

MH - Práctica Alternativa

Imperialist Competitive Algorithm (ICA)



Santiago Herron Mulet - 54153965M

1. Introducción.	3
2. Motivación.	3
Contexto Histórico del Imperialismo	3
Elección del Algoritmo	4
3. Analogía.	4
Conceptos en el ICA	4
Paralelismo de las Etapas del ICA con la Realidad	4
1. Inicialización de Imperios:	5
2. Asimilación:	5
3. Competencia Imperialista:	5
4. Revolución:	5
5. Colapso Imperial:	5
4. ICA.	6
1. Generación de la Población Inicial:	8
2. Imperios iniciales:	8
3. Asimilación.	9
4. Revolución.	10
5. Intercambio de posiciones entre una colonia y un imperialista	10
6. Transferencia de la colonia más débil.	11
7. Colapso de los imperios más débiles	12
5. Implementación.	12
1. Diseño:	12
2. Inicialización de Imperios:	13
3. Asimilación.	14
4. Revolución.	15
5. Reubicación.	15
6. Transferencia de la colonia más débil del imperialista más débil.	15
6. Parámetros y complejidad.	16
Repaso de los Parámetros	16
Importancia del Ajuste de los Parámetros	17
Complejidad del Algoritmo	17
7. Propuesta de Mejora mediante Hibridación.	18
Justificación de la Mejora	18
Búsqueda Local	18
8. Propuesta de Mejora del Diseño de la Metaheurística - Guerra.	19
Beneficios de la Mejora	20
Funcionamiento de la Guerra	20
9. Experimentación.	21
- 10% de evaluaciones.	21
ICA:	21
BL-ICA:	23
ICA-WAR:	24
50%:	26
ICA:	26

BL-ICA:	27
ICA-WAR:	29
100%:	31
ICA:	31
BL-ICA:	32
ICA-WAR:	33
10. Análisis.	35
ICA vs PSO	35
Resultados Comparativos	35
Análisis de Resultados	35
ICA vs BL-ICA vs ICA-WAR	36
Resultados Comparativos	36
Análisis de Resultados	36

1. Introducción.

En el campo de la optimización, los algoritmos inspirados en procesos naturales han demostrado ser herramientas poderosas para resolver problemas complejos. Uno de estos algoritmos es el Algoritmo Imperialista Competitivo (ICA), introducido por Esmail Atashpaz-Gargari y Caro Lucas en 2007. El ICA se basa en una analogía con la competencia imperialista, donde diferentes imperios compiten por dominar territorios y absorber las colonias de otros imperios. Esta dinámica permite explorar y explotar el espacio de soluciones de manera eficiente, combinando procesos de asimilación y revolución para mejorar continuamente las soluciones.

El objetivo principal de esta práctica es implementar el Algoritmo Imperialista Competitivo para resolver un problema específico de optimización, en este caso se usará el software de Daniel Molina para la competición Cec2017. A lo largo del desarrollo de la práctica, se estudiará el funcionamiento interno del ICA, se ajustarán sus parámetros y se evaluará su rendimiento en comparación con otros algoritmos de optimización. Esta implementación no solo busca resolver el problema planteado, sino también proporcionar una comprensión profunda del ICA y su aplicabilidad en diversos contextos de optimización. Mediante este análisis, se espera destacar las ventajas del ICA, así como identificar posibles áreas de mejora y adaptación del algoritmo para futuros trabajos.

2. Motivación.

Contexto Histórico del Imperialismo

El imperialismo, como fenómeno histórico, se refiere a la política y práctica de extender el poder y dominio de un estado sobre territorios ajenos, generalmente mediante la conquista y control económico y político. Este proceso alcanzó su apogeo a finales del siglo XIX y principios del siglo XX, durante lo que se conoce como la "Edad del Imperialismo". Las potencias europeas, impulsadas por la Revolución Industrial y la búsqueda de nuevos mercados y recursos, se embarcaron en una carrera frenética por colonizar África, Asia y otras partes del mundo.

El imperialismo no solo transformó la geopolítica global, sino que también dejó profundas huellas en las estructuras sociales, económicas y culturales de los territorios colonizados. Las motivaciones detrás del imperialismo eran diversas: desde la necesidad de recursos naturales para alimentar industrias en auge, hasta la ambición de expandir la influencia política y la creencia en la superioridad cultural y racial de los colonizadores. Estos factores, combinados con avances en tecnología militar y de transporte, facilitaron la expansión imperialista a un ritmo sin precedentes.

Elección del Algoritmo

La elección del Algoritmo Imperialista Competitivo (ICA) para esta práctica está profundamente influenciada por mi interés en la historia. El ICA no solo es un método eficaz para resolver problemas de optimización, sino que también ofrece una fascinante analogía con los procesos históricos de imperialismo. Esta relación entre historia y algoritmos proporciona una perspectiva única y enriquecedora sobre cómo los principios históricos pueden ser aplicados en el ámbito de la inteligencia artificial y la optimización.

3. Analogía.

Conceptos en el ICA

En el Algoritmo Imperialista Competitivo, varios conceptos históricos se trasladan al ámbito de la optimización:

- Países: Representan soluciones posibles en el espacio de búsqueda. Cada país tiene una posición que indica su calidad o "fitness".
- Imperialistas: Son las mejores soluciones (países) y forman el núcleo de los imperios. Un imperialista es una solución con alta calidad que lidera un conjunto de otras soluciones (colonias).
- Colonias: Son soluciones subordinadas a un imperialista dentro de un imperio. Las colonias son influenciadas y mejoradas continuamente por su imperialista.
- Imperios: Son conjuntos de países formados por un imperialista y sus colonias. Los imperios compiten entre sí para aumentar su poder y absorber colonias de otros imperios.

En comparación con los algoritmos evolutivos tradicionales, donde se utiliza el término "población" para describir el conjunto de soluciones, el ICA introduce una estructura jerárquica más compleja. En los algoritmos evolutivos, la población evoluciona a través de operadores como selección, cruce y mutación. En el ICA, los imperios evolucionan mediante procesos de asimilación, competencia y revolución, emulando los comportamientos y dinámicas de los imperios históricos.

Paralelismo de las Etapas del ICA con la Realidad

El ICA se compone de varias etapas que emulan los procesos de asimilación, competencia y revolución observados en el imperialismo histórico. A continuación, se describen estas etapas y su paralelismo con la realidad:

1. Inicialización de Imperios:

- Realidad Histórica: En el contexto histórico, las potencias europeas comenzaron su expansión imperialista estableciendo colonias en territorios estratégicos.
- Algoritmo: En el ICA, este proceso se refleja en la inicialización de imperios donde las mejores soluciones (países) son seleccionadas como imperios, y el resto de las soluciones (colonias) se distribuyen entre estos imperios.

2. Asimilación:

- Realidad Histórica: Las potencias coloniales intentaban asimilar las culturas y economías de las colonias para integrar sus recursos y poblaciones dentro del imperio.
- Algoritmo: En el ICA, las colonias se mueven hacia el país imperialista en un proceso de asimilación, mejorando así su posición en el espacio de búsqueda de soluciones.

3. Competencia Imperialista:

- Realidad Histórica: Los imperios competían entre sí para expandir sus territorios y recursos, utilizando tanto la diplomacia como la fuerza militar.
- Algoritmo: En el ICA, los imperios compiten por controlar más colonias. Los imperios más fuertes absorben colonias de los más débiles, emulando la competencia y expansión imperialista.

4. Revolución:

- Realidad Histórica: Las colonias a menudo experimentaban revoluciones y movimientos de resistencia contra el control imperialista, llevando a cambios significativos.
- Algoritmo: El ICA introduce la revolución como un mecanismo de variación aleatoria en las colonias, lo cual permite escapar de óptimos locales y explorar nuevas áreas del espacio de soluciones.

5. Colapso Imperial:

- Realidad Histórica: Los imperios podían colapsar debido a guerras, conflictos internos y presiones económicas, resultando en la pérdida de sus colonias.

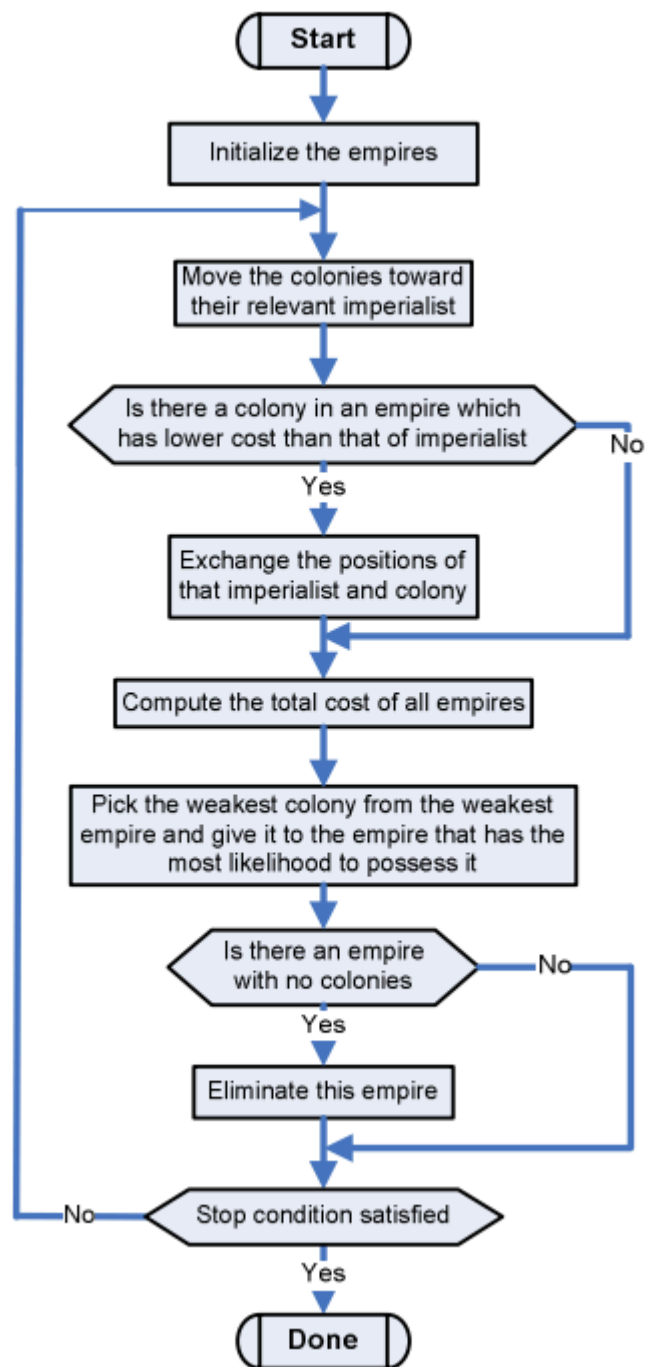
- Algoritmo: En el ICA, los imperios débiles eventualmente colapsan y sus colonias son redistribuidas entre los imperios restantes, asegurando que solo los más aptos sobrevivan.

