN': 'renta variable ocde', 'BIO': 'renta variable ocde', 'COM': 'commodities', 'CON': 'convertibles',

'COR': 'deuda corporativa grado de inversion', 'COS': 'renta variable ocde', 'DCO': 'renta variable ocde', 'DEF': 'deuda publica ocde', 'DRS': 'inmobiliario',

'DYN': 'renta variable emergentes', 'EM': 'bonos high yield', 'FIN': 'renta variable ocde', 'FLE': 'renta variable ocde', 'GAR': 'retorno absoluto',

'GOL': 'renta variable ocde', 'GOV': 'deuda publica ocde', 'HDG': 'retorno absoluto', 'HEA': 'renta variable ocde', 'HY': 'bonos high yield', 'IDT': 'renta variable ocde',

'INF': 'deuda corporativa grado de inversion', 'MMD': 'liquidez y renta fija a corto plazo', 'MMK': 'liquidez y renta fija a corto plazo', 'MX': 'renta variable emergentes',

'NAR': 'renta variable ocde', 'NEU': 'retorno absoluto', 'RFG': 'deuda corporativa grado de inversion', 'RFL': 'deuda corporativa grado de inversion',

'RFM': 'deuda corporativa grado de inversion', 'RFS': 'deuda corporativa grado de inversion', 'RST': 'renta variable ocde', 'RVA': 'ver zona geografica',

'RVG': 'ver zona geografica', 'SMD': 'ver zona geografica', 'TEC': 'renta variable ocde', 'TEL': 'renta variable ocde', 'TMA': 'deuda publica ocde',

'UTI': 'renta variable ocde', 'OTH': 'renta variable ocde', 'OTS': 'renta variable ocde',

}

subcategory = {

'ABR': 'alternativos\_liquidos', 'ACF': 'alternativos\_liquidos', 'ACU': 'alternativos\_liquidos', 'AED': 'alternativos\_liquidos', 'AGM': 'alternativos\_liquidos',

'ALS': 'alternativos\_liquidos', 'ALT': 'alternativos\_liquidos', 'AMF': 'alternativos\_liquidos', 'AMS': 'alternativos\_liquidos', 'ARV': 'alternativos\_liquidos',

'BAL': 'mixtos\_equilibrado', 'BIN': 'renta\_variable\_sectorial', 'BIO': 'renta\_variable\_sectorial', 'COM': 'otros', 'CON': 'renta\_fija\_convertibles',

'COR': 'renta\_fija\_largo\_plazo', 'COS': 'renta\_variable\_sectorial', 'DCO': 'renta\_variable\_sectorial', 'DEF': 'mixtos\_conservador', 'DRS': 'alternativos\_inmobiliario',

'DYN': 'mixtos\_agresivos', 'EM': 'renta\_fija\_emergente', 'FIN': 'renta\_variable\_sectorial', 'FLE': 'mixto\_flexible', 'GAR': 'renta\_fija\_largo\_plazo',

'GOL': 'renta\_variable\_sectorial', 'GOV': 'renta\_fija\_largo\_plazo', 'HDG': 'alternativos\_liquidos', 'HEA': 'renta\_variable\_sectorial', 'HY': 'renta\_fija\_high\_yield',

'IDT': 'renta\_variable\_sectorial', 'INF': 'renta\_fija\_largo\_plazo', 'MMD': 'monetario', 'MMK': 'monetario', 'MX': 'ver\_zona\_geografica',

'NAR': 'renta\_variable\_sectorial', 'NEU': 'alternativos\_liquidos', 'RFG': 'renta\_fija\_largo\_plazo', 'RFL': 'renta\_fija\_largo\_plazo',

'RFM': 'renta\_fija\_corto\_y\_medio\_plazo', 'RFS': 'renta\_fija\_corto\_y\_medio\_plazo', 'RST': 'renta\_variable\_sectorial', 'RVA': 'ver\_zona\_geografica',

