

UNIVERSIDAD NACIONAL DEL SANTA
FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas e Informática



ALGORITMOS EVOLUTIVOS - 1411-2278
LABORATORIO SEMANA 08

REPRESENTACIONES CROMOSÓMICAS EN
ALGORITMOS GENÉTICOS

ALUMNO:

- Dueñas Blas Joseph Ali Ricardo

DOCENTE:

- Ms. Ing. Johan Max Alexander López Heredia

Nuevo Chimbote – Perú

2025

ÍNDICE

1. INTRODUCCIÓN	3
2. COMPARACIÓN DE REPRESENTACIONES	3
3. RESULTADOS DE LAS ACTIVIDADES.....	4
4. REFLEXIÓN: CUÁNDO USAR CADA REPRESENTACIÓN.....	5
5. RESPUESTAS A PREGUNTAS DE REFLEXIÓN.....	5
6. CONLCUSION	6

1. INTRODUCCIÓN

En esta práctica se implementaron tres representaciones cromosómicas clásicas en Algoritmos Genéticos (binaria, real y permutacional), aplicadas al problema de distribución equitativa de 39 estudiantes en exámenes. El objetivo fue evaluar el equilibrio académico entre grupos A, B y C, observando sus promedios de nota, desviaciones y varianzas. Posteriormente, se propuso un escenario extendido que distribuye a los estudiantes en 4 exámenes para evaluar la escalabilidad del enfoque.

2. COMPARACIÓN DE REPRESENTACIONES

Se presentan los resultados reales observados al ejecutar los programas con las tres representaciones cromosómicas, según la salida mostrada en consola:

Representación Binaria

- El algoritmo intentó asignar los 39 alumnos en grupos de 13 para A, B y C, pero no logró cumplir esta condición.
- En la mejor solución, se obtuvieron 15 alumnos en A (promedio: 14.87), 11 en B (15.73) y 13 en C (15.77).
- La desviación estándar entre promedios fue **0.4159**, indicando un desequilibrio notable.
- El fitness se mantuvo constante en **-1000**, ya que la condición de tamaño de grupo no se cumplía.

Representación Permutacional

- Esta representación usa permutaciones, lo que garantiza que cada grupo tenga exactamente 13 alumnos.
- Examen A: promedio 15.38 | Examen B: 15.38 | Examen C: 15.46.
- Desviación estándar entre promedios: **0.0363**.
- Fitness alcanzado: **0.2637**, el más alto entre todas las representaciones.
- El algoritmo converge rápidamente desde las primeras generaciones, sin necesidad de muchas iteraciones.

Representación Real

- Utiliza pesos normalizados para asignar probabilidades a cada grupo por alumno.
- Logró un equilibrio casi perfecto con promedios: A=15.38, B=15.46, C=15.38.
- Varianzas: A=10.39, B=7.63, C=13.62.
- Desviación estándar entre promedios: **0.0363**.
- Fitness final: **-1.0911**, con convergencia estable desde la generación 30.
- También se observó buena distribución de notas y diversidad interna.

Comparación resumen:

- La representación **binaria** presenta limitaciones estructurales para lograr grupos equitativos y, sin una verificación explícita o penalización correcta, no cumple los requisitos del problema.
- La **permutacional** es precisa, garantiza el tamaño correcto y ofrece el mejor fitness, ideal para problemas con restricciones estrictas de tamaño.
- La **real** es altamente flexible, alcanzando un equilibrio fino y permitiendo control sobre la diversidad del grupo mediante ajustes de varianza, aunque requiere más parámetros.