El Algoritmo Imperialista Competitivo no solo ofrece una metodología eficaz para resolver problemas de optimización, sino que también proporciona una rica analogía con procesos históricos que han moldeado el mundo moderno. A través de esta práctica, se busca no solo aplicar el ICA para resolver un problema específico, sino también explorar y entender cómo los principios de la historia pueden influir en el diseño y aplicación de algoritmos en la inteligencia artificial. A continuación se pasará a detallar las diferentes etapas más en detalle.

4. ICA.

Decidí no partir de implementaciones ya existentes que se pueden encontrar en internet y, en su lugar, opté por implementar todas las partes del algoritmo yo mismo. Para ello, utilicé como base el artículo original del algoritmo e hice las suposiciones necesarias para adaptar el algoritmo a las restricciones específicas de nuestro problema concreto. Este problema se desarrolló para la competición CEC2017, que cuenta con 30 funciones de optimización (minimización). Para agilizar los cálculos, solo se experimentó con soluciones de dimensión 10, lo que significa que cada solución es un vector de 10 números reales entre -100 y 100, y tiene asociado un valor de fitness. Esto es importante para la toma de decisiones durante la implementación del algoritmo.

El Algoritmo Imperialista Competitivo (ICA) tiene una estructura general similar a otros algoritmos evolutivos. Comienza con una población inicial aleatoria, donde cada individuo de la población representa un país. Los países dentro del ICA son análogos a los cromosomas en los Algoritmos Genéticos (GA). En el paso de inicialización, algunos de los mejores países (menores costos) se seleccionan como países imperialistas, y el resto se considera colonias de los imperialistas. Las colonias se dividen entre los imperialistas en función del poder de estos últimos. Después de dividir todas las colonias entre los países imperialistas, las colonias se mueven hacia sus respectivos imperialistas dentro del espacio de estados culturales. Una colección de un país imperialista y algunas colonias forman un imperio, que compiten entre sí y se desplazan mutuamente. La supervivencia de un imperio depende de su poder para tomar colonias de otros competidores, y el poder de los imperios más grandes aumenta mientras que los imperios con menos poder colapsan. Después de muchas iteraciones de competencia entre imperios, el poder de las colonias se aproxima al de los países imperialistas, lo que indica convergencia. El límite extremo de la competencia imperialista ocurre cuando solo queda un imperialista junto con algunas colonias, que están cerca del país imperialista en términos de posición. La estructura quedaría tal que así:



A continuación se describe el ICA paso a paso.

1. Generación de la Población Inicial:

La población inicial de soluciones se genera de forma aleatoria, donde cada solución es un vector de 10 números reales dentro del rango [-100, 100].

2. Imperios iniciales:

Los mejores países (soluciones con menor costo) se seleccionan como imperialistas y el resto se consideran colonias. Las colonias se dividen entre los imperialistas proporcionalmente a su poder relativo.

Cálculo del Coste Normalizado: El coste normalizado de cada imperialista se determina por:

$$C_n = \frac{c_n - \max(c)}{\min(c) - \max(c)}$$

donde c_n es el coste del n-ésimo imperialista, y C_n es el coste normalizado del n-ésimo imperialista.

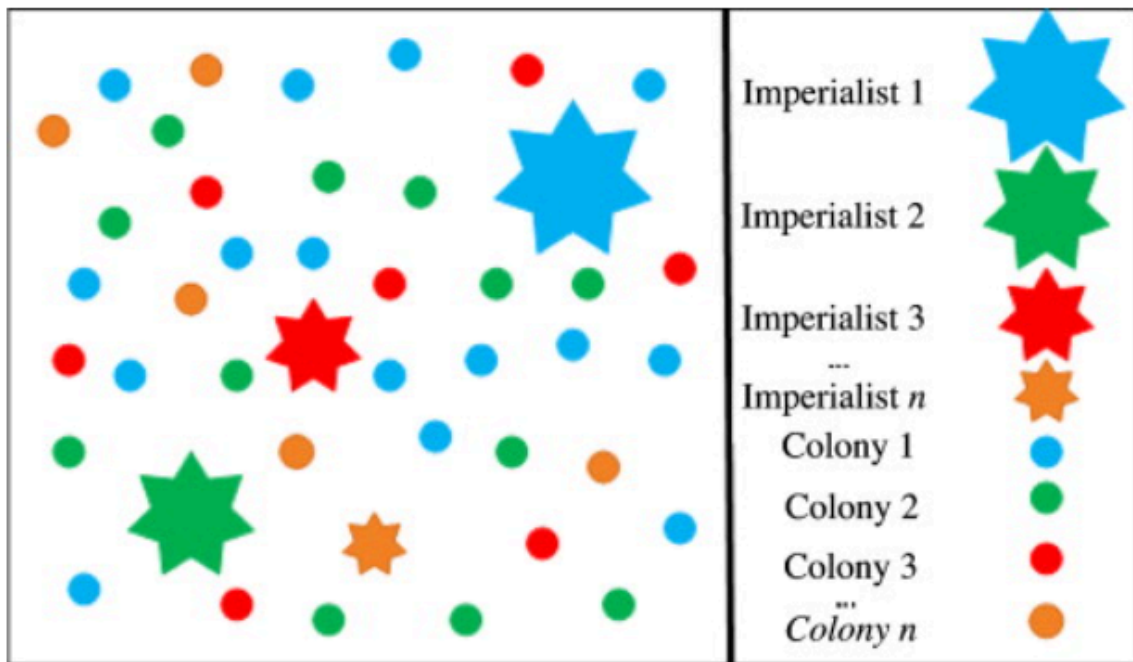
Cálculo del Poder de los Imperialistas: El poder de cada imperialista se calcula en función del coste normalizado:

$$p_n = \frac{1}{C_n}$$

Asignación de Colonias a Imperialistas: El número inicial de colonias que posee cada imperialista se calcula por:

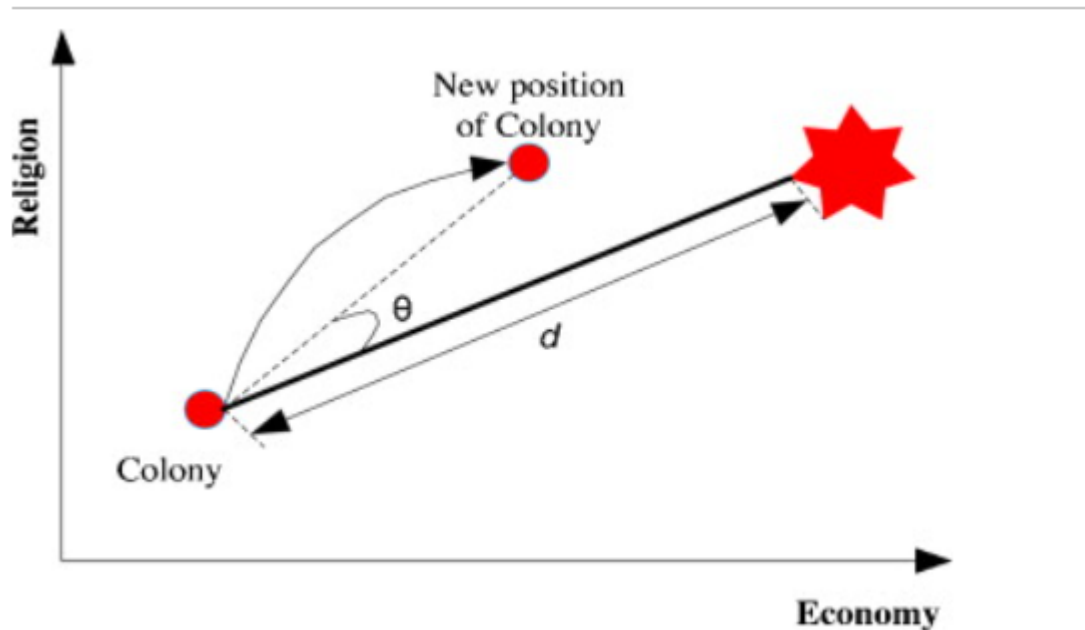
$$NC_n = \text{round} \left(\frac{p_n}{\sum p} \times N_{\text{col}} \right)$$

donde NC_n es el número de colonias del n-ésimo imperialista.



3. Asimilación.

La colonia se mueve una distancia x a lo largo de una dirección d hacia su imperialista. La distancia de movimiento x es un número aleatorio generado por una distribución aleatoria dentro del intervalo $[0, \beta \cdot d]$.



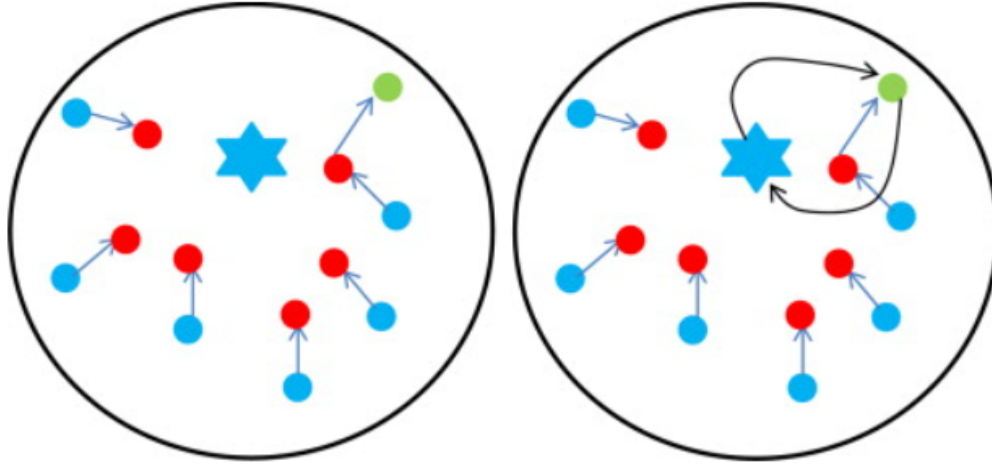
Sin embargo, la historia ha demostrado que las colonias son absorbidas parcialmente, no perfectamente, en el país imperialista, lo que indica la presencia de una desviación en el proceso de asimilación. El parámetro de desviación se muestra como θ , que sigue una distribución uniforme.

