
Algoritmos Exactos y Metaheurísticas

Primer Semestre 2025

Universidad Diego Portales
Prof. Víctor Reyes Rodríguez

Objetivos

- Sintonización de parámetros
-

MHs: Decisiones de diseño

- Tabu-Search
 - Tamaño de la lista
 - Simulated Annealing
 - Temperatura Inicial, parámetro de enfriamiento, cantidad de iteraciones para enfriamiento, Cantidad de iteraciones para calentamiento (opcional).
 - GA
 - Probabilidad de Cruzamiento/Mutación, Operadores, % Elitismo.
 - ACO
 - Tasa de evaporación.
 - PSO
 - Ponderación global best-local best-inertia.
-

Decisiones de diseño

- Debemos siempre tomar en cuenta:
 - Una buena técnica de búsqueda comienza explorando el espacio de búsqueda para, después, explotar aquellas zonas del espacio que parecen favorables
 - Elementos interrelacionados, por lo tanto el proceso de diseño es más complejo.
-

¿Qué son los parámetros?

- Elementos numéricos, a los cuales es necesario asignarle un conjunto/rango de valores.
 - Surge un problema...GRAN PROBLEMA...**El problema de asignación de valores a parámetros.** A esto se le llama sintonización de parámetros.
-

Asignación de valores a parámetros

- Proceso costoso en el tiempo
 - Parámetros propios de cada problema
 - Parámetros interrelacionados.
 - Se determinan a partir de un conjunto de benchmarks.
-

Sintonización vs Control

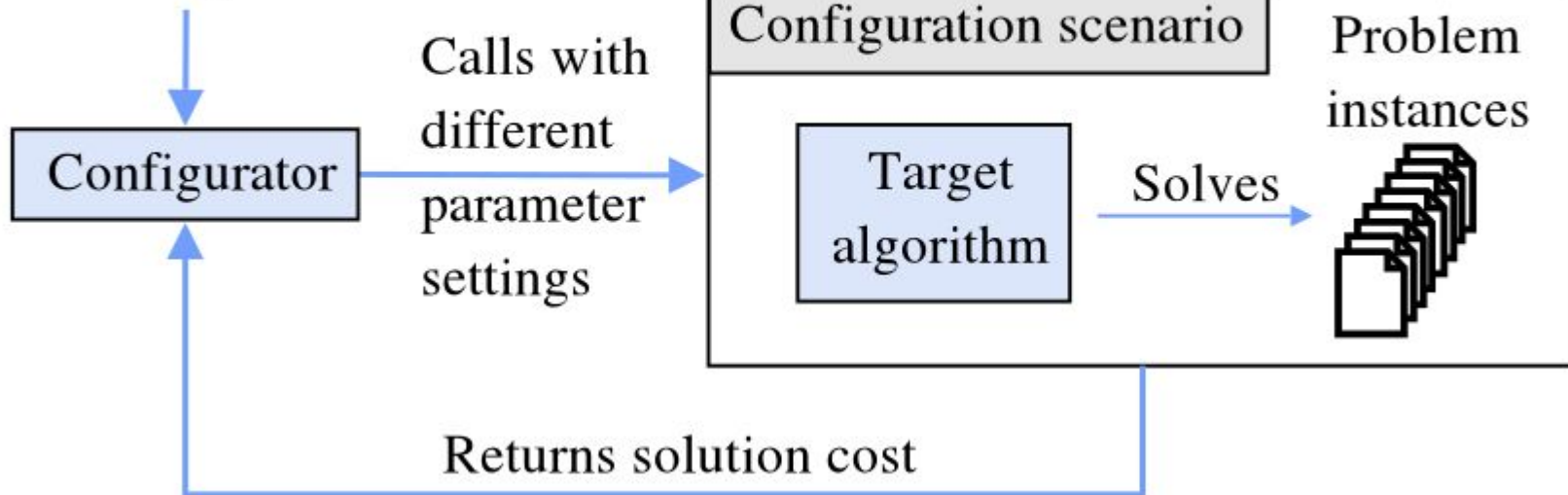
- Sintonización: Proceso previo a la ejecución del algoritmo.
 - Altamente consumidor de tiempo
 - Parámetros útiles para el problema en particular
 - Valores fijos durante la ejecución
 - Control: Proceso conjunto a la ejecución del algoritmo.
 - Tiempo de diseño
 - Tiempo adicional durante la ejecución
 - Valores variables durante la ejecución del algoritmo
-

Sintonización de parámetros

- Manual
 - Estadística: Racing
 - Meta-Algoritmos: ParamILS, Meta-Genético.
-

Sintonización de parámetros

Parameter domains
& starting values



Sintonización de parámetros: Racing

- Se basa en un testeo estadístico, en donde cada algoritmo es evaluado iterativamente a lo largo de la carrera.
 - La performance de un algoritmo se define como la habilidad de hacer evolucionar a una población, medido a través del fitness del campeón.
 - Siempre que se encuentre evidencia estadísticamente significativa sobre la inferioridad de un algoritmo, este algoritmo será eliminado de la carrera.
 - Es utilizado el test de Friedman (se encuentra en R, python, etc)
-

