UNIVERSITAT POLITÈCNICA DE VALÈNCIA

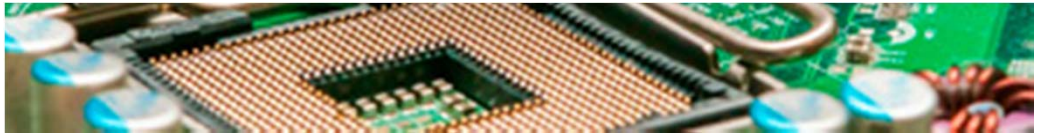
etsinf

Tema 3: Optimización inteligente

Miguel A. Salido

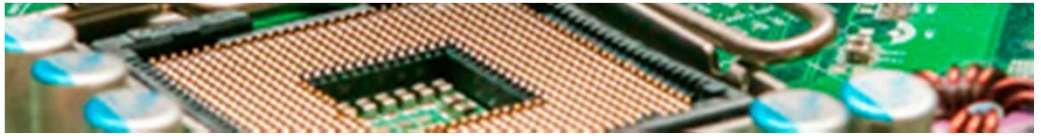
Máster Oficial Universitario en
Ingeniería Informática
muiinf.webs.upv.es

MUIInf



Contenido

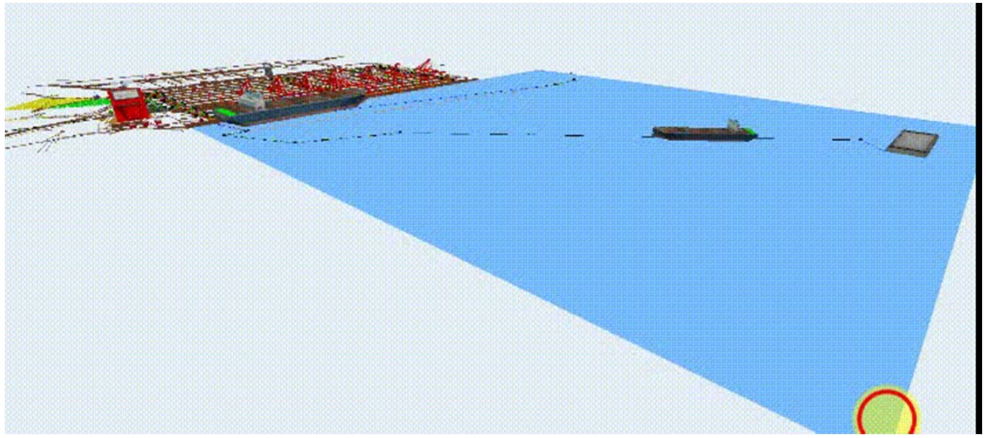
1. Introducción
2. Scheduling Inteligente
3. Problema de Satisfacción de Restricciones
4. Técnicas de Optimización Inteligente: Los Algoritmos Genéticos



1. Introducción

- Planificación y Scheduling representa un área de gran relevancia en Inteligencia Artificial.

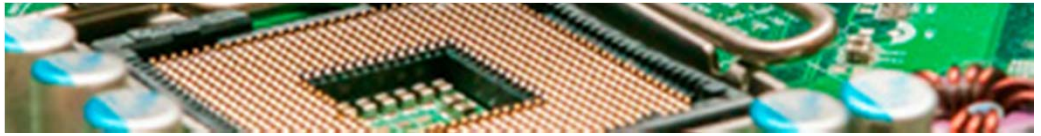
- Muchos problemas reales se modelan como problemas de P&S:



1. Introducción

- Planificación: es un proceso de deliberación que escoge y organiza acciones anticipando sus resultados o consecuencias.

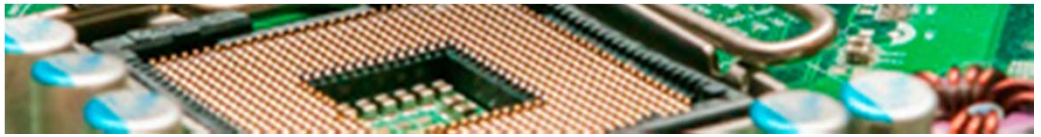
- "Planificación es una tarea de encontrar una secuencia de acciones posible para pasar de un estado inicial a un estado objetivo" (Tema 2)



1. Introducción

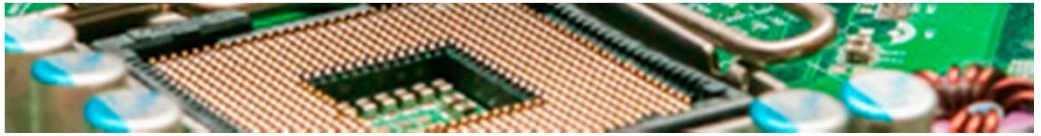
- Sheduling: es un proceso de asignación de recursos a tareas sobre el tiempo para optimizar uno o más objetivos.

PLANNING	SCHEDULING
La tarea de planificación determina **QUÉ** acciones son necesarias llevar a cabo para alcanzar un estado objetivo.	La tarea de scheduling se centra en **CUANDO/COMO** llevar a cabo las acciones para optimizar el problema

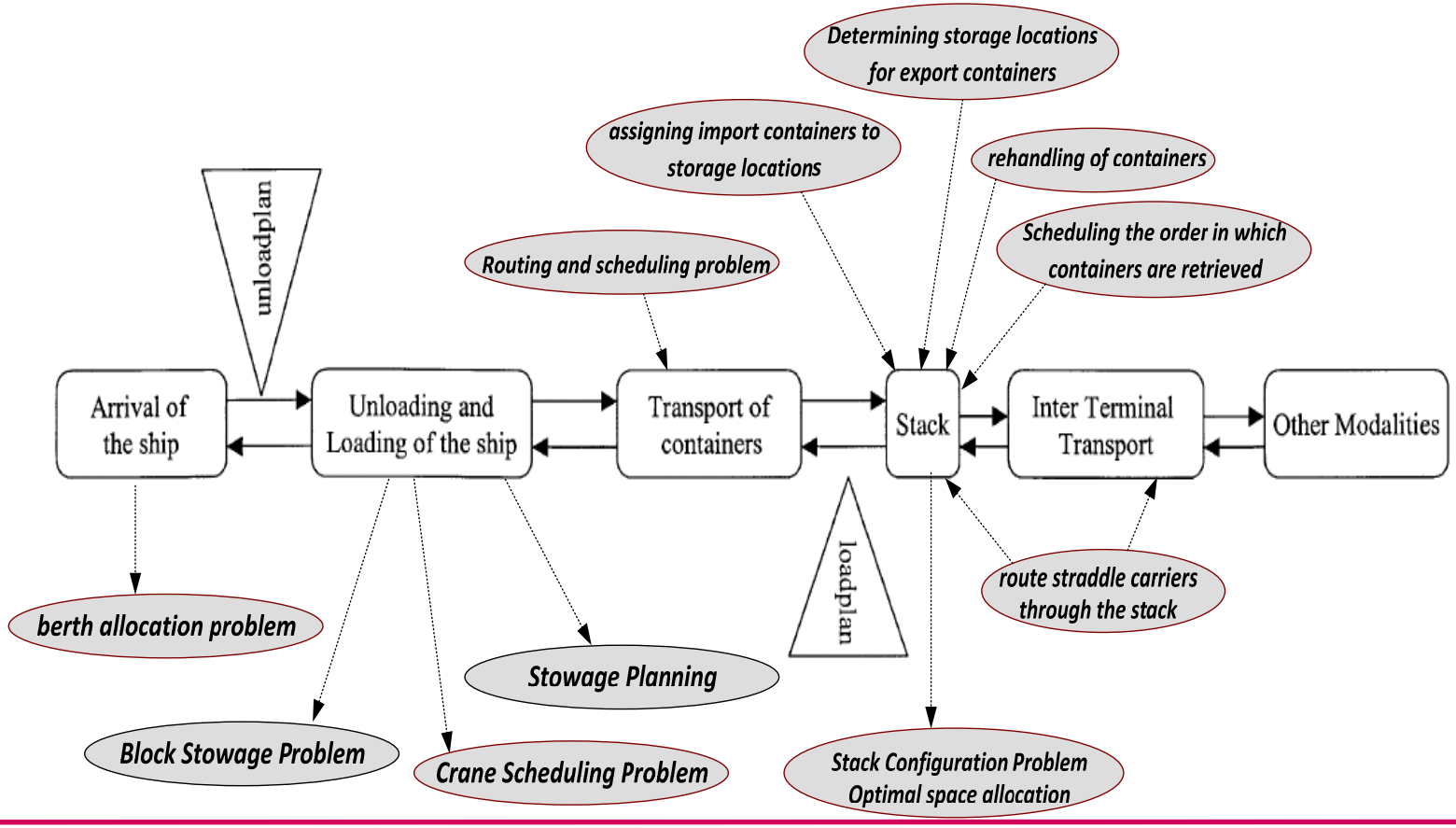


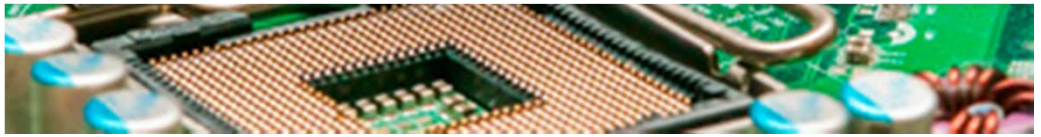
Scheduling

- ¿Qué es un problema de scheduling?
  - Un problema de scheduling consiste en organizar en el tiempo un conjunto de actividades que compiten por el uso limitado de recursos
  - Normalmente hay restricciones temporales, espaciales y otras
  - Y se trata de optimizar una o varias medidas de calidad
- ¿Dónde hay problemas de scheduling?
  - En todas partes: talleres, centros docentes, centros sanitarios, plantas de producción, cadenas de montaje, transporte de mercancías y viajeros, gestión de almacenes, logística, etc
- ¿Cómo se resuelven los problemas de scheduling?
  - En general son problemas NP-hard
  - Luego las instancias pequeñas se pueden resolver de forma exacta, pero las grandes no, en general.

