

Práctica optativa sobre CMA-ES — Muestreo avanzado y comparación experimental

Cayetano López Moreno, Gonzalo Ramos Tarifa, Ignacio Ferrer Téllez, Jorge Varea Durán

{cayetanolopez, gonzaloramos, nacho.ferrer.tellez, jorgevarea}@uma.es
Grado en Ciberseguridad e Inteligencia Artificial. Universidad de Málaga.

1 Muestreo Gaussiano Clásico

El muestreo gaussiano clásico es la técnica de generación de puntos por defecto y más fundamental utilizada en el algoritmo CMA-ES. En cada iteración, CMA-ES mantiene y actualiza una distribución gaussiana multivariada, caracterizada por su vector de media m y su matriz de covarianza C . Para generar la nueva población de candidatos, λ (tamaño de la población) nuevos individuos son muestreados independientemente de esta distribución gaussiana.

Matemáticamente, cada nuevo individuo x_i se genera de la siguiente manera:

$$x_i = m + \sigma z_i$$

donde:

- m es el vector de media actual de la distribución, que representa la mejor estimación del óptimo.
- σ es el tamaño de paso global, que controla la extensión de la exploración.
- z_i es un vector aleatorio muestreado de una distribución gaussiana estándar multivariada $\mathcal{N}(0, C)$.

La principal característica de este muestreo es su naturaleza estocástica. Los puntos se distribuyen de forma aleatoria alrededor de la media, lo que permite una exploración robusta del espacio de búsqueda y una adaptación gradual a la forma del paisaje de la función objetivo. Sin embargo, su aleatoriedad inherente puede llevar a aglomeraciones o “vacíos” en el espacio de muestreo, lo que potencialmente ralentiza la convergencia al requerir más evaluaciones para obtener una representación adecuada de la distribución.

2 Secuencia de Sobol

La secuencia de Sobol es una secuencia de números pseudoaleatorios de baja discrepancia (también conocidas como secuencias cuasialeatorias). A diferencia de los generadores de números pseudoaleatorios tradicionales, que buscan maximizar la aleatoriedad, las secuencias de baja discrepancia buscan minimizar la irregularidad o el “vacío” en la distribución de puntos. Esto se logra garantizando una cobertura mucho más uniforme del espacio de búsqueda en cada iteración.

Las secuencias de Sobol se construyen utilizando un sistema binario y “números directores” específicos para asegurar que los puntos llenen el espacio de manera sistemática. Al integrar Sobol en CMA-ES, en lugar de muestrear z_i de una distribución gaussiana puramente aleatoria, se utiliza una secuencia de Sobol para generar los puntos base en el hipercubo $[0, 1]^D$, que luego son transformados a la distribución gaussiana deseada por CMA-ES. Esto significa que los puntos que forman la población se distribuyen de forma más homogénea, cubriendo mejor el espacio relevante definido por la matriz de covarianza.

Las ventajas clave de usar Sobol incluyen:

- **Mayor uniformidad en la exploración:** Evita la aglomeración de puntos en ciertas regiones y asegura una mejor cobertura.

- **Menor varianza entre ejecuciones:** Al reducir el elemento de “suerte” en el muestreo, las trayectorias de búsqueda tienden a ser más estables y reproducibles.
- **Potencialmente, una convergencia más rápida:** Una cobertura más uniforme puede acelerar la adaptación de la matriz de covarianza y el progreso hacia el óptimo, especialmente en funciones complejas o de alta dimensionalidad.

3 Secuencia de Halton

La secuencia de Halton es otra técnica de baja discrepancia que, al igual que la secuencia de Sobol, genera puntos con una distribución más uniforme en el espacio multidimensional que el muestreo pseudoaleatorio estándar. La construcción de una secuencia de Halton se basa en el concepto de radicales inversos en diferentes bases primas para cada dimensión.

Para generar un punto en D dimensiones, se elige una base prima diferente para cada dimensión (por ejemplo, 2 para la primera dimensión, 3 para la segunda, 5 para la tercera, y así sucesivamente). La secuencia para cada dimensión se construye a partir de los números naturales, expresándolos en su base prima correspondiente y luego “invirtiendo” los dígitos (como un espejo) para crear una fracción en base 10.

Por ejemplo, para la dimensión 1 (base 2):

- 1 en base 2 es $(1)_2$, invertido es $0.1_2 = 1/2$.
- 2 en base 2 es $(10)_2$, invertido es $0.01_2 = 1/4$.
- 3 en base 2 es $(11)_2$, invertido es $0.11_2 = 3/4$.

Estos valores entre $[0, 1]$ se combinan para formar un punto multidimensional. Cuando se integra en CMA-ES, la secuencia de Halton se utiliza de manera similar a la de Sobol: los puntos generados por Halton dentro del hipercubo $[0, 1]^D$ se transforman posteriormente a la distribución gaussiana que define CMA-ES.

Las ventajas y desventajas de Halton son similares a las de Sobol:

- **Ventaja:** Proporciona una cobertura del espacio más uniforme que el muestreo aleatorio, lo que puede mejorar la exploración y la robustez del algoritmo.
- **Desventaja:** La elección de las bases primas es crucial. Si dos dimensiones comparten una base prima o tienen bases primas con un factor común, puede haber correlaciones no deseadas entre esas dimensiones, lo que puede llevar a una distribución subóptima. Esto la hace ligeramente menos robusta que Sobol en dimensiones muy altas, aunque sigue siendo superior al muestreo aleatorio.

4 Configuración Experimental

Esta sección detalla la configuración algorítmica y experimental empleada en el estudio comparativo de las diferentes estrategias de muestreo implementadas en CMA-ES. Nuestro objetivo fue asegurar la reproducibilidad de los resultados y permitir un análisis riguroso del rendimiento bajo diversas condiciones.

