

Modelos y algoritmos para logística y transporte

Agustín Pecorari

Rodrigo Maranzana

Algoritmos genéticos

- **Inicialización:** Se genera aleatoriamente la población inicial, que está constituida por un conjunto de cromosomas los cuales representan las posibles soluciones del problema.
- **Evaluación:** A cada uno de los cromosomas de esta población se aplicará la función de aptitud para saber cómo de "buena" es la solución que se está codificando.
- **Condición de término:** se usan dos criterios: correr el AG un número máximo de iteraciones (generaciones) o detenerlo cuando no haya cambios en la población.

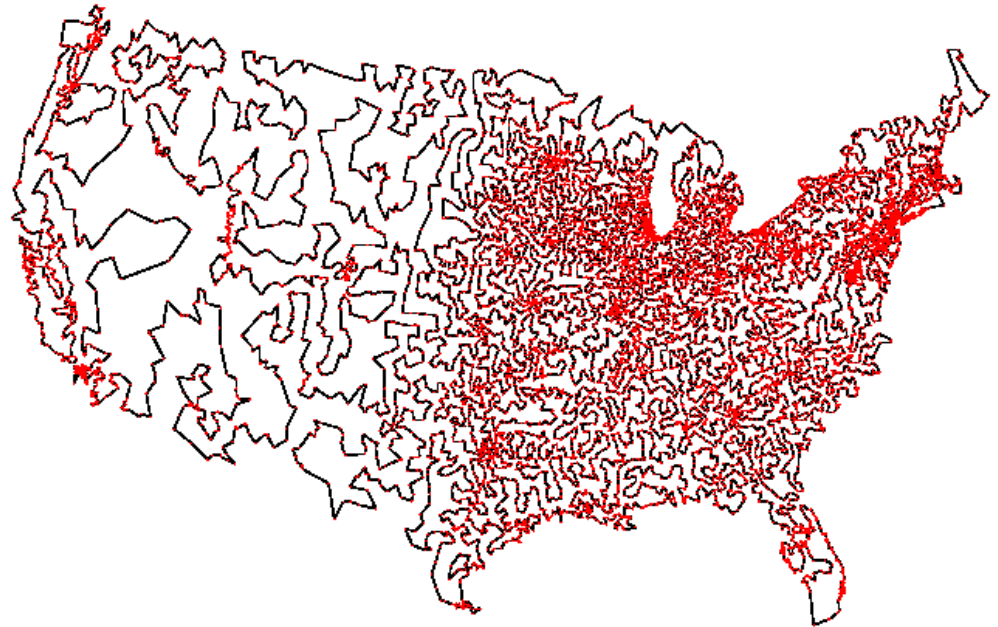
Algoritmos genéticos

Mientras no se cumpla la condición de término se hace lo siguiente:

1. **Selección.**
2. **Recombinación o cruzamiento.**
3. **Mutación.**
4. **Reemplazo.**

Algoritmos genéticos

Selección: Los cromosomas con mejor aptitud tienen mayor probabilidad de ser seleccionados.



Algoritmos genéticos

- **Recombinación o cruzamiento:** La recombinación es el principal operador genético, representa la reproducción sexual, opera sobre dos cromosomas a la vez para generar dos descendientes donde se combinan las características de ambos cromosomas padres.

Algoritmos genéticos

Mutación: Modifica al azar parte del cromosoma de los individuos, y permite alcanzar zonas del espacio de búsqueda que no estaban cubiertas por los individuos de la población actual.

Algoritmos genéticos

Reemplazo: Una vez aplicados los operadores genéticos, se seleccionan los mejores individuos para conformar la población de la generación siguiente.

GRASP

El **greedy randomized adaptive search procedure** (GRASP) consiste en iteraciones hechas de construcciones sucesivas de una solución aleatoria codiciosa y las mejoras iterativas subsiguientes de la misma a través de una búsqueda local.

GRASP

Las soluciones aleatorias codiciosas se generan agregando elementos al conjunto de soluciones del problema a partir de una lista de elementos clasificados por una función codiciosa de acuerdo con la calidad de la solución que lograrán.

GRASP

Para obtener variabilidad en el conjunto candidato de soluciones codiciosas, los mejores candidatos se colocan en una lista de candidatos restringida (RCL), y se eligen al azar cuando se construye la solución.

Constraint Programming

La optimización de restricciones, o la programación de restricciones (CP), es el nombre que se le da a la identificación de soluciones factibles a partir de un conjunto muy grande de candidatos, donde el problema se puede modelar en términos de **restricciones arbitrarias**.

El CP se basa en la **viabilidad** (encontrar una solución viable) en lugar de la optimización (encontrar una solución óptima) y se centra en las restricciones y variables en lugar de la función objetivo. De hecho, es posible que un problema de PC ni siquiera tenga una función objetivo.

Constraint Programming

La CP funciona particularmente bien para problemas de Scheduling, sobre todo los más difíciles.

Algunos solvers de CP tienen variables y funciones específicas para modelización de problemas de Scheduling.

Ejemplos de restricciones:

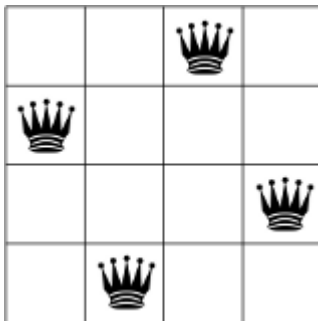
- Alldiff.
- StartAtEnd / StartBeforeEnd,....
- If ... then
- Operadores lógicos.

Constraint Programming

Si un problema se puede modelar con un objetivo lineal y restricciones lineales, entonces tiene un problema de programación lineal.

CP funciona como algoritmo de búsqueda con **propagación y backtracking**.

Un problemas clásico de CP es el problema de las N-reinas.



<https://developers.google.com/optimization/cp/queens>

Constraint Programming