'RVG': 'ver\_zona\_geografica', 'SMD': 'ver\_zona\_geografica', 'TEC': 'renta\_variable\_sectorial', 'TEL': 'renta\_variable\_sectorial', 'TMA': 'mixtos\_equilibrado',

'UTI': 'renta\_variable\_sectorial', 'OTH': 'mixto\_flexible', 'OTS': 'mixto\_flexible',

}

Hacemos también un mapping sobre la variable geo\_zone

full\_name = {

'BR': 'BRASIL', 'CA': 'CANADA', 'CAR': 'CARIBE', 'CL': 'CHILE', 'LAM': 'LATINOAMERICA MISC', 'LAT': 'LATINOAMERICA', 'MX': 'MEXICO', 'PAM': 'PANAMERICA',

'US': 'USA', 'ASE': 'ASIAN', 'ASP': 'ASIA PASIFICO', 'ATA': 'AUSTRALASIA', 'AU': 'AUSTRALIA', 'AXJ': 'ASIA EX JAPON', 'B': 'BAHREIN', 'CHI': 'CHINA',

'EAU': 'EMIRATOS ARABES UNIDOS', 'EGY': 'EGIPTO', 'EIJ': 'FAR EAST', 'EXJ': 'FAST EAST EX-JAPON', 'GCC': 'GCC', 'GCH': 'GREATER CHINA', 'HK': 'HONG KONG',

'IL': 'ISRAEL', 'IN': 'INDIA', 'IND': 'INDONESIA', 'JO': 'JORDANIA', 'JP': 'JAPON', 'KR': 'KOREA', 'KWT': 'KUWAIT', 'LE': 'LIBANO', 'MDE': 'ORIENTE MEDIO',

'MNA': 'MENA', 'MO': 'MARRUECOS', 'MSA': 'MALASIA', 'OM': 'OMAN', 'PH': 'FILIPINAS', 'SAU': 'ARABIA SAUDITA', 'SG': 'SINGAPUR',

'SYM': 'SINGAPUR Y MALASIA', 'TH': 'THAILANDIA', 'TU': 'TUNEZ', 'TW': 'TAIWAN', 'VNM': 'VIETNAM', 'AT': 'AUSTRIA', 'BA': 'PAISES BALTICOS', 'BE': 'BELGICA',

'BNL': 'BENELUX', 'CH': 'SUIZA', 'CZE': 'REPUBLICA CHECA', 'DE': 'ALEMANIA', 'DK': 'DINAMARCA', 'EE': 'EUROPA EMERGENTE',

'EES': 'EUROPA EXSUIZA', 'EMI': 'EUROPA MISC', 'ES': 'ESPAÑA', 'EU': 'EUROPA', 'EUK': 'EUROPA EX-UK', 'EUR': 'ZONA EURO', 'FI': 'FINLANDIA', 'FR': 'FRANCIA',

'GR': 'GRECIA', 'HUN': 'HUNGRIA', 'IBE': 'IBERICA', 'IE': 'IRLANDA', 'ISL': 'ISLANDIA', 'IT': 'ITALIA', 'LIE': 'LIETCHTENSTEIN', 'LUX': 'LUXEMBURGO',

'LU': 'LUXEMBURGO', 'MLT': 'MALTA', 'NL': 'PAISES BAJOS', 'NO': 'NORUEGA', 'NOR': 'PAISES NORDICOS', 'POL': 'POLONIA', 'PT': 'PORTUGAL', 'RS': 'RUSIA',

'SE': 'SUECIA', 'SKA': 'ESCANDINAVIA', 'SVK': 'ESLOVAQUIA', 'TR': 'TURQUIA', 'UK': 'REINO UNIDO', 'GB': 'REINO UNIDO', 'AFR': 'AFRICA', 'BRC': 'BRIC',

'GEA': 'GLOBAL EX AUSTRALIA', 'GEM': 'GLOBAL EMERGENTE', 'GEU': 'GLOBAL EX-US', 'GLB': 'GLOBAL', 'NZL': 'NUEVA ZELANDA', 'SAF': 'SUDAFRICA', 'EME': 'EMEA'