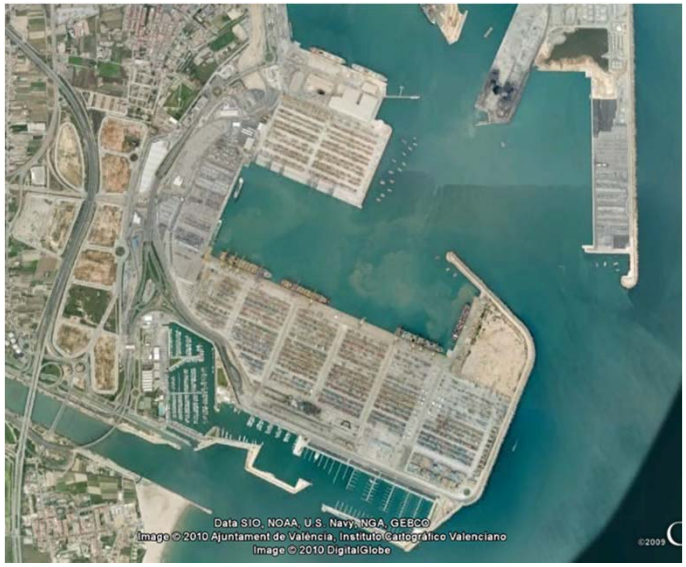


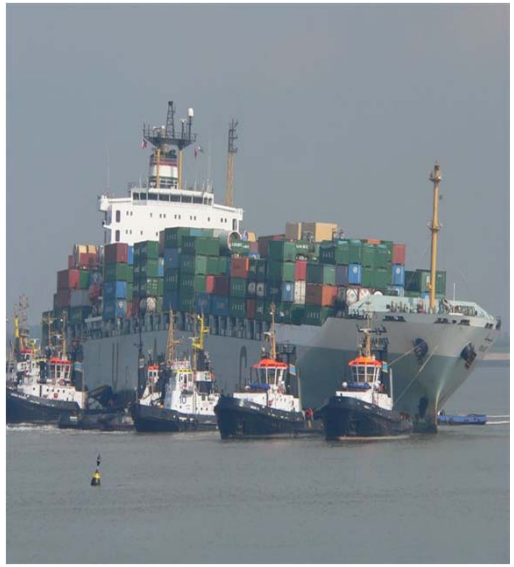
Problemas de Optimización en Terminal de contenedores

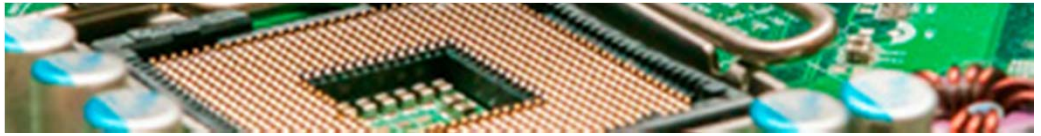




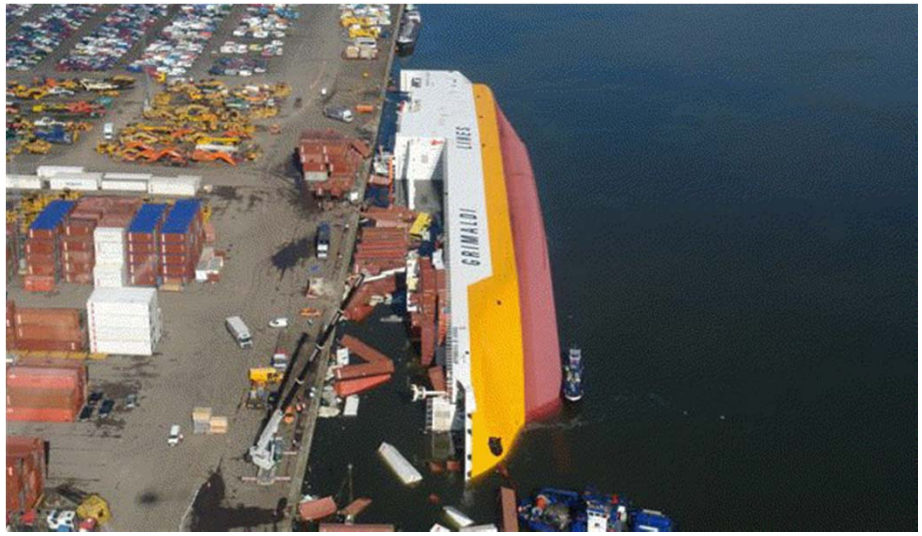
El problema de asignación de busques a puerto





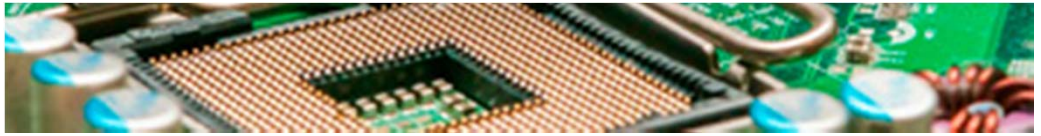


Problemas de Optimización en Terminal de contenedores

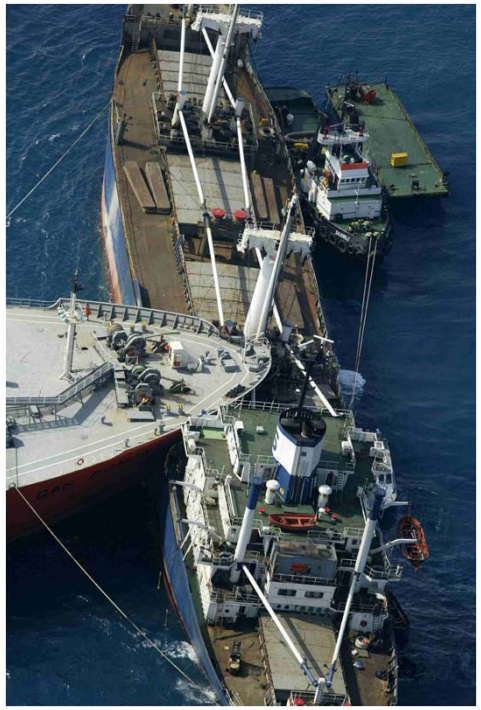


Problema de estabilidad

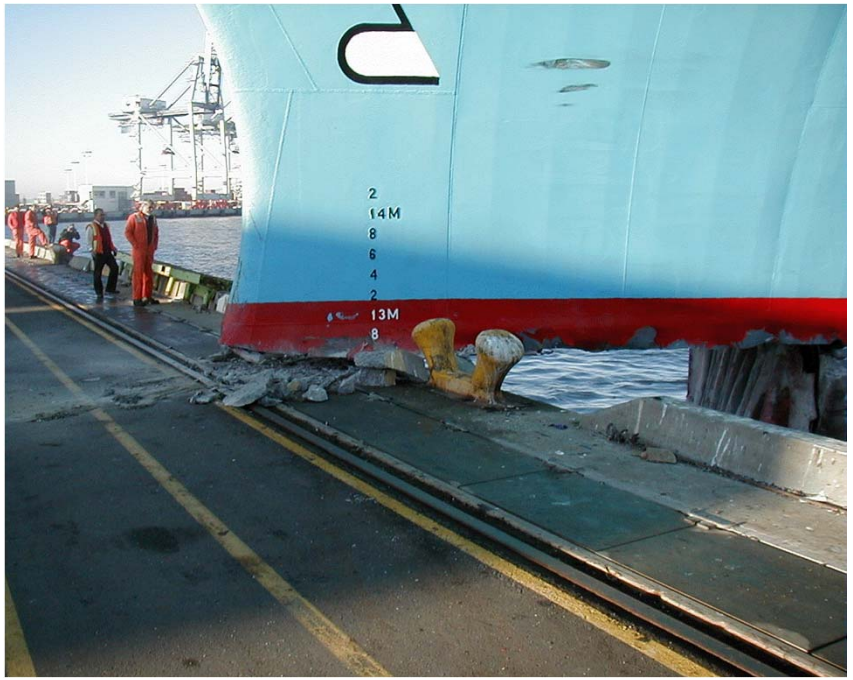
Problema de scheduling de contenedores



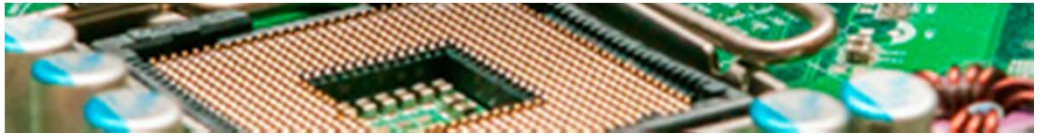
Problemas de Optimización en Terminal de contenedores



