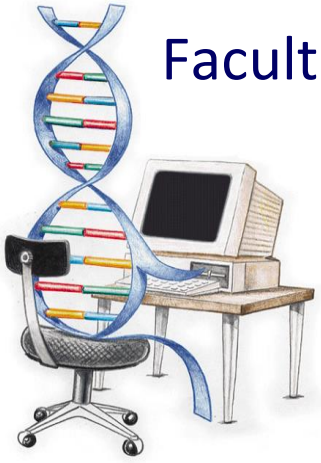
ALGORITMOS EVOLUTIVOS
Curso 2024

Tema 3: Algoritmos genéticos

Centro de Cálculo, Instituto de Computación
Facultad de Ingeniería, Universidad de la República, Uruguay



cecal

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Contenido

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

1. Introducción
   - Conceptos biológicos
   - Conceptos de optimización
2. Algoritmos Genéticos
   - El Algoritmo Genético Simple (AGS – Goldberg, 1989)
   - Representación
   - Operadores evolutivos
     - Operadores de selección, cruzamiento y mutación
3. Ejemplo de AGS
4. Resolviendo un problema: genotipo y fitness
5. Ejemplos de aplicación:
   - Planificación de sistemas heterogéneos
   - Diseño de redes overlay
6. Conceptos sobre el mecanismo de búsqueda

Conceptos biológicos

- El ADN es el material genético fundamental de todos los organismos vivos
  - La molécula de ADN está formado por secuencias de las bases: Adenina (A), Timina (T), Citosina (C) y Guanina (G)
- Un gen es una sección de ADN que codifica una cierta función bioquímica
- El gen es fundamentalmente una unidad de herencia
- Dependiendo de su especie, un organismo puede tener un número variable de genes en su ADN (desde una docena de genes hasta decenas de miles)

Conceptos biológicos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Se denomina cromosoma a cada una de las cadenas de ADN que se encuentran en el núcleo de las células
- Los cromosomas son los responsables de la transmisión de información genética
- Cada gen es capaz de ocupar una única región en particular de un cromosoma, se lo llama locus
- Pueden existir formas o valores alternativos del gen llamadas alelos
- Los gametos son las células que llevan información genética de los padres con el propósito de efectuar la reproducción sexual (esperma y óvulos en el ser humano)

Conceptos biológicos

- Las células que tienen un único conjunto de cromosomas (cada conjunto consiste en una única seceuencia de genes), se denominan haploides
- Las células que contienen dos copias con los mismo genes en la misma secuencia de cada cromosoma se denominan diploides
- La mayoría de las especies capaces de reproducirse sexualmente tienen estructuras celulares diploides
- Se denomina genotipo a la información contenida en el genoma de un individuo.
  - El genotipo puede verse como lo que potencialmente puede llegar a ser un individuo
- Los rasgos específicos y observables de un individuo constituyen su fenotipo. A partir del genotipo y del desarrollo se origina el fenotipo de un individuo

Conceptos biológicos

- La aptitud de un individuo evalúa su capacidad de adaptación a las condiciones de su entorno. Se relaciona con la probabilidad de que el individuo sobreviva para reproducirse, y tiene dependencia directa con su número de descendientes
- La selección es el proceso mediante el cual algunos individuos en una población son seleccionados para reproducirse basados en su aptitud.
- Se denomina “selección dura” cuando solamente los mejores individuos se mantienen para generar progenia
- Se denomina “selección blanda” cuando se utilizan mecanismos probabilísticos para mantener como padres a individuos que tengan aptitudes relativamente bajas

Conceptos biológicos

- La presión de selección (o *presión selectiva*) determina la intensidad con la que el entorno tiende a dar ventajas adaptativas a organismos o a eliminar su información genética
  - El neo-darwinismo divide los procesos evolutivos en tres categorías:
    1. Selección estabilizada o normalizada: tiende a eliminar cromosomas con valores extremos
    2. Selección direccional: incrementa o decrementa el valor medio de la población
    3. Selección quebrantada: tiende a eliminar cromosomas con valor moderado
- La mutación consiste en modificaciones individuales de nucleótidos en el proceso de copia (de padre a hijo)
- En general, las mutaciones son provocadas por errores (probabilísticos) en el mecanismo de replicación del ADN

Conceptos biológicos

- La reproducción consiste en la creación de un nuevo individuo a partir de dos progenitores (*reproducción sexual*) o de un único progenitor (*reproducción asexual*)
- Durante la reproducción sexual ocurre la recombinación o cruzamiento
- En el caso de individuos haploides, se intercambian los genes entre los cromosomas de los dos padres
- En el caso de individuos diploides, para cada padre se intercambian los genes entre cada par de cromosomas, formando un gameto
- Luego, los gametos de los dos padres se aparean para formar un único conjunto de cromosomas diploides

Conceptos de optimización

- Se denomina explotación al proceso de utilizar la información obtenida de puntos del espacio de búsqueda previamente visitados para determinar los puntos que conviene visitar a continuación
  - La explotación involucra movimientos finos y es un mecanismo provechoso para que un algoritmo encuentre óptimos locales
- Se denomina exploración al proceso de visitar nuevas regiones del espacio de búsqueda para tratar de encontrar soluciones prometedoras
  - La exploración involucra grandes saltos en el espacio de búsqueda y es un mecanismo útil para evitar que un algoritmo quede atrapado en óptimos locales

Algoritmos genéticos

Características

- Técnica de búsqueda estocástica (probabilística)
- En general utilizan selección probabilística
- Trabajan sobre una representación de soluciones (en su version más simple, es binaria)

1	0	1	1	0	0	1	0	1

  - A cada posición de la cadena se le denomina “gen”y al valor dentro de la posición “alelo”
  - A la representación se le denomina usualmente “cromosoma” o “individuo”
- Operadores evolutivos:
  - Cruzamiento (operador de explotación - *principal*)
  - Mutación (operador de exploración - *secundario*)
  - Selección basada en función de fitness (adecuación a la resolución del problema)

Algoritmos genéticos

Conceptos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Sugeridos por John Holland en la década de 1970
  - Presentó el mecanismo evolutivo en “Adaptation in Natural and Artificial Systems” (1975)
- De Jong (1975) presentó experimentos computacionales para optimizar un conjunto de funciones que se consolidó como el estándar para evaluar técnicas evolutivas
- Los AG fueron formalizados, sistematizados y popularizados por David Goldberg en 1989
  - En los últimos 25 años se han extendido los análisis teóricos, se han propuesto variantes y nuevos modelos y se han abordado con AG una amplia gama de problemas de optimización, búsqueda y aprendizaje en aplicaciones en las áreas de diseño industrial, inteligencia artificial, telecomunicaciones, biología computacional, y otras

John Holland

David Goldberg

Conceptos

- Un AG tiene cinco componentes básicos:
  1. Una representación de las soluciones potenciales del problema
  2. Un procedimiento para crear una población inicial de posibles soluciones (mediante un proceso aleatorio o aleatorizado)
  3. Una función de evaluación que representa al "ambiente", clasificando las soluciones en términos de su aptitud
  4. Un conjunto de operadores de evolución que alteran la composición de los individuos de la población a través de las generaciones
  5. Una configuración paramétrica (tamaño de la población, probabilidad de cruzamiento, probabilidad de mutación, criterio de parada, etc.)

Algoritmos genéticos

Sobre la configuración paramétrica

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- En general los AG trabajan sobre una población de tamaño fijo
- El tamaño de la población debe ser un número tal que permita mantener diversidad en los individuos solución, sin sacrificar la eficiencia computacional del mecanismo de búsqueda
- Las probabilidades de aplicación de los operadores evolutivos definen el balance entre la exploración del espacio de búsqueda y la explotación de buenas soluciones
- El criterio de parada puede especificar un esfuerzo prefijado (en número de generaciones o tiempo de ejecución) o involucrar aspectos dinámicos de la evolución (variaciones en los valores, mejor o promedio, de la función de fitness)
- Usualmente se determinan los valores apropiados mediante análisis empíricos y procedimientos estadísticos

Algoritmos genéticos

Conceptos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Existen múltiples propuestas y variantes de algoritmos genéticos
- Estudiaremos la propuesta original de Goldberg (1989), conocida como ALGORITMO GENÉTICO SIMPLE (AGS)
- Características:
  - Representación binaria
  - Selección proporcional (implementada mediante *rueda de ruleta*)
  - Cruzamiento de un punto
  - Mutación de inversión de valor de bit

Algoritmo Genético Simple

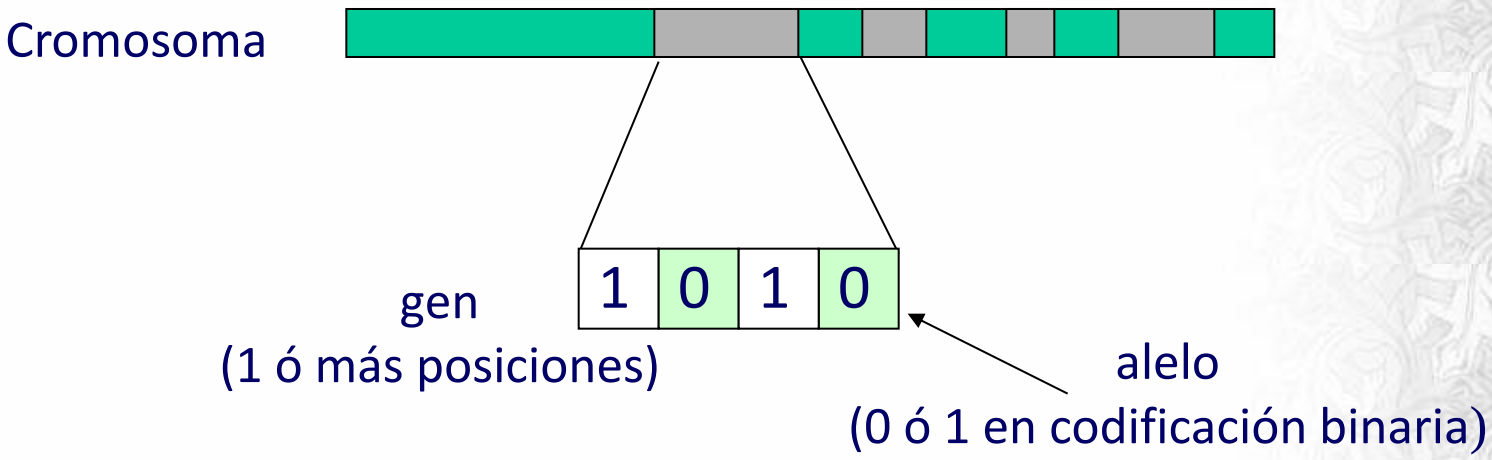
Representación

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- La representación tradicionalmente utilizada es una cadena binaria

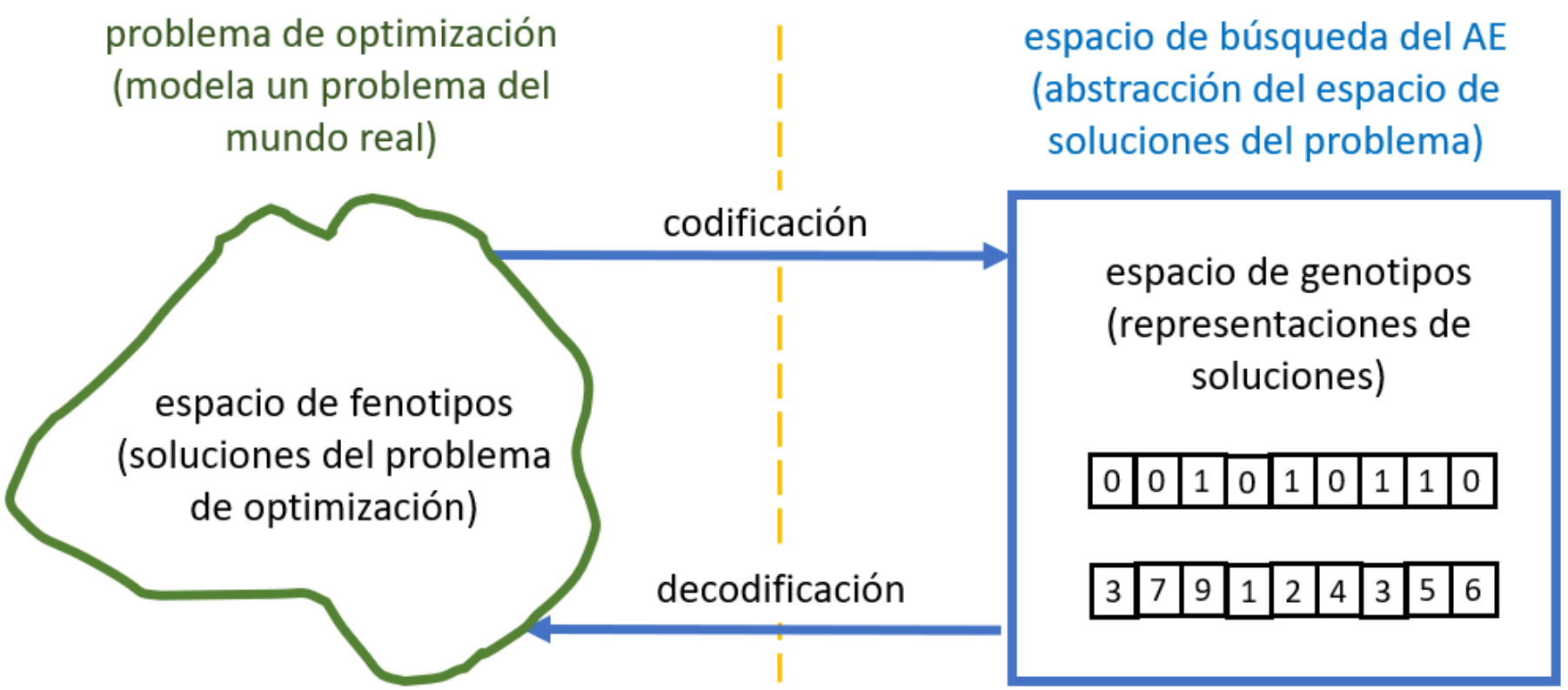
1	0	1	1	0	0	1	0	1

- A la cadena general se le llama cromosoma
- A cada subcadena en la cadena general se le denomina gen y al valor dentro de cada posición se le llama alelo



