

UNIVERSIDAD NACIONAL DEL SANTA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas e Informática



UNS
UNIVERSIDAD
NACIONAL DEL SANTA

ALGORITMOS EVOLUTIVOS

Representaciones Cromosómicas en Algoritmos Genéticos

Estudiante

OSORIO JUAQUIN, Edwin Fidencio

Docente:

Ms. Ing. LOPÉZ HEREDIA, Johan Max

NUEVO CHIMBOTE – PERÚ

2025

1. Comparación de las tres representaciones

- **Representación Binaria.**

Cada alumno es codificado mediante 3 bits (o 4, en la versión extendida). Es directa, fácil de implementar, pero menos eficiente al manejar restricciones o estructuras secuenciales.

- *Ventaja:* Control total sobre la asignación.
- *Desventaja:* Espacio de búsqueda grande y mutaciones poco eficientes.

- **Representación Permutacional**

Se usa una permutación de los índices de los alumnos. Esta secuencia se divide en grupos de tamaño fijo para cada examen.

- *Ventaja:* Naturalmente asegura la equidad numérica.
- *Desventaja:* No controla explícitamente el balance de notas.

- **Representación Real**

Cada alumno se representa con tres valores reales que indican la probabilidad de ser asignado a un examen.

- *Ventaja:* Flexible y adaptable a diferentes problemas.
- *Desventaja:* Problemas discretos como este no se benefician de codificación continua.

2. Resultados y conclusiones de las actividades realizadas

2.1. Conclusiones Generales

- La elección de la representación cromosómica es crítica para el desempeño del algoritmo genético. La representación permutacional demostró ser la más eficiente en este problema de asignación equitativa, debido a su estructura natural para distribuir elementos en grupos iguales.
- La representación binaria funciona, pero con limitaciones. Aunque logra distribuir a los alumnos correctamente en número, no siempre optimiza la distribución de notas. La convergencia es más lenta y la mejora del fitness limitada, especialmente en configuraciones complejas como 4 exámenes.
- La representación real no es adecuada para este tipo de problema discreto. Al ser continua, no maneja bien la asignación exclusiva de alumnos a un solo examen. La mejora fue mínima y se estancó rápidamente.

- La visualización es una herramienta clave para evaluar el rendimiento del algoritmo. Los gráficos de evolución, histogramas y boxplots permitieron validar de manera objetiva qué tan equilibradas eran las asignaciones.
- Extender el problema a 4 exámenes aumenta la complejidad, pero es factible. Con una codificación binaria de 4 bits por alumno y una población suficientemente grande, se logró una desviación estándar muy baja, lo que demuestra que el modelo puede adaptarse a otros escenarios.

2.2.Recomendaciones

- Usar representación permutacional para problemas de asignación donde el número de elementos por grupo debe ser igual o muy similar. Ideal para tareas de distribución equitativa.
- Evitar la representación real en problemas donde se requiere una decisión categórica única (por ejemplo, un alumno debe estar en un solo examen), ya que la naturaleza continua puede generar ambigüedad.
- Usar binaria solo cuando se requiera control detallado de las asignaciones o se busque adaptar el modelo a restricciones particulares (por ejemplo, capacidad máxima por examen).
- Acompañar los resultados con visualizaciones para tomar decisiones informadas. La interpretación visual complementa la información numérica y revela patrones ocultos.
- Realizar pruebas con distintos tamaños de población y generaciones si se trabaja con representaciones más complejas o número impar de elementos por grupo (como 39 alumnos en 4 exámenes).

3. Reflexión sobre cuándo y por qué usar cada tipo de representación

Representación	Cuándo usarla	Por qué utilizarla
Binaria	Cuando se requiere control individual y restricciones rígidas	Es simple y flexible, pero puede ser menos eficiente
Permutacional	Cuando se necesita un reparto equitativo y ordenado	Excelente para dividir en grupos iguales sin violar restricciones
Real	En problemas continuos o de optimización multiobjetivo	Ideal para flexibilizar asignaciones y pesos, pero no en problemas discretos