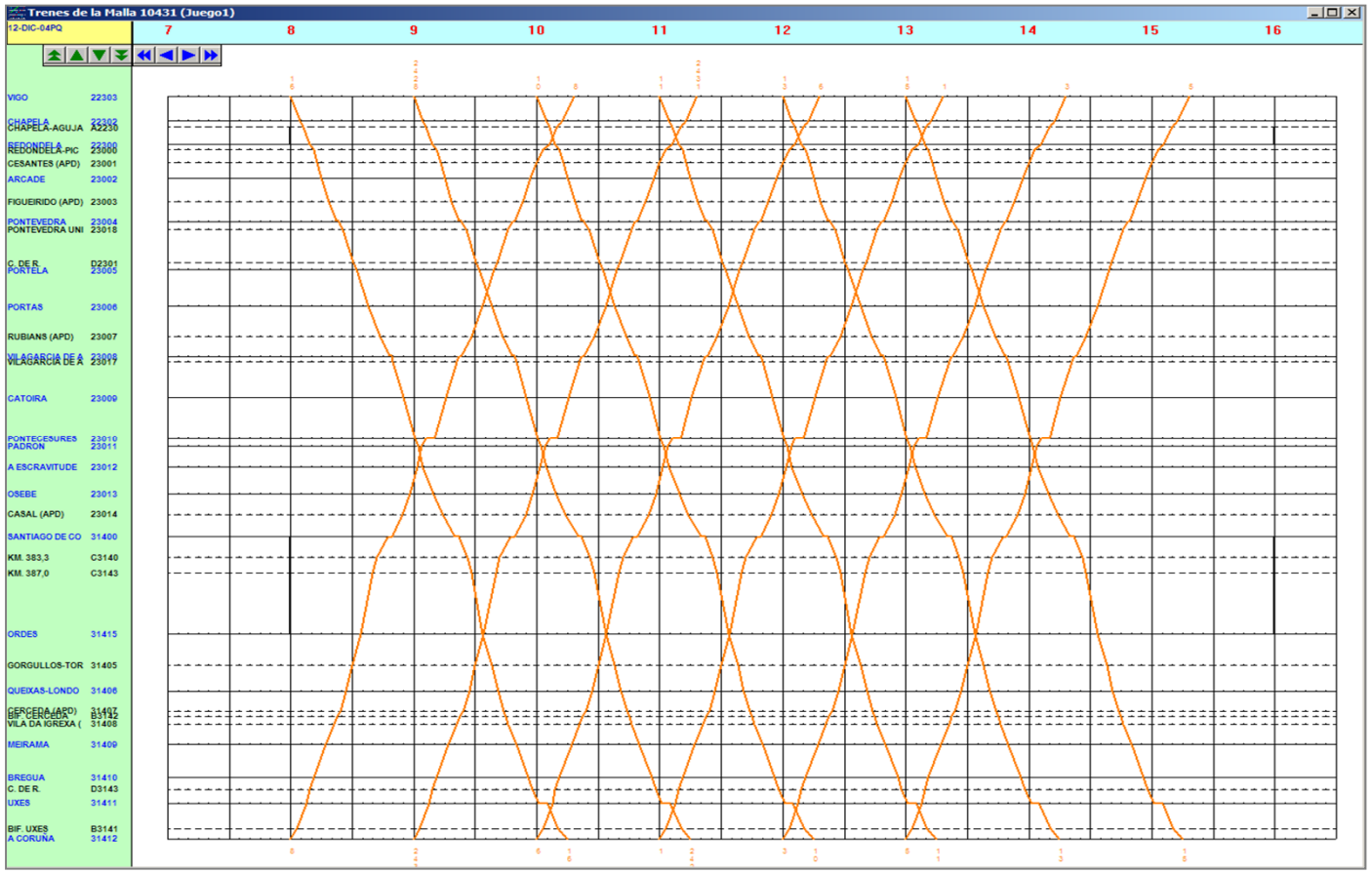
Problema de rutas

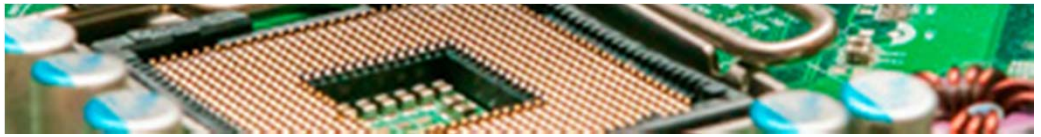


Problema de atraque



El problema de asignación de rutas ferroviarias

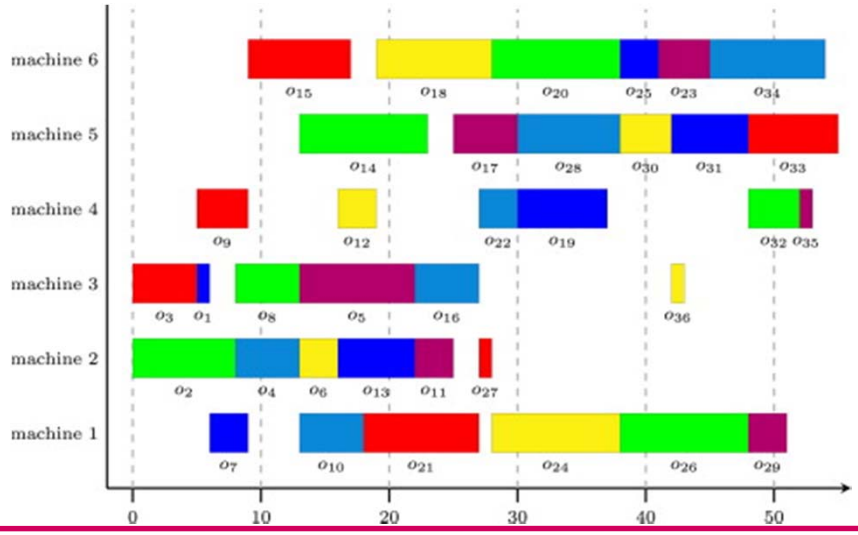


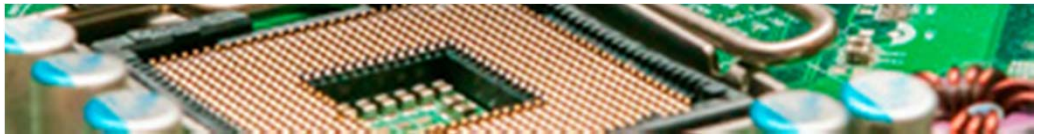


Scheduling

El Job-Shop Scheduling Problem (JSP)

- Es un paradigma de los problemas de scheduling: es muy simple de enunciar y muy difícil de resolver
- Es un “banco de pruebas” para todas las técnicas exactas y aproximadas de optimización combinatoria
- Está muy estudiado en la literatura. Casi tanto como el TSP, el problema de la mochila, ...

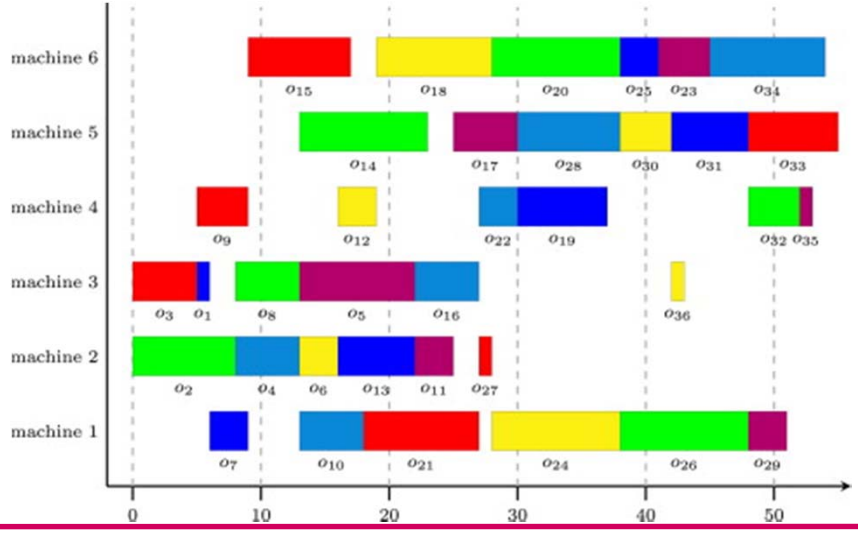


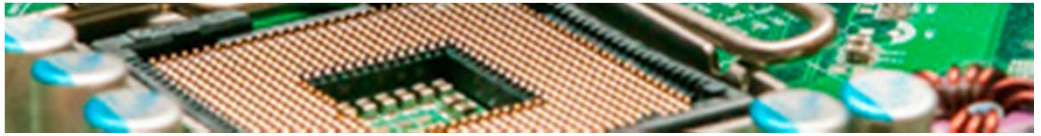


Scheduling

El Job-Shop Scheduling Problem (JSP):

- n trabajos, cada uno compuesto por un cto ordenado de tareas;
- m maquinas donde se procesas las tareas;
- Cada tarea debe ser procesada en una única máquina durante un tiempo determinado y en un orden prefijado;
- El objetivo es minimizar makespan: instante de finalización de la última tarea.





