ISI437 - Inteligencia Artificial

Algoritmos Genéticos

Ing. Jose Eduardo Laruta Espejo

31 de marzo de 2020

Universidad La Salle - Bolivia

Contenido i

Algoritmos Evolutivos

Algoritmos Evolutivos

Los algoritmos evolutivos son un conjunto de algoritmos de optimización metaheurística basados en poblaciones genéricas. Los AE, utilizan mecanismos inspirados por la evolución biológica tales como reproducción, mutación, recombinación y selección.

Algoritmos Evolutivos

En un algoritmo evolutivo, existen **soluciones candidatas** que juegan el papel de individuos en una población y una función de competencia o *fitness function* determina la calidad de la solución. Luego, se desarrolla una evolución de la población luego de la aplicación repetida de los operadores anteriormente mencionados.

Algoritmos Evolutivos

Los algoritmos evolutivos se usan para resolver problemas de optimización y búsqueda de todo tipo de manera efectiva ya que no se necesitan suposiciones sobre la calidad de la solución.

Implementación

De una forma general, un algoritmo evolutivo sigue el siguiente esquema:

1. Generar una población de individuos aleatoriamente,
2. Repetir lo siguiente para cada generación:
   - 2.1 Evaluar la competencia (fitness) de cada individuo.
   - 2.2 Seleccionar individuos para reproducción.
   - 2.3 Generar una nueva camada de individuos.
   - 2.4 Evaluar la competendia de cada individuo.
   - 2.5 Reemplazar los individuos más débiles con los nuevos.

Tipos de algoritmos evolutivos

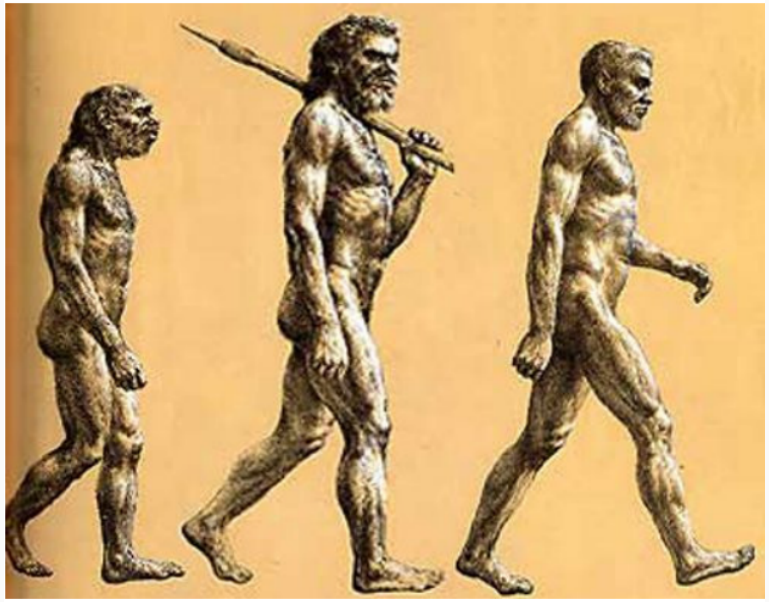
Existen distintos algoritmos y técnicas que siguen estos principios.

- Algoritmos genéticos.
- Programación genética.
- Programación evolutiva.
- Evolución diferencial.
- Neuroevolución.

Introducción a los algoritmos genéticos

Principio de selección natural

Selecciona al mejor, descarta al resto.



Un ejemplo

Las jirafas tienen cuellos largos.

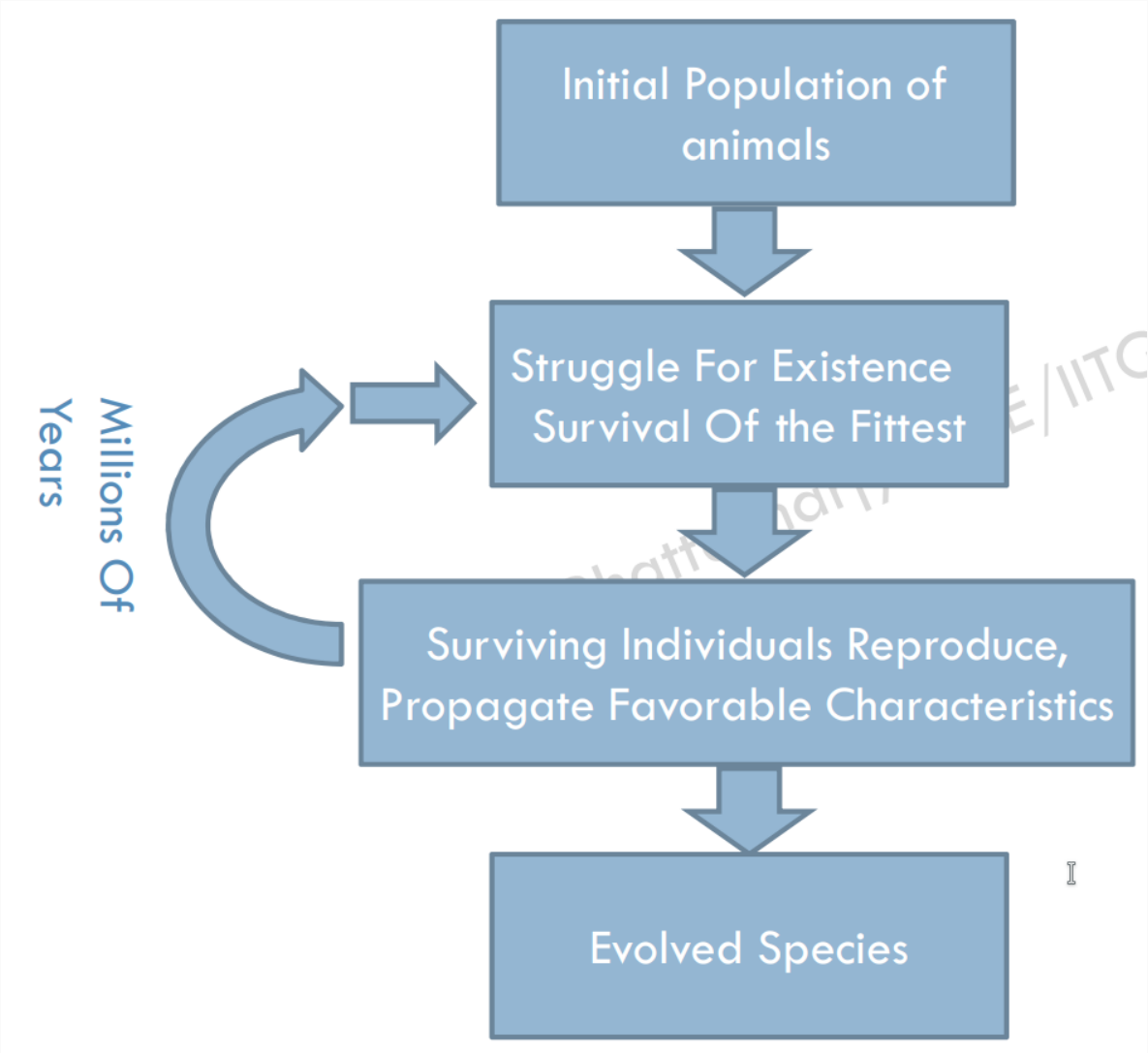
- Las jirafas con cuellos más largos pueden alimentarse de hojas más altas.
- Tienen más chance de sobrevivir.
- Características favorables se propagan a través de las generaciones.
- La especie evolucionada tiene cuellos largos.

