

Introduccion a los Algoritmos Evolutivos

Contenido

- Recapitulacion de las metaforas evolutivas
- Esquema basico para un AE
- Componentes basicos
 - Representacion/Evaluacion/Poblacion
 - Seleccion de ancestros / Recombinacion / Mutacion
 - Seleccion de sobrevivientes/ Terminacion
- Ejemplos: ocho reinas , problema de la mochila
- Comportamiento tipico de un AE
- AE en el contexto de optimisacion global

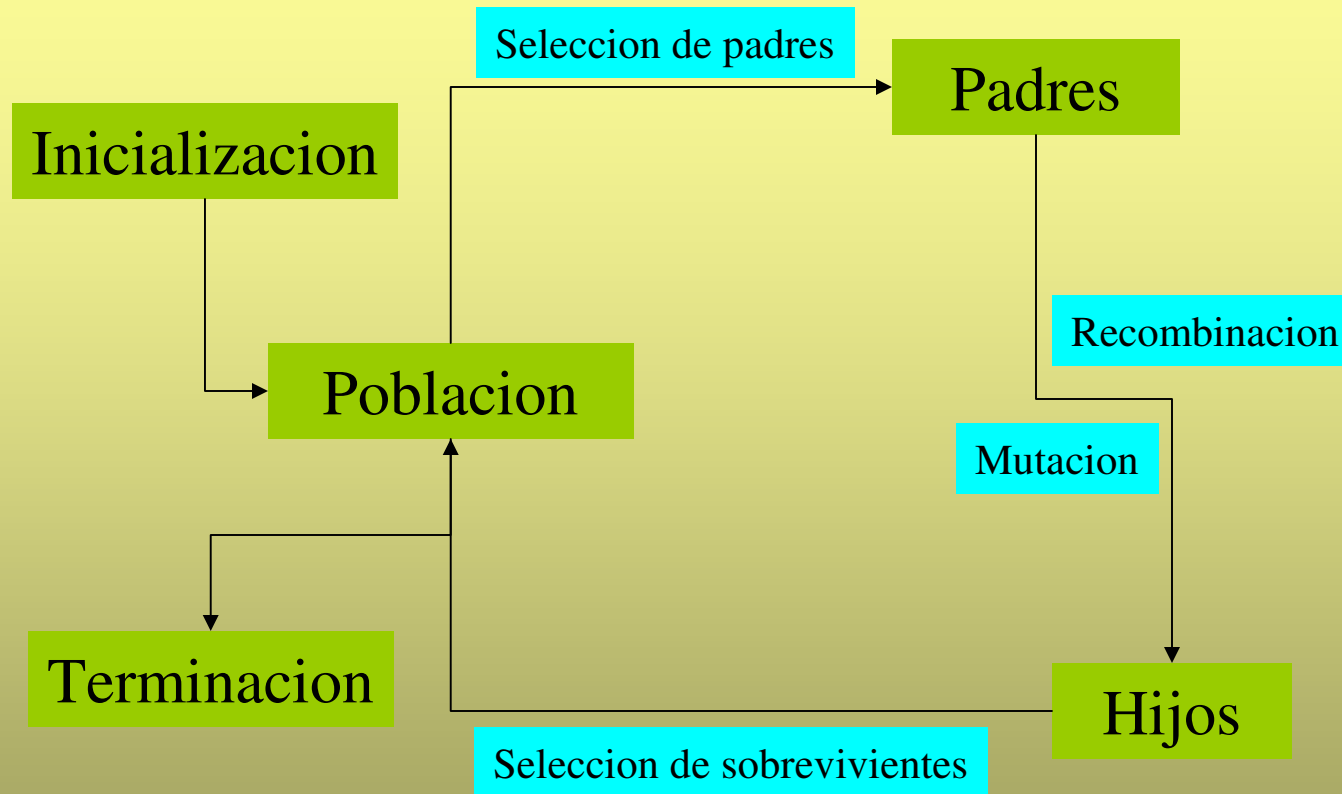
Recapitulacion de la metafora (1)