Datos: Job-Shop Scheduling Problem (JSP):

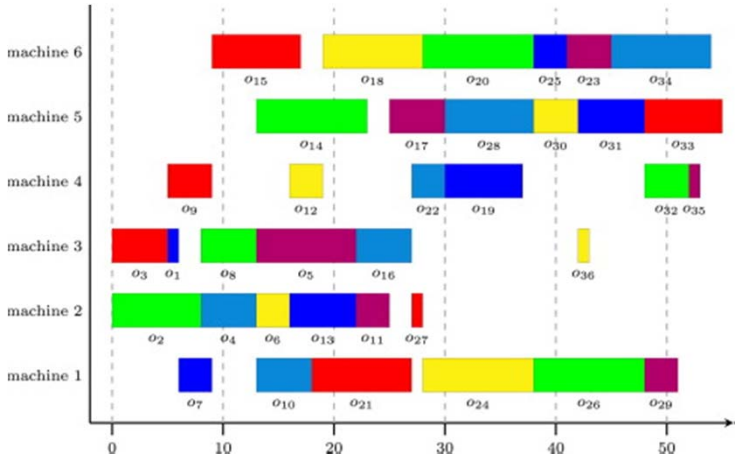
- p_{ij} es el tiempo de proceso de la tarea i en el trabajo j;

Variables:

- st_{ij} es el tiempo de inicio de tarea i en el trabajo j;

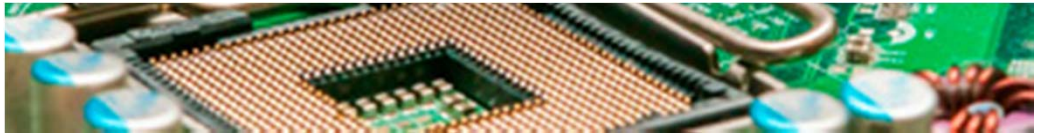
Restricciones

- Secuential: $st_{ij} + p_{ij} \leq st_{i(j+1)}$
- Capacidad: $st_{ij} + p_{ij} \leq st_{kl} \vee st_{kl} + p_{kl} \leq st_{ij}$
- Sin interrupción: $C_{ij} = st_{ij} + p_{ij}$



Objetivo:

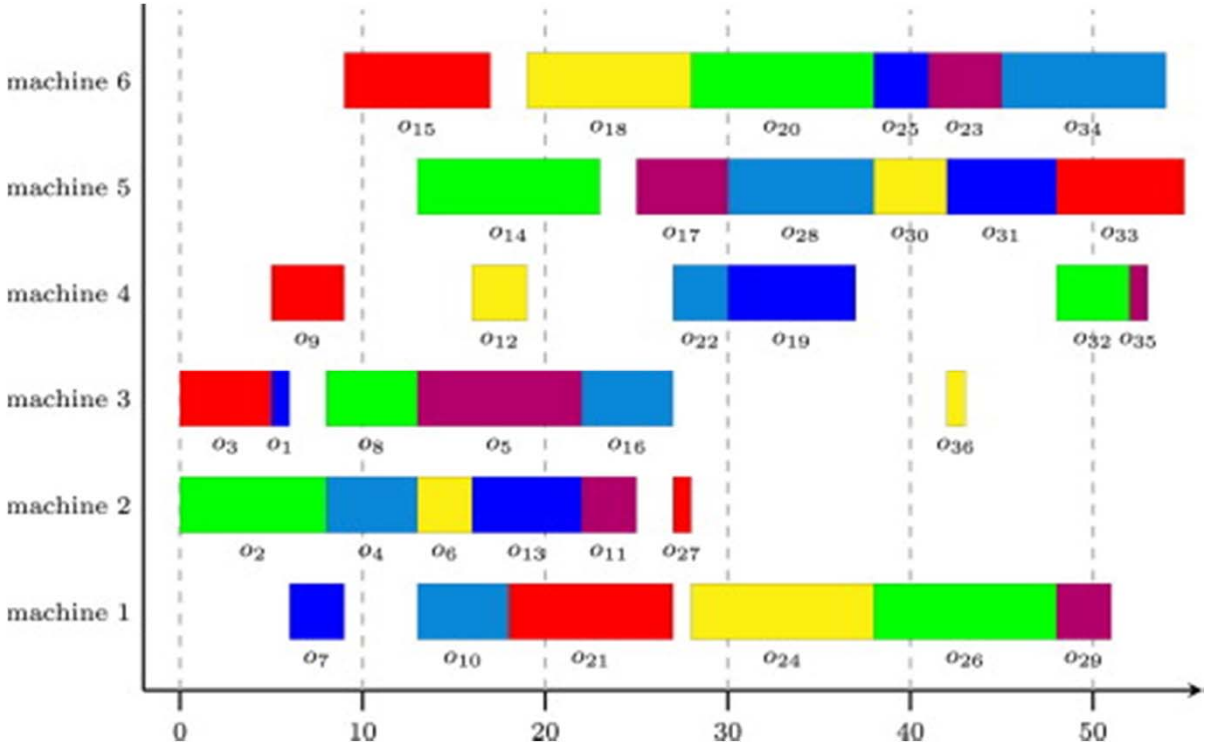
- Construir una ordenación de las tareas en el tiempo de manera que se satisfagan todas las restricciones sobre cada máquina.

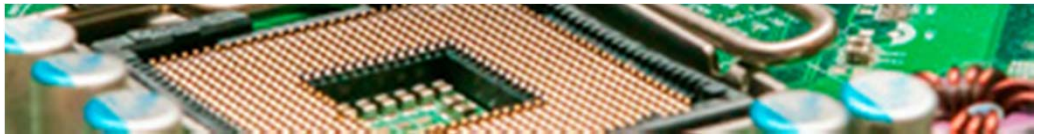


Scheduling

Objetivo:

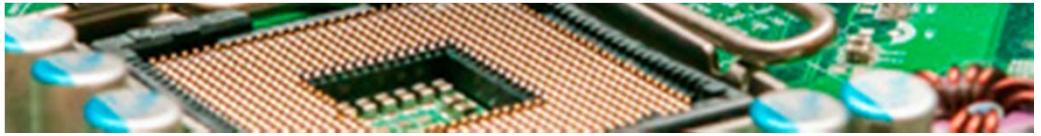
- Mimimize the makespan: $C_{max} = \max C_{ij}$
- Minimize total flow time: $C_{sum} = \Sigma C_i$





Como resolver un Job-Shop Scheduling Problem (JSP)?

- Búsqueda exhaustiva (completa)
- Razonamiento con restricciones
- Metaheuristicas
- Resolutores comerciales
- . . .



Problemas de Satisfacción de Restricciones (CSP)

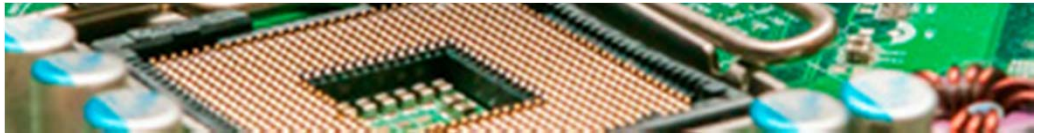
"Constraint Satisfaction, a simple but powerful idea"
Rina Dechter, In 'Constraint Processing' Morgan Kaufmann Pub. (2003)

Muchos problemas pueden ser expresados mediante:

- Un conjunto de variables,
- Un dominio de interpretación (valores) para las variables.
- Un conjunto de restricciones entre las variables.

tal que la solución al problema es una asignación válida de valores a las variables.

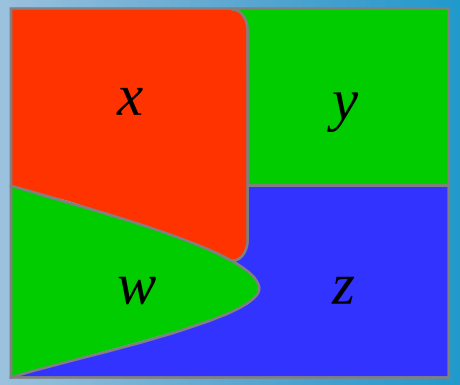
- Problemas de Empaquetamiento, cadenas montaje,
- Problemas de Rutas, transporte, logística,
- Problemas de Scheduling, compartición de recursos,
- Problemas de Razonamiento Temporal,
- Diseño, Planificación, Control, etc.

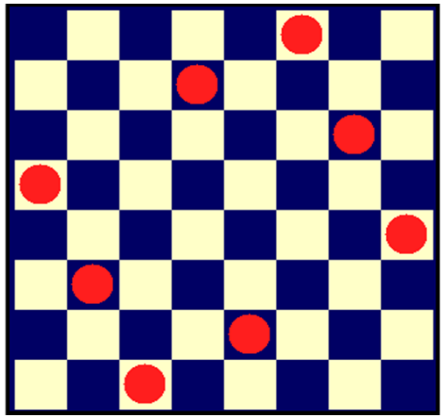


Coloreado de Mapas

- Variables: x,y,z,w
- Dominios: x,y,z,w :{r,v,a}
- Restricciones: *binarias*

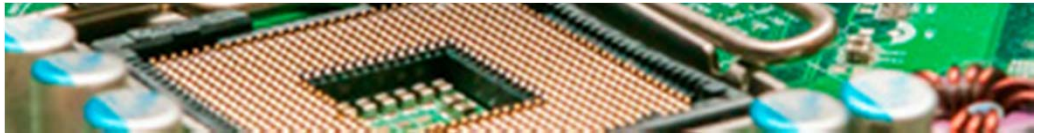
$x \neq y,\ y \neq z,\ z \neq x,\ \ldots$





El Problema de las 8 reinas

Sudoku

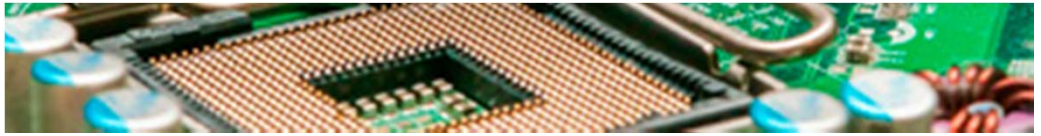


Lectores y Periódicos

Existen 3 periódicos (P1, P2, P3) y 4 lectores (L1, L2, L3, L4), que desean leer los periódicos en el mismo orden.
Todos deben empezar a partir del ready-time y acabar antes del due-time, según la tabla siguiente:

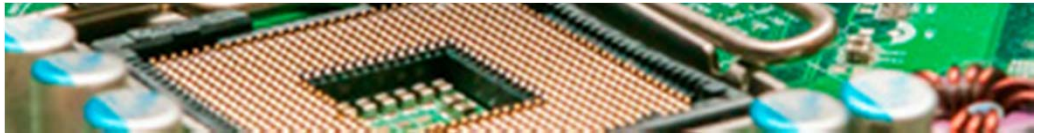
	Ready-Time	P1	P2	P3	Due-Time
L1	0	5'	10'	2'	30'
L2	0	2'	6'	5'	20'
L3	0	10'	15'	15'	60'
L4	0	3'	5'	5'	15'

Objetivo: Obtener la asignación óptima (scheduling) de lectura.