Un ejemplo

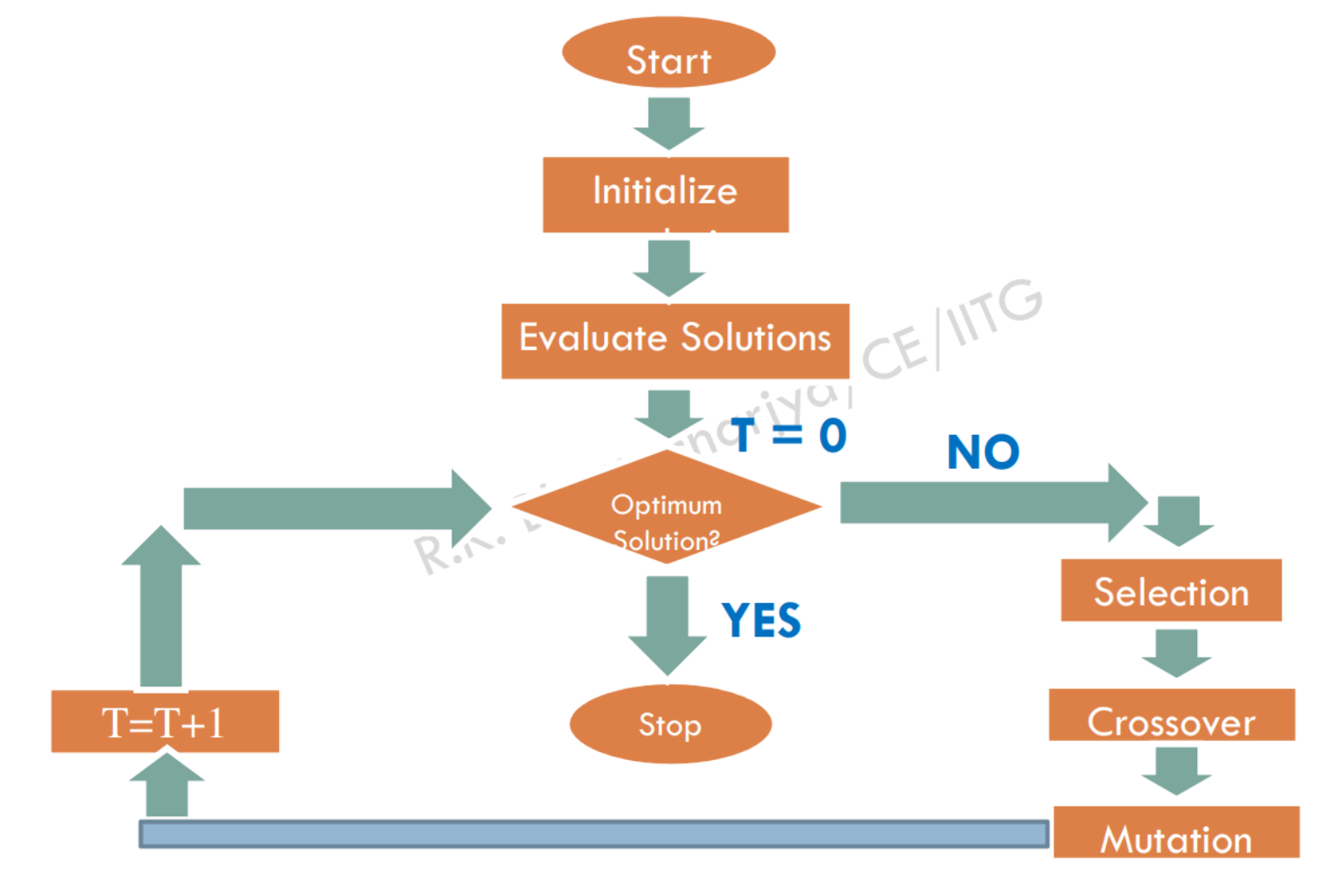
Las jirafas tienen cuellos largos.