- Una poblacion de individuos existe en un ambiente con limitados recursos.
- La *competicion* por esos recursos causa la *seleccion* de los individuos mas aptos.
- Estos individuos actuan como semillas para las sucesivas generaciones por medio del cruzamiento y las *mutaciones*.
- Los nuevos individuos son evaluados y compiten entre si (y probablemente con sus padres) por la supervivencia.
- Con el tiempo la seleccion natural causa que la poblacion se adapte mejor a su entorno (mejor fitness)

Recapitulacion de la metafora (2)

- AE entran en la categoria de “generate and test”
- Son estocasticos y basados en poblaciones
- Operadores de variacion (recombinacion y mutacion) crean la diversidad necesaria para facilitar la exploracion del espacio de individuos factibles.
- La seleccion reduce la diversidad y “enfoca” la poblacion en aquellos individuos que son mas aptos

Esquema General de un AE



Pseudocodigo para un AE Tipico

Begin

Inicializar al azar poblacion con soluciones candidatas

Evaluar cada solucion

Repetir hasta (condicion finalization == verdadero) Do

1 Seleccionar padres

2 Recombinar pares de padres

3 Mutar los hijos resultantes

4 Evaluar los nuevos individuos

5 Seleccionar individuos para la generacion proxima

Od

End

Tipos de AE

- Historicamente se han asociado “dialectos” distintos de AE de acuerdo a la representacion usada:
 - Cadenas binarias : Algoritmos Geneticos
 - Vectores de valores reales : Estrategias Evolutivas
 - Automatas de Estados Finitos: Programacion Evolutiva
 - Arboles Lisp: Programacion Genetica
- Estas distinciones son en gran medida irrelevantes, la mejor estrategia es:
 - elegir una representacion que se adecue al problema
 - elegir los operadores de variacion que se adecuen a la representacion
- Los operadores de seleccion solo usan el fitness por lo tanto son independientes de la representacion

Representaciones

- Soluciones candidatas (individuos) existen en el espacio de fenotipos
- Estas están codificadas en “cromosomas”, que existen en el espacio de genotipos
 - Codificación : fenotipo \Rightarrow genotipo (no necesariamente uno a uno)
 - Decodificación : genotipo \Rightarrow fenotipo (debe ser uno a uno)
- Los cromosomas contienen **genes**, que están en (usualmente) posiciones fijas llamadas **loci** (sing. locus) y adoptan un valor (**alelo**)

Para poder encontrar el óptimo este debe estar incluido en el espacio de representaciones posibles

Funcion de Evaluacion (Fitness)

- Representa los requisitos a los cuales la poblacion debe adaptarse
- tambien se llama funcion de evaluacion, funcion de fitness, funcion objetivo
- Asigna un valor singular real a cada fenotipo y forma la base para la seleccion
 - Cuanto mejor discrimine distintos fenotipos mejor
- Generalmente se habla de “la maximizacion del fitness” pero algunos problemas son mas bonitos minimizando (la conversion es trivial)

Poblacion

- Contiene representaciones de posibles soluciones
- Generalmente es de tamaño fijo y es un multi set
- Algunos EA usan una estructura espacial.
- Los operadores de seleccion generalmente consideran a toda la poblacion, esto es, las chances reproductivas son relativas a la poblacion actual.
- **Diversidad** se refiere al numero de genotipos, fenotipos o fitnesses distintos en la poblacion

Mecanismo de Selección de Padres

- La probabilidad de dejar prole depende del fitness del individuo
- Generalmente la selección es probabilística
 - Individuos con buen fitness son más propensos a ser padres
 - Pero no está garantizado
 - Aun el peor individuo puede dejar algún hijo
- Esta estocasticidad permite escaparse de óptimos locales

Operadores de Variacion

- El rol es el de generar nuevas soluciones
- Generalmente se dividen en dos tipos
 - **Mutacion**: generalmente de aridad 1 y “ciegos”
 - Recombinacion: generalmente de aridad >1
 - Se llaman **crossover**
- Mucho debate sobre la importancia de ambos
 - Casi la mayoría de los EA usa ambos tipos de operadores
 - Estos operadores dependen intimamente del tipo de representacion