Existen 3 periódicos (P1, P2, P3) y 4 lectores (L1, L2, L3, L4), que desean leer los periódicos en el mismo orden. Todos deben empezar a partir del ready-time y acabar antes del due-time, según la tabla siguiente:

	Ready-Time	P1	P2	P3	Due-Time
L1	0	5' - I11	10' – I12	2' – I13	30'
L2	0	2' – I21	6' - I22	5' – I23	20'
L3	0	10' – I31	15' – I32	15' - I33	60'
L4	0	3' – I41	5' – I42	5' – I43	15'



Existen 3 periódicos (P1, P2, P3) y 4 lectores (L1, L2, L3, L4), que desean leer los periódicos en el mismo orden. Todos deben empezar a partir del ready-time y acabar antes del due-time, según la tabla siguiente:

VARIABLES

\vi : I11a, I11b, I12a, I12b, ... 0..60;

RESTRICCIONES

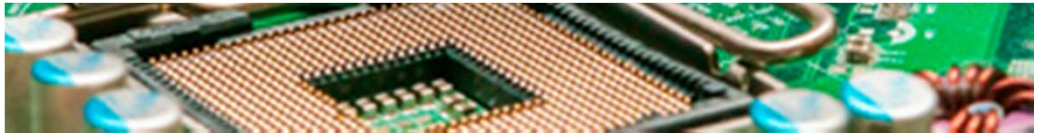
DUE TIME
\ci: duetime13, I13b <=30;
….

DURACIONES
\ci : duracion11, I11b = I11a + 5;
……

Flow-SHOP
\ci : patron1, I11b <= I12a;
….

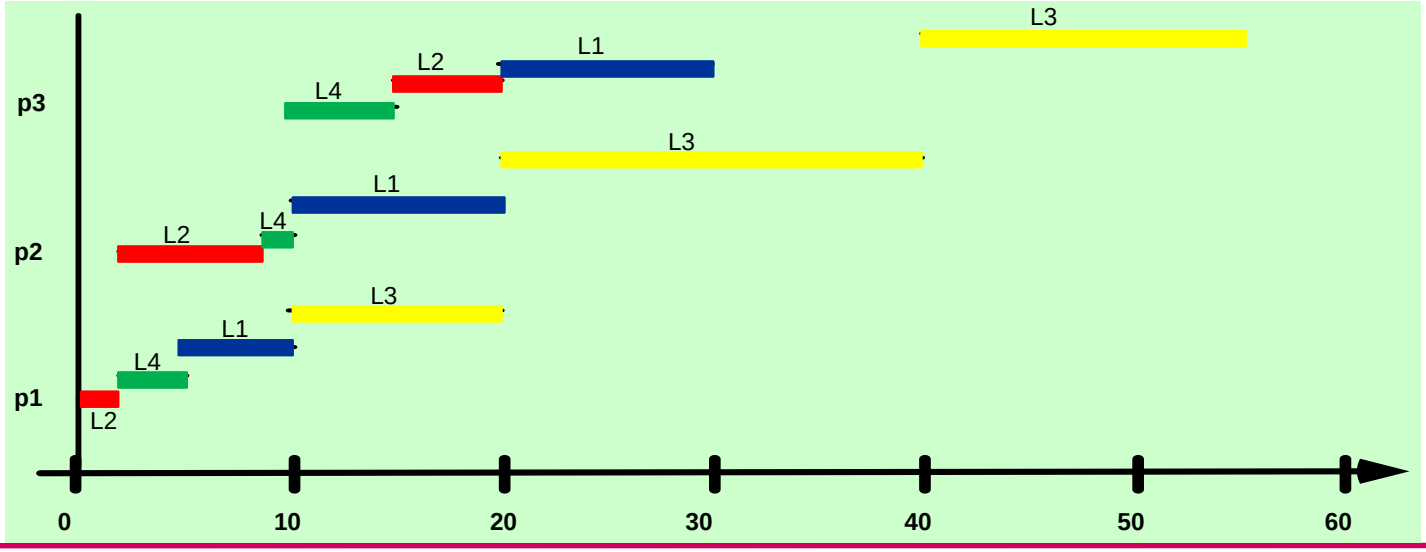
Recursos-No-simultaneos
\doc: nointerseccionI11I21
\ci: C1121a, I11b <= I21a;
  \or
\ci: C1121b, I11b >= I21a + 7;;

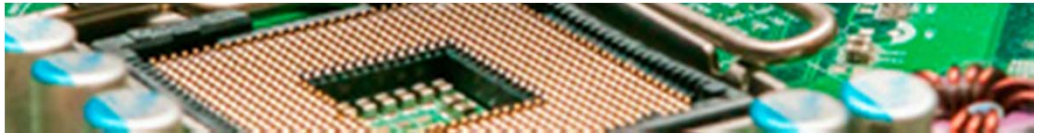
\doc: nointerseccionI11I31
\ci: C1131a, I11b <= I31a;
  \or
\ci: C1131b, I11b >= I31a + 15;;
……..



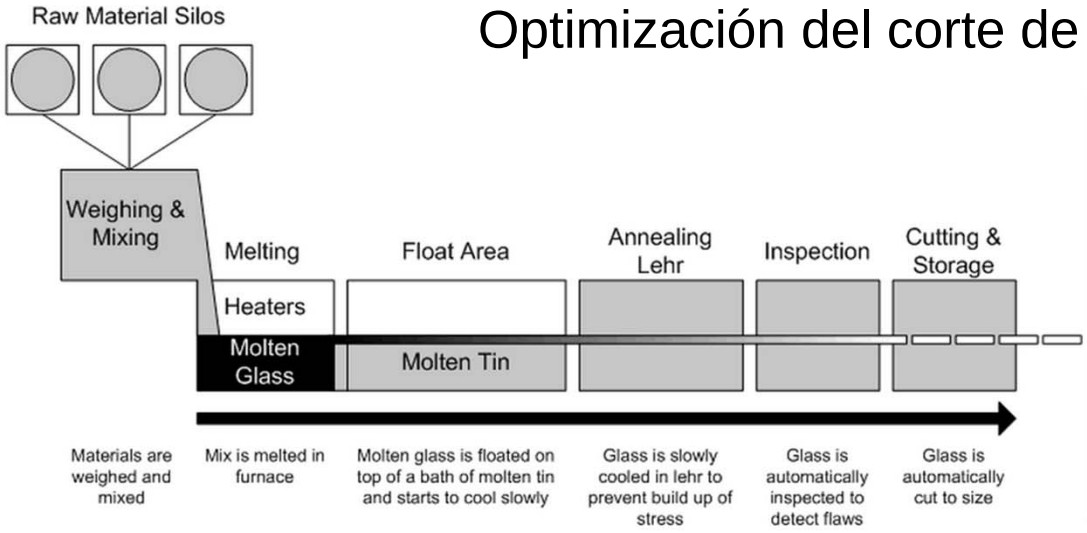
CSPs

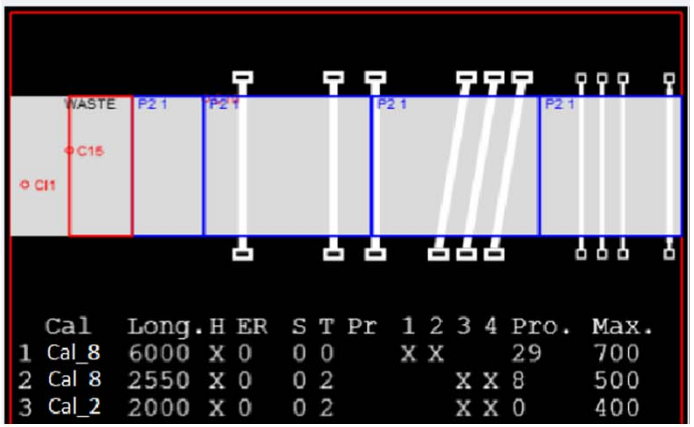
		p1		p2		p3		due-time
		on	off	on	off	on	off	
L1	5	5	10	10	20	20	22	<30
L2	2	0	2	2	8	15	20	<20
L3	10	10	20	20	40	40	55	<60
L4	3	2	5	8	10	10	15	<15