4. Revolución.

La revolución representa un proceso de cambio repentino en la posición de algunas colonias en el espacio de búsqueda. Este proceso se asemeja a la operación de mutación en los algoritmos genéticos (GA). La revolución en el ICA se implementa para aumentar la exploración y evitar la convergencia prematura hacia óptimos locales.

5. Intercambio de posiciones entre una colonia y un imperialista

Una vez que se realizan las operaciones de asimilación y revolución en las colonias de un imperio en el algoritmo Imperialista Competitivo (ICA), las funciones de costo de las nuevas posiciones de las colonias se comparan con la función de costo de la posición del imperialista. Si encontramos alguna colonia cuya función de costo sea menor que la función de costo del imperialista, entonces intercambiamos al imperialista con esa colonia.



6. Transferencia de la colonia más débil.

El poder de un imperio se calcula en función del poder de su imperialista y una fracción del poder de sus colonias.

$$TC_n = (1 - \zeta) \cdot CN_n + \zeta \cdot \frac{1}{NC_n} \sum_{i=1}^{NC_n} CN_i$$

Donde TC_n es el costo total del n -ésimo imperio, ζ es un número positivo entre 0 y 1, generalmente cercano a cero. Un valor pequeño de ζ enfatiza una mayor influencia del poder imperialista en la determinación del poder total del imperio, mientras que un valor grande de ζ indica la influencia de la potencia media de las colonias en la determinación del poder total del imperio.

Durante la competencia entre los países imperialistas, los imperios más débiles colapsarán gradualmente. Esto significa que los imperios más débiles perderán sus colonias con el tiempo, mientras que los imperios más fuertes poseerán las colonias de los imperios más débiles, aumentando así su poder. Por lo tanto, una o algunas de las colonias más débiles pertenecientes al imperio más débil se asignarán a un imperio diferente en función de la competencia que ocurre entre todos los imperios. Los imperios más fuertes tienen una mayor probabilidad de poseer la colonia más débil.

Para modelar el proceso de competencia entre los imperios, necesitamos calcular el costo total normalizado del imperio mediante:

$$NTC_n = \frac{TC_n}{\sum_{i=1}^N TC_i}$$

Luego, la probabilidad de poseer una colonia se calcula por:

$$P_n = 1 - NTC_n$$

Se necesita un mecanismo similar a la Ruleta en GA para distribuir las colonias más débiles entre los imperios en función de sus probabilidades de posesión. El ICA introduce un nuevo mecanismo de distribución que requiere menos cálculo que la Ruleta en GA, que requiere un cálculo bastante pesado para la Función de Distribución Acumulativa (CDF). El mecanismo en el ICA solo requiere una Función de Densidad de Probabilidad (PDF), que se aborda a continuación:

El vector P con el tamaño de $1 * N_{imp}$ contiene la probabilidad de posesión de una colonia por parte de los imperios de la siguiente manera:

$$P=[P_1,P_2,...,P_{N_{imp}}]$$

Luego, se forma el vector R con el mismo tamaño de P en el que sus elementos se generan utilizando una distribución uniforme dentro del intervalo [0,1].

A continuación, se define el vector D como: $D=P-R$ $D = P - R$

Una vez que se calcula el vector D, la colonia más débil se asigna al imperio con el índice más grande. La ausencia de cálculo de CDF acelerará el proceso de distribución del ICA en comparación con la Ruleta en GA.

7. Colapso de los imperios más débiles

Los imperios más débiles pierden gradualmente sus colonias ante los imperios más fuertes, que a su vez se vuelven más poderosos y causan que los imperios más débiles colapsen con el tiempo.

5. Implementación.

1. Diseño:

El diseño se basa en la modularidad y la jerarquía de clases para representar las entidades principales del algoritmo. Aquí hay una descripción del diseño y sus aspectos más importantes:

1. Clases Principales:

Country:

Representa tanto a los imperios como a las colonias.

Contiene atributos comunes como la solución y la aptitud, así como un enum que indica el tipo de país.

Imperialist:

Subclase de Country.

Representa un imperialista que tiene colonias bajo su dominio.

Colony:

Subclase de Country.

Representa una colonia que pertenece a un imperialista.

2. Clase ICA:

Atributos Relevantes:

CountryList: Lista de países.

Imperialists: Lista de imperialistas.

dim: Dimensión del problema.

max_evals: Número máximo de evaluaciones de la función objetivo.

ImperialistPercentage: Porcentaje de países que serán imperialistas.

numEmpires: Número de imperialistas.

numColonies: Número de colonias.

phi, beta, xi, revolution_rate: Parámetros del algoritmo.

evals: Número de evaluaciones realizadas.

seed: Semilla para la generación de números aleatorios.

gen: Generador de números aleatorios.

Métodos Relevantes:

initializeEmpires(): Inicializa los imperios con sus colonias correspondientes.

Assimilation(), Revolution(), Relocation(): Etapas del algoritmo ICA.

WeakestColonyTransfer(): Transfiere la colonia más débil al imperio más fuerte.

EmpireElimination(): Elimina los imperios más débiles.

Competition(): Realiza la competencia entre los imperios.

2. Inicialización de Imperios:

Se ordenan los países por su aptitud en orden ascendente (menor fitness primero).

Se inicializan los imperialistas con los países de menor fitness.

Se calcula el poder de los imperialistas.

Se distribuyen las colonias entre los imperialistas basándose en su poder.

Se manejan los imperios que no tienen colonias.

Se distribuyen las colonias restantes entre los imperialistas.

Función initializeEmpires():

```
countries = Copiar(CountryList)
```

```
Ordenar(countries) // Ordena los países por aptitud ascendente
```

```
Para i desde 0 hasta numEmpires:
```

```

    imp = Nuevo Imperialist(con la solución y aptitud del primer país en countries)
    Añadir imp a Imperialists
    Eliminar el primer país de countries

    power = ImperialistPower()
    num_colonies_imperialist = []
    Para i desde 0 hasta numEmpires:
        Añadir redondear(power[i] * numColonies) a num_colonies_imperialist

    Para i desde 0 hasta numEmpires:
        Para j desde 0 hasta num_colonies_imperialist[i]:
            Si countries no está vacío:
                index = AleatorioEntre(0, tamaño de countries - 1)
                col = Nueva Colony(con la solución y aptitud del país en countries en la
posición index)
                Añadir col a las colonias del Imperialist en la posición i
                Eliminar el país en la posición index de countries

```

3. Asimilación.

Se realiza la asimilación entre imperialistas y sus colonias.

Se calcula una distancia euclidiana entre el imperialista y cada colonia.

Se aplican pasos de asimilación a las colonias.

Se actualiza la solución y la aptitud de cada colonia.

Es importante definir el concepto de distancia en nuestro problema, en mi caso la definí como la distancia Euclídea.

Función Assimilation():

```

    Para cada Imperialist en Imperialists:
        Para cada col en las colonias del Imperialist:
            Si evals >= max_evals:
                Salir del bucle
            Calcular la distancia d entre el Imperialist y la col
            Generar un número aleatorio x entre 0 y beta*d
            Generar un ángulo aleatorio theta entre -phi y phi
            Calcular el número de pasos steps = d/10
            Inicializar step a 0
            Mientras step sea menor que steps:
                Avanzar new_sol hacia imp->getSolution() en 10 unidades
                Desviar new_sol en función de theta con un paso de 1 unidad
                Incrementar step
            Calcular la nueva aptitud new_fitness de new_sol
            Incrementar evals en 1
            Actualizar la solución y la aptitud de la col con new_sol y new_fitness

```

4. Revolución.

Se realiza la revolución en las colonias de los imperialistas.

Se aplica una tasa de revolución a cada colonia.

Se generan mutaciones en la solución de una colonia con una probabilidad determinada.

Se evalúa la nueva solución y se actualiza la colonia con la nueva solución y su aptitud.

Función Revolution():

Para cada Imperialist en Imperialists:

Para cada col en las colonias del Imperialist:

Si la probabilidad aleatoria es menor que la tasa de revolución:

Si evals \geq max_evals:

Salir del bucle

Obtener la solución actual new_sol de la colonia

Calcular el número de cambios changes como el máximo entre 1 y el 5% de las dimensiones

Para i desde 0 hasta changes:

Seleccionar un índice aleatorio index

Aplicar una mutación aleatoria a la new_sol en el índice seleccionado

Calcular la nueva aptitud new_fitness de la nueva solución

Incrementar evals en 1

Actualizar la solución de la colonia con new_sol

Actualizar el fitness de la colonia con new_fitness

5. Reubicación.

Se realiza la reubicación de las colonias hacia el imperialista si la aptitud de la colonia es menor que la del imperialista.

Se intercambian las soluciones y los valores de aptitud entre el imperialista y la colonia si es necesario

Función Relocation():

Para cada Imperialist en Imperialists:

Para cada col en las colonias del Imperialist:

Si la aptitud de la colonia es menor que la del imperialista:

Intercambiar las soluciones y los valores de aptitud entre el imperialista y la colonia

6. Transferencia de la colonia más débil del imperialista más débil.

Busca la colonia más débil entre todos los imperialistas y transfiere esta colonia al imperialista con la mayor potencia.

Función `WeakestColonyTransfer()`:

`weakestColony` = Colonia con menos fitness del Imperio más débil

`weakestImperialist` = Imperialista con el menor fitness.

Quitar la colonia `weakestColony` de `weakestImperialist`

Calcular el poder total de los imperios y almacenarlo en el vector `p`

Generar números aleatorios uniformemente distribuidos entre 0 y 1 y almacenarlos en el vector `r`

Calcular la diferencia entre el poder total y los valores aleatorios generados, y almacenarlos en el vector `d`

Encontrar el índice del imperialista con la mayor diferencia en el vector `d` y almacenarlo en la variable `index`

Añadir la colonia `weakestColony` al imperialista en `Imperialists[index]`

6. Parámetros y complejidad.

El éxito de cualquier metaheurística, incluido el ICA, depende en gran medida de una configuración adecuada de sus parámetros. La correcta sintonización de estos parámetros puede mejorar significativamente la flexibilidad y robustez del algoritmo, aunque requiere una inicialización cuidadosa. La llamada al método ICA con todos sus parámetros es la siguiente:

```
ICA(int dim, int max_evals, double ImperialistPercentage,  
vector<Country> CountryList, double phi, double beta, double xi,  
double revolution_rate, int seed)
```

Repaso de los Parámetros

1. **dim**: Dimensionalidad del problema.
2. **max_evals**: Número máximo de evaluaciones de la función objetivo.
3. **ImperialistPercentage**: Porcentaje de países que se inicializan como imperialistas.
4. **CountryList**: Lista de países (soluciones) iniciales.
5. **phi**: Parámetro de asimilación de desviación.
6. **beta**: Parámetro de asimilación de dirección.
7. **xi**: Coeficiente asociado con el poder promedio de las colonias del imperio.
8. **revolution_rate**: Tasa de revolución (equivalente a la tasa de mutación en algoritmos genéticos).
9. **seed**: Semilla para el generador de números aleatorios.