Mutacion

- Generalmente actua en un genotipo y devuelve otro
- Randomisacion es esencial y la distingue de otros operadores unarios (por ejemplo busqueda local)
- La mutacion debiese ser ciega si esperamos tener una probabilidad > 0 de llegar al optimo global
- La relevancia depende de la representacion y del “dialecto”
 - Ags binarios – es un operador de “background” que introduce variacion y preserva diversidad
 - PE para ASF – unico operador
 - PG – raramente usado
- Puede garantizar la conectividad del espacio de busqueda y permitir las pruebas de convergencia

Recombinacion

- Combina informacion de los dos padres
- La seleccion de “que” combinar es estocastica
- La mayoria de los hijos son peores o en el mejor de los casos iguales a los padres
- Algunos podrian ser mejores y sus “traits” explotados por el AE
- Este principio ha sido utilizado desde el principio de la civilizacion (agricultura/ganaderia) para mejorar las cosechas y el ganado

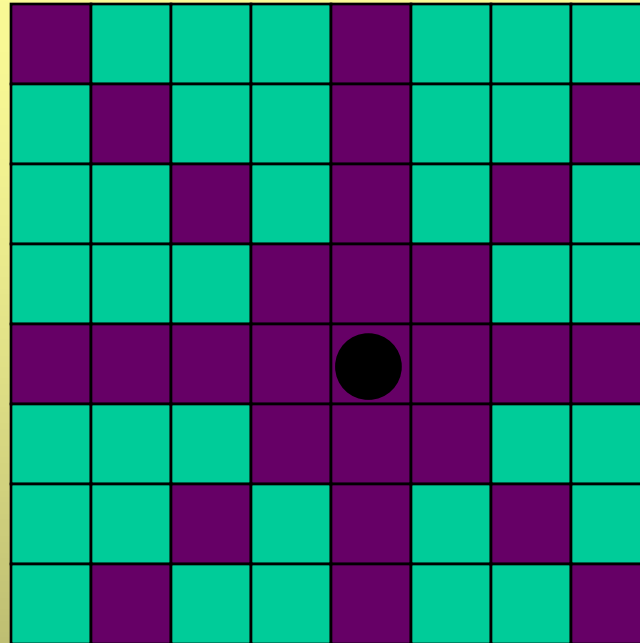
Selección de Sobrevivientes

- También conocido como reemplazo
- La mayoría de los AE usan una población de tamaño fijo de tal manera que se necesita una manera de pasar de (Padres+Hijos) a (nuevos padres)
- Generalmente determinístico
 - Basado en el fitness : se rankean los padres y los hijos y se seleccionan los mejores
 - Basado en la edad: se crean tantos hijos como padres haya y se elimina los padres
- Algunas veces se usa una combinación (elitismo)

Inicialisation / Terminacion

- Generalmente se inicializa al azar,
 - Se debe asegurar una buena mezcla y provision de valores para los alelos.
 - Se puede “sembrar” la poblacion con soluciones pre-existentes, calculadas heuristicamente.
- Condiciones de Terminacion
 - Se alcanzo algun fitness conocido o esperado
 - Se alcanzo el numero maximo de generaciones/evaluaciones de la funcion de fitness
 - Se alcanzo un minimo de diversidad
 - No se mejoro el fitness de la poblacion en las ultimas X generaciones

Ejemplo: Las 8 reinas

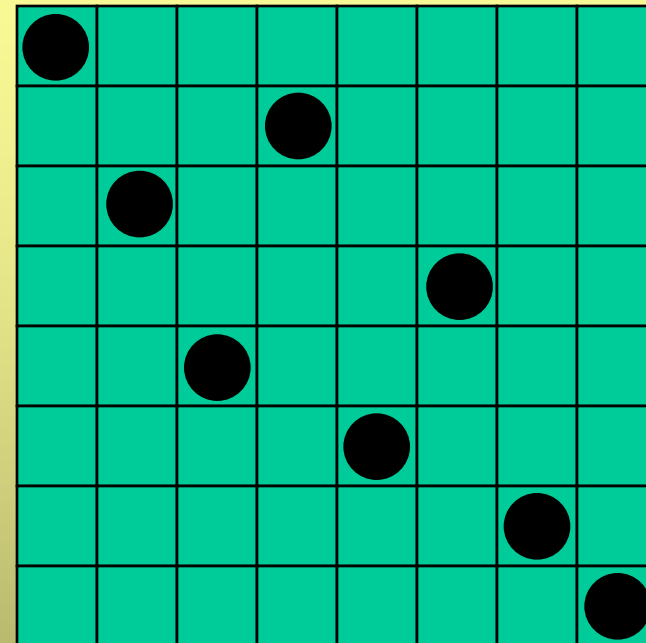


Ubicar 8 reinas en el tablero de tal manera que no se “ataquen”