}

nested\_family = {

'BR': 'renta variable emergentes', 'CA': 'renta variable ocde', 'CAR': 'renta variable emergentes', 'CL': 'renta variable emergentes',

'LAM': 'renta variable emergentes', 'LAT': 'renta variable emergentes', 'MX': 'renta variable emergentes', 'PAM': 'renta variable emergentes',

'US': 'renta variable ocde', 'ASE': 'renta variable emergentes', 'ASP': 'renta variable emergentes', 'ATA': 'renta variable emergentes', 'AU': 'renta variable ocde',

'AXJ': 'renta variable emergentes', 'B': 'renta variable emergentes', 'CHI': 'renta variable emergentes', 'EAU': 'renta variable emergentes',

'EGY': 'renta variable emergentes', 'EIJ': 'renta variable emergentes', 'EXJ': 'renta variable emergentes', 'GCC': 'renta variable emergentes',

'GCH': 'renta variable emergentes', 'HK': 'renta variable emergentes', 'IL': 'renta variable emergentes', 'IN': 'renta variable emergentes',

'IND': 'renta variable emergentes', 'JO': 'renta variable emergentes', 'JP': 'renta variable emergentes', 'KR': 'renta variable emergentes',

'KWT': 'renta variable emergentes', 'LE': 'renta variable emergentes', 'MDE': 'renta variable emergentes', 'MNA': 'renta variable emergentes',

'MO': 'renta variable emergentes', 'MSA': 'renta variable emergentes', 'OM': 'renta variable emergentes', 'PH': 'renta variable emergentes',

'SAU': 'renta variable emergentes', 'SG': 'renta variable emergentes', 'SYM': 'renta variable emergentes', 'TH': 'renta variable emergentes',

'TU': 'renta variable emergentes', 'TW': 'renta variable emergentes', 'VNM': 'renta variable emergentes', 'AT': 'renta variable ocde',

'BA': 'renta variable emergentes', 'BE': 'renta variable ocde', 'BNL': 'renta variable ocde', 'CH': 'renta variable ocde', 'CZE': 'renta variable ocde',

'DE': 'renta variable ocde', 'DK': 'renta variable ocde', 'EE': 'renta variable emergentes', 'EES': 'renta variable ocde', 'EMI': 'renta variable ocde',

'ES': 'renta variable ocde', 'EU': 'renta variable ocde', 'EUK': 'renta variable ocde', 'EUR': 'renta variable ocde', 'FI': 'renta variable ocde',

'FR': 'renta variable ocde', 'GR': 'renta variable ocde', 'HUN': 'renta variable ocde', 'IBE': 'renta variable ocde', 'IE': 'renta variable ocde',

'ISL': 'renta variable ocde', 'IT': 'renta variable ocde', 'LIE': 'renta variable ocde', 'LUX': 'renta variable ocde', 'LU': 'renta variable ocde',

'MLT': 'renta variable ocde', 'NL': 'renta variable ocde', 'NO': 'renta variable ocde', 'NOR': 'renta variable ocde', 'POL': 'renta variable ocde',

'PT': 'renta variable ocde', 'RS': 'renta variable emergentes', 'SE': 'renta variable ocde', 'SKA': 'renta variable ocde', 'SVK': 'renta variable ocde',

'TR': 'renta variable emergentes', 'UK': 'renta variable ocde', 'GB': 'renta variable ocde', 'AFR': 'renta variable emergentes', 'BRC': 'renta variable emergentes',

'GEA': 'renta variable ocde', 'GEM': 'renta variable emergentes', 'GEU': 'renta variable ocde', 'GLB': 'renta variable ocde', 'NZL': 'renta variable ocde',

'SAF': 'renta variable emergentes', 'EME': 'renta variable ocde'

}

# Exploración y optimización en entornos financieros altamente dinámicos

Entornos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| * + - Moneda / país     - Categoría del fondo     - Tamaño     - Gestión (activa, pasiva)     - Correlación     - Ratings     - Composición     - Volatilidad | * + - Máximo drawdown     - Tiempo de recuperación     - Tracking error     - Omega     - Calmar     - Sortino     - Information ratio     - Beta |  |  |  |  |
|  |  |  |  |

3 castas

* Exploradoras: su misión es valorar cada fondo para cada perfil. Modifican el entorno, mandando los fondos que se adecúan a los perfiles más agresivos arriba, los moderados a las capas centrales y los conservadores a las capas inferiores. Tienen N iteraciones predefinidas con el entono, antes de que este cambie.