Importancia del Ajuste de los Parámetros

El ajuste adecuado de los parámetros del ICA es crucial para su rendimiento. No es evidente definir a priori cuál debería ser la configuración óptima de parámetros, ya que esto depende en gran medida del tamaño y tipo del problema, así como del tiempo de búsqueda que el usuario está dispuesto a invertir. Una sintonización óptima permite al algoritmo equilibrar adecuadamente la exploración y explotación del espacio de soluciones.

1. **phi (θ):** Este parámetro define el compromiso entre la exploración y la explotación. Valores pequeños favorecen la explotación, mientras que valores grandes promueven la exploración. Un ajuste adaptativo de ϕ , comenzando con un valor grande y reduciéndolo gradualmente, puede ser beneficioso.
2. **beta:** Determina qué tan cerca una colonia se acerca a su imperialista durante la asimilación. Valores pequeños de β promueven la explotación, mientras que valores grandes fomentan la exploración.
3. **ImperialistPercentage:** Este parámetro define la proporción de imperialistas en la población total. Un porcentaje típico es alrededor del 10-13%. Un número pequeño de imperialistas puede reducir el esfuerzo computacional, pero también la calidad de la solución.
4. **xi:** Ajusta el efecto del poder acumulado de las colonias en el poder total del imperio. Valores grandes de ξ enfatizan la contribución del poder de las colonias, mientras que valores pequeños indican que el poder del imperio depende casi por completo del imperialista.
5. **revolution_rate:** Este parámetro controla la tasa de mutación. Una tasa de revolución alta refuerza la exploración, mientras que una baja favorece la explotación.

Complejidad del Algoritmo

El ICA es una metaheurística altamente compleja debido a la cantidad de parámetros que afectan su comportamiento y rendimiento. Cada parámetro puede ajustarse en un rango continuo de valores reales, lo que crea un espacio de búsqueda vasto y multidimensional para la sintonización de parámetros.

1. **Dependencia de los Parámetros:** La interdependencia entre los parámetros aumenta la complejidad, ya que el ajuste de uno puede influir en la eficacia de otro. Por ejemplo, el ajuste de ϕ y β debe equilibrarse para mantener una exploración y explotación adecuadas.
2. **Dificultad de Sintonización:** La dificultad de encontrar la configuración óptima de parámetros se incrementa con la dimensionalidad del problema y la variedad de instancias de problemas posibles. La sintonización manual puede ser extremadamente laboriosa y es posible que no siempre conduzca a los mejores resultados.
3. **Impacto en el Rendimiento:** La sensibilidad del algoritmo a los parámetros significa que pequeñas variaciones en estos pueden tener un gran impacto en el rendimiento, lo que requiere un enfoque cuidadoso y meticuloso para su ajuste.

7. Propuesta de Mejora mediante Hibridación.

En esta sección se propone una mejora sencilla pero efectiva para el algoritmo de optimización basado en Imperialismo Competitivo (ICA). La mejora consiste en aplicar una búsqueda local a los países imperialistas con una cierta probabilidad después de cada iteración del algoritmo. La idea detrás de esta hibridación es combinar la exploración global del espacio de soluciones, característica del ICA, con la explotación intensiva de soluciones locales a través de la búsqueda local.

Justificación de la Mejora

El ICA es conocido por su capacidad para explorar amplios espacios de soluciones mediante la competencia entre imperios y la asimilación de colonias. Sin embargo, uno de los posibles inconvenientes del ICA es que puede quedarse atrapado en óptimos locales debido a la falta de explotación intensiva de las soluciones encontradas. La hibridación con técnicas de búsqueda local puede ayudar a mitigar este problema, mejorando así la capacidad del algoritmo para encontrar soluciones más óptimas.

Al aplicar una búsqueda local específicamente a los países imperialistas, se potencia la calidad de los líderes de cada imperio, incrementando así la presión selectiva y mejorando la convergencia del algoritmo. La búsqueda local puede refinar las soluciones de los imperialistas, permitiéndoles alcanzar soluciones más óptimas en su vecindad y, en consecuencia, guiar mejor a sus colonias hacia regiones más prometedoras del espacio de búsqueda.

Esta mejora es sencilla de implementar y puede integrarse fácilmente en el ciclo iterativo del ICA sin un costo computacional significativo, dado que solo se aplica a los países imperialistas, que son una fracción pequeña del total de países. Esta estrategia equilibrada entre exploración y explotación tiene el potencial de mejorar significativamente el rendimiento del algoritmo.

Búsqueda Local

1. Inicialización: La función comienza obteniendo la aptitud actual del país imperialista (fitness) y estableciendo un contador de evaluaciones (evals).
2. Bucle de Búsqueda Local: La búsqueda local se realiza mientras el número de evaluaciones (evals) no exceda el límite especificado (evaluaciones). Dentro de este bucle, se inicializa la mejor aptitud (bestFitness) y la mejor solución (bestSolution) como la aptitud y solución actuales.
3. Generación de Soluciones Vecinas: Para cada dimensión de la solución actual, se genera una nueva solución (newSolution) modificando aleatoriamente el valor de la dimensión en el rango $[-1, 1]$. Se verifica que la nueva solución esté dentro del rango válido $[-100, 100]$.

4. Evaluación de Soluciones Vecinas: Se calcula la aptitud de la nueva solución (newFitness). Si esta aptitud es mejor que la mejor aptitud encontrada hasta ahora (bestFitness), se actualizan bestFitness y bestSolution.
5. Actualización de la Solución: Después de evaluar todas las soluciones vecinas, se actualizan la solución y la aptitud del país imperialista con la mejor solución encontrada en la iteración.

Función BúsquedaLocal(max_evaluaciones)

fitness_actual ← obtenerFitness()

evaluaciones ← 0

Mientras evaluaciones < max_evaluaciones

mejor_fitness ← fitness_actual

mejor_solucion ← solucion_actual

Para cada dimensión de la solución

nueva_solucion ← crearCopia(solucion_actual)

nueva_solucion[d] ← nueva_solucion[d] + valorAleatorio(-1, 1)

nueva_solucion[d] ← limitarRango(nueva_solucion[d], -100, 100)

nuevo_fitness ← calcularFitness(nueva_solucion)

evaluaciones ← evaluaciones + 1

Si nuevo_fitness < mejor_fitness

mejor_fitness ← nuevo_fitness

mejor_solucion ← nueva_solucion

fitness_actual ← mejor_fitness

solucion_actual ← mejor_solucion

FinFunción

8. Propuesta de Mejora del Diseño de la Metaheurística - Guerra.

La analogía del imperialismo ofrece un marco intuitivo y potente para la metaheurística, donde las naciones más fuertes (imperialistas) dominan y asimilan a las más débiles (colonias). Sin embargo, en la realidad geopolítica, las interacciones entre imperios no se limitan a la dominación pasiva; a menudo, se enfrentan directamente por la supremacía y el control de territorios. Esta observación nos lleva a proponer una mejora significativa: introducir el concepto de "guerra" entre imperios en nuestro algoritmo. La guerra permite que los imperios compitan activamente por las colonias en su zona de influencia, emulando

más fielmente la dinámica real y potencialmente mejorando la exploración y explotación del espacio de soluciones.

Beneficios de la Mejora

- Aumento de la Diversidad: Al forzar enfrentamientos entre imperios, las guerras pueden redistribuir las colonias de manera más efectiva, evitando la convergencia prematura en soluciones subóptimas.
- Mejora de la Exploitation: Los enfrentamientos directos permiten que las colonias de mayor aptitud sean absorbidas por los imperios más fuertes, optimizando así el proceso de refinamiento de soluciones.
- Intensificación de la Competencia: La guerra introduce una presión selectiva adicional, asegurando que solo los imperios más adaptativos y competitivos sobrevivan y prosperen.
- Adaptación Dinámica: La probabilidad de guerra basada en la proximidad entre imperios introduce un mecanismo adaptativo, donde las interacciones se vuelven más frecuentes y relevantes en áreas de alta densidad de soluciones.

Funcionamiento de la Guerra

El algoritmo de guerra se activa cuando dos imperialistas están suficientemente cerca uno del otro, determinada por la distancia euclídea entre ellos. La probabilidad de guerra es inversamente proporcional a esta distancia, aumentando la probabilidad de conflicto cuanto más cerca estén los imperialistas.

- Cálculo de la Probabilidad de Guerra: Se calcula la distancia euclídea entre dos imperialistas. A partir de esta distancia, se determina la probabilidad de guerra usando una fórmula que ajusta la probabilidad de acuerdo a la proximidad.
- Iniciación de la Guerra: Si la probabilidad de guerra es mayor que un valor aleatorio generado, los dos imperios entran en guerra.
- Competencia por Colonias: Las colonias de ambos imperios son barajadas y comparadas una a una. La colonia con mejor aptitud en cada par es absorbida por el imperio del rival.
- Resultado del Conflicto: El imperio ganador se queda con la colonia del rival, fortaleciendo su posición.

Para cada par de imperialistas (imp1, imp2) en Imperialists

Calcular la distancia euclídea entre imp1 y imp2

Calcular la probabilidad de guerra basada en la distancia

```

Si la probabilidad de guerra es mayor que un valor aleatorio
  Barajar las colonias de imp1 y imp2

Para cada par de colonias (col1, col2) en las colonias barajadas
  Si la aptitud de col1 es mejor que la de col2
    imp1 absorbe col2 y la quita de imp2
  De lo contrario
    imp2 absorbe col1 y la quita de imp1
  Fin Para
Fin Si
Fin Para

```

9. Experimentación.

Para llevar a cabo la experimentación se usó el software de Daniel Molina con los benchmarks proporcionados en [tacolab](#). Se decidió experimentar con dimensión 10 para que la complejidad computacional no fuera muy alta, se tomaron los datos con el 10%, 50% y 100% de evaluaciones usadas y con las 30 funciones del benchmark. Por otro lado, se decidió comparar los algoritmos implementados (ICA, BL_ICA (propuesta mejora con hibridación), ICA-WAR (propuesta mejora del diseño) con el Particle Swarm Optimization ya que una metaheurística explicada en clase y además con el resto de metaheurísticas del benchmark perdía el sentido hacer comparaciones ya que eran muy superiores ya que estas fueron creadas especialmente para esta competición en 2017.