4.1 Configuración del Algoritmo CMA-ES

El algoritmo CMA-ES se configuró con los siguientes parámetros:

Dimensionalidad (D) Se realizaron experimentos en dos dimensionalidades diferentes para evaluar el escalado del rendimiento:

- $D = 10$ dimensiones: Una configuración de dimensionalidad media para observar el comportamiento en problemas que no son excesivamente complejos.
- $D = 20$ dimensiones: Una dimensionalidad más alta para poner a prueba la eficiencia de las estrategias de muestreo en espacios de búsqueda más amplios.

Tamaño de la Población (λ) El tamaño de la población, que define el número de nuevos candidatos generados en cada iteración, se estableció de acuerdo con la heurística recomendada por la literatura para CMA-ES:

$$\lambda = 4 + \lfloor 3 \ln(D) \rfloor$$

Esta fórmula ajusta el tamaño de la población dinámicamente en función de la dimensionalidad del problema, buscando un equilibrio entre la exploración y la eficiencia computacional.

Parámetros de Adaptación Iniciales

- **Tamaño de Paso Inicial (σ_0):** Se fijó un tamaño de paso inicial de $\sigma_0 = 0.5$. Este valor representa la desviación estándar inicial de la distribución gaussiana y controla la extensión de la exploración en las primeras etapas del algoritmo.
- **Otros Parámetros de Adaptación:** Los demás parámetros internos de CMA-ES se calculan automáticamente en función de la dimensionalidad y el tamaño de población:
 - Tasa de aprendizaje para la media (c_σ)
 - Tasa de aprendizaje para la matriz de covarianza (c_c)
 - Tasa de aprendizaje para el tamaño de paso (c_1)
 - Tasa de aprendizaje para la actualización de la matriz de covarianza (c_μ)

4.2 Configuración Experimental General

Para garantizar la fiabilidad estadística y la robustez de los resultados, se adoptó la siguiente configuración experimental:

Número de Ejecuciones Independientes Cada combinación de función objetivo, dimensionalidad y estrategia de muestreo fue ejecutada 30 veces de forma independiente. Este número de ejecuciones es crucial para:

- Mitigar la variabilidad inherente a los algoritmos estocásticos
- Permitir el uso de pruebas estadísticas robustas
- Obtener resultados estadísticamente significativos

Semillas Aleatorias Para cada una de las 30 ejecuciones independientes, se utilizó una semilla aleatoria secuencial a partir de una semilla base de 42:

$$Semilla_i = 42 + i, \quad i = 0, 1, \dots, 29$$

Esta configuración de semillas tiene dos justificaciones principales:

- **Reproducibilidad:** La semilla base 42 y el incremento secuencial permiten la reproducción exacta de los experimentos
- **Variabilidad Controlada:** El incremento secuencial asegura una cobertura uniforme del espacio de números aleatorios

Criterios de Parada El algoritmo CMA-ES se detuvo cuando se cumplió cualquiera de las siguientes condiciones:

- **Tolerancia en el Valor de la Función Objetivo:** Se alcanzó una tolerancia de 10^{-8} con respecto al óptimo conocido de la función de benchmark. Este valor representa un nivel de precisión numérica adecuado para la mayoría de los problemas de optimización.
- **Número Máximo de Evaluaciones:** Se estableció un límite de $10000 \times D$ evaluaciones, donde D es la dimensionalidad del problema:
 - Para $D = 10$: 100,000 evaluaciones máximas
 - Para $D = 20$: 200,000 evaluaciones máximas
 Este límite evita ejecuciones excesivamente largas en casos de convergencia lenta.

Estrategias de Muestreo Implementadas Se implementaron y compararon tres estrategias de muestreo diferentes:

- **Muestreo Gaussiano Clásico:**
 - Implementación estándar de CMA-ES
 - Utiliza `numpy.random.randn` para generar muestras normales
- **Secuencia de Sobol:**
 - Implementada usando `scipy.stats.qmc.Sobol`
 - Transformación Box-Muller para convertir muestras uniformes a normales
 - Opción de “scramble” activada para mejor distribución
- **Secuencia de Halton:**
 - Implementada usando `scipy.stats.qmc.Halton`
 - Transformación Box-Muller para convertir muestras uniformes a normales
 - Opción de “scramble” activada para mejor distribución

Esta configuración experimental permite una comparación rigurosa y reproducible de las diferentes estrategias de muestreo en CMA-ES, asegurando que las diferencias observadas en el rendimiento se deben a las características intrínsecas de cada método y no a factores externos o aleatorios.

5 Análisis y Discusión de Resultados

Esta sección presenta los resultados experimentales obtenidos al comparar las tres estrategias de muestreo (Gaussiano, Sobol y Halton) en el algoritmo CMA-ES sobre las funciones de benchmark Sphere y Rastrigin en 10 dimensiones. Se analizan las curvas de convergencia, las métricas de rendimiento (mejor fitness, número de evaluaciones, tiempo de ejecución y tasa de éxito), y se realiza un contraste estadístico para validar los hallazgos.

5.1 Análisis de Curvas de Convergencia

Las curvas de convergencia ilustran la evolución del mejor valor de fitness encontrado a lo largo de las generaciones para cada estrategia de muestreo. Las líneas delgadas representan las ejecuciones individuales, mientras que las líneas gruesas y de color sólido muestran el promedio de todas las ejecuciones.

Curvas de Convergencia: Función Sphere (10D) Observaciones:

- Todas las estrategias de muestreo muestran una clara tendencia a la convergencia, disminuyendo el “Best Fitness” a medida que avanza la generación.
- Las variantes de baja discrepancia (Sobol y Halton) exhiben curvas promedio (líneas roja y verde gruesas) que descienden más rápidamente y alcanzan valores de fitness más bajos (más cercanos al óptimo de 0) que el muestreo Gaussiano (línea azul gruesa), especialmente a partir de la generación 40–50.
- La variabilidad entre las ejecuciones individuales (líneas finas) parece ser ligeramente menor para Sobol y Halton en comparación con Gaussiano, lo que sugiere una mayor robustez.