3. RESULTADOS DE LAS ACTIVIDADES

- **Actividad 1:** Se ejecutaron los tres programas y se analizó su rendimiento. La representación permutacional logró una desviación estándar mínima entre los promedios de los exámenes A, B y C (≈ 0.036), mostrando un alto grado de equilibrio. En cambio, la representación binaria generaba asignaciones no uniformes, muchas veces con más o menos de 13 estudiantes por grupo, lo que dejaba el fitness en -1000.
- **Actividad 2:** Se modificó la función de fitness de la binaria para incluir una penalización por varianza alta y un incentivo por diversidad. Al ejecutarse el algoritmo con esta mejora, el fitness evolucionó de -1000 a aproximadamente -2.5. Esto indicó que el nuevo enfoque guiaba mejor la solución hacia un balance de notas sin descuidar la diversidad de rendimientos entre grupos.
- **Actividad 3:** Se creó un nuevo operador de mutación gaussiana para la representación real y se aplicó con diferentes valores de sigma. En todos los casos (sigma=0.01 a 0.5), el algoritmo logró converger a un fitness estable de -1.0911. El sigma=0.1 destacó como un buen compromiso entre variación e integridad cromosómica. Las distribuciones de notas mostraron buen equilibrio y dispersión moderada.
- **Actividad 4:** Se añadió una restricción a la representación permutacional: evitar que todos los alumnos con nota < 11 estuvieran en un mismo examen. Se modificó la función de fitness para penalizar estos casos. El algoritmo logró soluciones válidas que cumplían la restricción sin perder calidad, con un fitness final de 0.2637.
- **Actividad 5:** Se generaron visualizaciones con matplotlib y seaborn: curvas de evolución del fitness por generación, histogramas de notas y boxplots por examen. Estas gráficas mostraron que la permutacional converge en menos de 10 generaciones, y que sus grupos tienen rangos similares. La representación real mostró una leve variabilidad interna. La binaria, aunque más variable, mejoró visualmente tras la penalización por varianza.
- **Actividad 6:** Se extendió el problema para asignar a los 39 alumnos en 4 exámenes. Se adaptó la representación real para usar 4 pesos por alumno

(cromosoma de 156 genes). El reparto fue 10-10-10-9. El algoritmo alcanzó un fitness de -1.0684 y una desviación entre promedios de solo 0.0192. Esto demostró que la arquitectura es escalable y eficiente, incluso en espacios de búsqueda más complejos.

4. REFLEXIÓN: CUÁNDO USAR CADA REPRESENTACIÓN

- **Binaria:**
 - Ideal para problemas de selección con decisiones discretas (incluir o no incluir).
 - Más fácil de implementar y analizar, pero limitada cuando se requiere equilibrio estricto en la salida.
- **Real:**
 - Recomendable para problemas donde se manejen probabilidades, pesos o variables continuas.
 - Permite una exploración más fina del espacio de soluciones y es ideal para ajustes suaves.
- **Permutacional:**
 - Excelente para problemas de ordenamiento, asignación exacta o agrupamiento fijo.
 - Rápida y eficaz en alcanzar equilibrio entre subconjuntos. Menos adecuada cuando los valores no tienen orden natural.

5. RESPUESTAS A PREGUNTAS DE REFLEXIÓN

- **¿Por qué la representación binaria tuvo dificultades para lograr exactamente 13 alumnos por grupo?**

Porque su codificación no garantiza automáticamente que la cantidad de alumnos por grupo se mantenga fija. Es posible que algunos grupos tengan más o menos de 13 alumnos, generando penalizaciones severas en el fitness. A menos que se impongan restricciones adicionales, la aleatoriedad puede provocar desequilibrio.

- **¿Qué ventajas tiene usar valores reales normalizados vs. selección directa?**

Permiten representar distribuciones de probabilidad más flexibles, facilitando que el algoritmo explore soluciones intermedias y que se mantenga el equilibrio mediante normalización automática de pesos. Además, se mejora la diversidad genética sin romper la consistencia del cromosoma.

- **¿En qué casos la representación permutacional sería inadecuada?**

En problemas donde no importa el orden o donde se requiere asignar elementos a múltiples categorías con valores difusos o continuos, ya que la permutación fuerza una estructura de orden estricto. Además, no se adapta bien si hay que asignar elementos a varios grupos simultáneamente.

- **¿Cómo afecta el tamaño de la población y número de generaciones a la calidad de la solución?**

Una población mayor y más generaciones permiten una exploración más amplia del espacio de soluciones, aumentando la probabilidad de alcanzar resultados óptimos, aunque con mayor costo computacional. Poblaciones pequeñas pueden converger rápido pero quedar atrapadas en óptimos locales.

6. CONCLUSION

A Cada representación ofrece ventajas según el tipo de problema. La representación permutacional fue la más efectiva para este escenario específico, al lograr distribuciones exactas y equilibrio entre grupos en pocas generaciones. La representación real también mostró gran adaptabilidad, especialmente en escenarios extendidos.

Las actividades permitieron explorar y adaptar los algoritmos para mejorar su rendimiento, evaluar restricciones, comparar operadores genéticos y analizar visualmente los resultados. Esto demostró la importancia del diseño cromosómico en la resolución de problemas reales de optimización con poblaciones.

Se concluye que una elección adecuada de la representación, combinada con operadores de calidad y parámetros ajustados, permite obtener soluciones equilibradas y óptimas en menor tiempo, con menor riesgo de estancamiento evolutivo.