Representacion

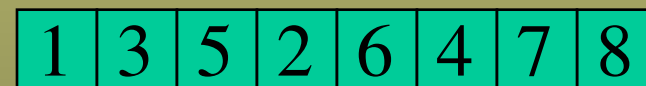
Fenotipo:

Una configuracion del tablero



Genotipo:

Una permutacion de los numeros 1-8



Mapeo obvio

Evaluacion de fitness

- Penalidad de una reina:
el numero de reinas que ataca
- Penalidad de la configuracion:
suma de las penalidades de todas las reinas
- debemos minimizar las penalidades
- Fitness de una configuracion:
inverso de la penalidad que debe ser maximisado

Mutacion

Pequeña variación en una configuración

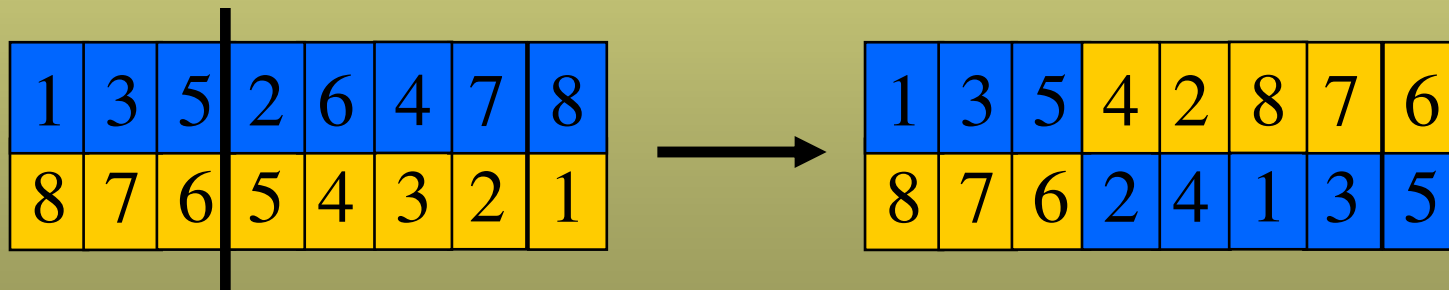
- intercambiando valores en dos posiciones random



Recombinacion

Combinar dos permutaciones en dos nuevas permutaciones:

- elegir al azar el punto de crossover
- copiar la primera parte al hijo
- crear la segunda parte insertando valores del otro padre:
 - en el orden que aparecen
 - empezando despues del punto de crossover
 - salteando los valores ya presentes

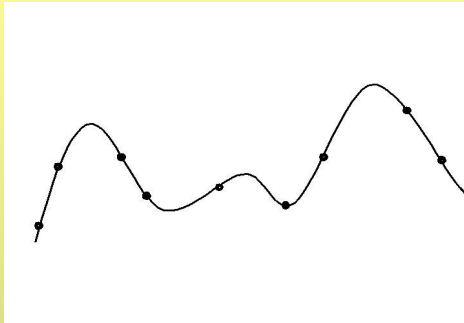


Selección

- Selección de padres:
 - Elegir 5 padres y de ellos seleccionar los dos mejores para reproducción
- Selección de sobrevivientes (reemplazamiento)
 - Al insertar un nuevo hijo en la población seleccionar a quien eliminar por medio de:
 - Ordenar la población por fitness decreciente
 - Iterando en la lista reemplazar el primero que tenga un fitness menor que el del nuevo hijo