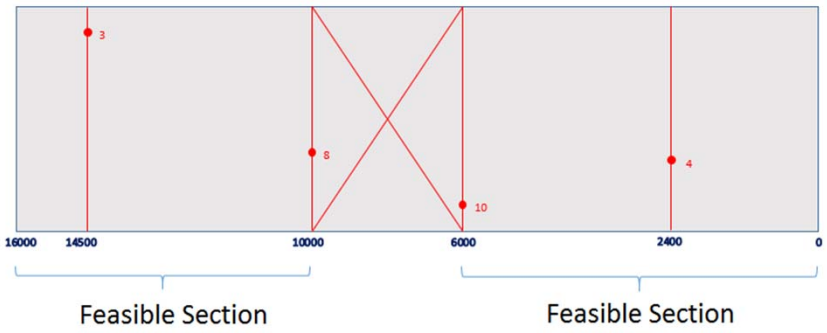


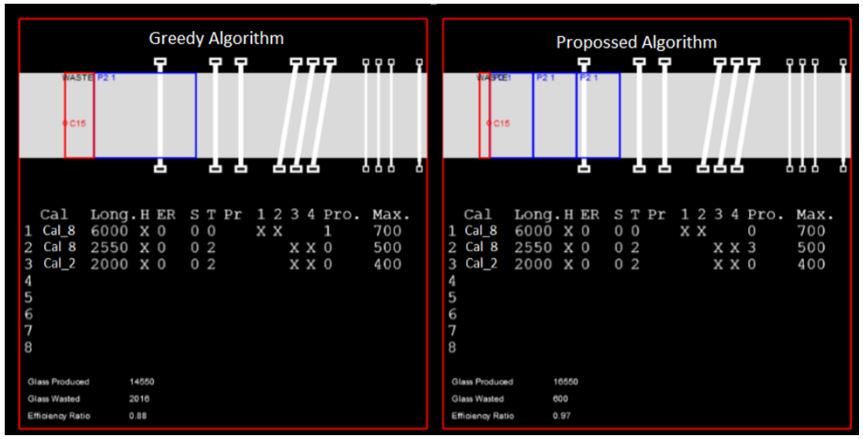


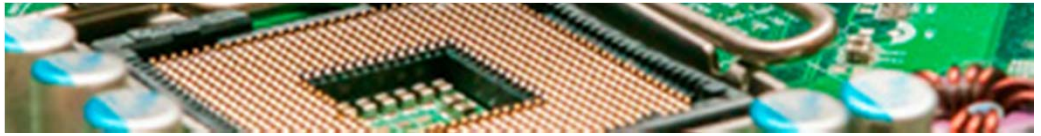
Optimización del corte de cristal continuo

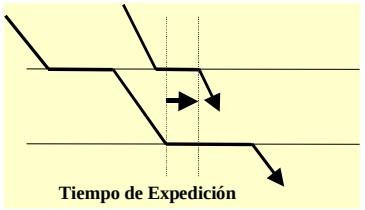


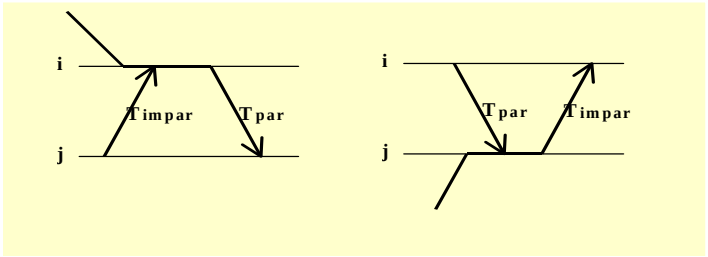


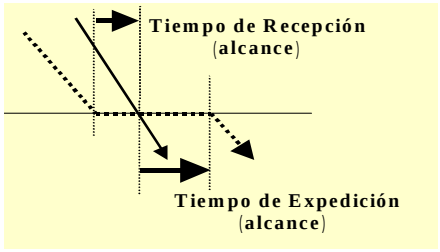


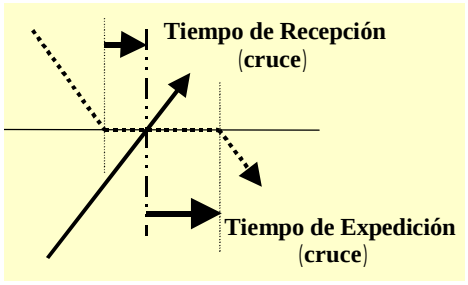


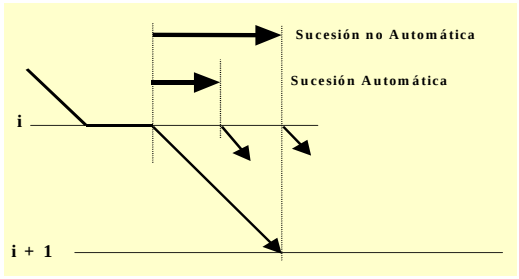


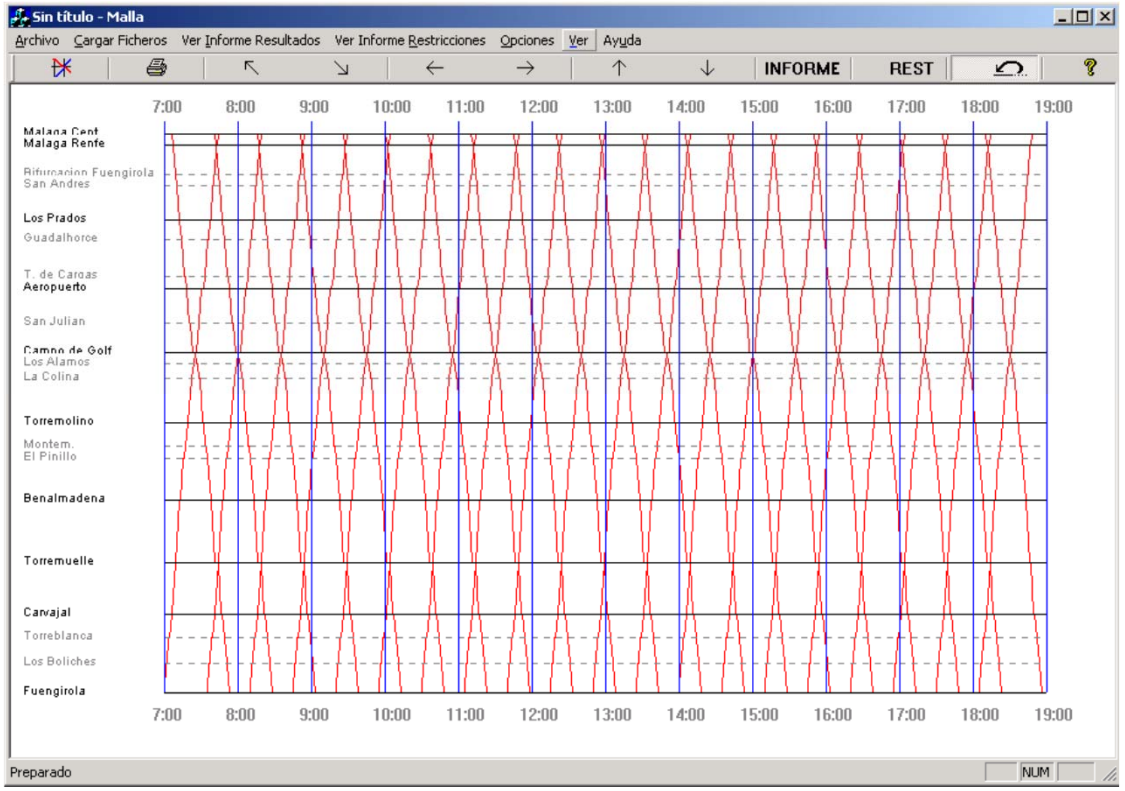

Tiempo de Expedición

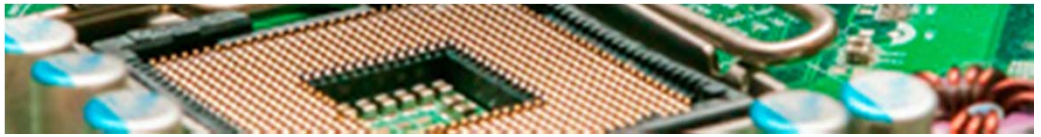

i
j
Timpar
Tpar
i
j
Tpar
Timpar


Tiempo de Recepción
(alcance)
Tiempo de Expedición
(alcance)


Tiempo de Recepción
(cruce)
Tiempo de Expedición
(cruce)

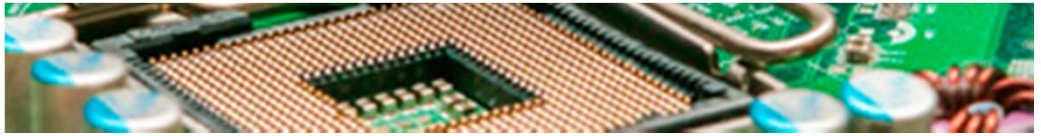

Sucesión no Automática
Sucesión Automática
i
i + 1


Sin título - Malla
Archivo Cargar Ficheros Ver Informe Resultados Ver Informe Restricciones Opciones Ver Ayuda
INFORME
REST
7:00 8:00 9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00
Malaga Cent
Malaga Renfe
Bifurcacion Fuengirola
San Andres
Los Prados
Guadalhorce
T. de Cargas
Aeropuerto
San Julian
Campo de Golf
Los Alamos
La Colina
Torremolino
Montem.
El Pinillo
Benalmadena
Torremuelle
Carvajal
Torreblanca
Los Boliches
Fuengirola
Preparado
NUM



¿Qué ocurre con la escalabilidad?

Uso de heurísticas y metaheurísticas



Optimización

En general, los problemas de optimización son particularmente difíciles de resolver.

- La complejidad NP-completa en factibilidad, resulta NP-dura en optimalidad

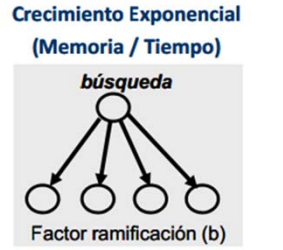
En muchos casos, no existe o no es viable un Método Algorítmico / Método exacto.

- No se puede garantizar encontrar la solución óptima en un tiempo computacional razonable.