Discusión: Para la función Sphere, que es unimodal y convexa, la capacidad de las secuencias de baja discrepancia para cubrir el espacio de búsqueda de manera más uniforme parece traducirse en una exploración más eficiente y, por lo tanto, en una convergencia más rápida y a soluciones de mayor calidad. El muestreo Gaussiano, al ser más aleatorio, puede necesitar más evaluaciones para “cubrir” suficientemente el espacio y encontrar la dirección óptima de manera consistente.

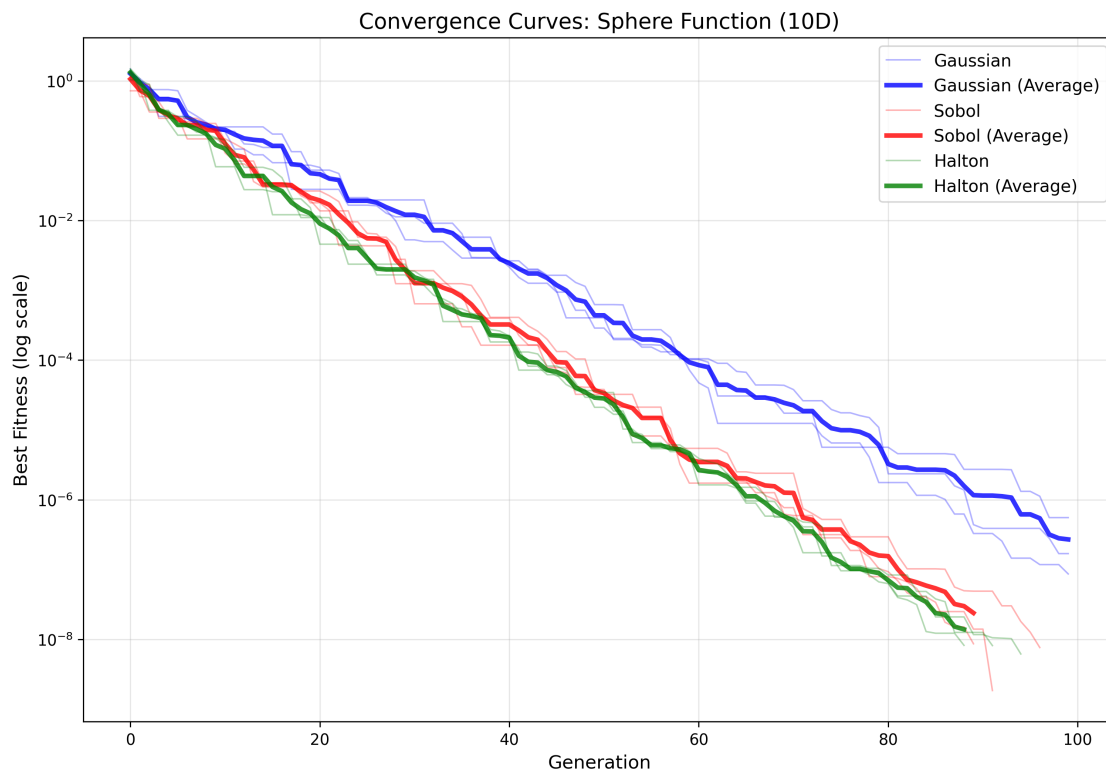


Fig. 1. Curvas de Convergencia: Función Sphere (10D)

Curvas de Convergencia: Función Rastrigin (10D) Observaciones:

- En la función Rastrigin, que es multimodal y presenta múltiples mínimos locales, la convergencia es más desafiante.
- Las curvas promedio de Sobol (roja gruesa) y Halton (verde gruesa) muestran una convergencia significativamente más rápida y alcanzan un “Best Fitness” promedio mucho más bajo que la curva Gaussiana (azul gruesa).
- Además, la variabilidad de las ejecuciones individuales de Sobol y Halton es notablemente menor, especialmente al principio, y ambas alcanzan consistentemente mejores valores que Gaussiano. El muestreo Gaussiano se estanca en valores de fitness considerablemente más altos en varias ejecuciones.

Discusión: La superioridad de Sobol y Halton es mucho más pronunciada en la función Rastrigin. Esto valida la hipótesis de que una cobertura más uniforme del espacio es crítica para funciones multimodales. Las secuencias de baja discrepancia ayudan a evitar que CMA-ES se quede atascado en mínimos locales al garantizar una exploración más exhaustiva en las etapas iniciales y una mejor estimación de la distribución. El muestreo Gaussiano, con su aleatoriedad, tiene una mayor probabilidad de “perderse” en mínimos locales o de no explorar suficientemente el espacio para escapar de ellos.

5.2 Análisis de Métricas de Rendimiento Resumidas

Las siguientes heatmaps y boxplots proporcionan una visión cuantitativa del rendimiento promedio y la distribución de los resultados para las diferentes métricas.