Algoritmo Genético Simple

Representación



Algoritmo Genético Simple

Representación

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- La representación es el genotipo que se corresponde con una solución al problema (fenotipo)
- Existe un proceso de codificación (y su inverso de decodificación) que permite pasar de fenotipo a genotipo y viceversa
- La codificación especifica una función de correspondencia

  $f_C : S \rightarrow \{0,1\}^*$ (siendo S el espacio de soluciones del problema)
- La función inversa es la decodificación $f_D : \{0,1\}^* \rightarrow S$ puede ser una función parcial
- La complejidad de f_C y de f_D dependerá de las características del problema y de las variables a codificar

Algoritmo Genético Simple

Representación

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- La longitud de la representación depende de las características del problema (número de variables, número de funciones objetivo, dimensión del dominio), y de características de la solución buscada (precisión deseada, por ejemplo)
- El tipo de representación depende de las características del problema a resolver
- Mecanismos de codificación binaria más utilizados:
  - Código binario
  - Códigos de Gray
  - Representación de punto flotante
  - Otras (dependientes del problema a resolver)

Algoritmo Genético Simple

Representación

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

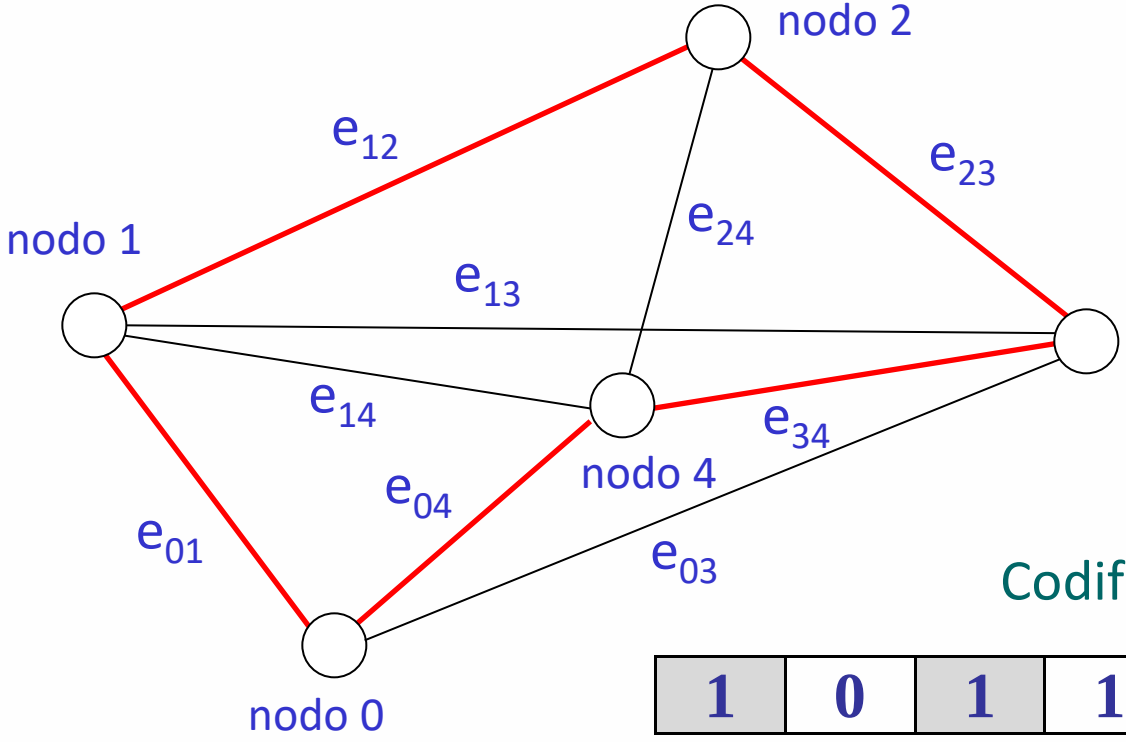
- Ejemplo 1:
- Se trabaja con un problema de optimización que utiliza una variable real x_j definida en el dominio $D_i = [a_j, b_j]$ y se requiere una precisión de n cifras significativas en las soluciones
- Para utilizar una representación binaria el dominio debe ser dividido en $(b_j - a_j)*10^n$ rangos de igual tamaño
- El número de bits requeridos (m_j) estará dado por:

$$2^{m_j - 1} < (b_j - a_j) * 10^n \leq 2^{m_j} - 1$$

Algoritmo Genético Simple

Representación

- Ejemplo 2: para un problema estilo TSP, codificación binaria basada en representar aristas presentes en una solución



Grafo $G=(V,E,C)$

Codificación binaria

1	0	1	1	0	0	1	0	1
e_{01}	e_{03}	e_{04}	e_{12}	e_{13}	e_{14}	e_{23}	e_{24}	e_{34}

Algoritmo Genético Simple

Operadores evolutivos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

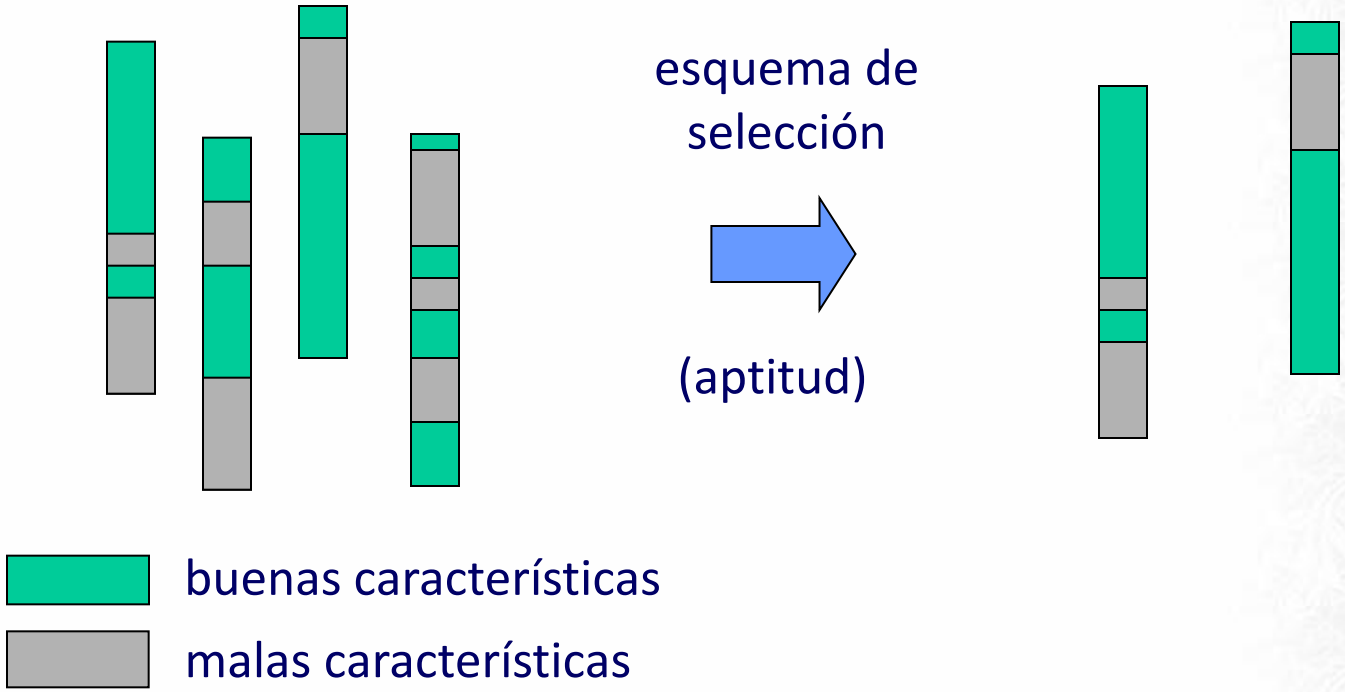
- En cada generación se aplican los operadores evolutivos
  - Se seleccionan padres de acuerdo a su valor de fitness
  - Se cruzan los padres para producir descendientes
  - Se mutan aleatoriamente los nuevos descendientes
  - Se insertan los nuevos individuos creados, que reemplazan a algunos de la generación anterior
- Son operadores probabilísticos
  - Se aplican de acuerdo a probabilidades
- Actúan sobre los genotipos y no sobre los fenotipos de los individuos

Algoritmo Genético Simple

Selección

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Objetivo: mantener las características de aquellos individuos mejor adaptados



Algoritmo Genético Simple

Selección

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- El mecanismo de selección determina fuertemente la operativa del mecanismo de búsqueda (dirige al AG hacia la exploración de secciones "prometedoras" del espacio de búsqueda)
- La presión de selección es crítica para el funcionamiento del AG
  - Una presión de selección alta puede ocasionar pérdida de diversidad y que la búsqueda termine prematuramente (en un óptimo local). Esta situación se denomina como convergencia prematura
  - Una presión de selección baja puede conducir a que la búsqueda avance mucho más lento de lo necesario
- Lo adecuado es intentar mantener un compromiso entre la exploración del espacio de búsqueda y la explotación de buenas soluciones
  - Utilizar una presión de selección baja en las generaciones tempranas, para lograr una amplia exploración
  - Utilizar una presión de selección alta en etapas finales de la evolución, para explotar las áreas más prometedoras

Algoritmo Genético Simple

Selección

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- El mecanismo de muestreo determina la selección de cromosomas
  - Determina un número esperado de descendientes en la próxima generación del algoritmo
- La precisión y eficacia de un algoritmo de muestreo se evalúa mediante ciertos parámetros:
  - Sesgo: la diferencia en valor absoluto entre el número real y el número esperado de descendientes de un individuo (el sesgo óptimo es 0)
  - Amplitud: el conjunto de posibles valores que puede tomar el número real de descendientes de un individuo. Se define la amplitud mínima como el valor que permite obtener un sesgo nulo
  - Complejidad computacional: establece la relación entre el tiempo empleado por el mecanismo de muestreo y los parámetros del algoritmo (tamaño de población, longitud de individuos, etc.). Es deseable que la complejidad sea lo más próxima posible a una función lineal
- Técnicas de muestreo: estocástico, determinista y mixto

Algoritmo Genético Simple

Selección

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Muestreo estocástico
  - Selección en dos fases: determinar el valor esperado de un cromosoma y luego convertir el valor esperado en un número de hijos
  - El más conocido es la selección proporcional (rueda de ruleta, Holland)
  - Muestreo Estocástico Universal (Baker, 1987): previene la aparición de cromosomas que dominan ampliamente y mantiene la diversidad
- Muestreo determinista
  - Ordena los individuos según aptitud y selecciona los mejores como padres
  - Selección truncada: los T% mejores individuos reciben 100/T copias
  - Selección en bloque: s copias para los n/s mejores individuos
  - Selección elitista: asegura la permanencia del (de los) mejor(es) individuo(s)
  - Reemplazo generacional: reemplaza todos los padres por sus hijos
  - Reemplazo elitista: reemplaza los peores individuos solamente

Algoritmo Genético Simple

Selección

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Muestreo mixto: incluye características deterministas y aleatorias
  - Selección por torneo (Goldberg): selecciona el (o los) mejor(es) individuos de un conjunto aleatorio (compiten en un *torneo*)
  - Selección por torneo estocástico: selección proporcional para seleccionar individuos, entre los cuales se efectúa el torneo
- ¿Cómo determinar la probabilidad de selección para cada individuo?
  - En un método proporcional, la probabilidad es proporcional a su aptitud
  - Inconvenientes: en generaciones tempranas unos pocos individuos muy adaptados dominan la selección, mientras que en las últimas generaciones (la población ha convergido) la competencia se hace débil y el comportamiento del AG simula a una búsqueda aleatoria
- Enfoques para resolver este problema
  - Mecanismos de escalado: determinan la probabilidad de supervivencia de un individuo de acuerdo a valores escalados de la función de fitness
  - Mecanismos de ordenamiento: en lugar de usar valores de fitness para determinar la supervivencia utilizan un ordenamiento de los individuos

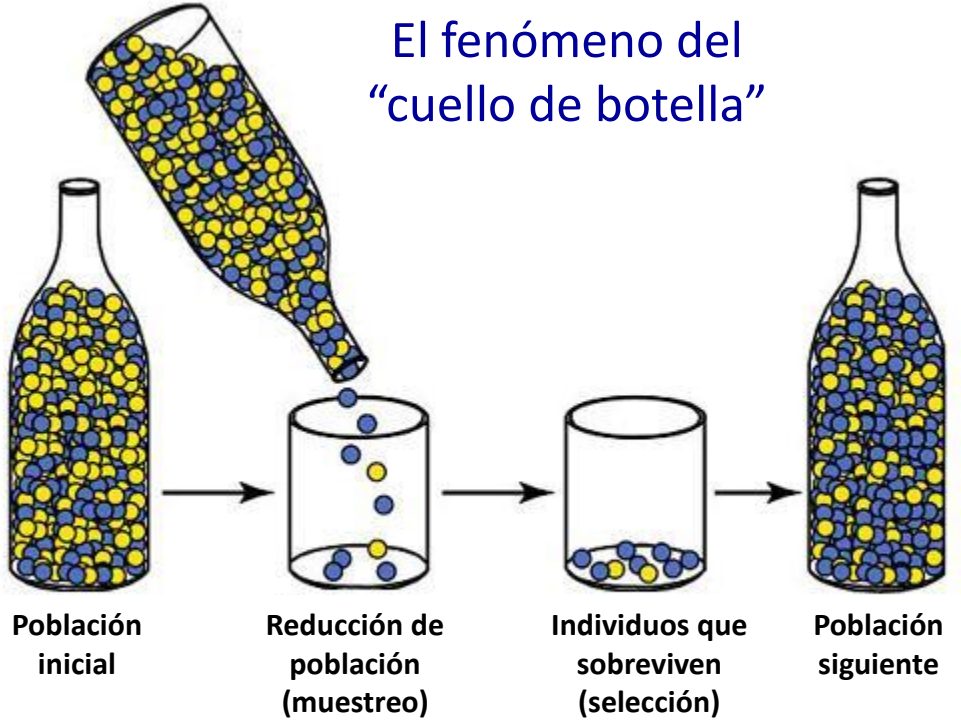
Algoritmo Genético Simple

Selección

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Deriva genética (*genetic drift*)
  - Fuerza evolutiva que actúa junto con la selección natural, cambiando las frecuencias alélicas de las especies en el tiempo
  - Es un efecto estocástico consecuencia del muestreo aleatorio en la selección y de la pérdida de alelos por azar y no por selección natural, que cambia la frecuencia de alelos de una generación a otra
  - Normalmente se da una pérdida de los alelos menos frecuentes y una fijación (frecuencia próxima al 100%) de los más frecuentes, resultando una disminución en la diversidad genética de la población