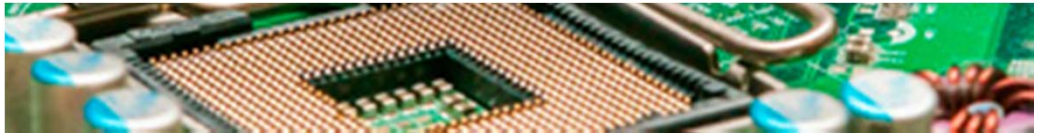
En general, basta con una solución razonablemente buena (factible, optimizada, pero no la óptima)

- Métodos aproximados (como alternativa a los métodos exactos)

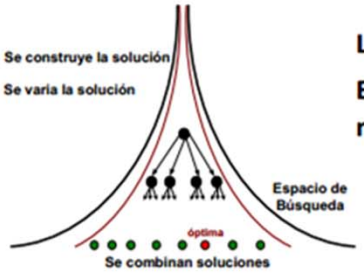
Búsqueda Búsqueda / Metaheurística: La solución es obtenida (generada / mejorada), mediante un proceso de búsqueda en un amplio espacio de estados/soluciones.



Se requieren métodos eficientes para encontrar soluciones (buenas soluciones)



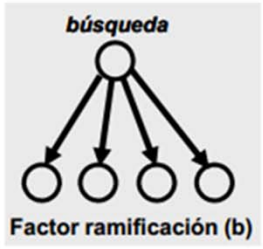
BÚSQUEDA ⇒ explosión combinatoria.

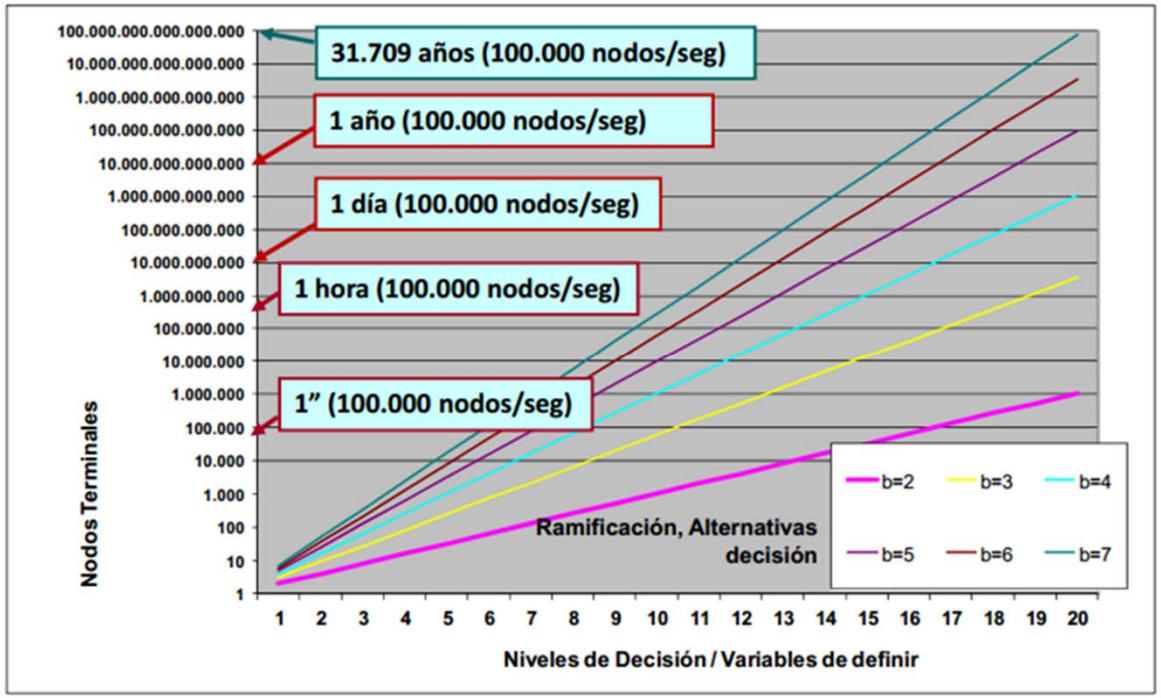


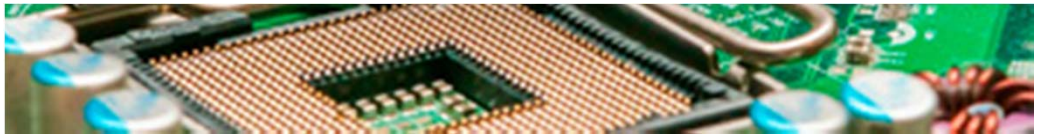
La explosión combinatoria depende del factor de ramificación (b) y niveles de decisión (n).

En dominios finitos (optimización combinatoria), el conjunto de soluciones es finito y numerable (pero muy grande).

Una fina hoja de papel (0.1 mm), doblada 50 veces, alcanza un espesor de….







Optimización

Estrategias de Búsqueda

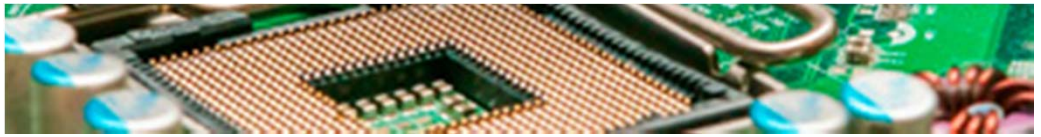
(Criterio para determinar el siguiente estado en la búsqueda: local/global)

- **Búsqueda Local :**
  busca solo en el ***entorno local actual***.
  (irrevocable, escalada, ...)
- **Búsqueda Sistemática (Global):**
  busca en ***todas*** las alternativas posibles.
  (tentativa, backtracking, algoritmos A, ...)

Métodos (procesos)

(búsqueda en un espacio de estados / soluciones)

- **Métodos Constructivos:** construyen una solución paso a paso
  Elección heurística de nuevo elemento de la solución.
  Espacio de búsqueda ⇐ espacio de estados.
  Búsqueda global/local.
- **Métodos de Mejora:** mejora iterativa de soluciones.
  Elección heurística de nueva solución cercana.
  Espacio de búsqueda ⇐ espacio de soluciones.
  Búsqueda local.
- **Métodos Evolutivos:** combinan soluciones previas
  Mejora de una población de soluciones utilizando información implícita en la sociedad/población.



Optimización: AG

Algoritmos Genéticos: Ideados por John Holland en 1975 inspirándose en la evolución natural de los seres vivos.

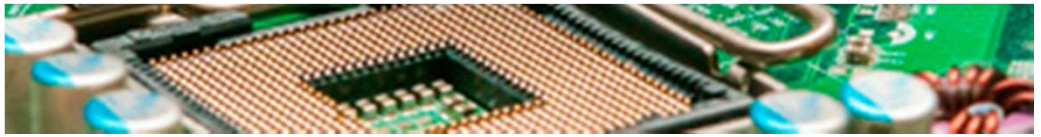
Idea subyacente:

- La población evoluciona de forma natural, mejorando su adaptación.
- En el transcurso de la evolución se generan poblaciones sucesivas, con información genética de los padres previos, y cuya adecuación al entorno va mejorando sucesivamente.

Los AG se inspiran en los procesos de Evolución Natural y Genética (métodos bioinspirados):

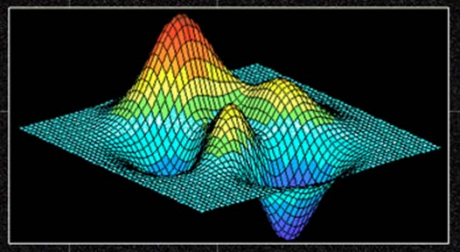
- Un AG evoluciona a partir de una población de soluciones inicial, intentando producir nuevas generaciones de soluciones que sean mejores que la anterior.
- El proceso evolutivo es guiado por decisiones probabilísticas *(componentes aleatorios)* basadas en la adecuación de los individuos (fitness).
- Al final del proceso, se espera obtener un buen individuo: buena solución global al problema.
- Orientados hacia la resolución (optimización) de problemas combinatorios.
- Los AG son el método metaheurístico más aplicado y con mejores resultados prácticos

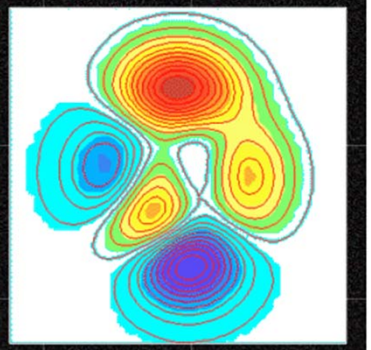
Población de Individuos	**⇔ Conjunto de Soluciones**
Evolución	**⇔ Búsqueda**
Mejor individuo	**⇔ Solución Final**



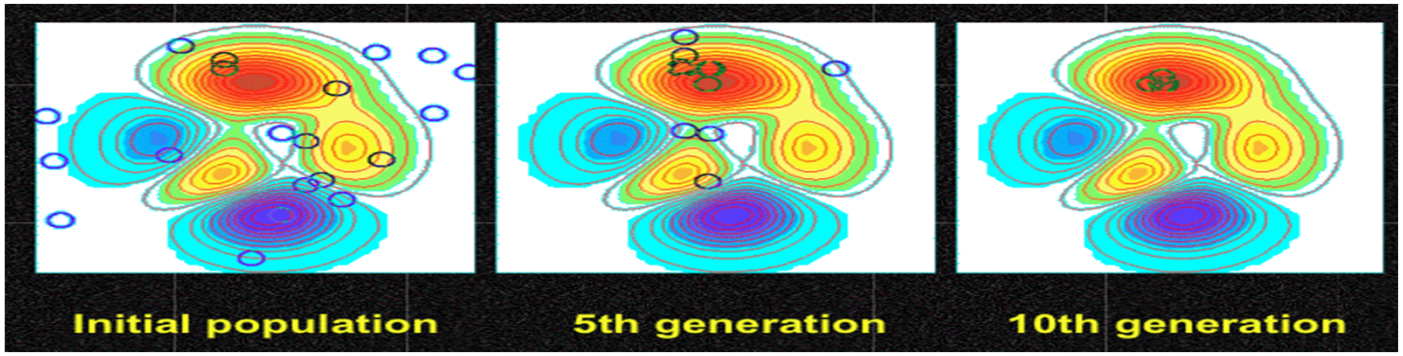
Un ejemplo de Algoritmo Genético

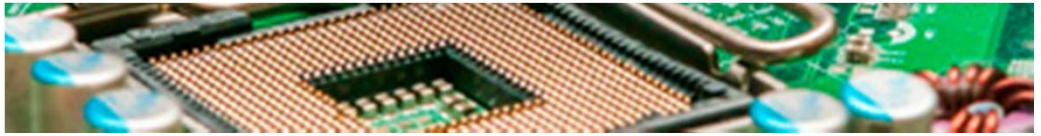
- Example: Find the max. of the "peaks" function
- z = f(x, y) = 3*(1-x)^2*exp(-(x^2) - (y+1)^2) - 10*(x/5 - x^3 - y^5)*exp(-x^2-y^2) -1/3*exp(-(x+1)^2 - y^2).