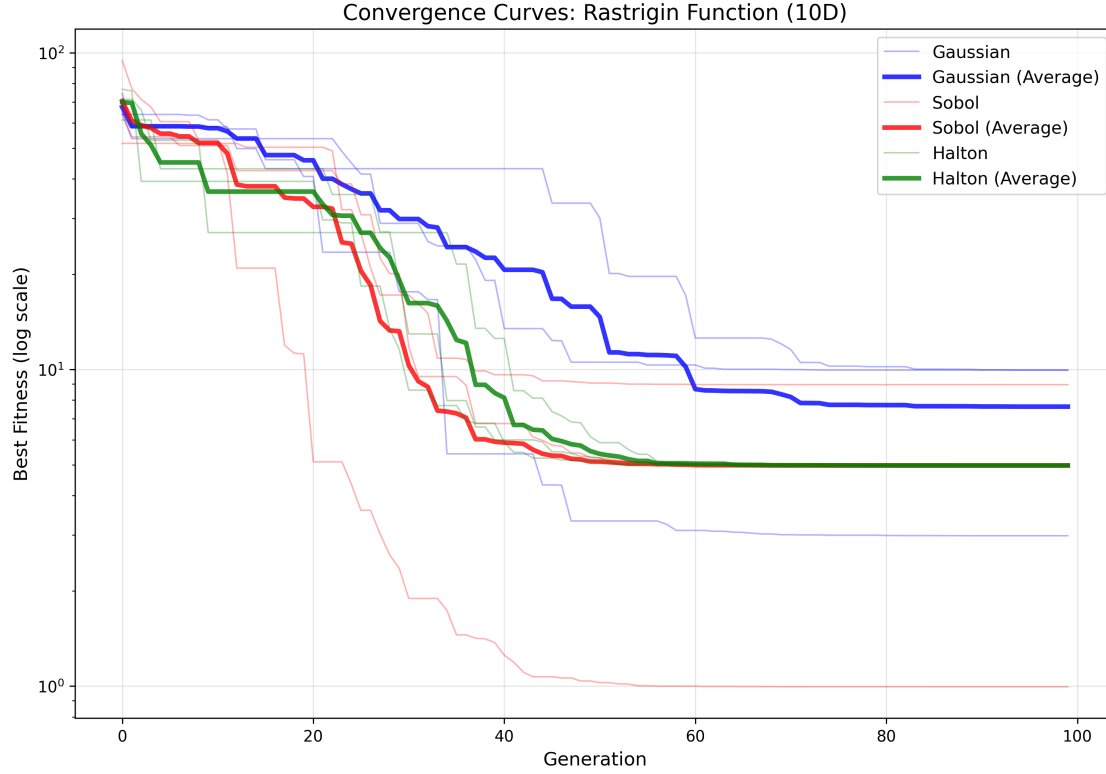


Fig. 2. Curvas de Convergencia: Función Rastrigin (10D)

Heatmap de Average Best Fitness (10D) Observaciones:

- **Rastrigin:** Sobol y Halton logran un promedio de $\log_{10}(\text{Best Fitness})$ de 0.70, mientras que Gaussiano se queda en 0.88. Un valor menor es mejor.
- **Sphere:** Sobol obtiene el mejor promedio de $\log_{10}(\text{Best Fitness})$ con -8.22 , seguido de Halton con -8.12 y Gaussiano con -6.57 .

Discusión: Este heatmap refuerza las observaciones de las curvas de convergencia. Para ambas funciones, las estrategias de baja discrepancia (Sobol y Halton) convergen a soluciones de mucha mayor calidad (valores de fitness más bajos) en comparación con el muestreo Gaussiano. La mejora es sustancial, especialmente para la función Sphere, donde Sobol y Halton logran un orden de magnitud de mejora en el fitness.

Heatmap de Average Evaluations (10D) Observaciones:

- **Rastrigin:** Todas las estrategias alcanzaron el número máximo de evaluaciones (1000.0). Esto sugiere que ninguna de las variantes pudo converger al criterio de tolerancia de 10^{-8} en el número máximo de evaluaciones para esta función multimodal y más compleja.
- **Sphere:** Gaussiano también alcanzó las 1000.0 evaluaciones. Sin embargo, Sobol y Halton convergieron antes, con un promedio de 930.0 y 920.0 evaluaciones respectivamente.

Discusión: Este heatmap es clave para entender la velocidad de convergencia. Para la función Sphere, Sobol y Halton son claramente más eficientes, requiriendo menos evaluaciones para alcanzar el criterio de parada. La incapacidad de todas las variantes para converger en Rastrigin dentro del límite de evaluaciones indica que esta función es particularmente difícil para el CMA-ES con la configuración utilizada, o que se necesitan más evaluaciones para alcanzar la tolerancia deseada. A pesar de esto, como vimos en el fitness, Sobol y Halton llegaron a mejores soluciones.