El fenómeno del "cuello de botella"



Algoritmo Genético Simple

Operadores de selección

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Selección proporcional
  - Implementada por la técnica de *rueda de ruleta*
  - Se asigna una probabilidad de selección proporcional al fitness relativo

$$ps_i = \frac{f(i)}{\sum_{j \in P} f_j}$$

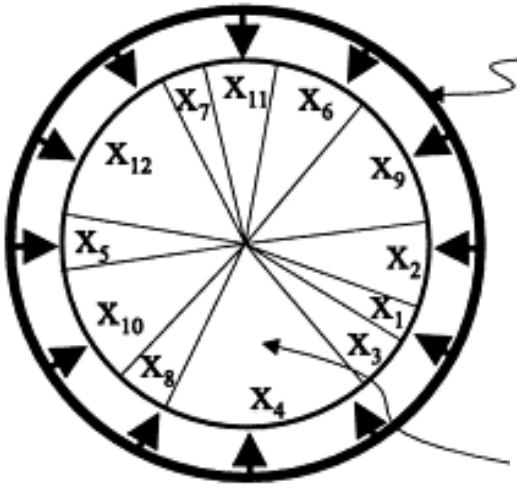
- Selección por rango (parámetro *k*)
  - Se ordenan los individuos según su fitness y se seleccionan los *k* mejores
- Selección por torneo (parámetros *k* y *r*)
  - Se eligen k individuos y se seleccionan los r de mejor fitness

Algoritmo Genético Simple

Operadores de selección

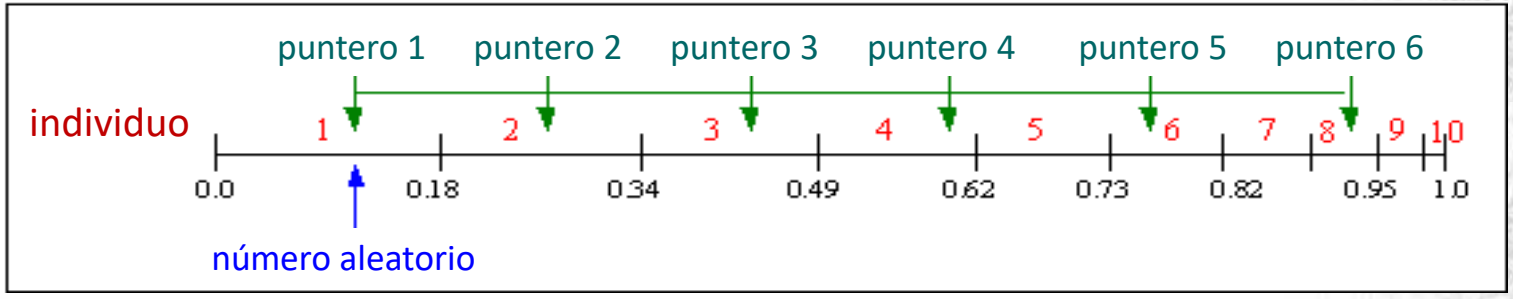
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Selección estocástica universal
  - Se implementa con una *ruleta con punteros equiespaciados* para eliminar el sesgo de la selección proporcional tradicional



Ruleta con punteros equiespaciados

Tamaño proporcional al fitness



Algoritmo Genético Simple

Cruzamiento

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Objetivo: garantizar la explotación de buenas regiones del espacio de búsqueda

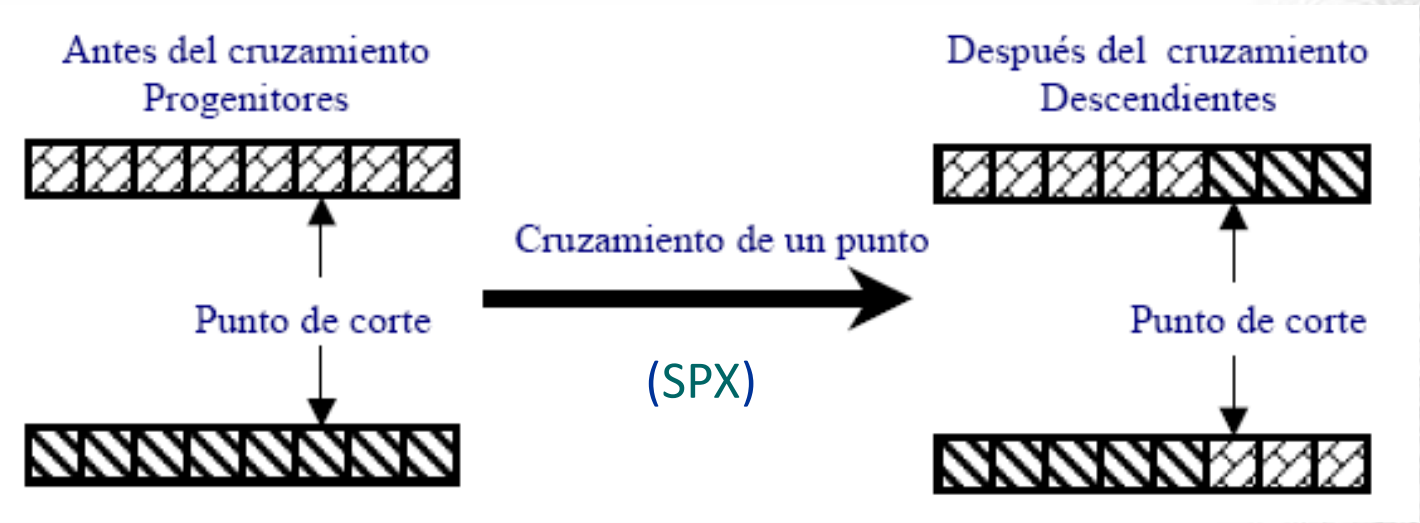
punto de cruce (cruzamiento de un punto)

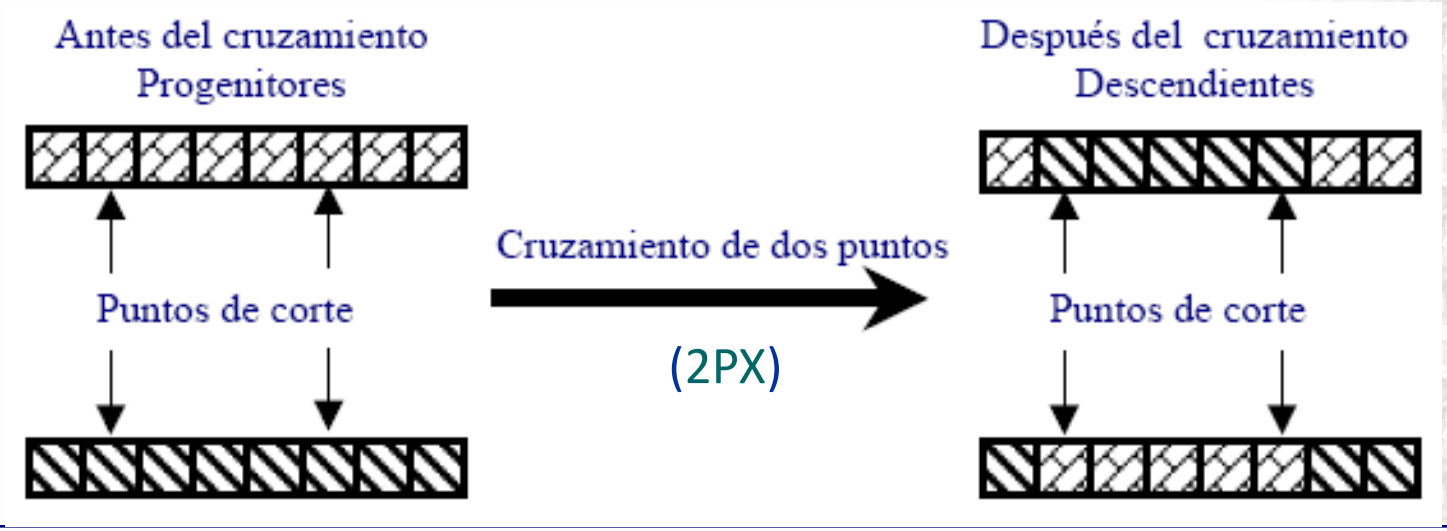
buenas características

malas características

Algoritmo Genético Simple

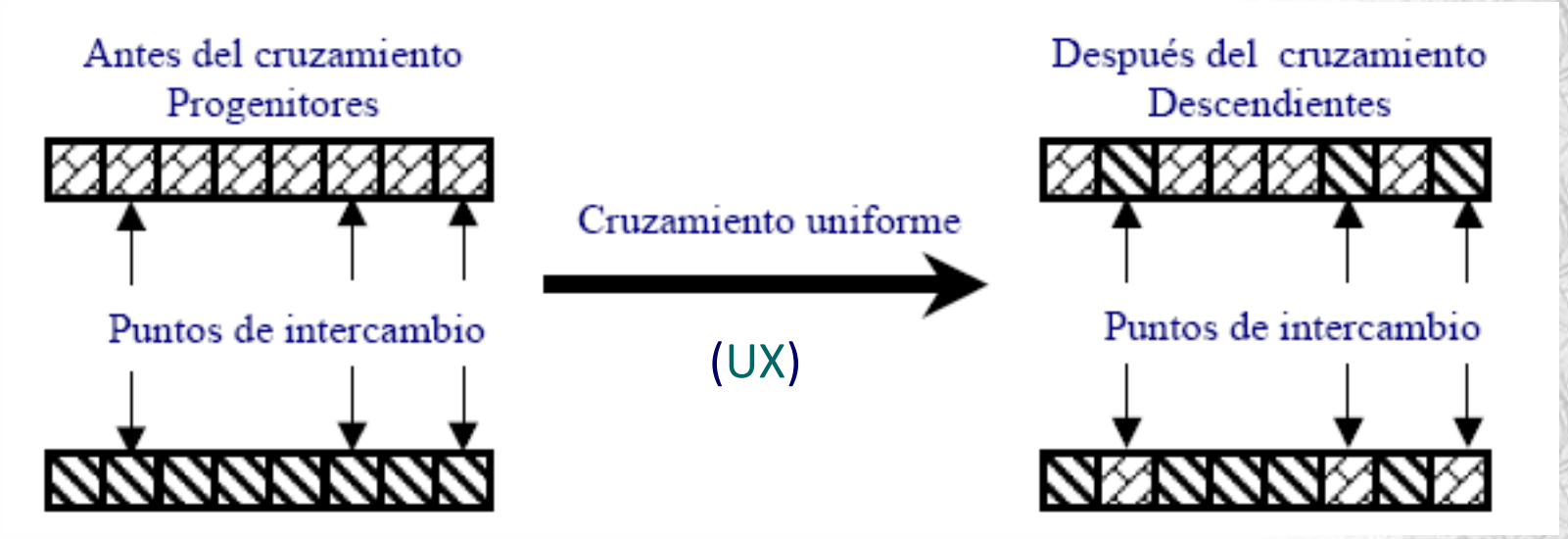
Cruzamiento de n puntos





Algoritmo Genético Simple

Cruzamiento uniforme



- Generalizado utilizando probabilidades de intercambio para cada punto de intercambio
- Más disruptivo que cruzamiento de n puntos
- Probabilidad de conservar grupos de bits es independiente del resto y de la posición en el individuo