Inicialmente cuellos más largos pudieron aparecer por efectos de la mutación, pero los resultados positivos se propagaron en futuras generaciones

Evolución de las especies



Algoritmos genético simple



Operadores de AG

- Selección.
- Cruce o *Crossover*.
- Mutación.

Selección

Este proceso determina cuáles soluciones se mantienen y se permiten reproducirse y cuáles merecen desaparecer.

El objetivo principal del operador de selección es el de destacar soluciones con buen desempeño y eliminar soluciones con mal desempeño mientras se mantiene el tamaño de la población constante.

Selecciona al mejor, descarta al resto.

Selección

El operador de selección debe cumplir las siguientes funciones:

- Identificar soluciones "buenas".
- Hacer copias múltiples de las buenas soluciones.
- Eliminar soluciones malas de la población para que las copias de las buenas puedan tomar su lugar.

Selección

El operador de selección debe cumplir las siguientes funciones:

- Identificar soluciones "buenas".
- Hacer copias múltiples de las buenas soluciones.
- Eliminar soluciones malas de la población para que las copias de las buenas puedan tomar su lugar.

Cómo identificar las buenas soluciones?

Función de competencia o Fitness function

- Se asigna un valor de competencia o *fitness* para evaluar cada solución.
- La función de competencia cuantifica la optimalidad de una solución.
- El valor es usado para rankear una solución dentro de las demás.
- El valor de competencia es asignado en función de su cercanía a la solución óptima al problema.

Implementación del operador de Selección

Existen distintas técnicas para implementar la selección en AG.

- Selección por torneo.
- Selección por la ruleta.
- Selección proporcional.
- Selección por ranking.
- Selección por estado estable.

Selección por torneo

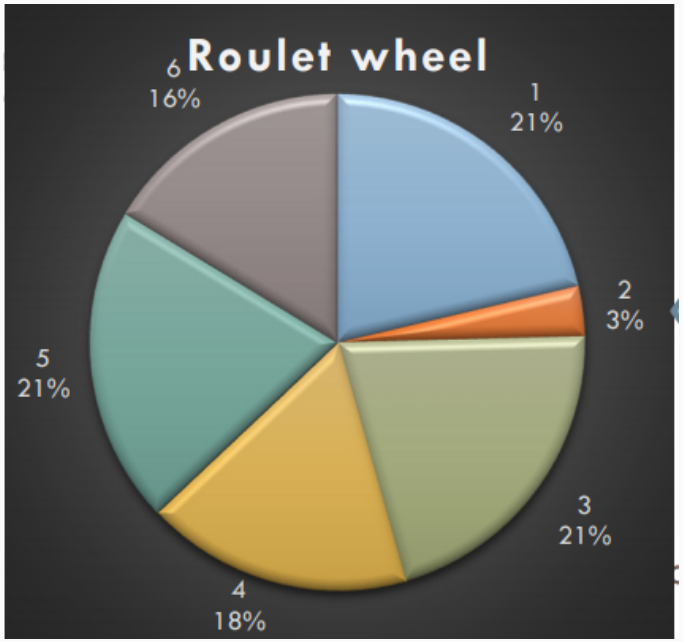
- Se realizan varios “torneos” o competencias entre algunos individuos seleccionados aleatoriamente.
- El ganador de cada torneo es seleccionado para la siguiente generación.
- La rigurosidad de la selección se puede ajustar con el tamaño del torneo.
- Los individuos débiles tienen menor chance en torneos grandes.

Selección por ruleta y proporcional

Se ordena a los individuos y se asigna un porcentaje en la ruleta. Los mejores individuos tienen más chances de ser seleccionados.

Chrom #	Fitness	% of RW	EC	AC
1	50	26.88	1.61	2
2	6	3.47	0.19	0
3	36	20.81	1.16	1
4	30	17.34	0.97	1
5	36	20.81	1.16	1
6	28	16.18	0.90	1
	186	100.00	6	6

Selección por ruleta y proporcional



Chrom #	Fitness	% of RW	EC	AC
1	50	26.88	1.61	2
2	6	3.47	0.19	0
3	36	20.81	1.16	1
4	30	17.34	0.97	1
5	36	20.81	1.16	1
6	28	16.18	0.90	1
	186	100.00	6	6

Selección por ranking

Se ordena a los individuos de acuerdo a su competencia y se asigna un ranking. Luego se procede a definir la cantidad de copias.