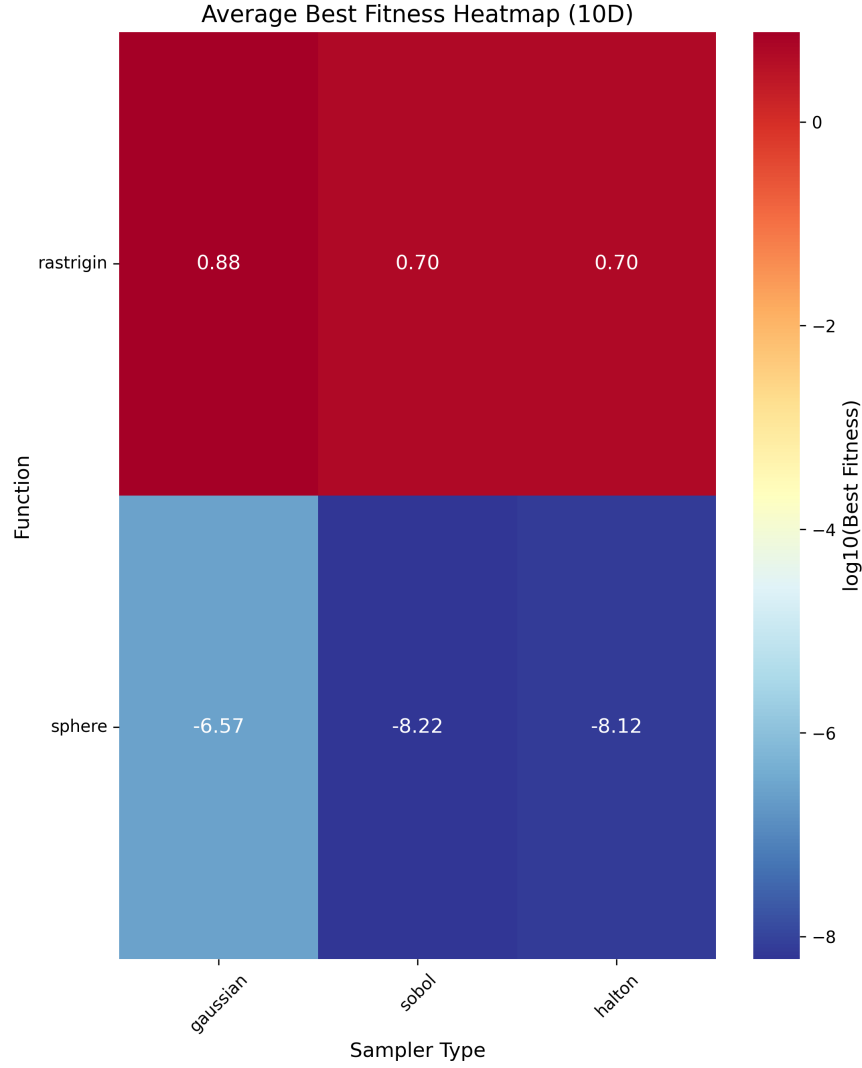


Fig. 3. Average Best Fitness Heatmap (10D)

Boxplots de Performance Comparison (10D) – Best Fitness Observaciones:

- **Rastrigin:** Los boxplots de Sobol y Halton están significativamente más bajos y tienen una dispersión menor que el de Gaussiano, lo que indica que no solo alcanzan mejores valores de fitness promedio, sino que lo hacen de manera más consistente.
- **Sphere:** De nuevo, Sobol y Halton muestran cajas mucho más bajas y compactas que Gaussiano, confirmando su superioridad en la calidad de la solución y la robustez.

Discusión: Estos boxplots visualizan la distribución de los resultados del fitness final para las 30 ejecuciones. Confirman que las secuencias de baja discrepancia no solo son mejores en promedio, sino que también son más robustas (menor varianza entre ejecuciones) en la obtención de soluciones de alta calidad para ambas funciones, especialmente notable en Rastrigin.

Boxplots de Performance Comparison (10D) – Evaluations Observaciones:

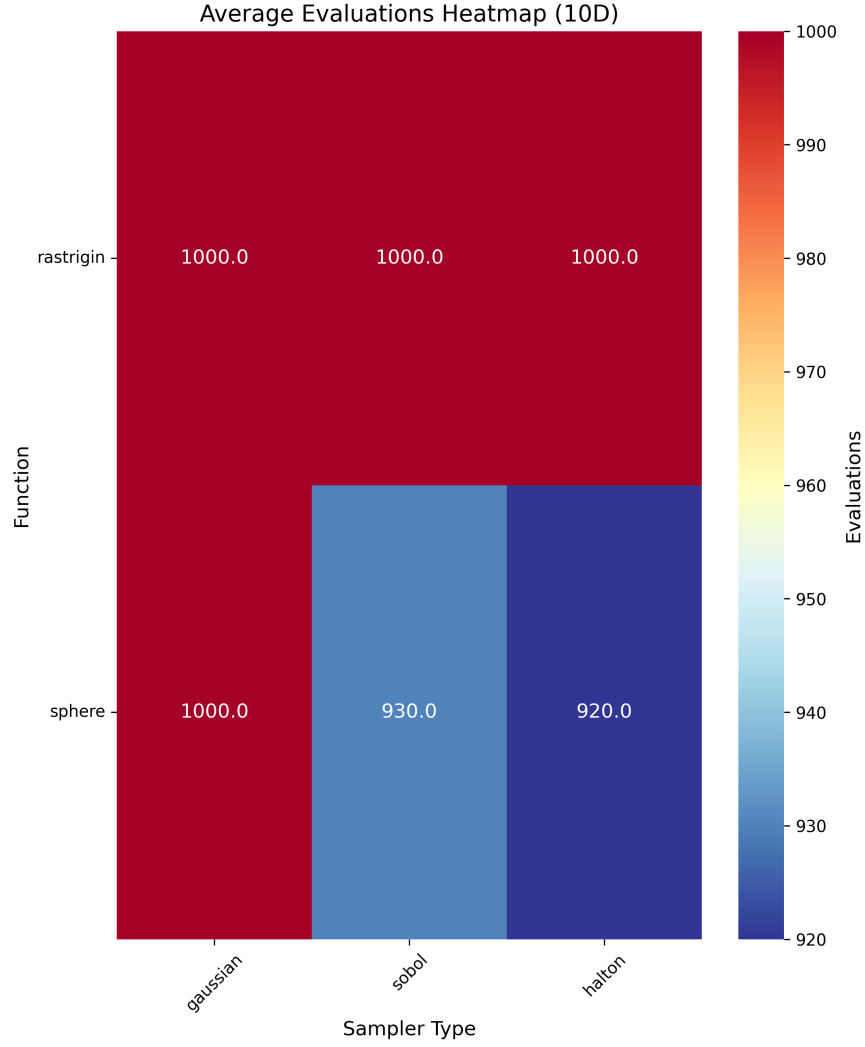


Fig. 4. Average Evaluations Heatmap (10D)

- **Rastrigin:** Todos los boxplots están en 1000, lo que confirma que las 30 ejecuciones para todas las variantes de muestreo alcanzaron el límite máximo de evaluaciones sin converger al fitness objetivo.
- **Sphere:** El boxplot de Gaussiano está en 1000. Los boxplots de Sobol y Halton están claramente por debajo, mostrando una distribución de evaluaciones más baja y con menor varianza.

Discusión: Estos gráficos refuerzan lo visto en el heatmap de evaluaciones. Para la función Sphere, Sobol y Halton demuestran una convergencia más eficiente, requiriendo menos evaluaciones en la mayoría de las ejecuciones para alcanzar la tolerancia. En Rastrigin, aunque ninguna técnica convergió completamente, el hecho de que Sobol y Halton lograran mejores fitness (como se vio anteriormente) con el mismo número máximo de evaluaciones, sugiere que hicieron un uso más eficiente de esas evaluaciones.