Algoritmo Genético Simple

Cruzamientos específicos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

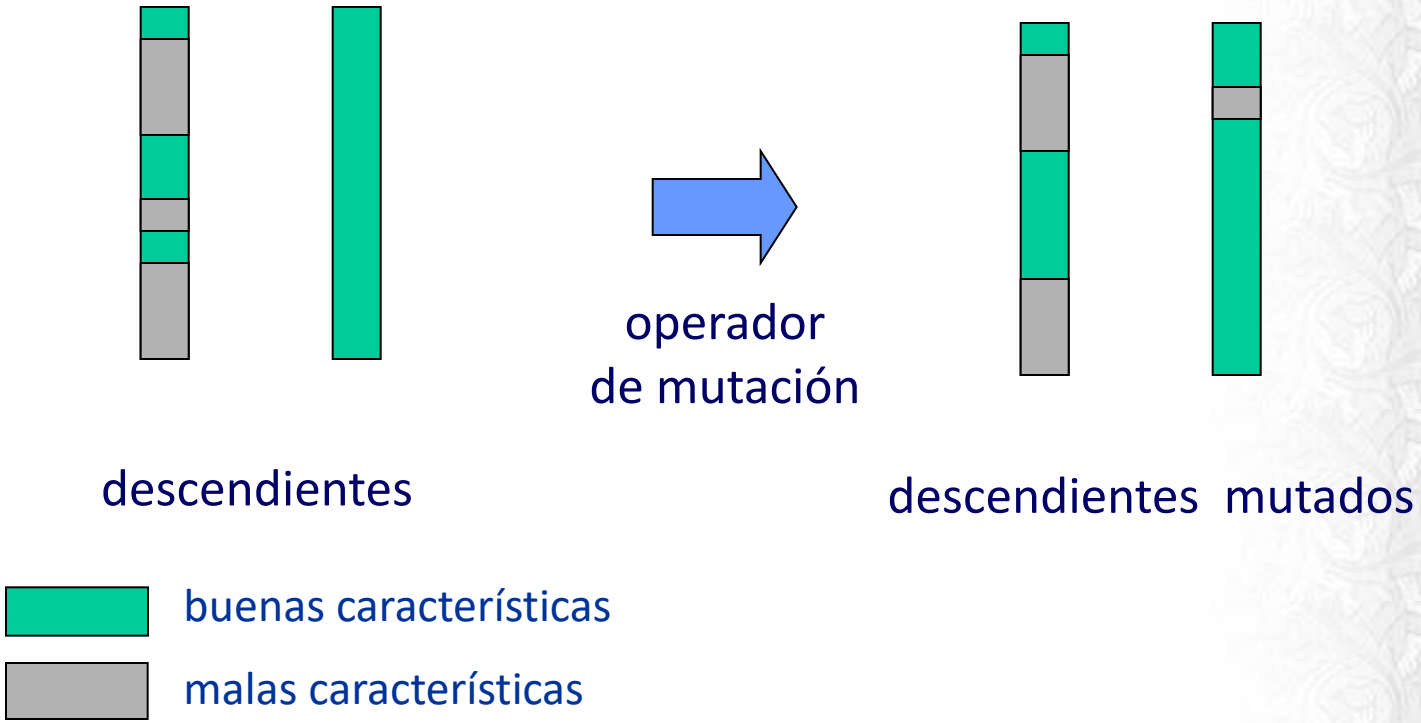
- Ciertas codificaciones requieren operadores de cruzamiento específicos
  - Para conservar las características de las soluciones del problema
  - TSP $\rightarrow$ solución debe ser un ciclo, cada solución está representada por una permutación, no es posible cruzar soluciones directamente
- Otras codificaciones admiten operadores de cruzamiento con operaciones específicas
  - Por ejemplo, la codificación con números reales admite cruzamientos aritméticos (promedio, combinación lineal, etc)
- En otros casos deben implementarse operadores de cruzamiento dependientes del problema
  - Para asegurar factibilidad de soluciones
  - Para conservar significado semántico
- Existen operadores de cruzamiento con varios padres y panmícticos
- Se presentarán operadores de cruzamiento para representaciones específicas en el Tema 6

Algoritmo Genético Simple

Mutación

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Objetivo: introducir diversidad (aleatoriamente) en los individuos de la población, posibilitando la exploración de diferentes secciones del espacio de búsqueda



Algoritmo Genético Simple

Mutación de inversión de valor de alelo

- Simple modificación de la codificación (material genético) en una posición determinada aleatoriamente
- Sigue la analogía con la mutación en la evolución natural

Operadores de mutación

- Ciertas codificaciones no admiten la mutación simple de un solo valor, y requieren operadores de mutación específicos
  - Para conservar las características de las soluciones del problema
  - TSP $\rightarrow$ solución debe ser un ciclo, cada solución está representada por una permutación, no es posible mutar un alelo (ciudad) en la permutación, pues un valor quedaría repetido.
- Otras codificaciones admiten operadores de mutación con operaciones específicas
  - Por ejemplo, la codificación con números reales admite mutación gaussiana (o utilizando otras distribuciones)
- En otros casos deben implementarse operadores de mutación dependientes del problema
  - Para asegurar factibilidad de soluciones
  - Para conservar significado semántico
- Se presentarán operadores de mutación para representaciones específicas en el Tema 7

Algoritmo Genético Simple

Ejemplo de resolución de problemas

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- One-Max: problema de maximización de función contínua o discreta, monótona creciente en un intervalo
- Caso de estudio: maximización de la función cuadrática $f(x) = x^2$ para $x \in Z$, en el intervalo D=[0,31]
- Operadores:
  - Representación binaria de enteros
  - Tamaño de población (como ejemplo para el análisis gráfico): 4 individuos
  - Operadores:
    - Selección proporcional
    - Cruzamiento de un punto
    - Mutación de inversión de bit

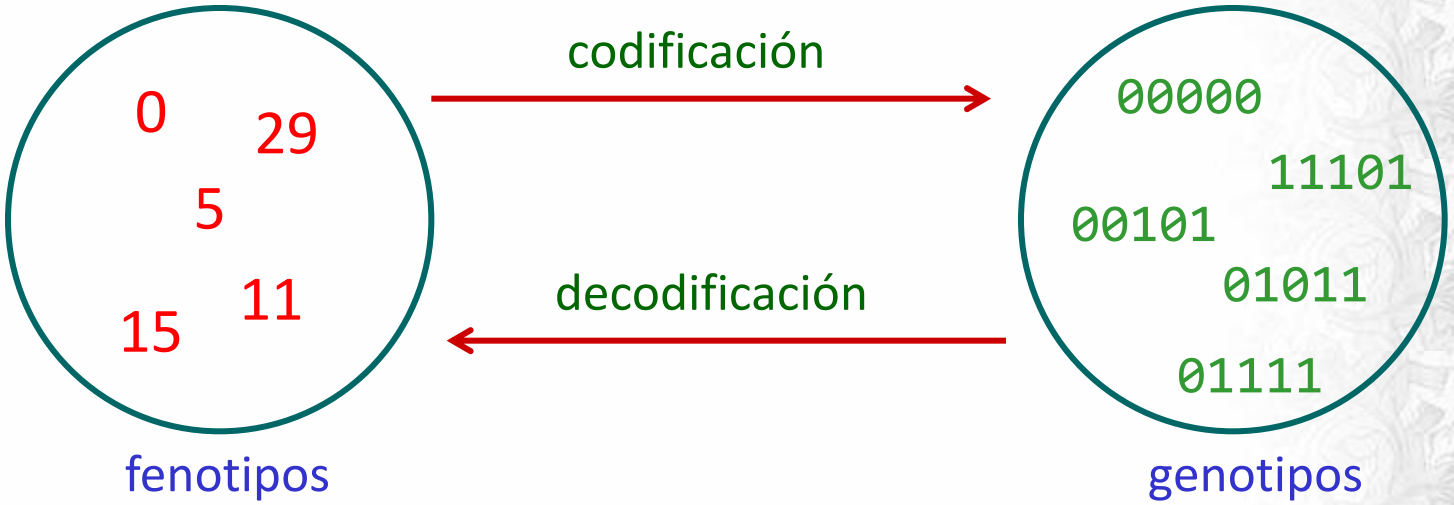
Algoritmo Genético Simple

Ejemplo: maximizar $f(x) = x^2$

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

1. Representación de soluciones
   - Código binario, representación de enteros
   - Dominio del problema: intervalo D=[0,31]
   - Se utilizan 5 bits para la representación de soluciones

0	1	0	1	1



Algoritmo Genético Simple

Ejemplo: maximizar $f(x) = x^2$

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

2. Evaluación de individuos
   - En este caso, se optimiza una función matemática, por lo cual la función de fitness es la propia función a optimizar
   - No es el caso general en problemas más complejos

índice	individuo	fitness	% del total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

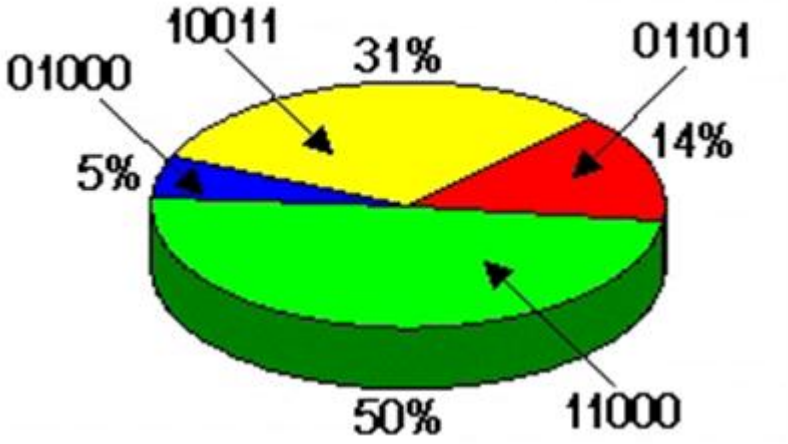
Algoritmo Genético Simple

Ejemplo: maximizar $f(x) = x^2$

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

3. Selección proporcional
   - Slots proporcionales a los valores de fitness relativo
   - Individuos mejor adaptados tendrán más posibilidades de obtener copias en la siguiente generación

$$ps_i = \frac{f(i)}{\sum_{j \in P} f_j}$$



Algoritmo Genético Simple

Ejemplo: maximizar $f(x) = x^2$

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

4. Recombinación
   - Combina características de los individuos seleccionados en la etapa anterior
   - Aplicado con una probabilidad $p_C \in [0.6, 1.0]$
   - Ejemplo: cruzamiento de un punto (SPX)
   - Selecciona uniformemente un punto de corte $l \in [1, \#i - 1]$

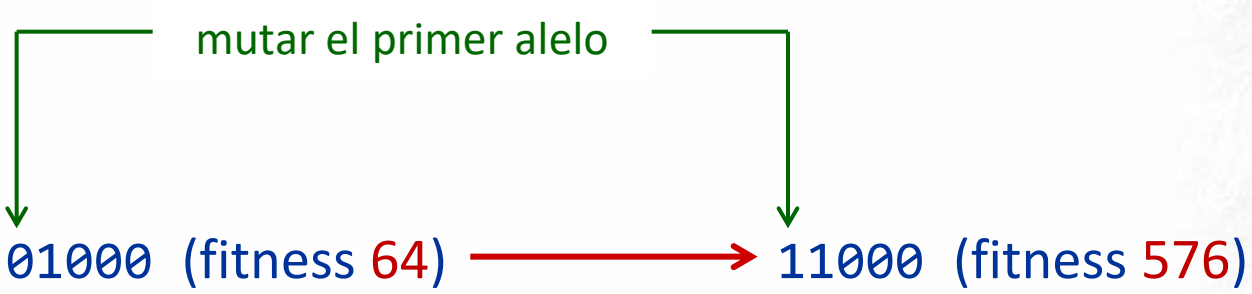
padres		hijos	
01 | 101	(fitness 169)	01000	(fitness 64)
11 | 000	(fitness 576)	11101	(fitness 841)

Algoritmo Genético Simple

Ejemplo: maximizar $f(x) = x^2$

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

5. Mutación
   - Modifica aleatoriamente (unos pocos componentes) de un individuo
   - Aplicado con una probabilidad $p_M \in [1\times10^{-3}, 0.1]$
   - Ejemplo: mutación de inversión de bit (FBM)
   - Muta un alelo con probabilidad p_M



Resolviendo un problema: genotipo y fitness

Genotipo: representación de soluciones

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Genotipo: cómo representar las soluciones del problema
- Deben definirse los mecanismos de codificación y decodificación
- La función de codificación es completa
  - Debe asociar una representación a cada posible solución
  - Pueden utilizarse codificaciones tradicionales (binaria, de enteros, real, etc.) o codificaciones específicas para determinados tipos de problemas (permutaciones, árboles, etc.)
  - La codificación es requerida para aplicar los operadores evolutivos, que trabajan sobre el genotipo
- La función de decodificación puede ser parcial
  - En caso de existir soluciones no factibles, restricciones o límites para las variables del problema
  - La decodificación debe aplicarse antes de evaluar las soluciones, ya que la función de fitness opera sobre los fenotipos

Resolviendo un problema: genotipo y fitness

Genotipo: representación de soluciones

- Representación de permutaciones
  - Ejemplo: TSP

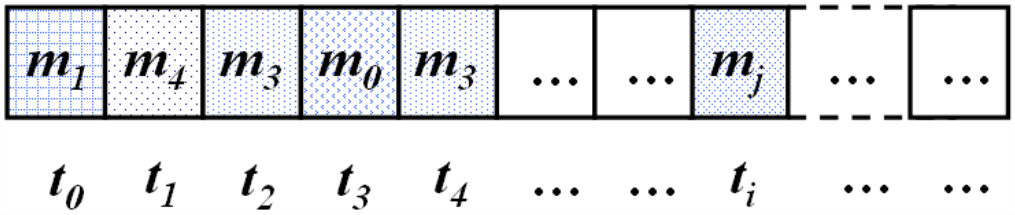
1	4	12	14	13	15	9	5	6	7	8	11	3	10

ciudad	id
Berlin	1
Bonn	2
Bremen	3
Dresden	4
Colonia	5
Düsseldorf	6
Duisburg	7
Essen	8
Frankfurt	9
Hamburg	10
Hannover	11
Leipzig	12
Munich	13
Nuremberg	14
Stuttgart	15

Resolviendo un problema: genotipo y fitness

Genotipo: representación de soluciones

- Representación de enteros
  - Ejemplo: problema de asignación de tareas a procesadores
  - Tareas $\{t_0, t_1, t_2, \ldots t_n\}$ a asignar en procesadores $\{m_0, m_1, m_2, \ldots m_n\}$