Selección por ranking

Chrom #	Fitness
1	37
2	6
3	36
4	30
5	36
6	28

Sort according to fitness

Chrom #	Fitness
1	37
3	36
5	36
4	30
6	28
2	6

Assign raking

Chrom #	Rank
1	6
3	5
5	4
4	3
6	2
2	1

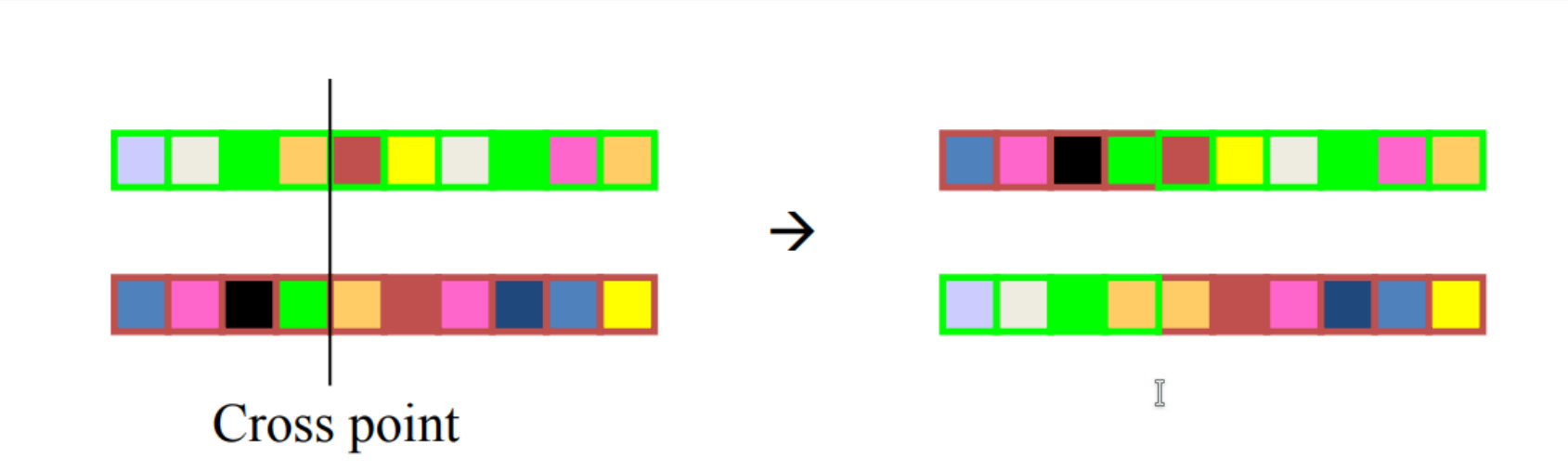
Chrom #	% of RW
1	29
3	24
5	19
4	14
6	10
2	5

Chrom #	EC	AC
1	1.714	2
3	1.429	1
5	1.143	1
4	0.857	1
6	0.571	1
2	0.286	0

Crossover

Se usa el operador de Crossover para crear nuevas soluciones en base a soluciones existentes disponibles despues de aplicar el operador de selección.

Este operador intercambia información genética entre las soluciones más fuertes.

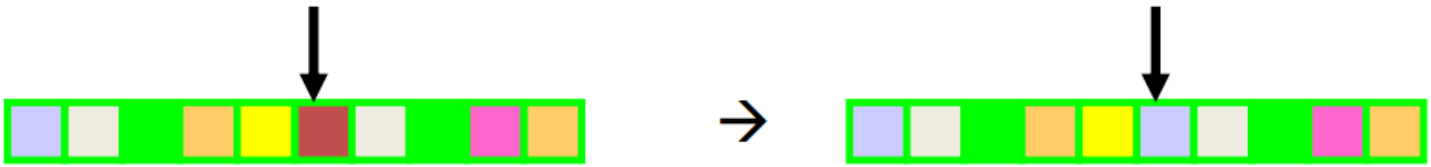


El *cross point* se suele elegir de forma aleatoria.

Mutación

La mutación refiere a la introducción ocasional de nuevas características en la naturaleza de las soluciones para mantener un nivel de diversidad.

Pese a que el crossover tiene la principal responsabilidad en la búsqueda de la solución óptima, la mutación también se usa con este propósito.



Elitismo

La combinación de crossover y mutación puede eliminar a la mejor solución de la población. Elitismo se define como la preservación de las mejores soluciones de la población como un porcentaje de la misma.

Equivalencias biológicas

Los componentes de un AG se pueden equiparar a distintos fenónemos en la naturaleza.

- Población: Conjunto de soluciones.
- Individuo: Solución a un problema.
- Fitness: Calidad de la solución.
- Cromosoma: Codificación de la solución.
- Gen: Parte unitaria de la codificación de la solución.

Ejemplo: Traveling Salesman Problem

En este problema se debe encontrar un recorrido en un conjunto de ciudades o locaciones en el cual:

- cada ciudad es visitada al menos una vez.
- la distancia total recorrida es mínima.

Representación

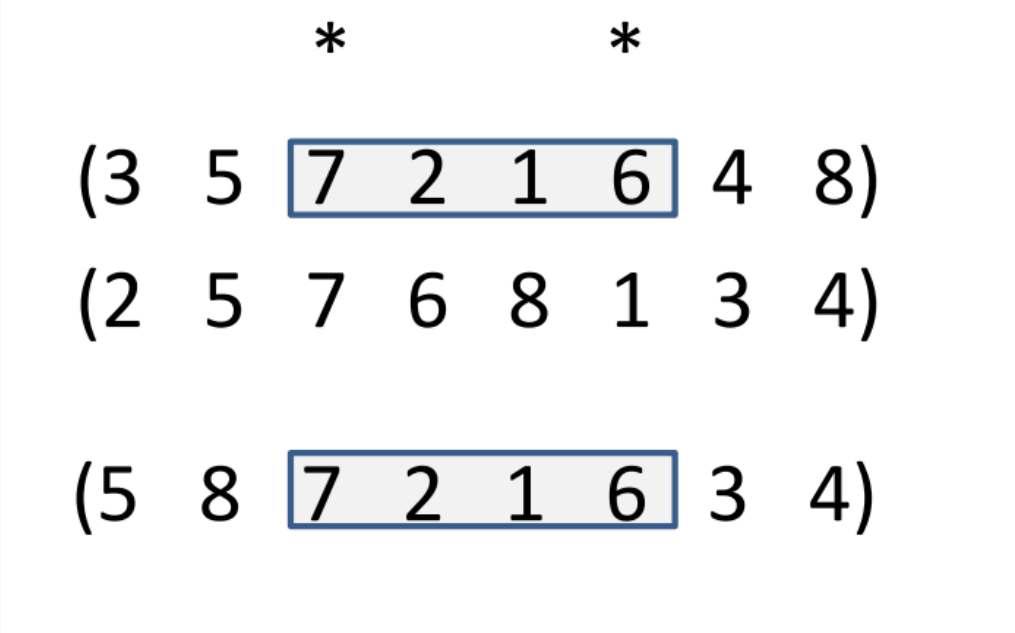
La Representación de las soluciones es una lista ordenada de ciudades representadas por un número. Este tipo de representación corresponde a un Algoritmo Genético basado en ordenamiento.