Sintonización de parámetros: Racing

- Ejemplos de este algoritmo:
 - <https://esa.github.io/pygmo/>
 - <https://github.com/cran/irace>
-

Sintonización de parámetros: Búsqueda local

- Uno podría hacer lo siguiente, de manera secuencial:
 - Comenzar con una configuración inicial aleatoria de parámetros
 - Experimentar cambiando el valor de un parámetro, aceptando la nueva configuración solo si hay un mejor desempeño
 - Repetir lo anterior hasta que no exista una configuración que mejore el desempeño.
 - Esto corresponde a una búsqueda local, por tanto uno podría usar esta idea y optimizar de mejor manera el proceso de búsqueda de parámetros → Iterated Local Search (ILS)
-

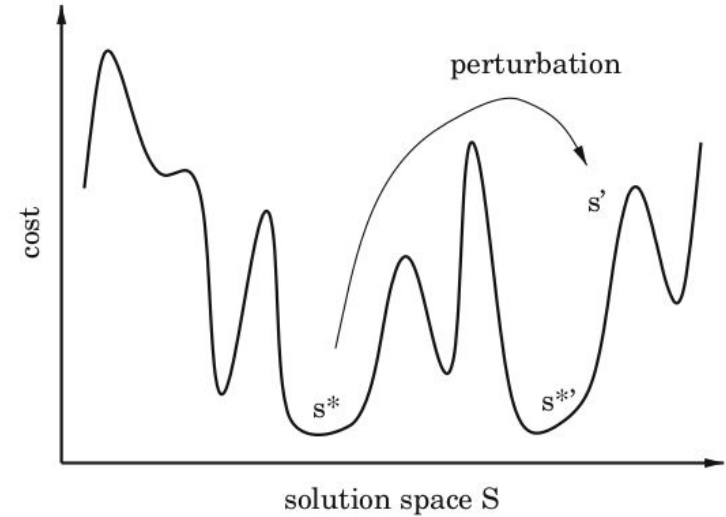
ILS: Recordatorio

- Dada una solución s' , se aplica una perturbación (random-walk) o pequeño cambio.
 - Luego, a partir de dicha nueva solución, se aplica una búsqueda local, es decir, con operadores de movimiento buscar en la vecindad por mejores soluciones. Con esto llegamos a s''
 - Si s'' cumple con ciertos criterios (por ejemplo es “mejor” que s'), entonces, será nuestra nueva solución y repetimos el proceso, si no, volvemos a s' .
 - La clave está en la perturbación: Si es pequeña probablemente no mejore mucho, si es muy grande será muy aleatoria la búsqueda.
-

ILS: Recordatorio

Algorithm 1 Iterated local search

```
1:  $s_0 = \text{GenerateInitialSolution}$   
2:  $s^* = \text{LocalSearch}(s_0)$   
3: repeat  
4:    $s' = \text{Perturbation}(s^*, \text{history})$   
5:    $s^{*'} = \text{LocalSearch}(s')$   
6:    $s^* = \text{AcceptanceCriterion}(s^*, s^{*'}, \text{history})$   
7: until termination condition met
```



Sintonización de parámetros: ParamILS

```
Input   : Parameter configuration space  $\Theta$ , neighbourhood  
          relation  $\mathcal{N}$ , procedure better (compares  $\theta, \theta' \in \Theta$ ).  
Output : Best parameter configuration  $\theta$  found.  
1  $\theta_0 \leftarrow$  default parameter configuration  $\theta \in \Theta$ ;  
2 for  $i \leftarrow 1 \dots R$  do  
3    $\theta \leftarrow$  random  $\theta \in \Theta$ ;  
4   if better( $\theta, \theta_0$ ) then  $\theta_0 \leftarrow \theta$ ;  
5  $\theta_{ils} \leftarrow \text{IterativeFirstImprovement}(\theta_0, \mathcal{N})$ ;  
6 while not TerminationCriterion() do  
7    $\theta \leftarrow \theta_{ils}$ ;  
   // ===== Perturbation  
8   for  $i \leftarrow 1 \dots s$  do  $\theta \leftarrow$  random  $\theta' \in \mathcal{N}(\theta)$ ;  
   // ===== LocalSearch  
9    $\theta \leftarrow \text{IterativeFirstImprovement}(\theta, \mathcal{N})$ ;  
   // ===== AcceptanceCriterion  
10  if better( $\theta, \theta_{ils}$ ) then  $\theta_{ils} \leftarrow \theta$ ;  
11  with probability  $p_{\text{restart}}$  do  $\theta_{ils} \leftarrow$  random  $\theta \in \Theta$ ;  
12 return overall best  $\theta$  found;  
13 Procedure IterativeFirstImprovement ( $\theta, \mathcal{N}$ )  
14 repeat  
15    $\theta' \leftarrow \theta$ ;  
16   foreach  $\theta'' \in \mathcal{N}(\theta')$  in randomised order do  
17     if better( $\theta'', \theta'$ ) then  $\theta \leftarrow \theta''$ ; break;  
18 until  $\theta' = \theta$ ;  
19 return  $\theta$ ;
```