1	4	5	3	5	1	9	…	…	…	7
t_0	t_1	t_2	t_3	t_4	t_5	t_6	…	…	…	t_n

Resolviendo un problema: genotipo y fitness

Fitness: evaluación de soluciones

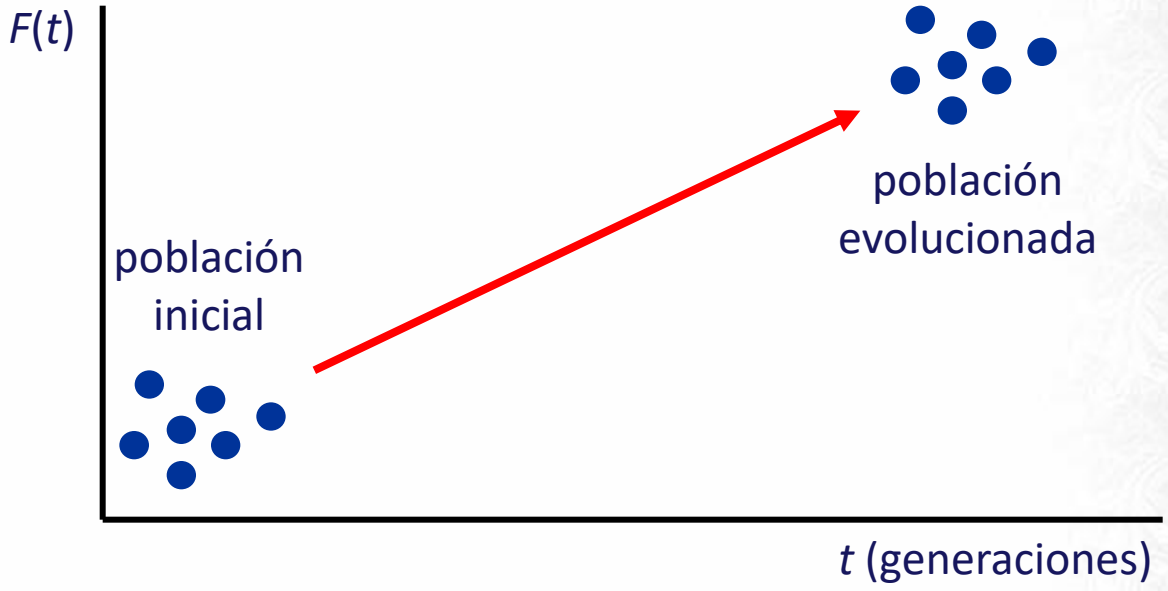
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Construcción de una función de fitness apropiada para el problema
  - La función de fitness depende del problema y del criterio de optimización
  - Opera directamente sobre las soluciones del problema (fenotipos)
  - Debe considerar las restricciones del problema
  - Puede definir objetivos múltiples (función vectorial, problemas multiobjetivo) o incorporar sub-objetivos
  - Puede cambiar dinámicamente a medida que un AE procede en la exploración (problemas dinámicos)
  - La función de fitness es una caja negra para un AE
    - Input: fenotipo
    - Output: valor de fitness

Resolviendo un problema: genotipo y fitness

Fitness: evaluación de soluciones

- La función de fitness guía el mecanismo de búsqueda a través del operador de selección
- La forma en que se produce la evolución se relaciona con el concepto de presión selectiva
- La influencia de la función de fitness es fundamental para determinar los individuos candidatos a sobrevivir (aquellos individuos a los que se aplicarán los operadores evolutivos)



Resolviendo un problema: genotipo y fitness

Fitness y diversidad

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Una presión selectiva alta (estrategias elitistas) tiende a reducir la diversidad de información genética en la población, al existir dominancia fuerte de los individuos más adaptados
- La pérdida de diversidad puede ser una desventaja o no, dependiendo fuertemente del problema a resover
  - Por ejemplo, la pérdida de diversidad es un inconveniente cuando se resuelve un problema multimodal, si un óptimo local se vuelve demasiado atractivo en el corto plazo
- Cuando la presión selectiva es baja o moderada, se mantiene (e incluso puede incrementarse) la diversidad
- La diversidad es un concepto fundamental para la resolución de problemas “difíciles”. Mantener diversidad en la información genética permite evitar los problemas de convergencia prematura
  - En estos casos, la diversidad es una medida de la robustez del AE

Resolviendo un problema: genotipo y fitness

Fitness: evaluación de soluciones

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- No existe una única función de fitness para un problema
- Es necesario considerar las características del problema de optimización a resolver
- Debe considerarse "el sentido" de la optimización. El formalismo de los AE propone maximizar el fitness, mientras que los problemas de optimización usualmente se plantean como una minimización de una función objetivo (costo)
- Es necesario transformar el problema de minimización de la función objetivo a uno de maximización de la función de fitness
- Por ejemplo, si el problema de optimización es min $f(x)$ se pueden considerar varias alternativas para definir una función de fitness $F(x)$:

1. $F(x) = - f(x)$ (opuesto)
2. $F(x) = 1/f(x)$ (inverso)
3. $F(x) = C - f(x)$ (respecto a costo máximo)
4. $F(x) = g(f(x))$ (genérica)

Resolviendo un problema: genotipo y fitness

Fitness: posibles dificultades

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Un posible problema es la generación de individuos no factibles durante la evolución
- Existen tres enfoques para tratar los individuos no factibles:
  1. Evitarlos en la codificación. En general no es un procedimiento sencillo, complica los procesos de codificación y decodificación
  2. Descartarlos es la opción más simple, pero conduce a la pérdida de características que podrían ser útiles para resolver el problema
  3. Penalizarlos en sus valores de fitness. Para problemas complejos puede ser dificultoso, al requerir estudios teóricos y empíricos para definir un modelo de penalización adecuado
- Pueden existir restricciones inherentes al problema
  - Por ejemplo, al trabajar con funciones discontinuas, existen individuos para los cuales no hay un valor de fitness asociado (en este caso la función de fitness es parcial)

Resolviendo un problema: genotipo y fitness

Fitness: posibles dificultades

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Pueden existir dificultades asociadas al propio mecanismo de evaluación de aptitud de los individuos
  - La función de fitness puede ser muy costosa de evaluar algorítmicamente, puede ser multivaluada, e inclusive puede tener una expresión analítica desconocida (por ejemplo, debe estimarse con simulaciones)
- Debe considerarse la posible variabilidad de la función de fitness
  - Como pueden existir varias funciones de fitness, pueden existir problemas asociados a las transiciones
  - Se pueden tomar medidas para evitar cambios bruscos en la evolución (por ejemplo, combinar linealmente varias funciones de fitness, o utilizar memoria)
- Operadores específicos podrían tener requerimientos sobre el fitness
  - Por ejemplo, la selección proporcional requiere valores positivos de fitness para calcular el fitness proporcional

Resolviendo un problema: genotipo y fitness

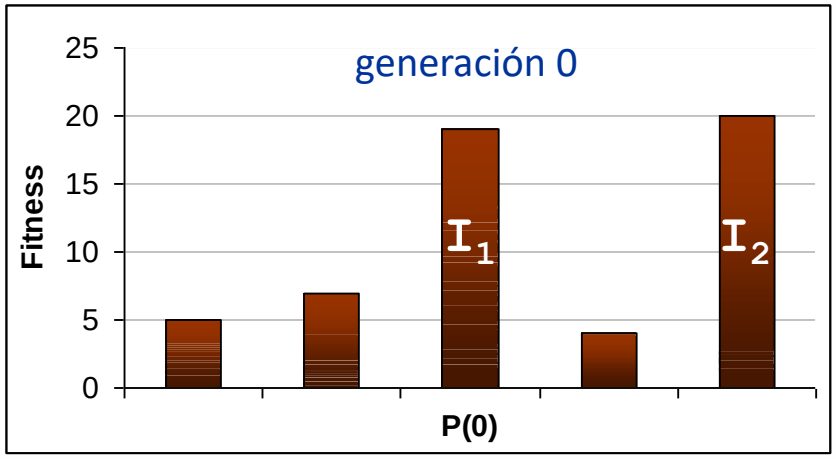
Escalado del fitness

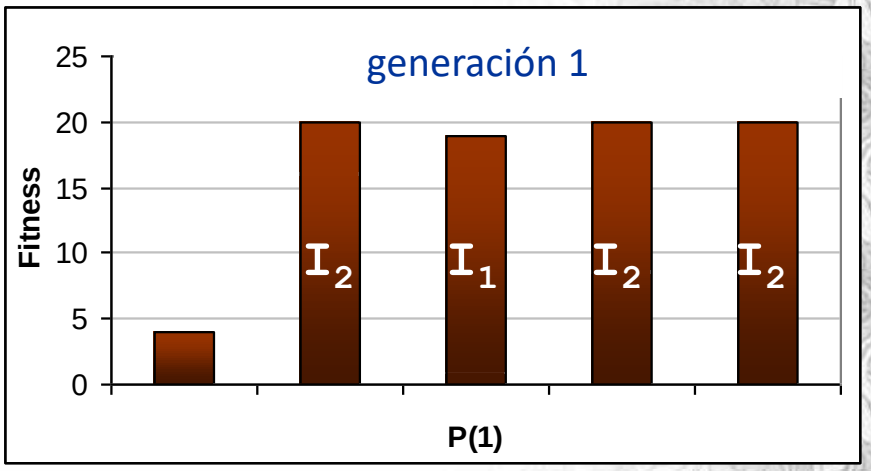
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Se suelen utilizar mecanismos avanzados que permitan mejorar la eficacia de la función de fitness
- Uno de los mecanismos más utilizados es el escalado
- El escalado busca solucionar dos problemas que tradicionalmente surgen cuando se utilizan mecanismos de selección proporcional a los valores de fitness:
  1. Dominancia prematura: dominancia de individuos muy adaptados en las etapas tempranas del proceso evolutivo
  2. Caminata aleatoria: (*random walk*) entre individuos con valores de fitness similares, que puede ocurrir en las etapas avanzadas de la evolución

Resolviendo un problema: genotipo y fitness

Escalado del fitness





dominancia inicial en etapas tempranas de la búsqueda

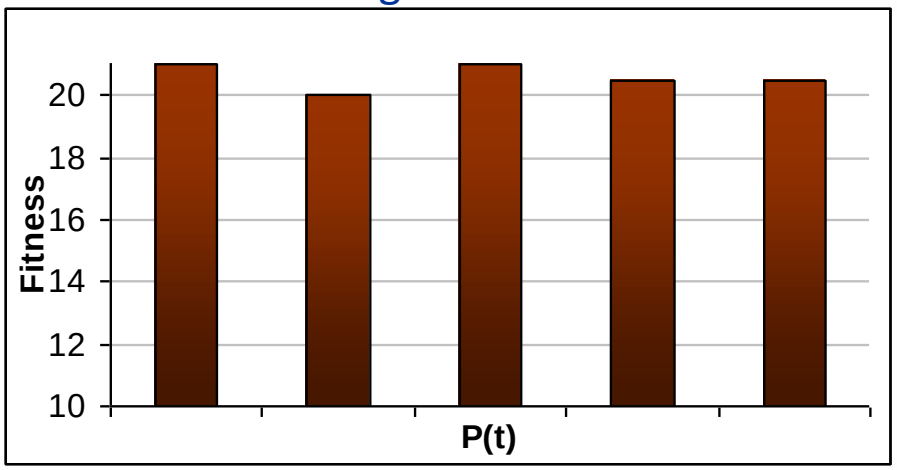
- En la selección hay una presión selectiva media
- Dominan los individuos I_1 e I_2, por tener altos valores de fitness relativo al resto de la población
- Se pierde rápidamente la diversidad genética

Resolviendo un problema: genotipo y fitness

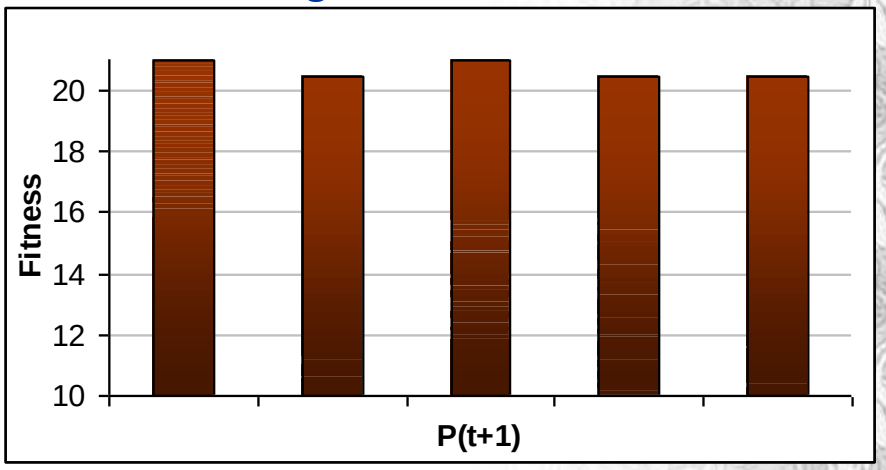
Escalado del fitness

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

generación t



generación t+1



caminata aleatoria en etapas avanzadas de la búsqueda

- En la selección hay una presión selectiva baja o media
- Las diferencias entre los valores de fitness son muy poco significativas
- La selección proporcional produce resultados estocásticos

Resolviendo un problema: genotipo y fitness

Escalado del fitness

- La idea consiste en resolver los problemas transformando la función de fitness, de forma de evitar las grandes o pequeñas diferencias de valores (según el caso)

$$F \xrightarrow{\sim} F_{Escalado}$$

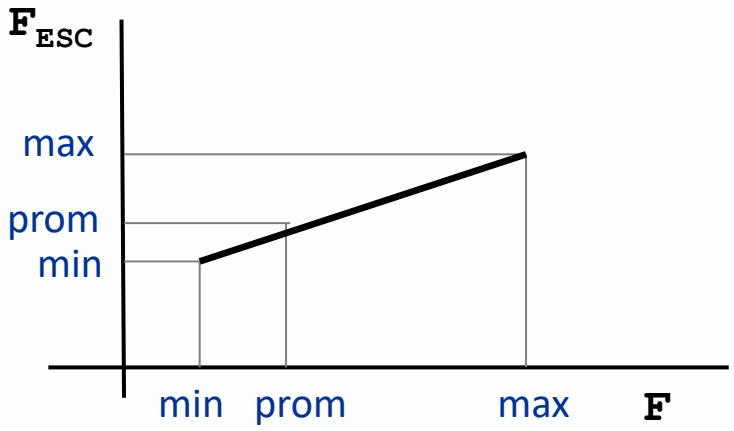
- Métodos para escalado:
  - Escalado lineal
  - Truncamiento Sigma
  - Escalado potencia
- Goldberg propone utilizar un enfoque diferente, el "*número esperado de copias*" como criterio para evitar la dominancia inicial