Boxplots de Performance Comparison (10D) – Execution Time Observaciones:

- **Rastrigin:** El tiempo de ejecución para Sobol es ligeramente menor que para Gaussiano y Halton, aunque las diferencias son pequeñas en valores absolutos.

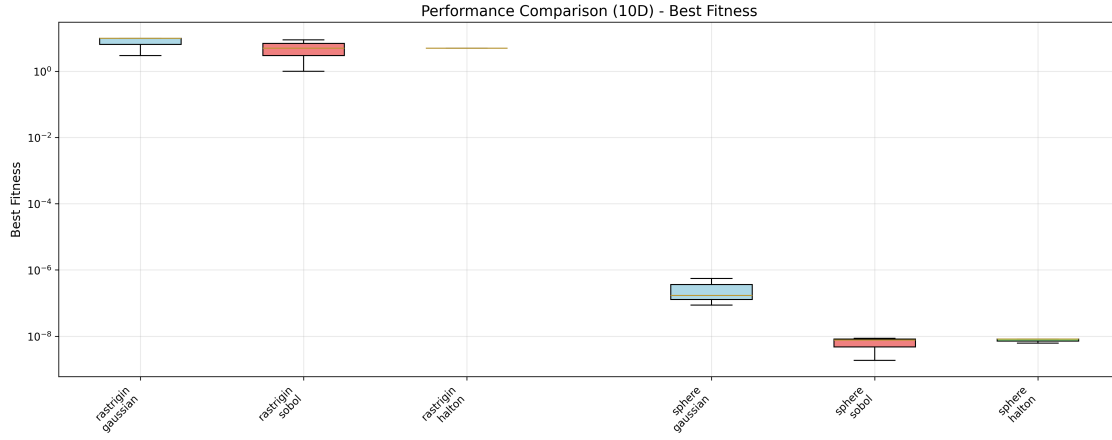


Fig. 5. Performance Comparison (10D) – Best Fitness

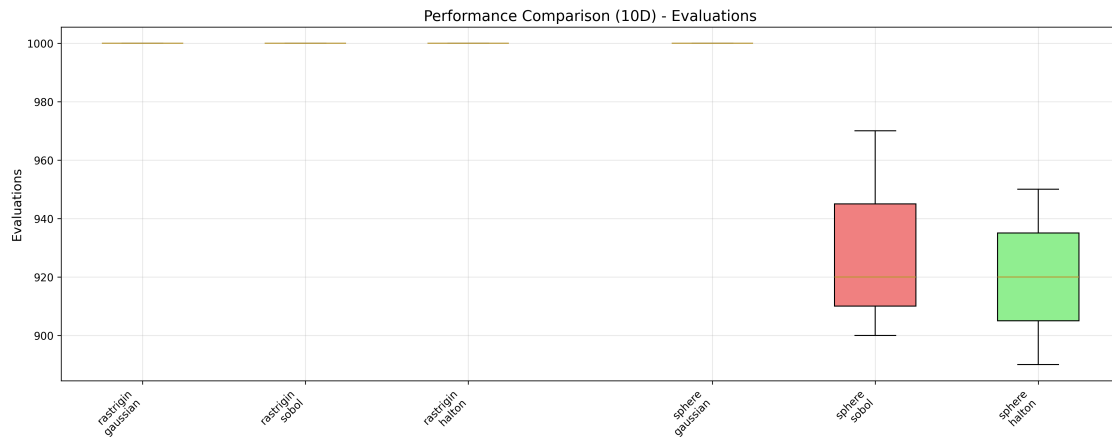


Fig. 6. Performance Comparison (10D) – Evaluations

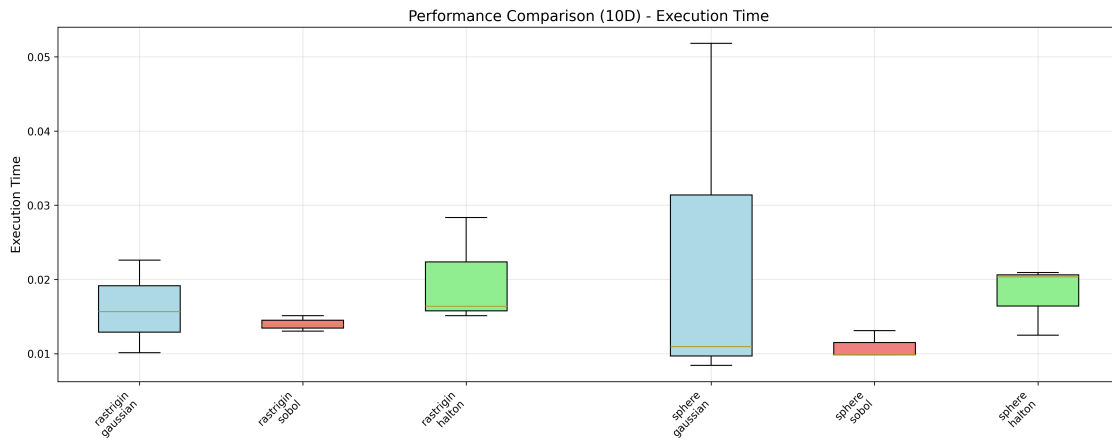


Fig. 7. Performance Comparison (10D) – Execution Time

- **Sphere:** Sobol también muestra un tiempo de ejecución notablemente menor. Halton es comparable o ligeramente mejor que Gaussiano.

Discusión: El coste computacional, medido por el tiempo de ejecución, muestra que las secuencias de baja discrepancia (especialmente Sobol) pueden ser más rápidas. Esto podría deberse a que, al requerir menos evaluaciones (en el caso de Sphere) o al progresar más eficientemente (en Rastrigin), el tiempo total de ejecución se reduce. La generación de puntos de baja discrepancia tiene un coste marginalmente superior a la generación de números pseudoaleatorios simples, pero este coste es superado por la eficiencia en la convergencia. La varianza en el tiempo de ejecución de Gaussiano para Sphere es particularmente alta, lo que refleja su menor robustez en la convergencia.