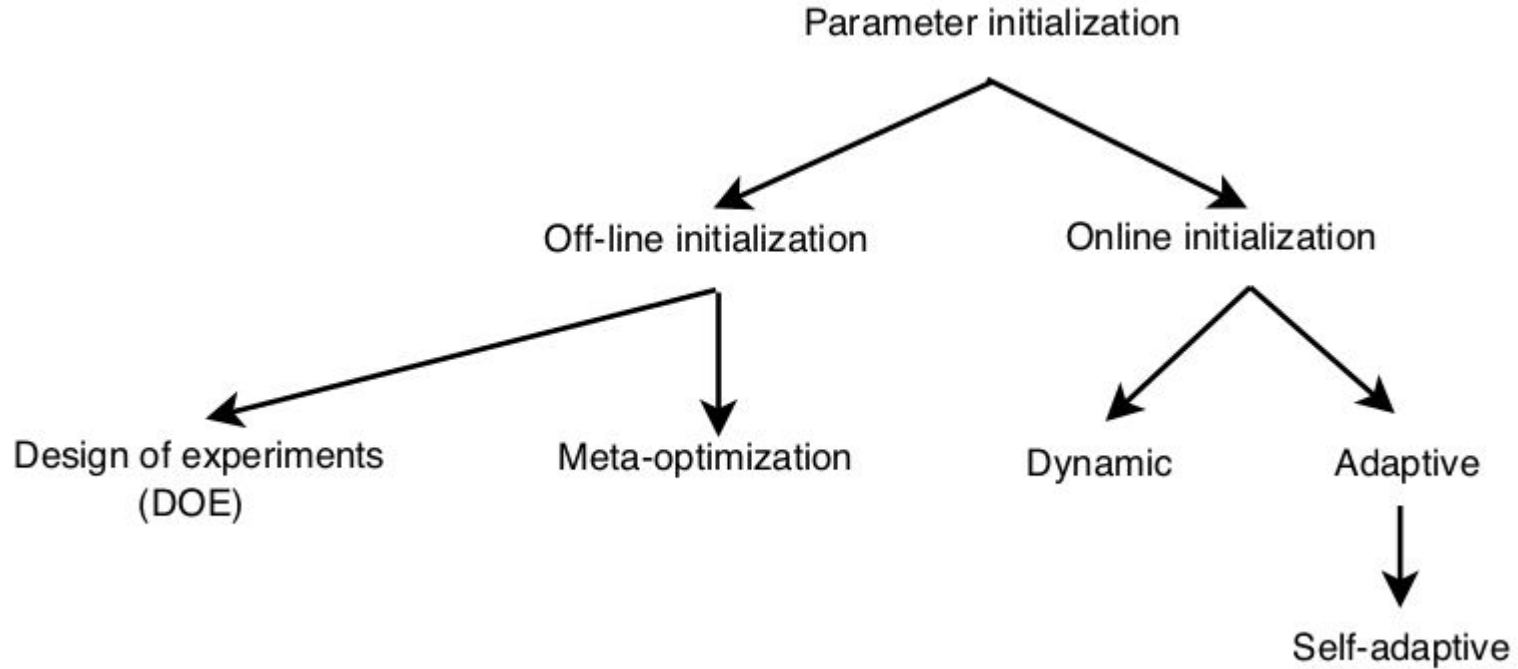
Sintonización de parámetros: ParamILS

- <https://www.cs.ubc.ca/labs/algorithms/Projects/ParamILS/>
-

Clasificación de Técnicas de Control

- Control Dinámico: Los cambios se realizan de forma determinista con respecto a alguna forma de medición de tiempo en el algoritmo. Cada n iteraciones disminuye el valor del parámetro x en 10%.
 - Control Adaptativo : Los cambios se realizan en base a la información de la búsqueda. La idea es monitorear ciertos elementos del algoritmo de forma de terminar cambios en los valores de los parámetros en base a cambios en dichos elementos. Los valores de los parámetros son globales para el algoritmo.
 - Control auto-Adaptativo: Lo mismo a lo anterior, sin embargo, los valores de los parámetros son propios de cada solución/individuo (algoritmos poblacionales).
-

Resumen



Análisis de desempeño en MHs

- Diseño experimental: Selección de benchmarks.
 - Mediciones: Análisis estadístico de resultados (tiempo de CPU/Valor de la f.o vs global). Debe en general compararse con técnicas del Estado-del-arte dedicadas al problema. Debe responder: ¿Cuántas ejecuciones? ¿Qué tan estable es mi técnica (min, prom, max)?
 - Reporte: Análisis de los datos. Gráficos de convergencia, gráficos de cajas, reporte de tiempo de CPU, generaciones, según sea el caso. Lo importante es asegurar la reproducibilidad de los experimentos.
-