Resolviendo un problema: genotipo y fitness

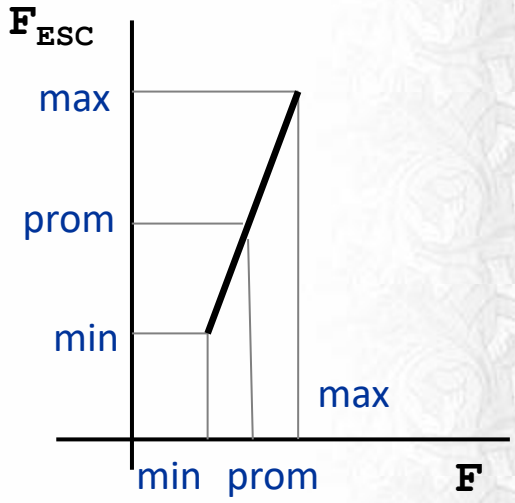
Escalado lineal

$$F_{Escalado} = a.F + b$$

- La pendiente de la recta queda determinada por *a* y *b,* que son parámetros del modelo y dependen del problema
- La idea funciona correctamente para ambos casos presentados
  - Pueden existir problemas debido a la generación de fitness negativos



baja pendiente: reduce diferencias entre valores de fitness



alta pendiente: aumenta diferencias entre valores de fitness

Resolviendo un problema: genotipo y fitness

Otros mecanismos de escalado

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Truncamiento sigma (Forrest, 1985)

$$F_{escalado} = F - (\bar{F} - c\sigma)$$

  - Incorpora información sobre la variación de los valores de fitness de la población (antes de realizar el escalado) mediante la desviación estándar
  - La constante *c* se utiliza para definir un múltiplo de la desviación estándar de la población (usualmente se utiliza *c* entre 1 y 3)
  - Los valores negativos son ajustados a 0 en forma arbitraria
  - Después de aplicar el truncamiento sigma, se puede proceder a realizar el escalado, sin el riesgo de obtener valores negativos

- Escalado potencia (Gillies, 1985)

$$F_{escalado} = F^k$$

  - El valor del parámetro *k* depende del problema a resolver
  - Se suele aplicar autoadaptación del valor de *k* durante la ejecución para aumentar o disminuir el rango de variación de los valores de fitness, según sea necesario

Resolviendo un problema

1 - Diseño de redes de comunicaciones confiables

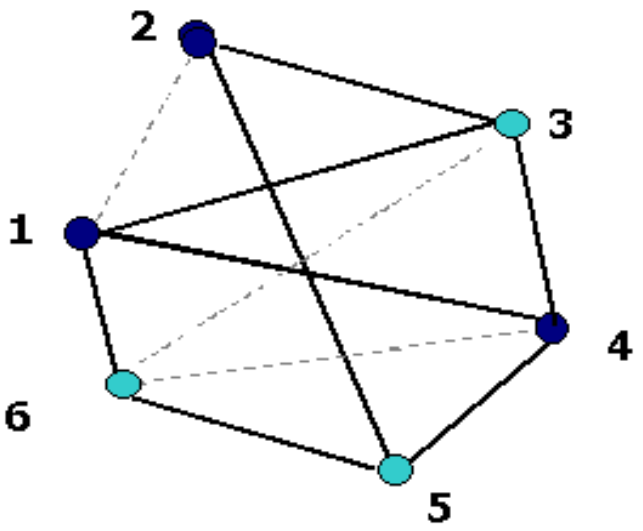
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Problema de Steiner generalizado
  - G=(V,E,C), no dirigido. V: conjunto de nodos, E: conjunto de aristas, C: costos asociados a las aristas de G
  - Subconjunto fijo $T \subseteq V$ (nodos *terminales*), $2 \leq n_T = |T| \leq n = |V|$.
  - Matriz n_T x n_T, simétrica $R=r_{ij} \in Z^+$ de *requisitos de conectividad* (caminos disjuntos entre todo par de nodos terminales $i,j \in T$)
- GSP: hallar subgrafo de costo mínimo $G_T \subseteq G$ tal que $i,j \in T$, sean r_{ij} arista conexos (r_{ij} *caminos disjuntos en aristas*) en G_T
- Sobre *nodos de Steiner* no se plantean requisitos de conectividad
  - Opcionalmente pueden ser utilizados para asegurar conectividad o reducir el costo de una solución

Resolviendo un problema

1 - Diseño de redes de comunicaciones confiables

- Representación:
  - Binaria simple basada en aristas (arreglo de bits indexado en 0,|E|-1)
  - Operadores sencillos de implementar, pero pueden generar soluciones no factibles.
  - El enfoque de descartar soluciones no factibles (Esbensen, 1994) evita cuantificar distancias al conjunto de soluciones factibles, y las complejidades de definir modelo de penalización para el fitness.



- Chequeo de factibilidad:
  1. verificar grados de nodos terminales
  2. Si son compatibles con requisitos se hallan caminos entre terminales (aplicando el algoritmo de Ford-Fulkerson)

Resolviendo un problema

1 - Diseño de redes de comunicaciones confiables

- Función de fitness
  - Evalúa el costo del grafo (diseño de red) representado por una solución

$$f = C_{ORIG} - \sum_{i=0}^{|E|-1} [EDGE(i) * C(i)]$$

    - C_{ORIG} : costo del grafo original (todas las aristas presentes)
    - C: $N \rightarrow R$ retorna el costo de una arista
    - $EDGE$: N $\rightarrow$ {0,1}, retorna el valor binario de una arista en la representación

- Operadores tradicionales:
  - Selección proporcional
  - Cruzamiento de dos puntos o uniforme
  - Mutación de inversión de bit
  - El método basado en Ford-Fulkerson puede usarse como mecanismo de corrección para soluciones no factibles

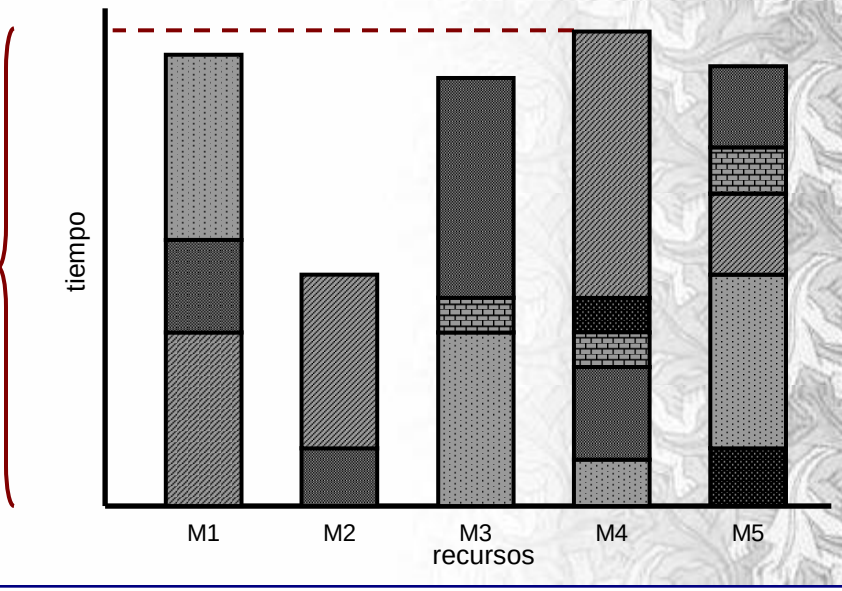
Resolviendo un problema

2 - Planificación de sistemas heterogéneos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Heterogeneous computing scheduling problem (HCSP)
  - Conjunto de máquinas *heterogéneas* $P = \{m_1, m_2, ..., m_M\}$
  - Conjunto de tareas $T = \{t_1, t_2, ..., t_N\}$ a ser ejecutadas en P
  - Tiempo de ejecución de tareas ET: $P \times T \rightarrow R$
    - Tiempo requerido para ejecutar la tarea t_i en la máquina m_j
  - Objetivo: encontrar una planificación (función f: $T^N \rightarrow S^M$) que minimiza el *makespan*

$$makespan = \max_{m_j \in P} \sum_{\substack{t_i \in T: \\ f(t_i)=m_j}} ET(t_i, m_j)$$

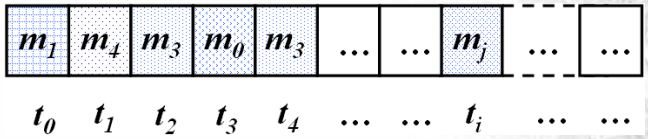


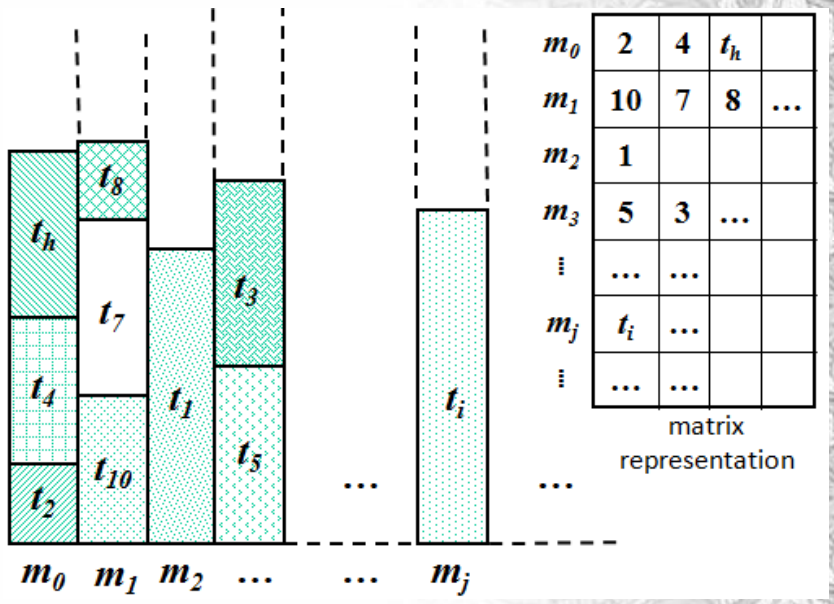
Resolviendo un problema

2 - Planificación de sistemas heterogéneos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Codificación: enteros representan tareas
  - Orientada a tareas (simple, 1D)
  - Orientada a máquinas (matricial, 2D, permite calcular eficientemente el makespan)





- Función de fitness
  - *fitness* = (-1) × makespan

- Inicialización de la población
  - Pueden utilizarse heurísticas simples de asignación de recursos

Resolviendo un problema

2 - Planificación de sistemas heterogéneos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

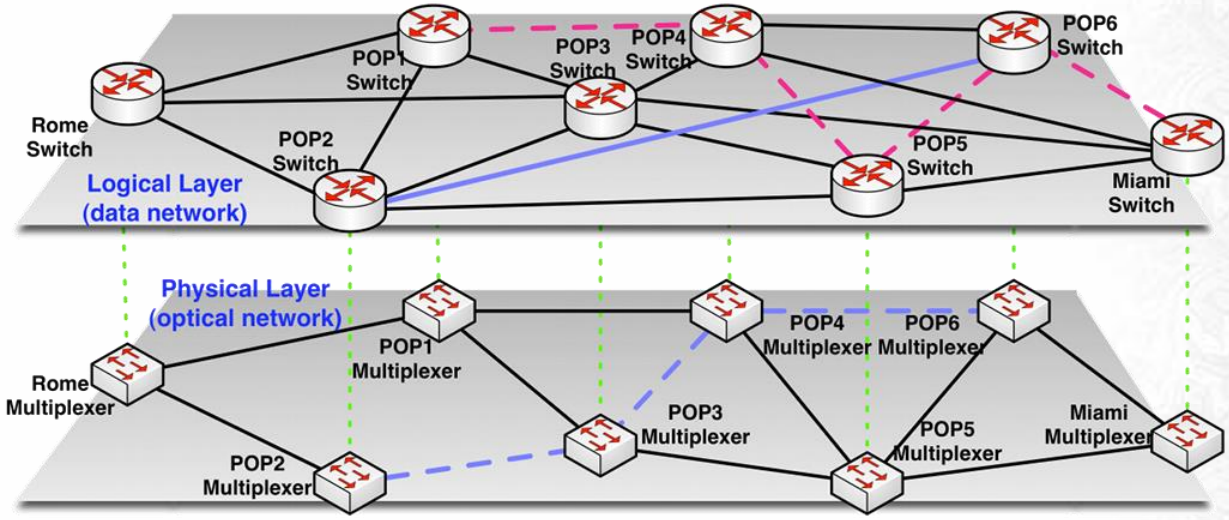
- Explotación: recombinación
  - Cruzamiento de un punto o cruzamiento uniforme
  - Reparación es requerida para evitar repeticiones de tareas
- Exploración: mutación
  - Movimientos e intercambios:
    1. Mover una tarea aleatoria desde *heavy* a *light*
    2. Mover la tarea más larga desde *heavy* a la mejor máquina
    3. Mover hacia *light* la mejor tarea
    4. Seleccionar una tarea de *heavy* y buscar la mejor máquina para moverla, tomando en cuenta la planificación actual
  - Intercambio: complementar la mutación con un moviemiento en sentido opuesto, aplicado probabilísticamente (probabilidad 0.5)

Resolviendo un problema

3 - Diseño de redes overlay

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Una red lógica (overlay) se construye sobre una red física