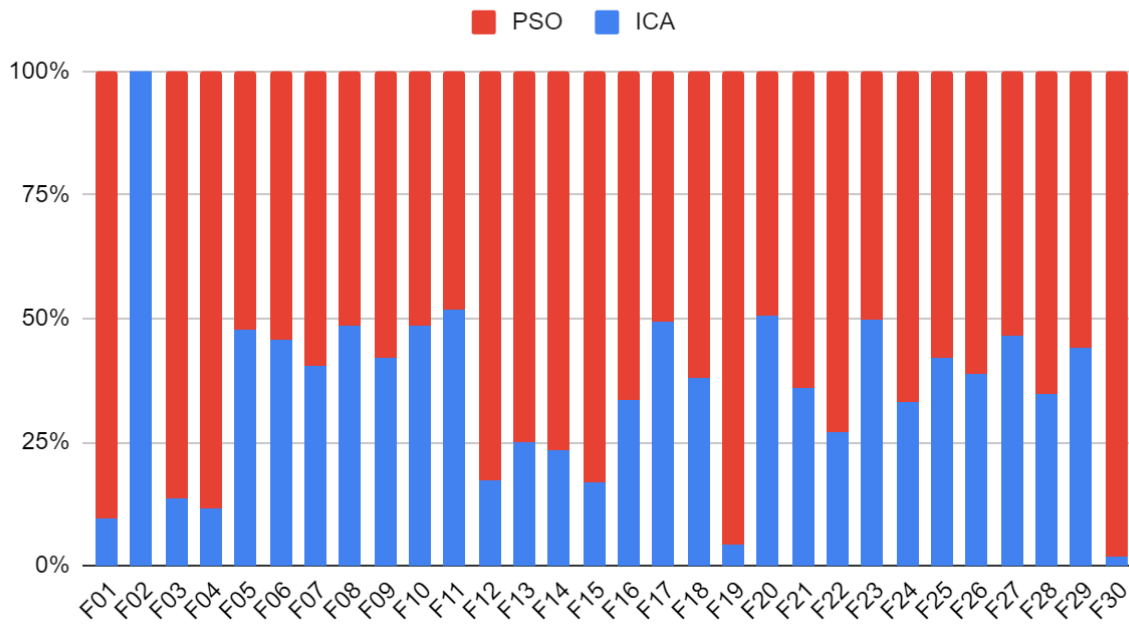
- 10% de evaluaciones.

ICA:

	ICA	PSO
Best	2,70E+01	3,00E+00
F01	8,94E+07	8,30E+08
F02	5,96E+05	1,00E+00
F03	2,00E+03	1,28E+04
F04	1,49E+01	1,16E+02
F05	6,45E+01	7,09E+01
F06	2,17E+01	2,59E+01
F07	6,04E+01	8,90E+01
F08	5,33E+01	5,69E+01
F09	2,16E+02	2,95E+02

F10	1,51E+03	1,59E+03
F11	1,26E+02	1,16E+02
F12	5,54E+06	2,63E+07
F13	4,88E+04	1,48E+05
F14	1,67E+02	5,51E+02
F15	1,77E+03	8,78E+03
F16	1,74E+02	3,42E+02
F17	1,27E+02	1,31E+02
F18	1,44E+05	2,36E+05
F19	1,02E+03	2,36E+04
F20	1,81E+02	1,78E+02
F21	1,07E+02	1,89E+02
F22	5,95E+01	1,61E+02
F23	3,57E+02	3,58E+02
F24	1,31E+02	2,66E+02
F25	3,52E+02	4,84E+02
F26	3,79E+02	6,01E+02
F27	3,88E+02	4,49E+02
F28	3,90E+02	7,36E+02
F29	3,38E+02	4,30E+02
F30	7,06E+04	3,77E+06

10% - ICA vs PSO

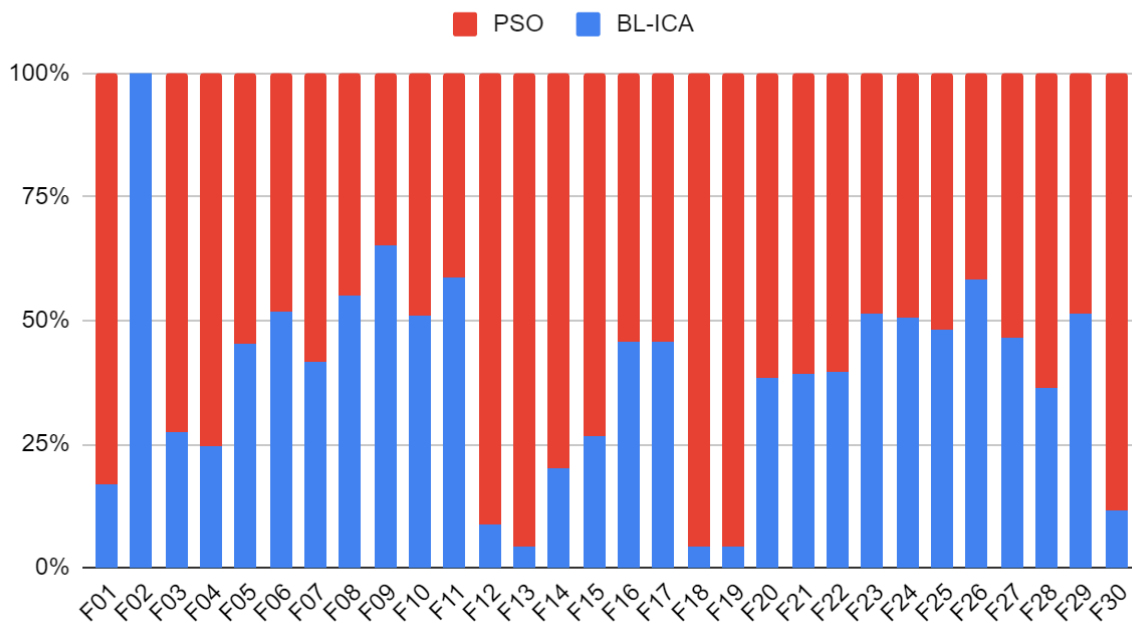


BL-ICA:

	BL-ICA	PSO
Best	2,00E+01	1,00E+01
F01	1,70E+08	8,30E+08
F02	6,22E+05	1,00E+00
F03	4,88E+03	1,28E+04
F04	3,78E+01	1,16E+02
F05	5,84E+01	7,09E+01
F06	2,79E+01	2,59E+01
F07	6,38E+01	8,90E+01
F08	7,00E+01	5,69E+01
F09	5,53E+02	2,95E+02
F10	1,64E+03	1,59E+03
F11	1,65E+02	1,16E+02
F12	2,50E+06	2,63E+07
F13	6,43E+03	1,48E+05
F14	1,38E+02	5,51E+02
F15	3,17E+03	8,78E+03
F16	2,87E+02	3,42E+02

F17	1,09E+02	1,31E+02
F18	1,10E+04	2,36E+05
F19	1,03E+03	2,36E+04
F20	1,10E+02	1,78E+02
F21	1,21E+02	1,89E+02
F22	1,05E+02	1,61E+02
F23	3,80E+02	3,58E+02
F24	2,70E+02	2,66E+02
F25	4,53E+02	4,84E+02
F26	8,40E+02	6,01E+02
F27	3,90E+02	4,49E+02
F28	4,23E+02	7,36E+02
F29	4,51E+02	4,30E+02
F30	492443,9	3769251

10% - BL-ICA vs PSO

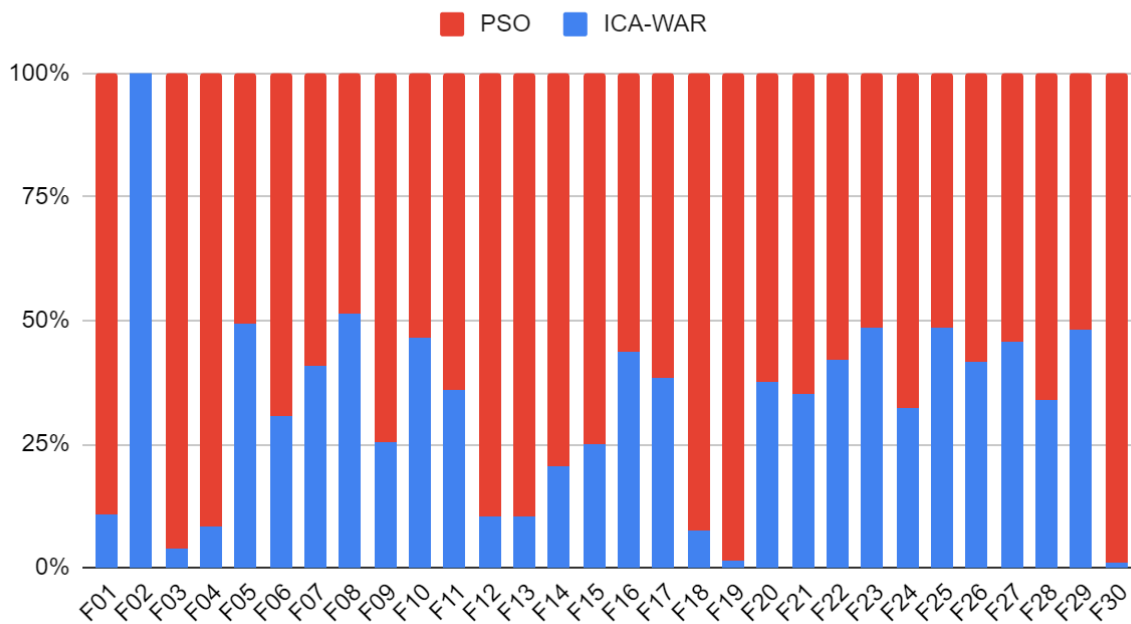


ICA-WAR:

	ICA-WAR	PSO
Best	2,80E+01	2,00E+00

F01	1,00E+08	8,30E+08
F02	2,94E+04	1,00E+00
F03	5,31E+02	1,28E+04
F04	1,03E+01	1,16E+02
F05	6,86E+01	7,09E+01
F06	1,15E+01	2,59E+01
F07	6,18E+01	8,90E+01
F08	6,02E+01	5,69E+01
F09	9,95E+01	2,95E+02
F10	1,39E+03	1,59E+03
F11	6,52E+01	1,16E+02
F12	3,03E+06	2,63E+07
F13	1,73E+04	1,48E+05
F14	1,41E+02	5,51E+02
F15	2,92E+03	8,78E+03
F16	2,65E+02	3,42E+02
F17	8,18E+01	1,31E+02
F18	1,98E+04	2,36E+05
F19	3,37E+02	2,36E+04
F20	1,08E+02	1,78E+02
F21	1,02E+02	1,89E+02
F22	1,17E+02	1,61E+02
F23	3,40E+02	3,58E+02
F24	1,26E+02	2,66E+02
F25	4,56E+02	4,84E+02
F26	4,26E+02	6,01E+02
F27	3,81E+02	4,49E+02
F28	3,80E+02	7,36E+02
F29	4,00E+02	4,30E+02
F30	3,59E+04	3,77E+06

10% - ICA-WAR vs PSO



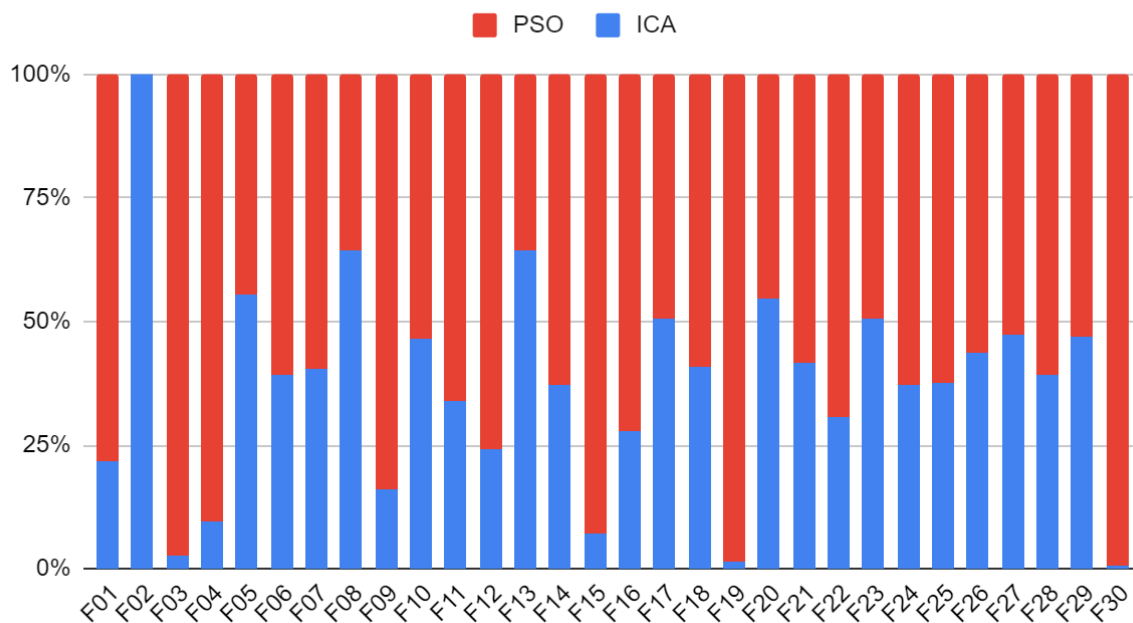
50%:

ICA:

	ICA	PSO
Best	2,30E+01	7,00E+00
F01	5,09E+07	1,85E+08
F02	3,71E+03	1,00E+00
F03	9,71E+01	3,46E+03
F04	6,79E+00	6,36E+01
F05	5,86E+01	4,73E+01
F06	8,72E+00	1,36E+01
F07	3,87E+01	5,70E+01
F08	5,33E+01	2,97E+01
F09	1,60E+01	8,36E+01
F10	1,15E+03	1,32E+03
F11	2,58E+01	5,04E+01
F12	1,49E+06	4,64E+06
F13	2,22E+04	1,23E+04

F14	9,75E+01	1,64E+02
F15	2,53E+02	3,23E+03
F16	7,12E+01	1,85E+02
F17	7,83E+01	7,63E+01
F18	1,83E+04	2,65E+04
F19	8,91E+01	5,86E+03
F20	1,32E+02	1,10E+02
F21	1,02E+02	1,42E+02
F22	4,56E+01	1,03E+02
F23	3,43E+02	3,37E+02
F24	1,21E+02	2,04E+02
F25	2,73E+02	4,54E+02
F26	3,22E+02	4,17E+02
F27	3,76E+02	4,20E+02
F28	3,39E+02	5,24E+02
F29	3,10E+02	3,51E+02
F30	1,05E+04	1,48E+06

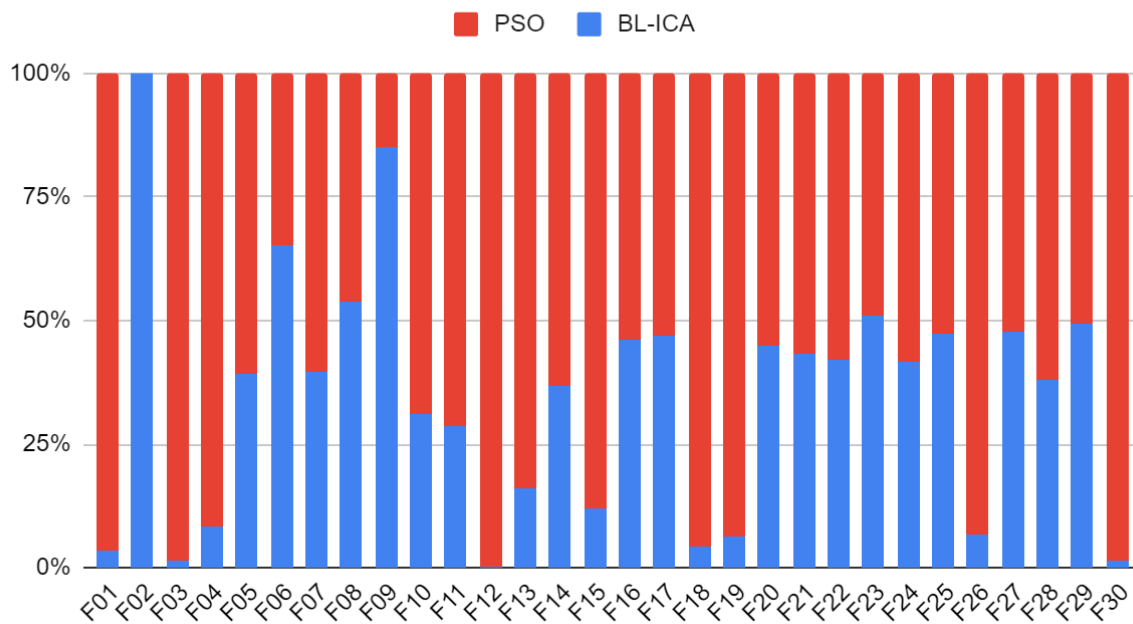
50% - ICA vs PSO



BL-ICA:

	BL-ICA	PSO
Best	2,50E+01	5,00E+00
F01	6,98E+06	1,85E+08
F02	1,00E+04	1,00E+00
F03	4,90E+01	3,46E+03
F04	5,87E+00	6,36E+01
F05	3,03E+01	4,73E+01
F06	2,54E+01	1,36E+01
F07	3,76E+01	5,70E+01
F08	3,46E+01	2,97E+01
F09	4,80E+02	8,36E+01
F10	5,99E+02	1,32E+03
F11	2,03E+01	5,04E+01
F12	1,21E+04	4,64E+06
F13	2,37E+03	1,23E+04
F14	9,48E+01	1,64E+02
F15	4,49E+02	3,23E+03
F16	1,58E+02	1,85E+02
F17	6,80E+01	7,63E+01
F18	1,21E+03	2,65E+04
F19	3,86E+02	5,86E+03
F20	9,02E+01	1,10E+02
F21	1,09E+02	1,42E+02
F22	7,50E+01	1,03E+02
F23	3,49E+02	3,37E+02
F24	1,45E+02	2,04E+02
F25	4,06E+02	4,54E+02
F26	2,97E+01	4,17E+02
F27	3,84E+02	4,20E+02
F28	3,19E+02	5,24E+02
F29	3,42E+02	3,51E+02
F30	2,05E+04	1,48E+06

50% - BL-ICA vs PSO

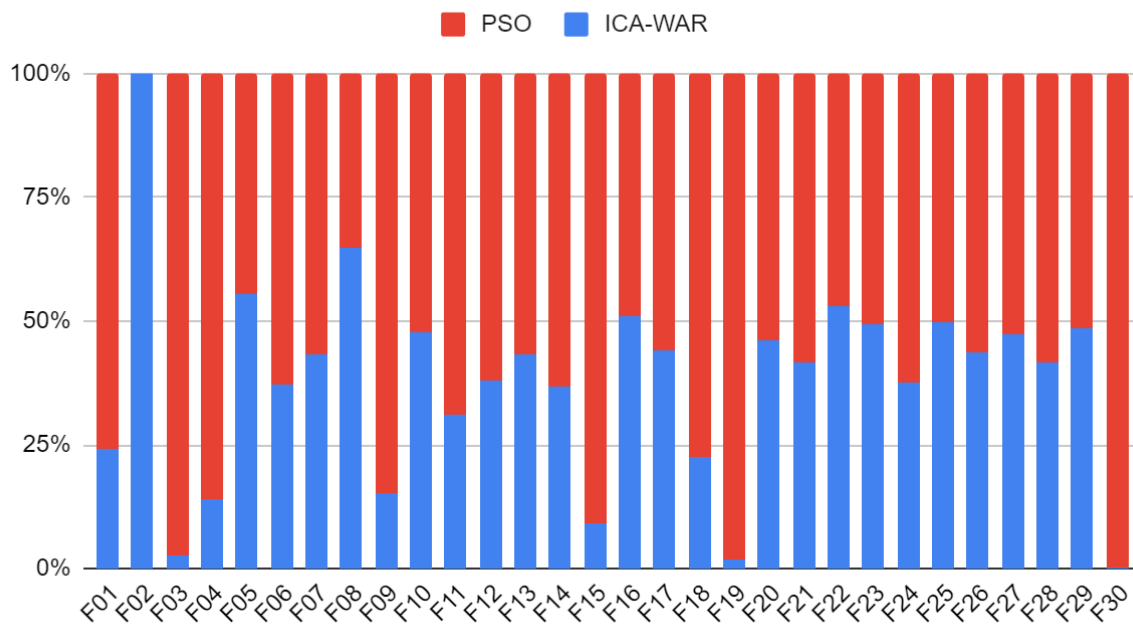


ICA-WAR:

	ICA-WAR	PSO
Best	2,50E+01	5,00E+00
F01	5,89E+07	1,85E+08
F02	1,62E+03	1,00E+00
F03	9,91E+01	3,46E+03
F04	1,03E+01	6,36E+01
F05	5,88E+01	4,73E+01
F06	8,02E+00	1,36E+01
F07	4,33E+01	5,70E+01
F08	5,42E+01	2,97E+01
F09	1,51E+01	8,36E+01
F10	1,21E+03	1,32E+03
F11	2,27E+01	5,04E+01
F12	2,86E+06	4,64E+06
F13	9,44E+03	1,23E+04
F14	9,56E+01	1,64E+02
F15	3,20E+02	3,23E+03
F16	1,93E+02	1,85E+02

F17	5,99E+01	7,63E+01
F18	7,68E+03	2,65E+04
F19	1,04E+02	5,86E+03
F20	9,44E+01	1,10E+02
F21	1,01E+02	1,42E+02
F22	1,17E+02	1,03E+02
F23	3,31E+02	3,37E+02
F24	1,23E+02	2,04E+02
F25	4,51E+02	4,54E+02
F26	3,24E+02	4,17E+02
F27	3,78E+02	4,20E+02
F28	3,74E+02	5,24E+02
F29	3,33E+02	3,51E+02
F30	6,06E+03	1,48E+06

50% - ICA-WAR vs PSO



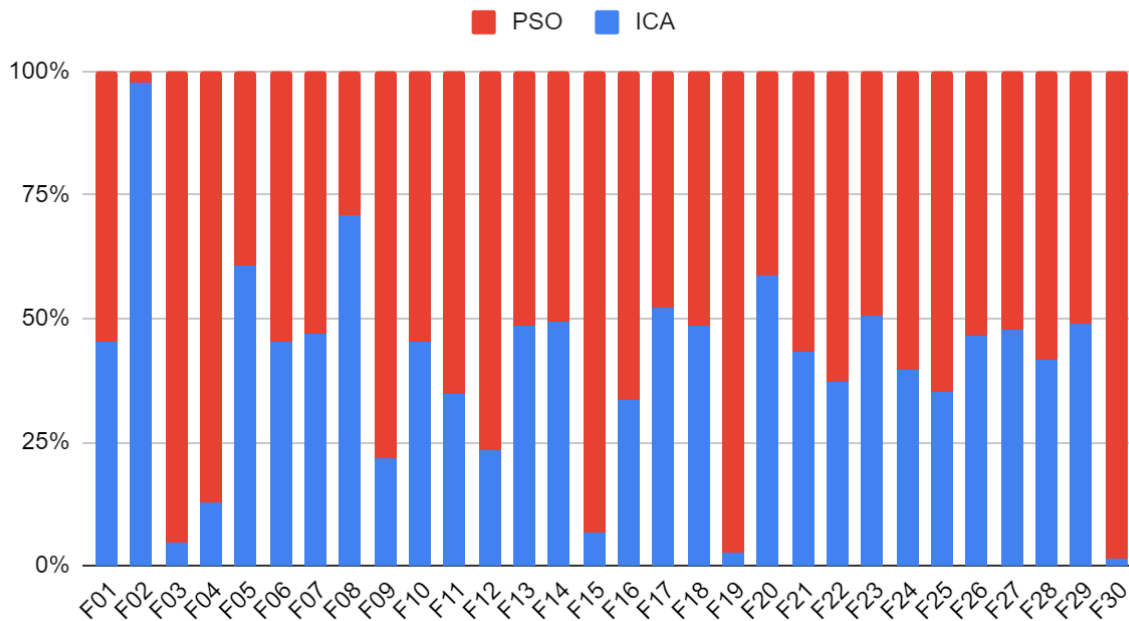
100%:

ICA:

	ICA	PSO
Best	2,40E+01	6,00E+00
F01	4,34E+07	5,26E+07
F02	4,00E+01	1,00E+00
F03	9,71E+01	1,99E+03
F04	6,79E+00	4,68E+01
F05	5,00E+01	3,21E+01
F06	8,27E+00	1,00E+01
F07	3,79E+01	4,28E+01
F08	5,33E+01	2,20E+01
F09	1,60E+01	5,69E+01
F10	8,94E+02	1,08E+03
F11	2,03E+01	3,84E+01
F12	7,63E+05	2,52E+06
F13	7,90E+03	8,41E+03
F14	9,75E+01	9,99E+01
F15	1,49E+02	2,07E+03
F16	7,12E+01	1,41E+02
F17	7,15E+01	6,50E+01
F18	1,39E+04	1,48E+04
F19	8,91E+01	3,22E+03
F20	1,20E+02	8,44E+01
F21	1,01E+02	1,32E+02
F22	4,56E+01	7,74E+01
F23	3,37E+02	3,30E+02
F24	1,19E+02	1,81E+02
F25	2,44E+02	4,48E+02
F26	3,22E+02	3,73E+02
F27	3,76E+02	4,13E+02
F28	3,36E+02	4,70E+02
F29	3,05E+02	3,19E+02

F30	1,05E+04	6,35E+05
------------	----------	----------

100% - ICA vs PSO

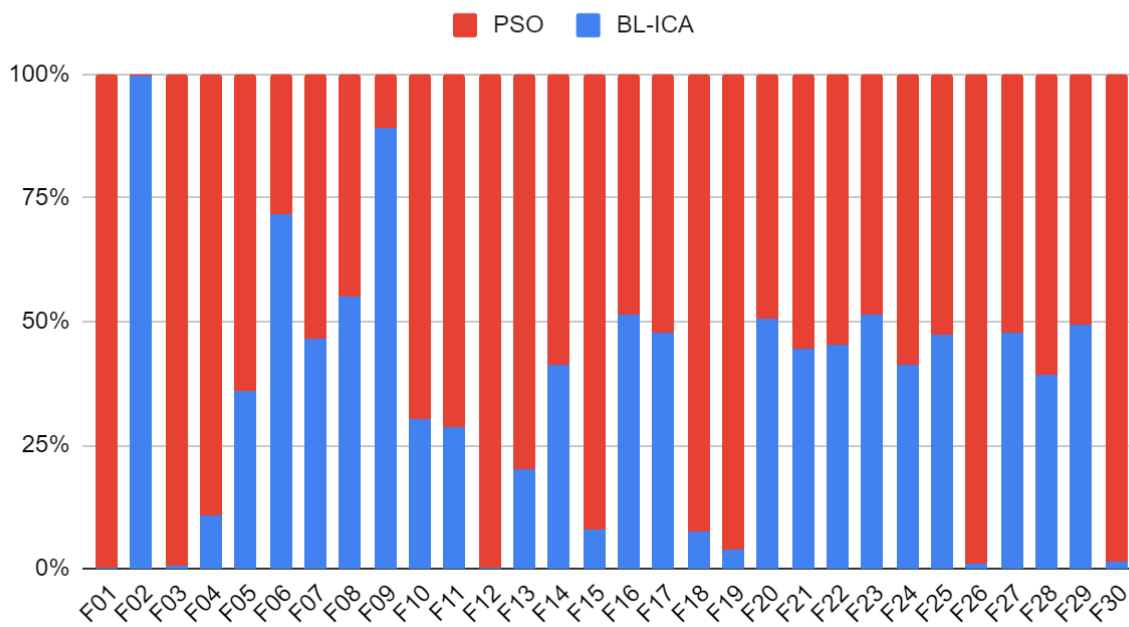


BL-ICA:

	BL-ICA	PSO
Best	2,30E+01	7,00E+00
F01	6,36E+04	5,26E+07
F02	3,85E+02	1,00E+00
F03	1,67E+01	1,99E+03
F04	5,55E+00	4,68E+01
F05	1,80E+01	3,21E+01
F06	2,53E+01	1,00E+01
F07	3,72E+01	4,28E+01
F08	2,69E+01	2,20E+01
F09	4,69E+02	5,69E+01
F10	4,68E+02	1,08E+03
F11	1,55E+01	3,84E+01
F12	8,10E+03	2,52E+06
F13	2,14E+03	8,41E+03
F14	7,02E+01	9,99E+01

F15	1,80E+02	2,07E+03
F16	1,50E+02	1,41E+02
F17	5,94E+01	6,50E+01
F18	1,19E+03	1,48E+04
F19	1,32E+02	3,22E+03
F20	8,65E+01	8,44E+01
F21	1,05E+02	1,32E+02
F22	6,42E+01	7,74E+01
F23	3,48E+02	3,30E+02
F24	1,28E+02	1,81E+02
F25	4,00E+02	4,48E+02
F26	3,41E+00	3,73E+02
F27	3,80E+02	4,13E+02
F28	3,03E+02	4,70E+02
F29	3,11E+02	3,19E+02
F30	1,04E+04	6,35E+05

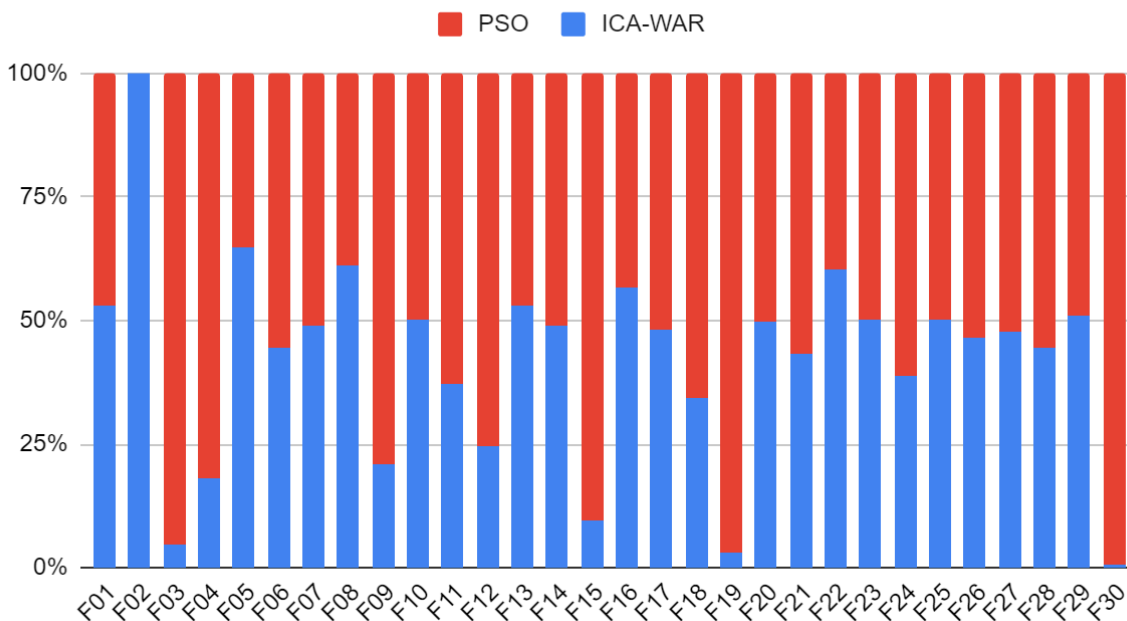
100% - BL-ICA vs PSO



ICA-WAR:

	ICA-WAR	PSO
Best	1,90E+01	1,10E+01
F01	5,89E+07	5,26E+07
F02	1,62E+03	1,00E+00
F03	9,91E+01	1,99E+03
F04	1,03E+01	4,68E+01
F05	5,88E+01	3,21E+01
F06	8,02E+00	1,00E+01
F07	4,09E+01	4,28E+01
F08	3,49E+01	2,20E+01
F09	1,51E+01	5,69E+01
F10	1,08E+03	1,08E+03
F11	2,27E+01	3,84E+01
F12	8,16E+05	2,52E+06
F13	9,44E+03	8,41E+03
F14	9,56E+01	9,99E+01
F15	2,20E+02	2,07E+03
F16	1,84E+02	1,41E+02
F17	5,99E+01	6,50E+01
F18	7,68E+03	1,48E+04
F19	1,04E+02	3,22E+03
F20	8,41E+01	8,44E+01
F21	1,01E+02	1,32E+02
F22	1,17E+02	7,74E+01
F23	3,31E+02	3,30E+02
F24	1,15E+02	1,81E+02
F25	4,50E+02	4,48E+02
F26	3,24E+02	3,73E+02
F27	3,76E+02	4,13E+02
F28	3,74E+02	4,70E+02
F29	3,33E+02	3,19E+02
F30	4,25E+03	6,35E+05

100% - ICA-WAR vs PSO



10. Análisis.

ICA vs PSO

Resultados Comparativos

En las evaluaciones realizadas, el Algoritmo de Colonización Imperialista (ICA) mostró un rendimiento significativamente superior al Particle Swarm Optimization (PSO) en todas las etapas del proceso de búsqueda. Con el 10% de las evaluaciones, el ICA superó al PSO en 27 de las 30 funciones del benchmark, mientras que el PSO solo destacó en 3 funciones. A medida que el número de evaluaciones aumentó al 50%, el ICA mantuvo su ventaja, siendo mejor en 23 funciones frente a 7 del PSO. Incluso al alcanzar el 100% de las evaluaciones, el ICA continuó mostrando un mejor desempeño en 24 funciones, comparado con las 6 en las que el PSO fue superior.

Análisis de Resultados

La superioridad del ICA sobre el PSO puede atribuirse a las diferencias fundamentales en la naturaleza de ambos algoritmos. El ICA, inspirado en el proceso de colonización imperialista, enfatiza tanto la explotación como la exploración mediante la asimilación de colonias y la competencia entre imperios. Esto le permite mantener un equilibrio entre la exploración de nuevas soluciones y la explotación de las mejores encontradas, adaptándose eficazmente a diversos paisajes de funciones.

En contraste, el PSO, basado en el comportamiento de enjambres, tiende a ser más eficiente en la explotación temprana, enfocándose en la convergencia hacia la mejor solución encontrada por el enjambre. Sin embargo, su capacidad de exploración puede verse limitada, especialmente en problemas de alta dimensionalidad o en paisajes de funciones complejas con múltiples óptimos locales. Esta limitación se hace más evidente en los resultados obtenidos, donde el PSO fue superado consistentemente por el ICA en la mayoría de las funciones, especialmente en las primeras etapas del proceso de búsqueda.

ICA vs BL-ICA vs ICA-WAR

Resultados Comparativos

10% de Evaluaciones:

- ICA vs BL-ICA: ICA fue mejor en 27 funciones, mientras que BL-ICA fue superior en 20.
- ICA vs ICA-WAR: ICA fue mejor en 28 funciones, mientras que ICA-WAR destacó en 28 funciones.
- BL-ICA vs ICA-WAR: BL-ICA fue mejor en 20 funciones, mientras que ICA-WAR destacó en 28.

50% de Evaluaciones:

- ICA vs BL-ICA: ICA fue mejor en 23 funciones, mientras que BL-ICA fue superior en 25.
- ICA vs ICA-WAR: ICA fue mejor en 25 funciones, mientras que ICA-WAR destacó en 25 funciones.
- BL-ICA vs ICA-WAR: BL-ICA fue mejor en 25 funciones, mientras que ICA-WAR destacó en 25.

100% de Evaluaciones:

- ICA vs BL-ICA: ICA fue mejor en 24 funciones, mientras que BL-ICA fue superior en 23.
- ICA vs ICA-WAR: ICA fue mejor en 19 funciones, mientras que ICA-WAR destacó en 19.
- BL-ICA vs ICA-WAR: BL-ICA fue mejor en 23 funciones, mientras que ICA-WAR destacó en 19.

Análisis de Resultados

La comparación entre ICA, BL-ICA, e ICA-WAR muestra cómo las mejoras propuestas afectan el rendimiento del algoritmo base en distintas etapas de la búsqueda.

ICA vs BL-ICA:

BL-ICA incorpora un mecanismo de búsqueda local aplicado a los países imperialistas después de cada iteración. Esto permite una mayor explotación de las soluciones cercanas a los mejores individuos del imperio, refinando las soluciones encontradas y mejorando la

precisión del algoritmo. Esto es particularmente útil en las primeras etapas de la búsqueda, donde la explotación de las soluciones prometedoras puede llevar a una convergencia más rápida hacia los óptimos locales. Los resultados muestran que BL-ICA tiende a superar al ICA en las etapas intermedias y finales del proceso de búsqueda, destacando su capacidad para mejorar la precisión de las soluciones finales.

ICA vs ICA-WAR:

ICA-WAR introduce un elemento competitivo adicional mediante guerras entre imperios, basadas en la proximidad de los imperialistas. Este mecanismo fomenta la diversidad y evita la convergencia prematura al redistribuir colonias entre los imperios en conflicto. Al mantener una mayor diversidad en la población, ICA-WAR mejora la exploración del espacio de búsqueda, lo que es particularmente efectivo en problemas con paisajes de funciones complejas y múltiples óptimos locales. Los resultados muestran que ICA-WAR supera consistentemente al ICA estándar, especialmente en las primeras etapas del proceso de búsqueda, donde la diversidad es crucial para evitar la trampa en óptimos locales.

BL-ICA vs ICA-WAR:

La comparación directa entre BL-ICA e ICA-WAR resalta las diferentes fortalezas de las mejoras propuestas. BL-ICA es más efectivo en la explotación, mejorando la precisión de las soluciones en las etapas intermedias y finales de la búsqueda. Por otro lado, ICA-WAR sobresale en la exploración, manteniendo una mayor diversidad y evitando la convergencia prematura, lo que es especialmente beneficioso en las primeras etapas de la búsqueda.

En resumen, las mejoras propuestas no solo incrementaron la robustez y la eficiencia del ICA, sino que también destacaron las ventajas de incorporar mecanismos adicionales de exploración y explotación. La comparación entre ICA, BL-ICA, e ICA-WAR subraya la importancia de un equilibrio adecuado entre estos dos aspectos en el diseño de metaheurísticas, mostrando cómo las adaptaciones específicas pueden abordar las limitaciones inherentes de cada enfoque.

11. Conclusión.

En esta práctica, se ha profundizado en el Algoritmo Imperialista Competitivo (ICA) y sus variantes, evaluando su rendimiento en un conjunto de funciones de benchmark del cec17. El ICA, inspirado en el proceso de imperialismo y colonización, ha demostrado ser una herramienta poderosa en el campo de la optimización. Este algoritmo utiliza una analogía directa con la geopolítica, donde los países más fuertes (imperialistas) dominan a los más débiles (colonias), absorbiéndolos y asimilándolos para formar imperios más poderosos. A lo largo del tiempo, este proceso de asimilación y competencia conduce a la mejora continua de las soluciones.

Una de las principales mejoras introducidas en esta práctica fue el BL-ICA, que incorpora un mecanismo de búsqueda local, aplicando refinamientos adicionales a las soluciones de los países imperialistas. Este enfoque híbrido permite un balance más preciso entre la

exploración global y la explotación local, mejorando la precisión de las soluciones encontradas y demostrando ser especialmente efectivo en las etapas intermedias y finales del proceso de optimización.

Además, se introdujo el ICA-WAR, una variante que simula conflictos bélicos entre imperios. Esta mejora añade una capa adicional de dinamismo al algoritmo, permitiendo que los imperios compitan por las colonias cuando se encuentran en proximidad. Este mecanismo de guerra no solo fomenta la diversidad de las soluciones, sino que también previene la convergencia prematura hacia óptimos locales al mantener una competencia activa entre diferentes regiones del espacio de búsqueda.

La experimentación comparativa entre el ICA, BL-ICA, ICA-WAR y el algoritmo de optimización por enjambre de partículas (PSO) demostró que el ICA y sus variantes superan consistentemente al PSO en la mayoría de las funciones de benchmark. Esto se debe a la capacidad del ICA para equilibrar la exploración y la explotación de manera efectiva, adaptándose a la estructura del problema y ajustando sus parámetros dinámicamente.

En resumen, el ICA, con su inspiradora analogía al imperialismo, ofrece una metodología robusta y eficaz para la optimización. Las mejoras propuestas en esta práctica, incluyendo el BL-ICA y el ICA-WAR, han demostrado aumentar significativamente la eficiencia y efectividad del algoritmo, abordando tanto la precisión en la búsqueda de soluciones como la diversidad necesaria para evitar la convergencia prematura. Estos resultados destacan la versatilidad del ICA y su potencial para enfrentar diversos desafíos de optimización, subrayando la importancia de la innovación continua en el diseño y ajuste de algoritmos metaheurísticos.

Fuentes:

https://www.researchgate.net/publication/224302036_Imperialist_Competitive_Algorithm_An_Algorithm_for_Optimization_Inspired_by_Imperialistic_Competition

<https://www.sciencedirect.com/science/article/abs/pii/S1568494614003895?via=ihub>

Trabajo realizado por Santiago Herron Mulet