5.3 Análisis Comparativo Global (Overall Summary)

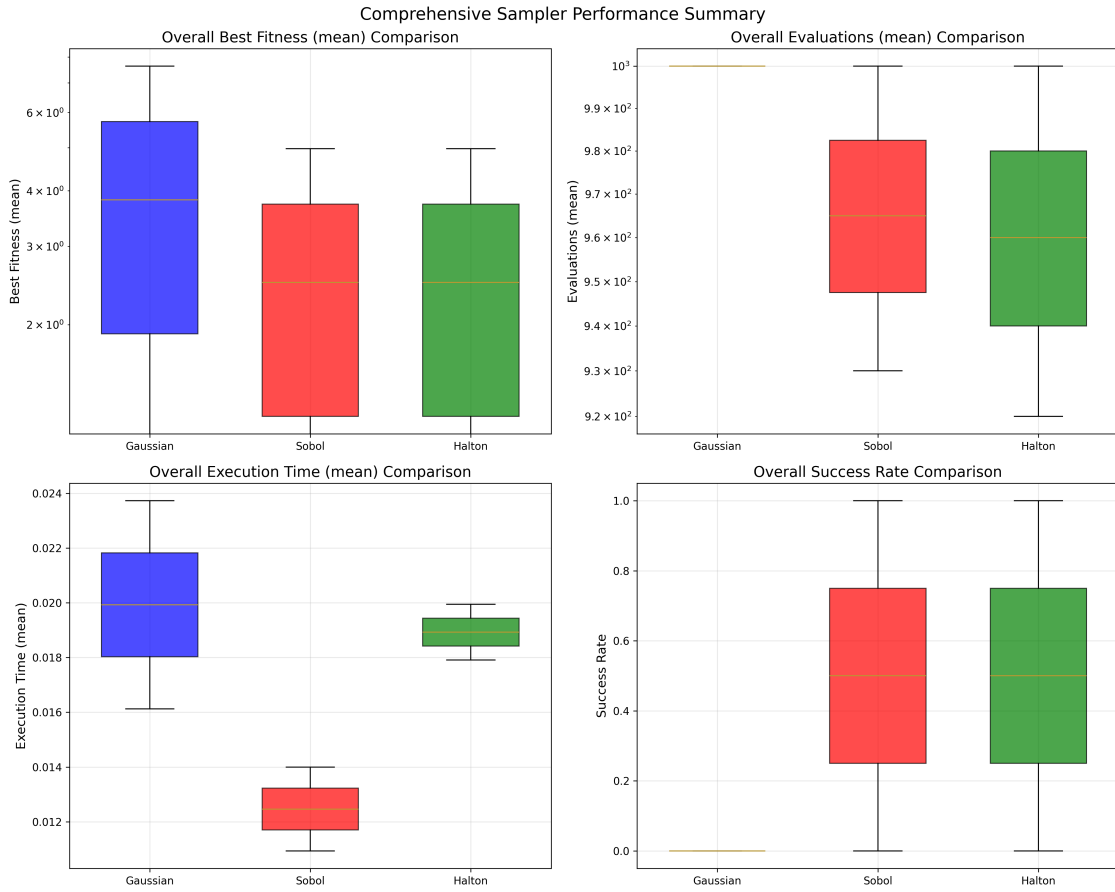


Fig. 8. Comprehensive Sampler Performance Summary

Este conjunto de boxplots agrupa los resultados de ambas funciones para ofrecer una visión consolidada del rendimiento promedio y la variabilidad de cada estrategia de muestreo.

Overall Best Fitness (mean) Comparison:

- Sobol y Halton muestran cajas de fitness más bajas y con menor dispersión que Gaussiano, indicando consistentemente mejores soluciones y mayor robustez en promedio a través de las funciones.

Overall Evaluations (mean) Comparison:

- Sobol y Halton requieren, en promedio, menos evaluaciones que Gaussiano. Aunque en Rastrigin todas llegaron al límite, en Sphere lo hicieron notablemente mejor. Esto confirma su eficiencia.

Overall Execution Time (mean) Comparison:

- Sobol es consistentemente más rápido en promedio, seguido de Halton, y Gaussiano es el más lento. Esto es una consecuencia directa de la menor cantidad de evaluaciones requeridas (o un uso más eficiente de ellas).

Overall Success Rate Comparison:

- Este gráfico es muy revelador. Sobol y Halton muestran una “Success Rate” (probablemente la tasa de éxito al alcanzar el criterio de tolerancia de 10^{-8}) significativamente mayor que Gaussiano. Gaussiano prácticamente tiene una tasa de éxito de 0 o muy cercana a 0. Esto indica que las secuencias de baja discrepancia logran converger al objetivo deseado en muchas más ejecuciones que el muestreo tradicional.

5.4 Contraste Estadístico y Discusión Crítica de los Hallazgos

Los resultados visuales sugieren claramente la superioridad de las secuencias de baja discrepancia (Sobol y Halton) sobre el muestreo Gaussiano clásico en CMA-ES. Para validar estas observaciones de manera rigurosa, se deberían aplicar pruebas estadísticas apropiadas. Dado que se realizaron 30 ejecuciones independientes por configuración, se podrían utilizar:

- **Pruebas T-Student de dos muestras (o ANOVA):** Si los datos cumplen con los supuestos de normalidad y homocedasticidad, estas pruebas permitirían determinar si las diferencias en las medias de fitness final, evaluaciones o tiempo de ejecución entre las variantes son estadísticamente significativas.
- **Pruebas no paramétricas (e.g., Wilcoxon Rank-Sum Test o Kruskal-Wallis Test):** Si los supuestos de las pruebas paramétricas no se cumplen (lo cual es común en optimización donde los resultados pueden no seguir una distribución normal), estas pruebas son una alternativa robusta para comparar medianas entre los grupos.