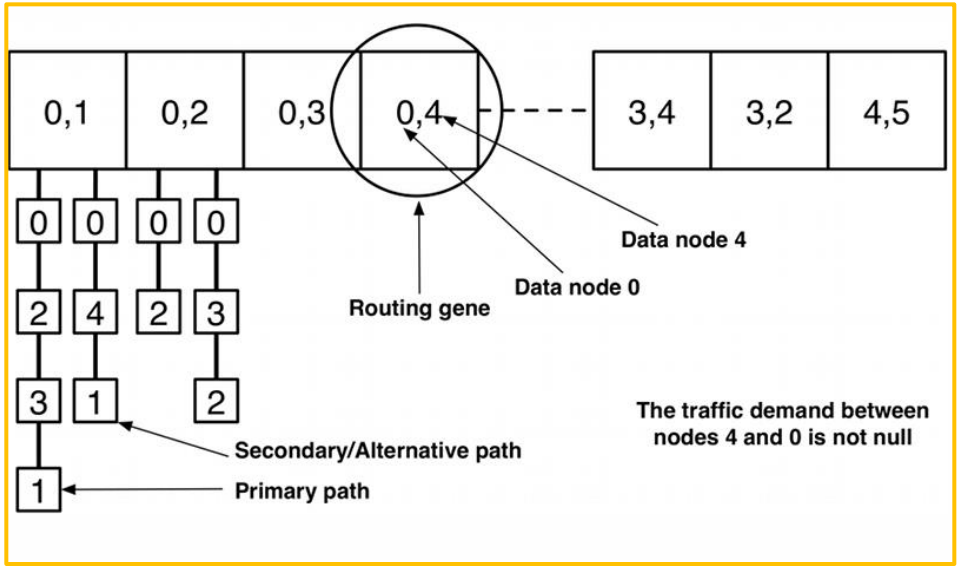
- Sus nodos se conectan por enlaces virtuales, construidos usando enlaces físicos reales de la red de base, que son fijos
- El tráfico de datos sigue un camino (túnel) entre pares de nodos
- Los túneles pueden reconfigurarse ante fallos en los enlaces lógicos
- Los enlaces tienen restricciones de capacidad

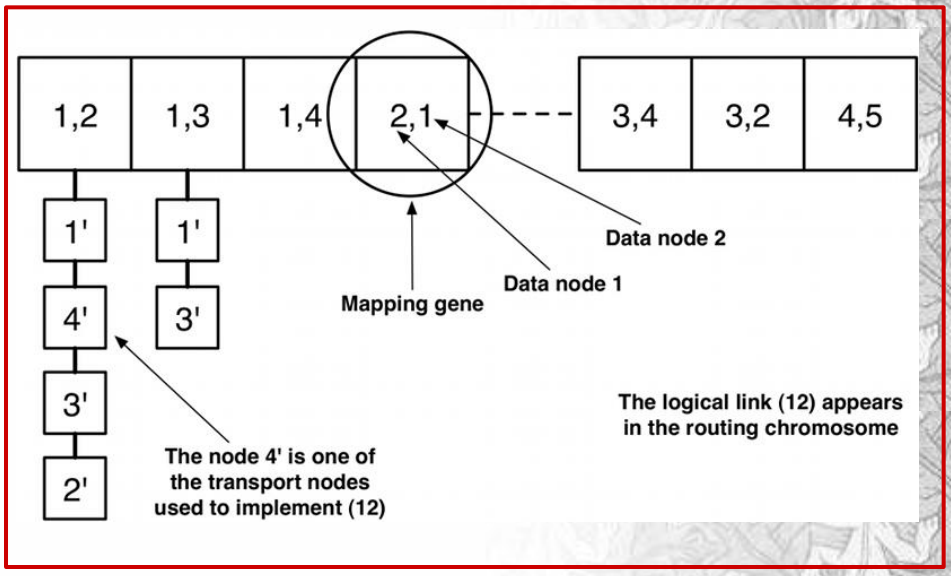
Resolviendo un problema

3 - Diseño de redes overlay

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Representación: dos cromosomas se usan para codificar una solución
  - Codificación de rutas (*routing gene*): contiene una lista de pares de nodos lógicos y la lista de nodos lógicos que conforman el túnel activo y el túnel secundario (disjuntos)
  - Codificación de mapeo (*mapping gene*): contiene la lista de nodos físicos usados para implementar un enlace lógico (*lightpath*)



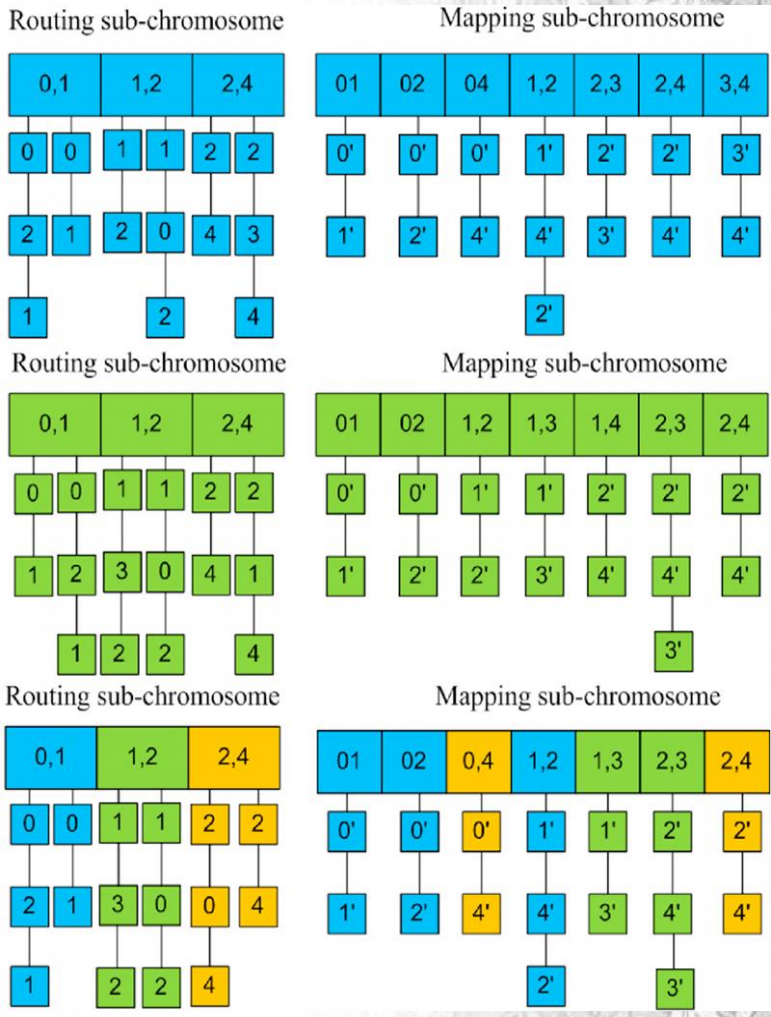


Resolviendo un problema

3 - Diseño de redes overlay

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Cruzamiento específico
  - Hijo copia las rutas y todos los mapeos correspondientes de un padre elegido al azar
  - Se verifica que los túneles primario y alternativo sean disjuntos
  - Se chequean las restricciones de capacidad en los enlaces
  - El proceso se repite hasta que las rutas están completas o no hay más información en los padres
  - Si el hijo no está completo, se aplica como corrector una construcción iterativa greedy



3 - Diseño de redes overlay

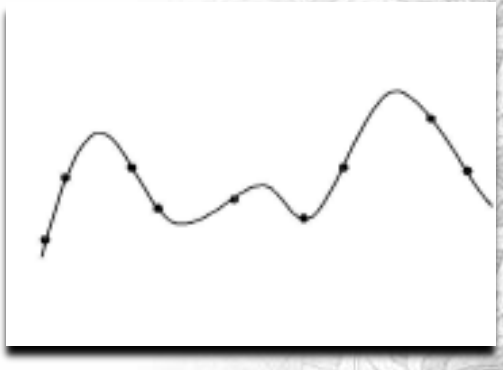
- Mutación: 5 operadores
  - Mutación a nivel lógico: reconstruye un gen aleatorio de la codificación de rutas usando el generador/corrector greedy
  - Mutación a nivel físico: cambia la codificación de mapeo seleccionando aleatoriamente un conjunto de genes de mapeo y buscando un nuevo camino físico
  - Mutación de enlace Tabu: selecciona aleatoriamente un enlace de datos "tabu" que se elimina de la solución, junto con sus genes asociados. Luego, el generador/corrector greedy se aplica para reconstruir la solución
  - Mejor mutación lógica : aplica una búsqueda local (acotada en iteraciones) en la vecindad de la codificación de rutas
  - Mejor mutación física: análogo a la anterior, pero operando sobre la codificación de mapeo

Algoritmos evolutivos

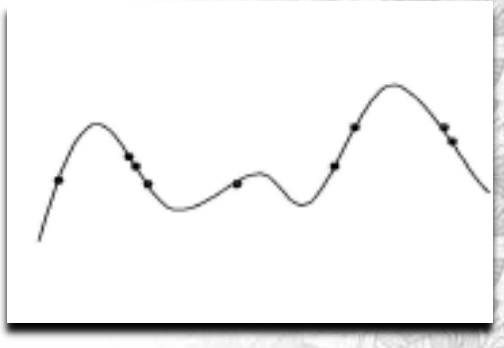
Comportamiento típico de la búsqueda

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

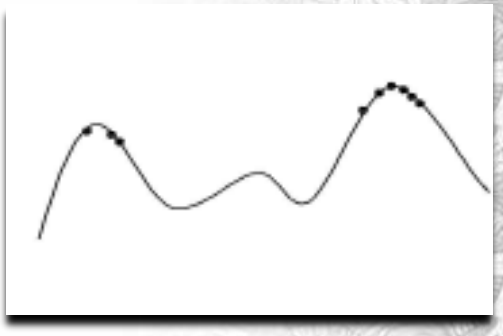
- Fase temprana:
  - Distribución de individuos cuasi-aleatoria



- Fase intermedia:
  - La poblacion comienza a concentarse sobre/alrededor de las colinas