Los fondos son rankeados por cada característica. Una característica podría hacer que un fondo sea valorado en función de su volatilidad como agresivo. Por lo que, en el entorno de volatilidad, ese fondo estará en las capas más altas. Y el mismo fondo puede ser calificado como conservador, por su TE, por lo que sería enviado a las capas más bajas en el entorno de TE. De esa manera, un mismo fondo podría ser empleado a la vez por constructoras agresivas y conservadoras a la vez.

La adecuación de la característica principal del entorno (país, rating, correlación, TE… la que sea), hace que el fondo sea movido hacia arriba, o hacia abajo.

Las variables rentabilidad y riesgo de cada fondo, en cada instante, hace que sea desplazado izq / der (riesgo), adelante / atrás (rentabilidad), en el plano.

Al llegar a una nueva casilla, primero ordena verticalmente (adecuación de la característica al perfil), después izq/der (riesgo) y, por último, ade/atr (rentabilidad).

Movimiento: Las obreras no pueden cambiar de capa (se mueven en un plano). Cuando ha hecho un cambio, marca la casilla con feromona y decide a dónde ir en función de la cantidad de feromona que hay a su alrededor. A menor cantidad, mayor probabilidad de que se desplace a esa casilla. Para poder evitar quedarse “atrapada en una zona”, utiliza el método de enfriamiento de metal para “saltar” de zona, dentro de su capa (de su plano).

* Constructoras de carteras (agresiva, moderada, conservadora): Serían semejantes a usuarios con perfiles asociados. Pueden visitar únicamente las zonas correspondientes a sus perfiles (3 capas). Son capaces de ver a su alrededor en todos los entornos a la vez.

Podríamos hacer que cada una tuviera intereses, gustos, preferencias distintas, para evitar que todas convergieran en la misma cartera. Ej: tener mayor interés en Asia. Cada constructora debe tener su vector de importancia para cada entorno. Es decir, le da más importancia a la descorrelación, luego al TE…

Movimiento: por la manera en la que ordenamos los entornos habrá zonas calientes. ¿Usamos una zona de baile?, ¿un enjambre de partículas?

Incorporan nuevos activos a la cartera que ya tienen (en función de la importancia que le dan a cada entorno) y realizan N optimizaciones de Markowitz, para decidir su cartera en cada instante.

Cada constructora debe tener un sistema de valoración ¿TE, ranking entre ellas? (en función de su histórico). Si el ranking que tiene es muy malo, es que está en una zona de baile mala. Debe cambiar a otra zona.

* Reina: No se desplaza. Es omnisciente. Es capaz de conocer qué activos tienen todas las constructoras. Optimiza una cartera para cada perfil, con el conocimiento global de toda la colmena.

**Esquema del algoritmo**

Generación de los entornos (colocación aleatoria) / actualización entornos

Proceso de organización del entorno (N iteraciones)

Para cada Exploradora

Evaluar movimientos posibles

Decidir movimiento

Reordenar entorno

Vertical: adecuación de la característica del entorno al perfil

Izq / der: Nivel de riesgo

Ade / atr: Rentabilidad

Proceso de optimización de carteras (N iteraciones)

Para cada Constructora

Evaluar movimientos posibles

Decidir movimiento

Incorporar activos de alrededor, de cualquier entorno, en función de las preferencias

Optimizar cartera: incorporar nuevos activos, descartar los que no tengan peso

Reina

Recopilar activos no repetidos, para cada perfil

Generar cartera óptima, para cada perfil