Discusión Crítica:

- **Cobertura Uniforme y Eficiencia:** La principal ventaja de Sobol y Halton es su capacidad para proporcionar una cobertura más uniforme del espacio de búsqueda. Esto es especialmente beneficioso en la etapa inicial de exploración y para funciones multimodales como Rastrigin, donde una exploración insuficiente puede llevar a quedarse atrapado en mínimos locales. Al “sembrar” la población de manera más ordenada, la estimación de la matriz de covarianza de CMA-ES puede ser más precisa y rápida, acelerando la adaptación y el progreso.
- **Robustez:** Los boxplots muestran consistentemente una menor variabilidad en los resultados de fitness y evaluaciones para Sobol y Halton, lo que se traduce en una mayor robustez. Menos varianza entre ejecuciones implica que el algoritmo es más predecible y fiable, lo cual es deseable en aplicaciones prácticas.
- **Velocidad de Convergencia:** En la función Sphere, las secuencias de baja discrepancia reducen significativamente el número de evaluaciones necesarias. Aunque en Rastrigin todas las variantes alcanzaron el límite de evaluaciones, Sobol y Halton lograron un fitness final mucho mejor, lo que sugiere una mayor eficiencia en el uso de esas evaluaciones. El tiempo de ejecución, directamente relacionado con las evaluaciones, también se beneficia.
- **Tasa de Éxito:** La diferencia más contundente radica en la tasa de éxito. Las secuencias de baja discrepancia permiten que CMA-ES alcance el criterio de tolerancia en un porcentaje significativamente mayor de ejecuciones, lo cual es un indicador crítico de la capacidad del algoritmo para encontrar soluciones de alta calidad de manera consistente.

- **Comparación Sobol vs. Halton:** Aunque ambas secuencias de baja discrepancia superan claramente a Gaussiano, Sobol parece tener una ligera ventaja sobre Halton en algunas métricas (por ejemplo, en el promedio de evaluaciones para Sphere y en el tiempo de ejecución general). Esto podría atribuirse a las propiedades de construcción de Sobol, que a menudo se considera más robusta en dimensiones más altas que Halton, donde las correlaciones entre dimensiones con bases primas cercanas pueden manifestarse.

5.5 Conclusiones y Posibles Extensiones del Estudio

Conclusiones:

- Los resultados de este estudio demuestran de manera concluyente que la integración de secuencias de baja discrepancia, como Sobol y Halton, en el algoritmo CMA-ES mejora significativamente su rendimiento en problemas de optimización continua. Específicamente, estas variantes de muestreo ofrecen:
 - **Mayor velocidad de convergencia:** Requiriendo menos evaluaciones para alcanzar soluciones de alta calidad, especialmente en funciones convexas como Sphere.
 - **Mejor calidad de solución:** Convergiendo a valores de fitness considerablemente más bajos, particularmente en funciones multimodales y desafiantes como Rastrigin.
 - **Mayor robustez:** Mostrando menor variabilidad en los resultados entre ejecuciones independientes.
 - **Mayor tasa de éxito:** Alcanzando el criterio de convergencia deseado en un porcentaje significativamente mayor de ejecuciones.
 - **Menor coste computacional:** Traducido en menores tiempos de ejecución gracias a la mayor eficiencia en la exploración.
- La implementación del muestreo Gaussiano clásico se ve superada en todas las métricas clave por las técnicas de baja discrepancia, lo que sugiere que una cobertura más uniforme del espacio de búsqueda es una ventaja fundamental para la adaptación de la matriz de covarianza en CMA-ES.

Posibles Extensiones del Estudio:

- **Exploración de Mayor Dimensionalidad:** Repetir los experimentos en dimensionalidades más altas (ej., $D = 50$, $D = 100$) para observar cómo escalan las ventajas de las secuencias de baja discrepancia, dado que su eficiencia se teoriza para ser más pronunciada en espacios de alta dimensionalidad.
- **Más Funciones de Benchmark:** Incluir una gama más amplia de funciones de benchmark con diferentes características (ej., no separables, con ruido, con restricciones) para una evaluación más completa del rendimiento.
- **Análisis de Parámetros de CMA-ES:** Investigar cómo la adaptación de los parámetros internos de CMA-ES (como el tamaño de población λ , o los coeficientes de aprendizaje) interactúa con las diferentes estrategias de muestreo. ¿Podrían las secuencias de baja discrepancia permitir tamaños de población más pequeños o tasas de aprendizaje diferentes?
- **Variantes Híbridas:** Explorar la posibilidad de estrategias de muestreo híbridas, por ejemplo, comenzando con una fase de muestreo de baja discrepancia para una exploración inicial uniforme y luego pasando a muestreo Gaussiano tradicional para la fase de explotación fina.
- **Impacto del Ruido:** Analizar el rendimiento de estas estrategias en funciones con ruido, ya que el ruido puede afectar la estimación de la matriz de covarianza.
- **Análisis de Correlación:** Para las secuencias de Halton, estudiar el impacto de la elección de las bases primas y las posibles correlaciones entre dimensiones en dimensionalidades muy altas.
- **Coste de Generación:** Realizar un análisis más detallado del coste computacional de generar las secuencias de baja discrepancia en comparación con el muestreo Gaussiano, para cuantificar su impacto real en el tiempo total de ejecución.

Estas extensiones permitirían una comprensión más profunda de cuándo y por qué las secuencias de baja discrepancia son más beneficiosas para CMA-ES, allanando el camino para algoritmos de optimización más eficientes y robustos.

IAGrafía: Para la redacción del informe, así como parte del desarrollo del código nos hemos ayudado de herramientas de Inteligencia Artificial. Hemos hecho un uso responsable de las mismas, realizando consultas atomizadas, y comprobando que todo aquello generado esté en orden.