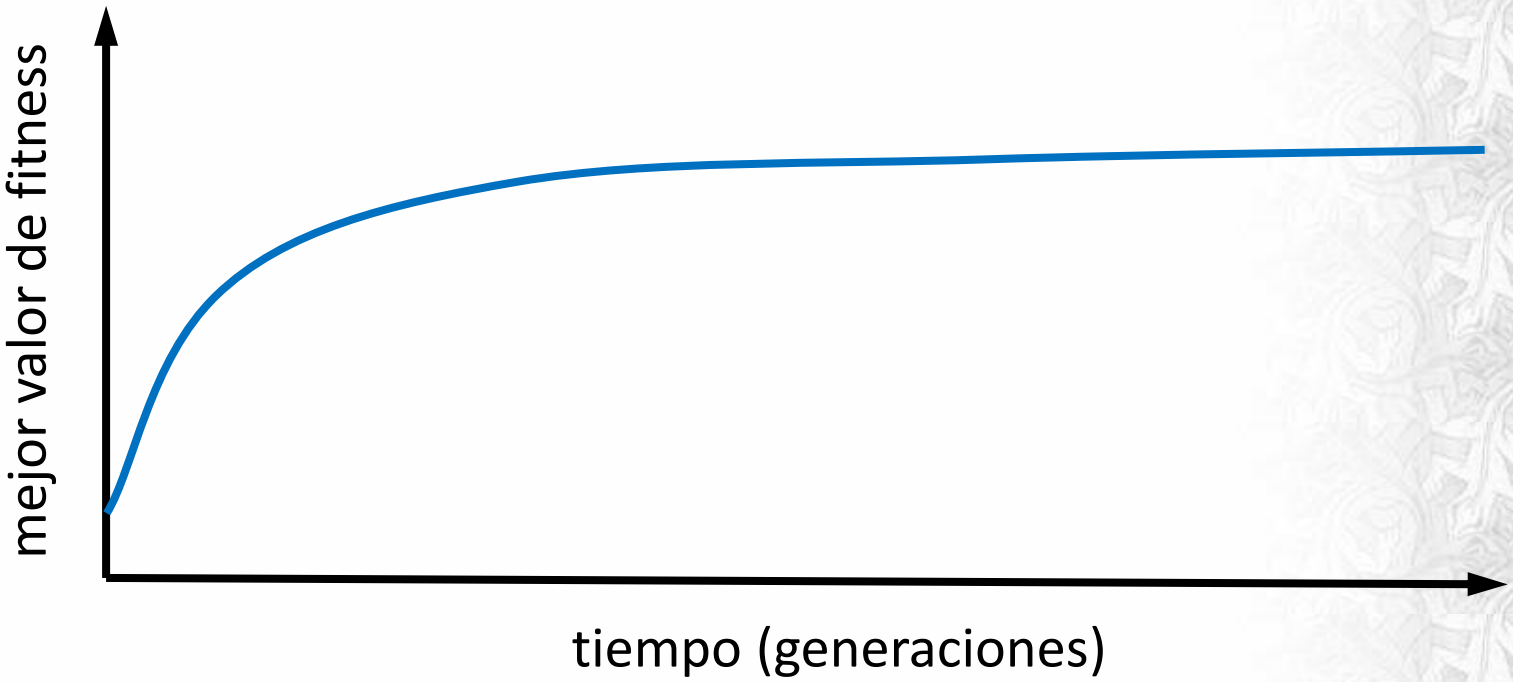


- Fase final:
  - La población se concentra sobre los óptimos locales



Comportamiento típico: fitness vs tiempo

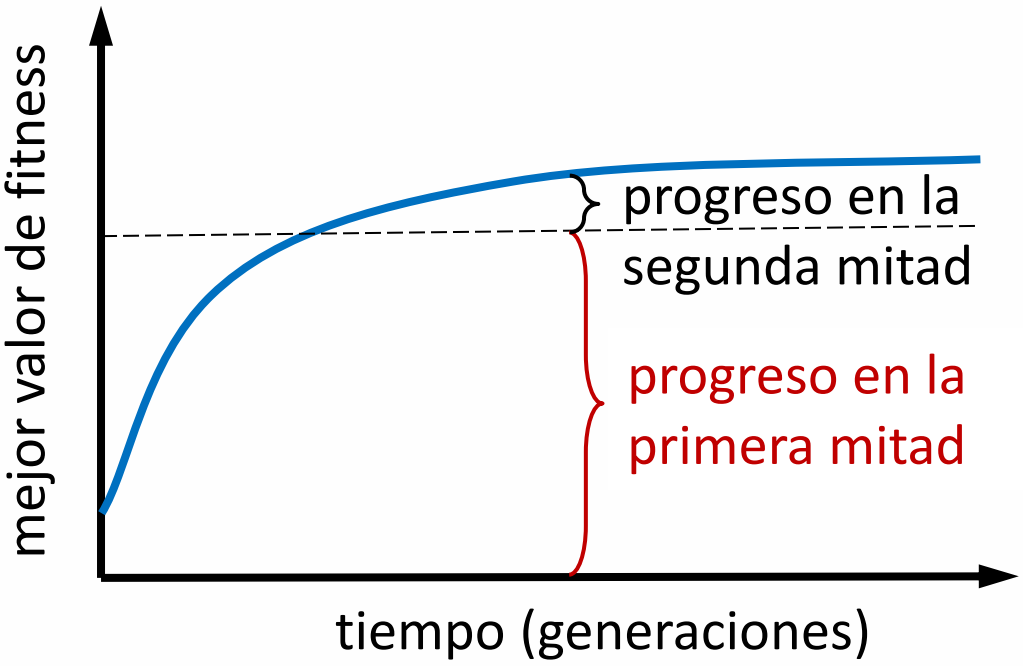
UNIVERSIDAD DE LA REPÚBLICA URUGUAY



Comportamiento típico de un AE

Algoritmos evolutivos

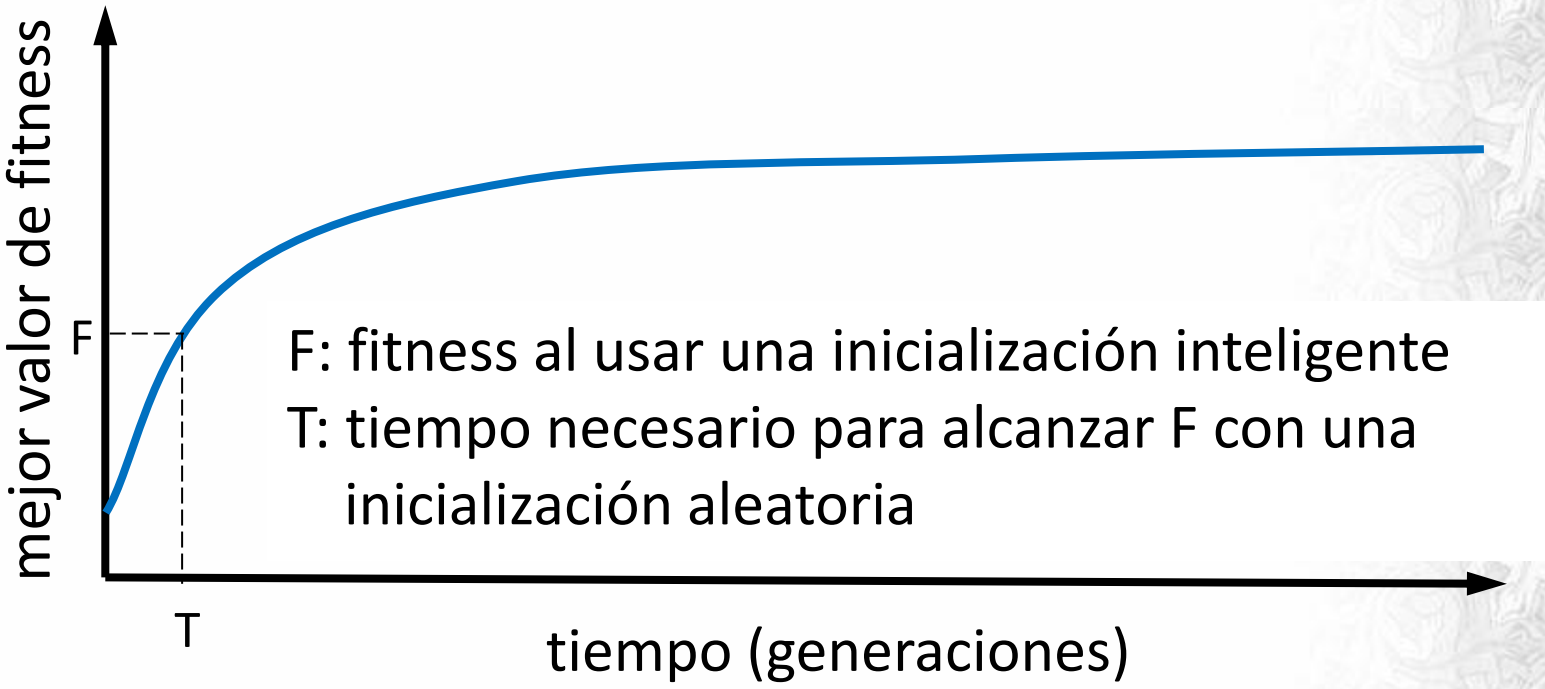
Comportamiento típico: fitness vs tiempo (generaciones)



- Son beneficiosas las ejecuciones largas?
  - Depende de cuanto valor se le asigna a los últimos progresos
  - Puede ser más beneficioso hacer muchas ejecuciones cortas

Algoritmos evolutivos

Comportamiento típico: fitness vs tiempo (inicializaciones)



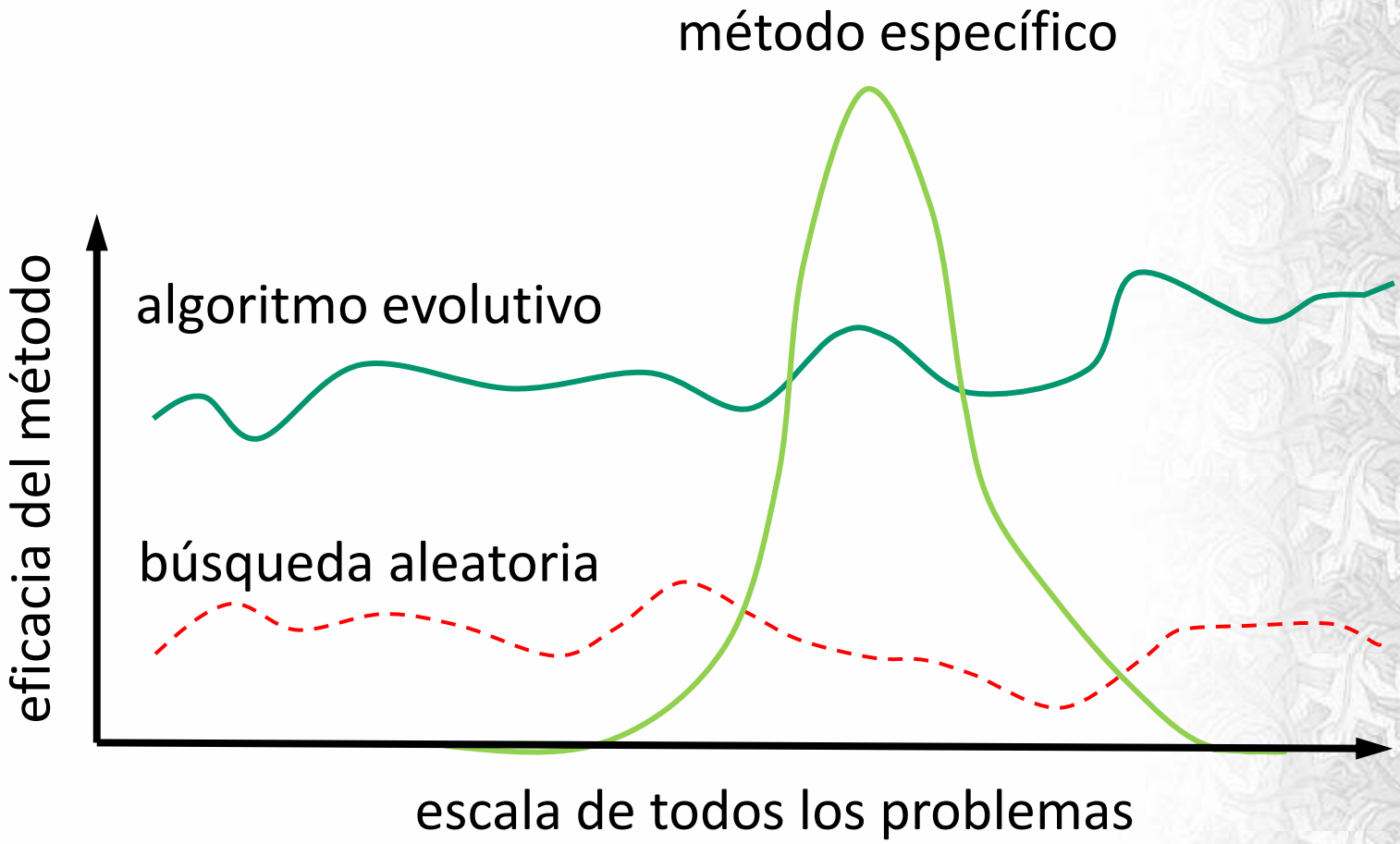
- Vale la pena invertir esfuerzo en inicializaciones inteligentes?
  - Enfoque útil si se conocen buenos métodos de inicialización para el problema (heurísticas que funcionen como “semilla” para la población inicial)
  - Debe validarse sobre una variedad de problemas e instancias

Aplicabilidad en problemas realistas

- Hay muchas opiniones sobre el uso de algoritmos genéticos en optimización
- Para la mayoría de los problemas un algoritmo específico puede:
  - Funcionar mejor que cualquier algoritmo genérico en la mayoría de las instancias
  - Pero tener limitada utilidad en otro dominio
  - No funcionar bien para algunas instancias
- El objetivo es proveer herramientas robustas que:
  - Funcionan de manera aceptable en una amplia cantidad de casos de aplicación
  - Aplicables sobre una variedad de problemas e instancias

Algoritmos evolutivos

Resolución de problemas (Goldberg, 1989)



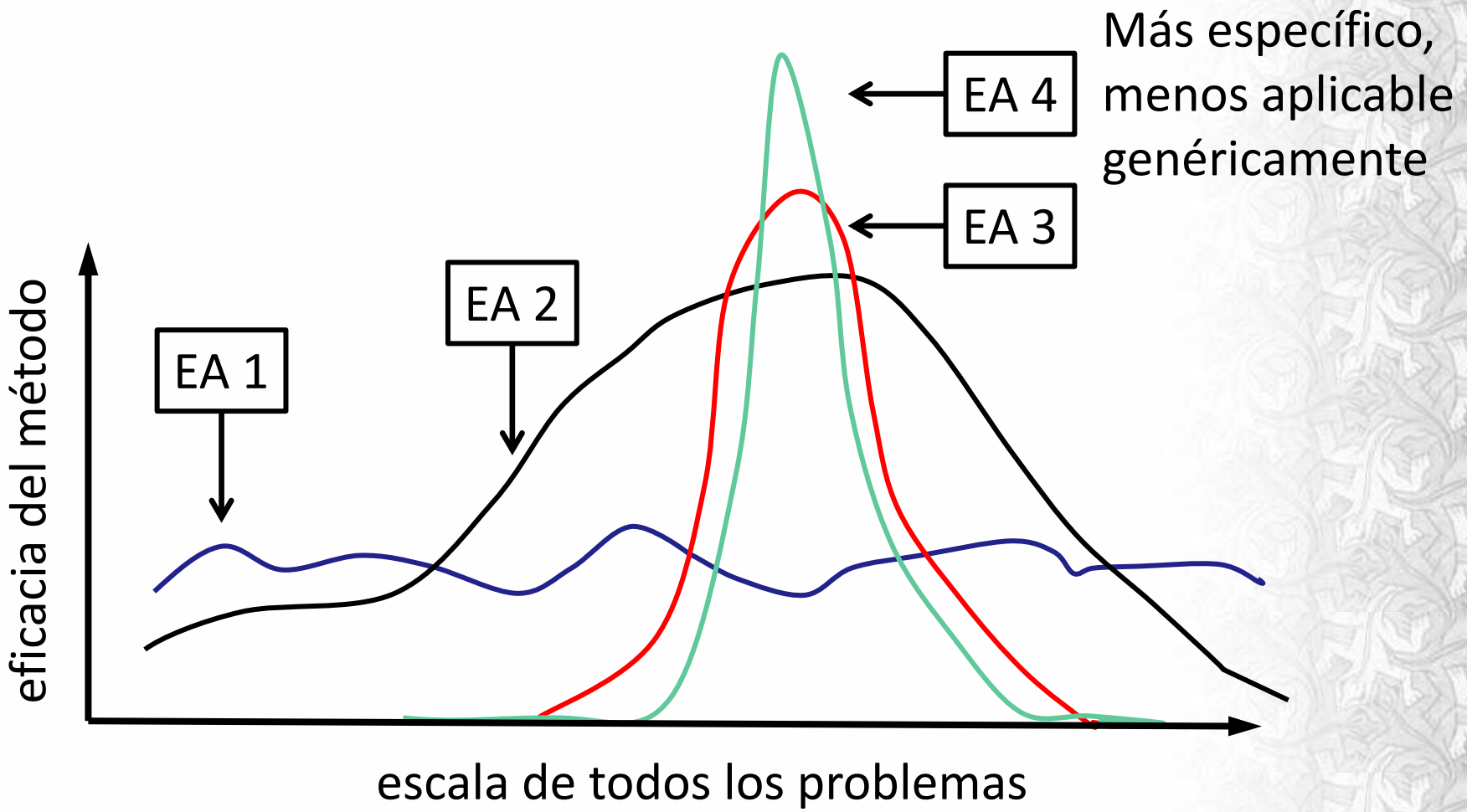
Goldberg presentó esta vista de aplicabilidad genérica en 1989

Resolución de problemas: conocimiento del dominio

- Moda de la década de 1990:
  - Agregar conocimiento del problema a los AE, mediante:
    - Operadores de mutación
    - Operadores de cruzamiento
    - Representaciones especiales
- Resultado: la curva de performance de los AE se deforma
  - Es mejor en algunos problemas de tipo definido
  - Pero es peor en problemas de otro tipo
  - Cantidad de conocimiento agregado es variable
- Teoría reciente (No Free Lunch) sugiere que la búsqueda de un algoritmo general para el conjunto de todos los problemas es infructuosa

Algoritmos evolutivos

Resolución de problemas (Michalewicz, 2002)



Diversos AE proporcionarán diversos patrones de búsqueda para diferentes problemas