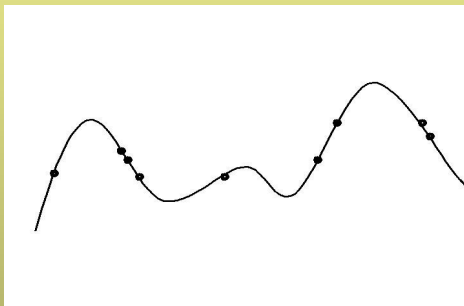
Comportamiento tipico de un AE

Fases de la optimisacion de un problema 1-dimensional



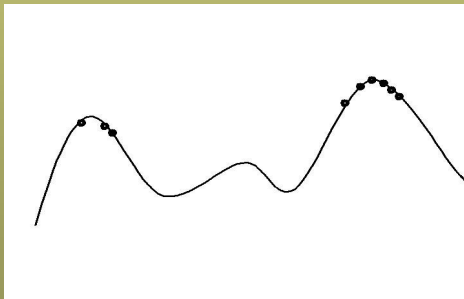
Fase temprana:

Distribucion quasi-randomica de la poblacion



Fase intermedia:

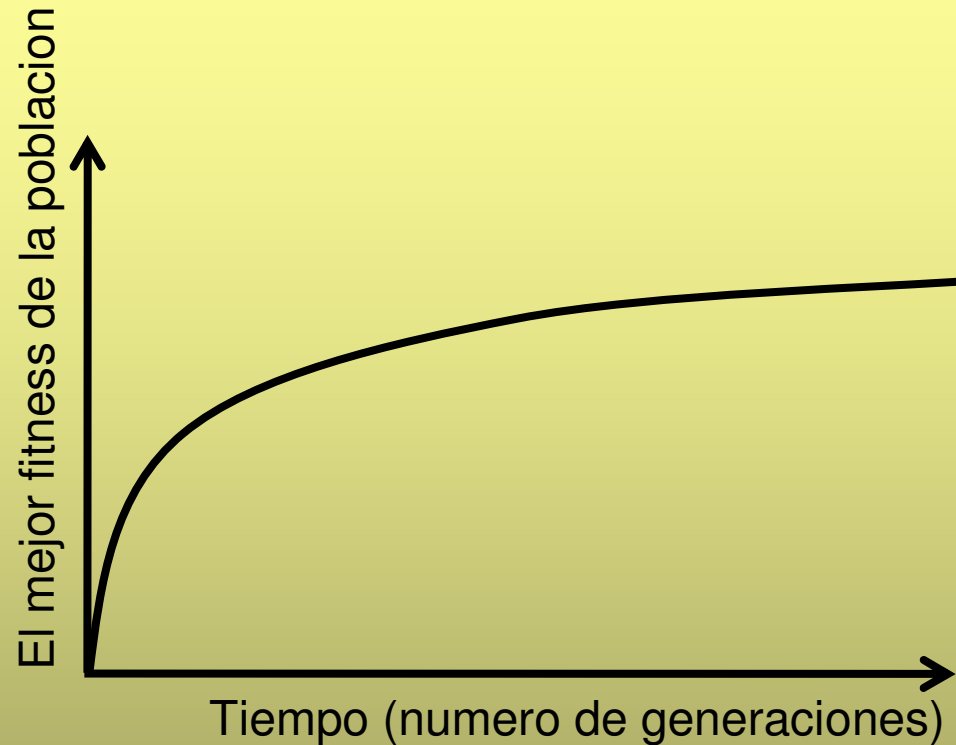
La poblacion empieza a concentrarse sobre /
alrededor de las colinas



Fase final:

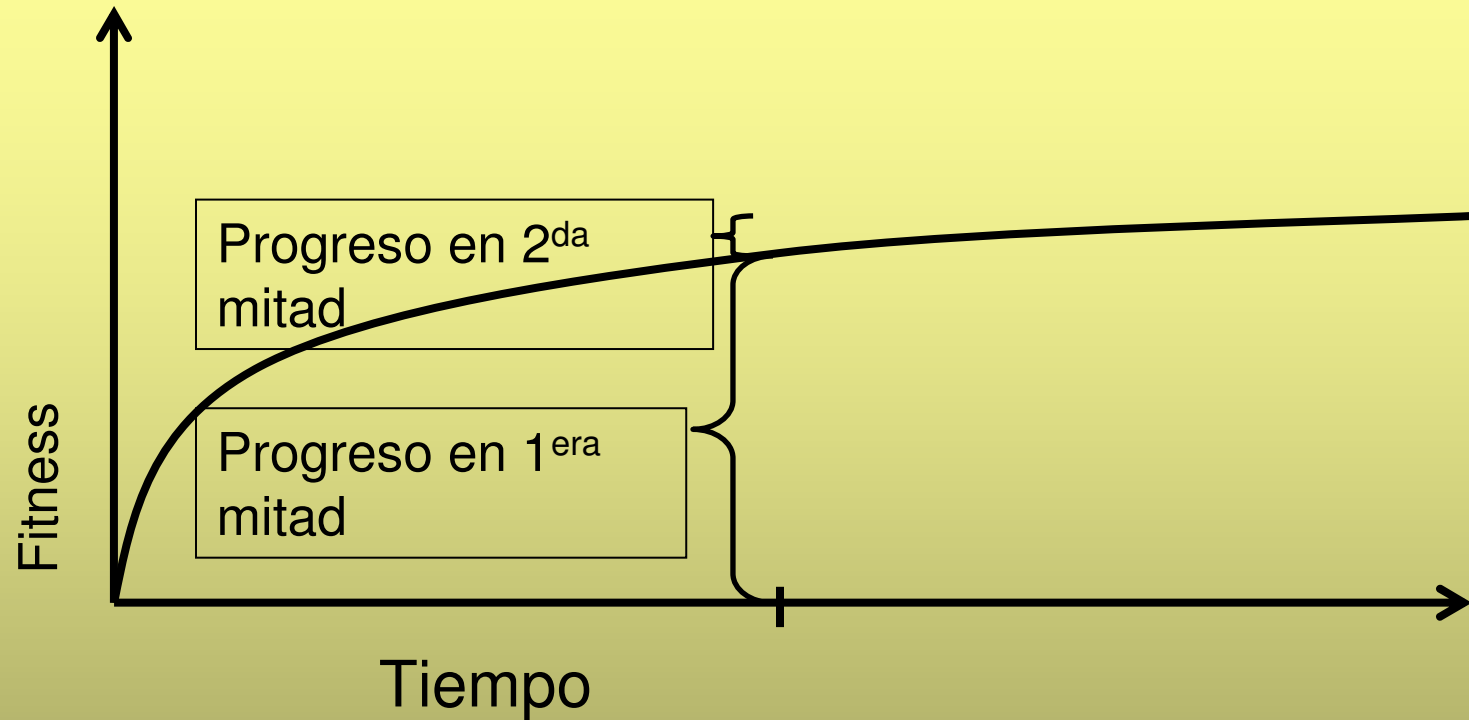
La poblacion se concentra en los optimos locales