- Lista1: (3 5 7 2 1 6 4 8)
- Lista2: (2 5 7 6 8 1 3 4)

1. La Paz
2. Santa Cruz
3. Oruro
4. Potosí
5. Cochabamba
6. Sucre
7. Trinidad
8. Tarija

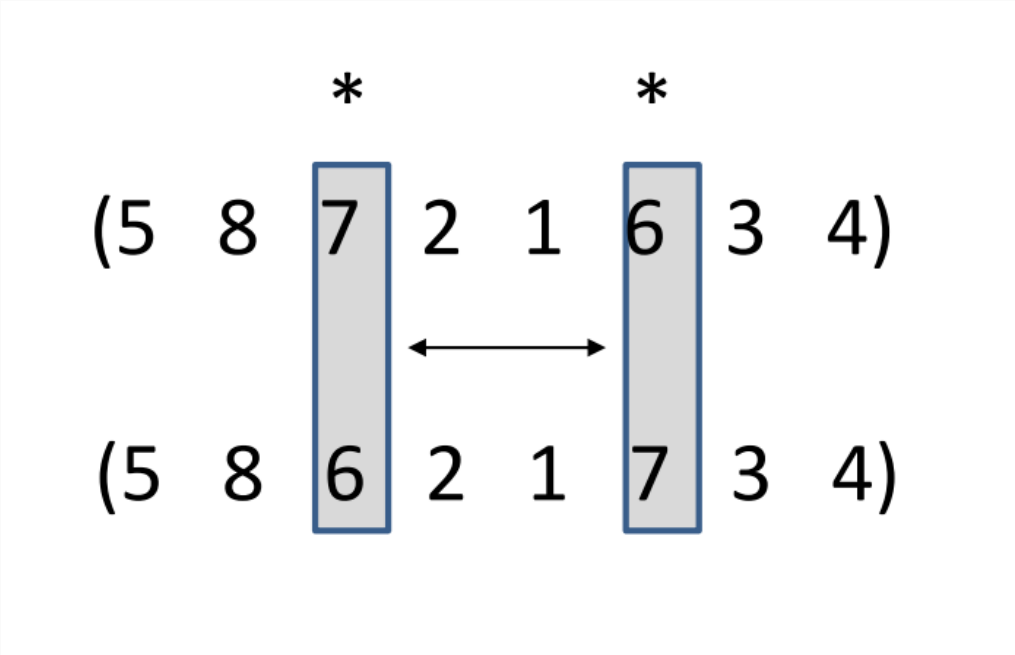
Crossover

Se combina inversión y recombinación:

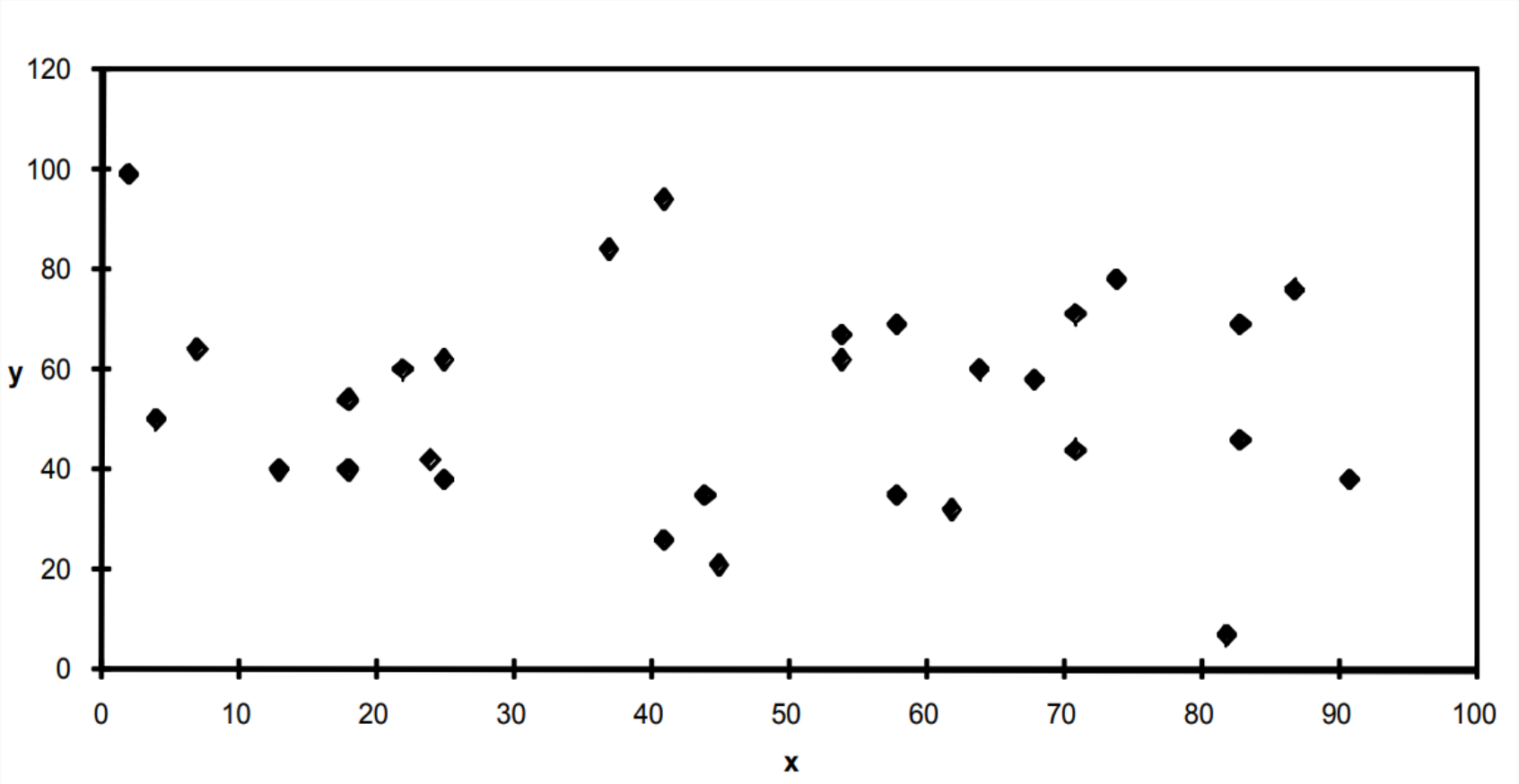


Mutación

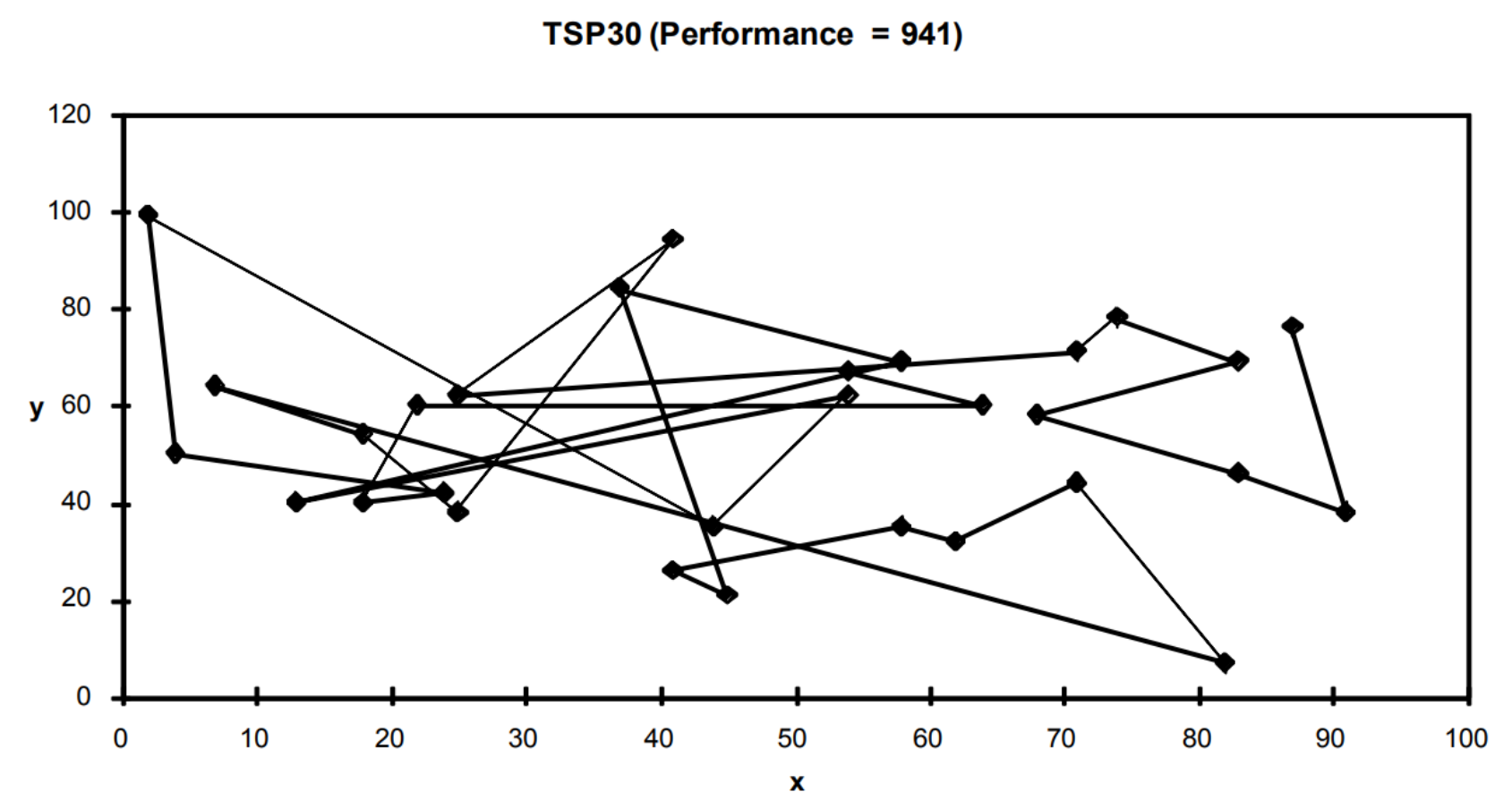
La mutación consiste en un reordenamiento.