Source: Neuro-Fuzzy and Soft Computing, by J.-S. R. Jang, C.-T. Sun, E. Mizutani





Conceptos principales de un Algoritmo Genético

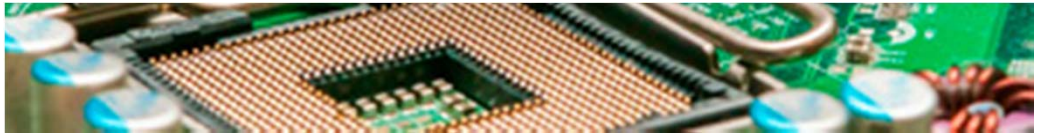
1. Una **representación** adecuada de las soluciones del problema (individuos). Típicamente binarias. **Cromosomas, genes, genotipo, fenotipo**.

2. Una forma de crear una **población de soluciones inicial** (individuos iniciales). A menudo, aleatoria.

3. Una **función de evaluación** capaz de medir la adecuación de cualquier solución (individuo) al problema. ***Fitness***.

4. Un conjunto de **operadores evolutivos** para combinar las soluciones existentes con el objetivo de obtener nuevas soluciones (nuevos individuos): ***selección, cruce, mutación y reemplazo*** de individuos. Guían el proceso de la búsqueda.

5. Conjunto de **parámetros de entrada**: tamaño de la población, número de iteraciones (generaciones), probabilidades de selección, etc.

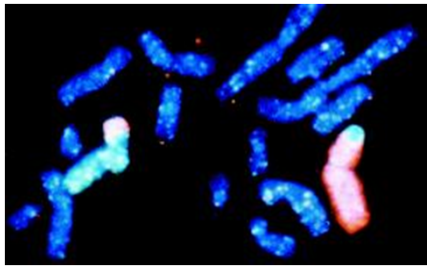


Individuo de la Población : Una solución del problema

Evolución Genética

Individuo ≡ Cromosoma: cadena de ADN que se encuentra en el núcleo de las células.

Los cromosomas son responsables de la transmisión de información genética.



Gen: sección de ADN que codifica una cierta función bioquímica.

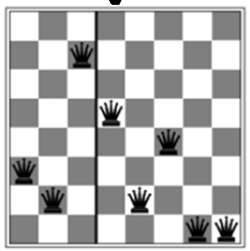
Algoritmo Genético

Individuo ≡ Cromosoma: estructura de datos que contiene una **secuencia de parámetros** que permite definir (o formar) una **solución** al problema.

Un cromosoma representa el **Genotipo** (representación interna de la instanciación):

3	2	7	5	2	4	1	1

Fenotipo: representación externa de una instanciación



Gen: subsección de un cromosoma que (usualmente) codifica el valor de un solo parámetro o aspecto del individuo.

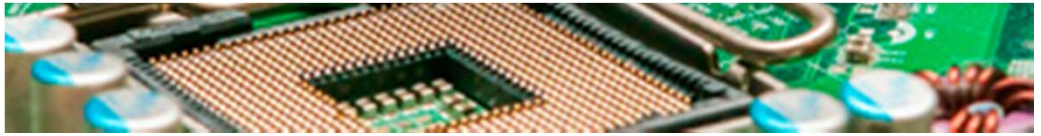
Optimización: Estructura AG

0. Codificación de Soluciones/instanciaciones.

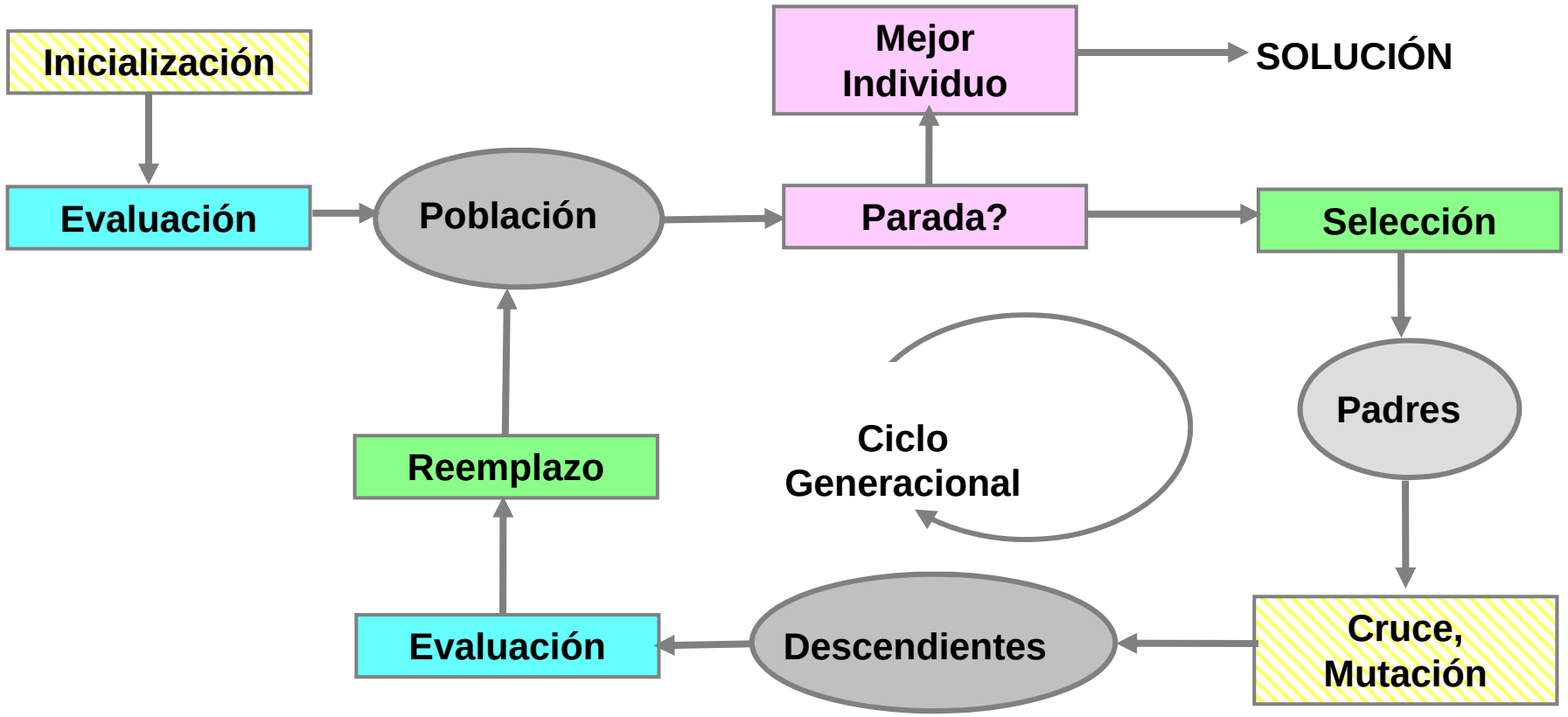
1. **[Población Inicial]:**
   - Generar población aleatoria de ***n*** individuos: {x} *(posibles soluciones del problema)*
   - Evaluar la aptitud o *fitness* ***f(x)*** de cada individuo.
2. **[Ciclo-Generacional]**:
   1. **[Selección]** Seleccionar dos padres de la población de acuerdo a su aptitud: Probabilidad de cruce ($\mathbf{p_c}$).
   2. **[Cruce]** Combinar los genes de los dos padres para obtener descendientes.
   3. **[Mutación]** Con una probabilidad de mutación ($\mathbf{p_{mut}}$), mutar cada gen de los cromosomas hijo.
   4. **[Reemplazo]** Añadir nuevos hijos y determinar nueva población.
3. Verificar condición de parada:

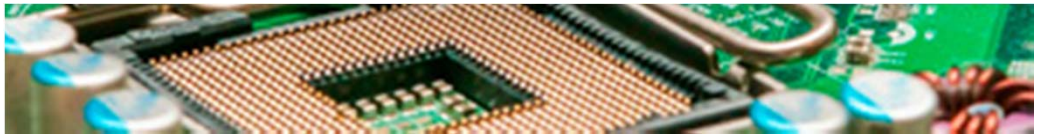
   [Parada]: proporcionar como solución el **individuo con mejor valor de aptitud *f(x)*.**

   [Ciclo]: Ir al paso **2**



Optimización: Proceso AG





Optimización: Proceso AG

Función de evaluación (8 reinas) = número de pares de reinas no atacadas

28 para una solución