Ejecucion tipica: Fitness VS Tiempo



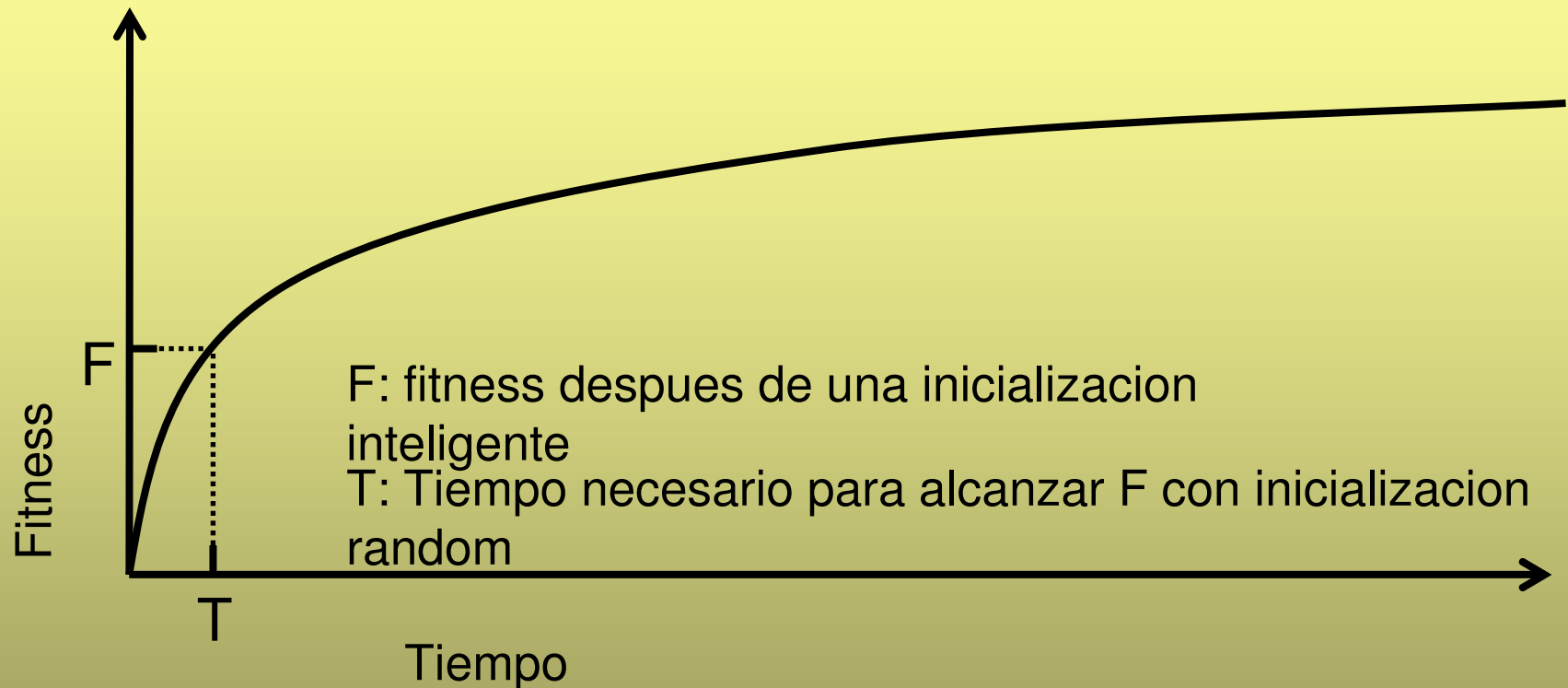
Ejecucion tipica de un AE

Son beneficiosas las ejecuciones largas?



- Respuesta:
 - depende de cuanto valor uno le asigna a los ultimos progresos
 - puede ser mas beneficioso hacer muchos ejecuciones cortas

Vale la pena invertir esfuerzo en inicializaciones inteligentes?



- Respuesta : depende de:
 - posiblemente si es que existen buenos metodos
 - hay que tener cuidado (discutiremos esto luego)

Algoritmos Evolutivos en Contexto

- Hay muchas opiniones sobre el uso de algoritmos evolutivos en optimisacion
- Para la mayoría de los problemas un algoritmo específico puede:
 - Funcionar mejor que cualquier algoritmo generico en la mayoría de las instancias,
 - Pero tener limitada utilidad en otro dominio,
 - No funcionar tan bien en algunas instancias
- El objetivo es proveer herramientas robustas que:
 - Funcionan mas o menos bien en mas o menos todos los casos
 - Sobre una variedad de problemas e instancias