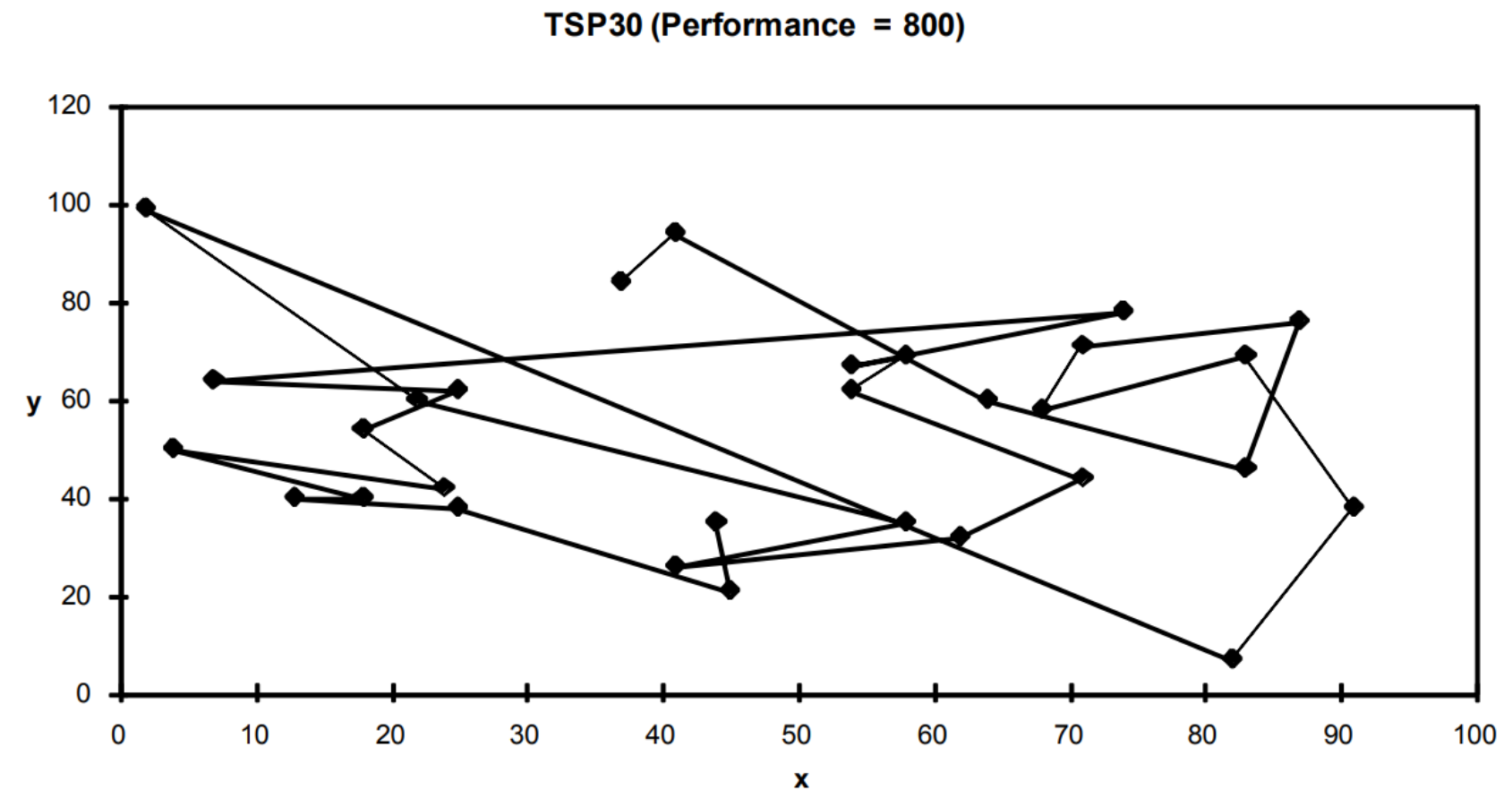
Ejemplo con 30 ciudades



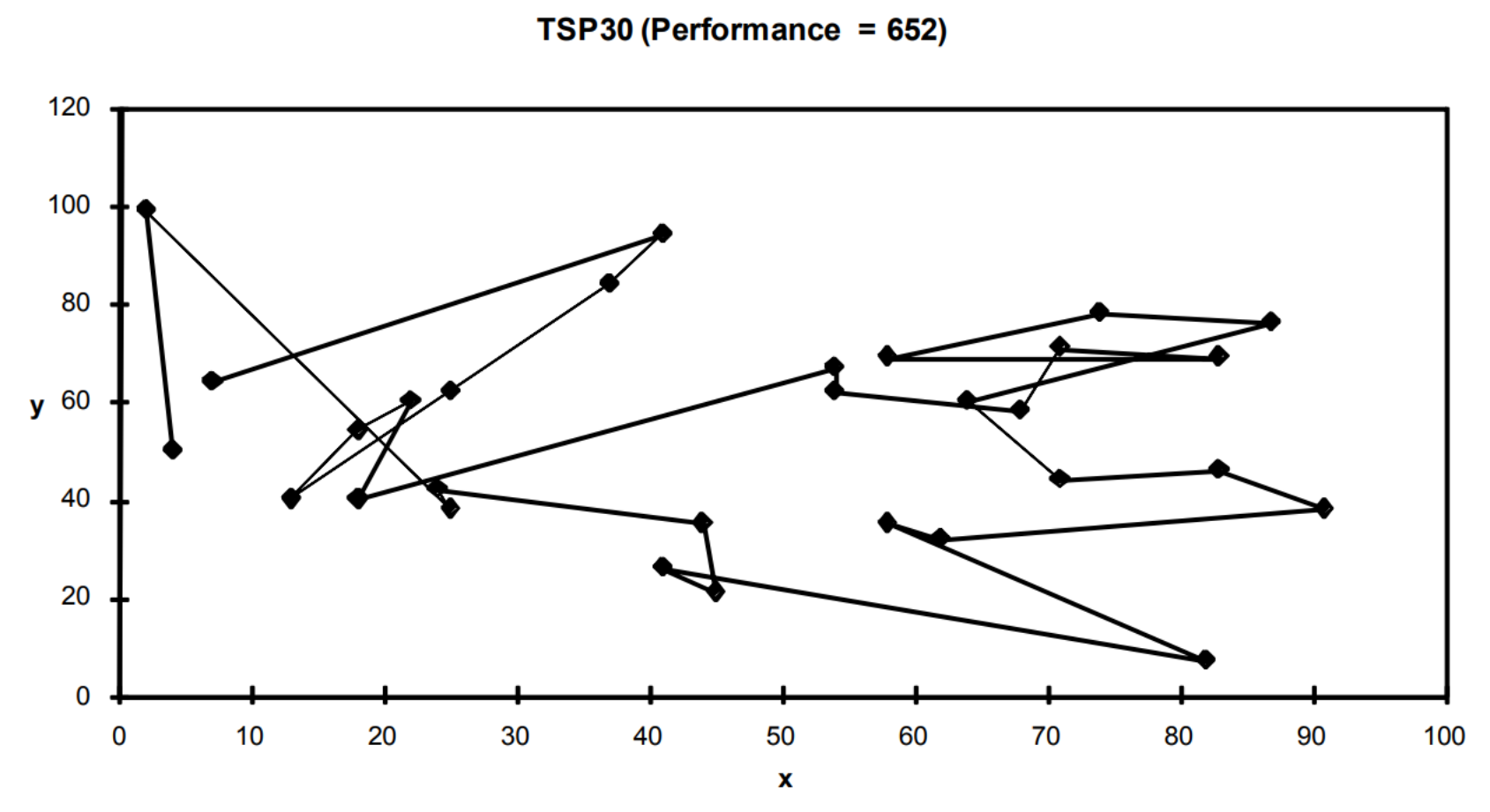
Ejemplo con 30 ciudades, distancia=941



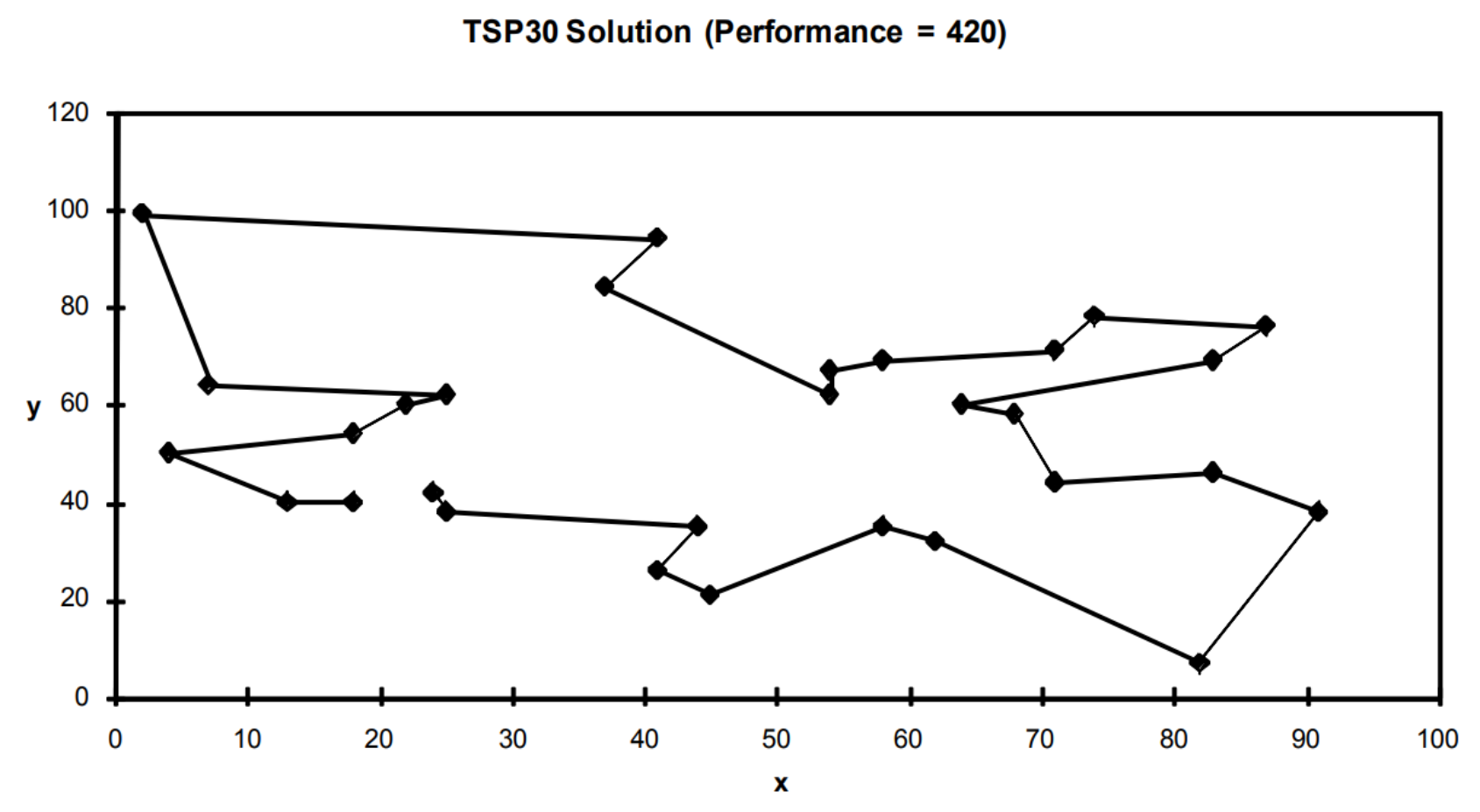
Ejemplo con 30 ciudades, distancia=800



Ejemplo con 30 ciudades, distancia=652



Ejemplo con 30 ciudades, distancia=420



Aplicaciones

Los AG tienen varias aplicaciones en diversas áreas:

- **Control:** Gasoductos, evasión de misiles.
- **Diseño:** Diseño de aeronaves, redes de comunicaciones.
- **Juegos:** Poker, damas, otros juegos adversarios.
- **Seguridad:** Encriptación.
- **Robótica:** Planeamiento de trayectorias.

Ventajas

- Conceptualmente fácil de entender.
- Modular, independiente de la aplicación.
- Soporta optimización multiobjetivo.
- Siempre presenta una solución, y ésta mejora con el tiempo.
- Se puede aprovechar soluciones previas o alternativas.
- Flexibilidad y extensibilidad para aplicaciones híbridas (Neuroevolución).

Demo Time

- **demo1**
- **demo2**
- **demo3**

Preguntas?