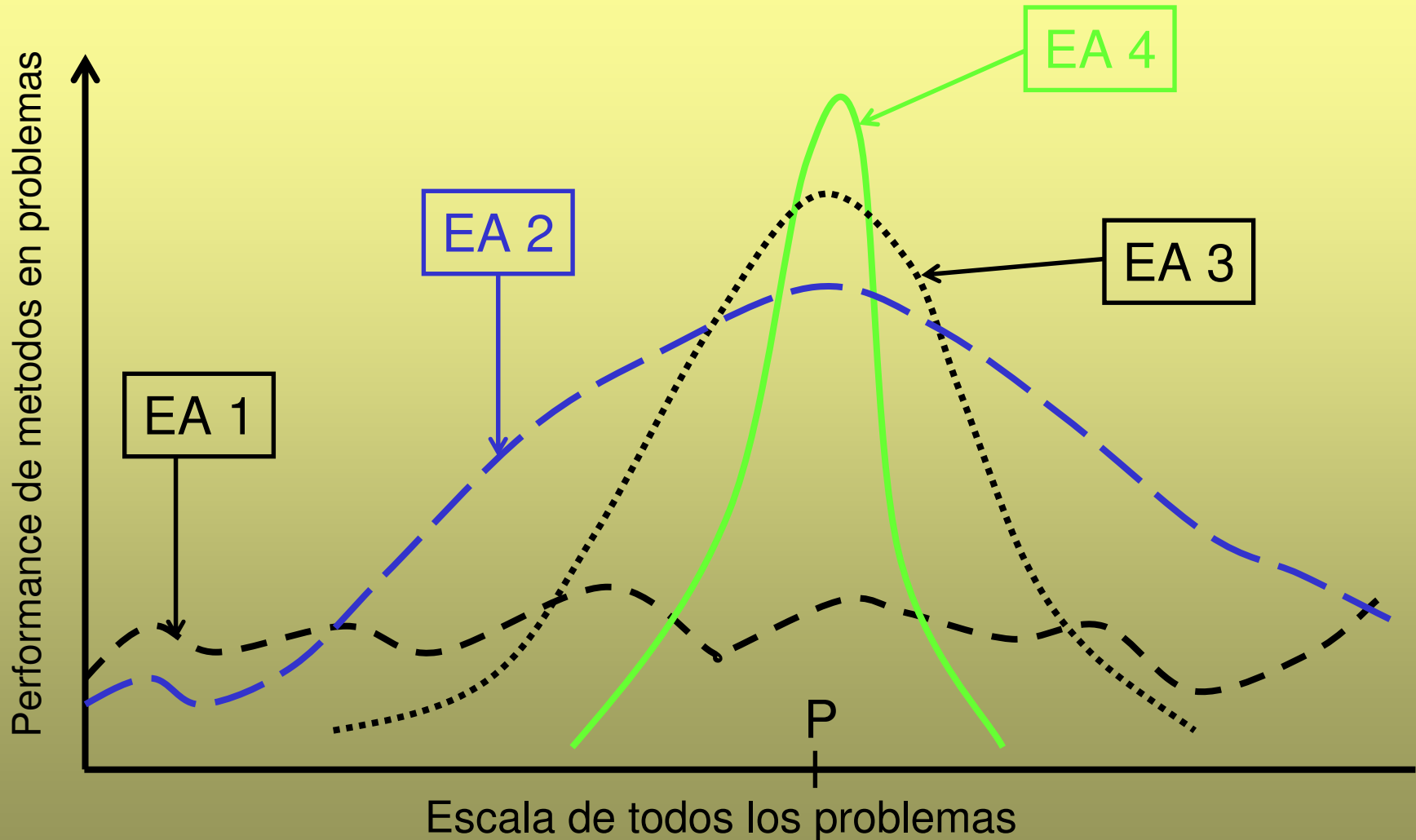
EAs as problem solvers: Goldberg's 1989 view



AE y conocimiento del dominio

- Moda de los 90's:
agregar conocimiento del problema a los AE
(operadores de mutacion, crossover, reparacion especiales)
- Resultdo: la curva de performance de los AE se deforma:
 - Mejor en algunos problemas de un tipo definido
 - Peor en problemas de otro tipo
 - Cantidad de conocimiento agregado es variable
- Teoria reciente sugiere que la busqueda de un algoritmo general para todos los problemas puede ser infructuosa.

La interpretacion de Michalewicz



AE y Optimisacion global

- Optimisacion global: buscar la mejor solucion x^* de algun conjunto S
- Metodos deterministicos
 - “box decomposition” (branch and bound, etc)
 - Garantizan el hallazgo de x^* , pero pueden resultar en tiempos superpolinomiales o exponenciales
- Metodos heuristicos (generar y probar)
 - Reglas para decidir cual $x \in S$ generar
 - No hay garantias de optimalidad global

AE y Búsqueda de vecindarios

- Muchas heurísticas imponen un vecindario a S
- A veces se puede garantizar *optimalidad local* e.g. Hill-Climbers:
 - **Sin embargo** todos los problemas interesantes tienen muchos óptimos locales
 - Generalmente se encuentran soluciones razonables rápido
 - Pero existen casos en que incluso encontrar un óptimo local puede llevar tiempo exponencial
- Los AE se distinguen por:
 - Uso de la población,
 - Uso de varios operadores (estocásticos)
 - Operadores de variación con aridad mayor a 1
 - Selección estocástica