
Algoritmos Exactos y Metaheurísticas

Primer Semestre 2025

Universidad Diego Portales
Prof. Víctor Reyes Rodríguez

Objetivos

- Iterated Local Search (ILS) y Large Neighborhood Search (LNS)
 - Metodología experimental:
 - Benchmarks
 - Número de ejecuciones, métricas, etc
 - Gráficos
 - Análisis de resultados
 - Repaso
-

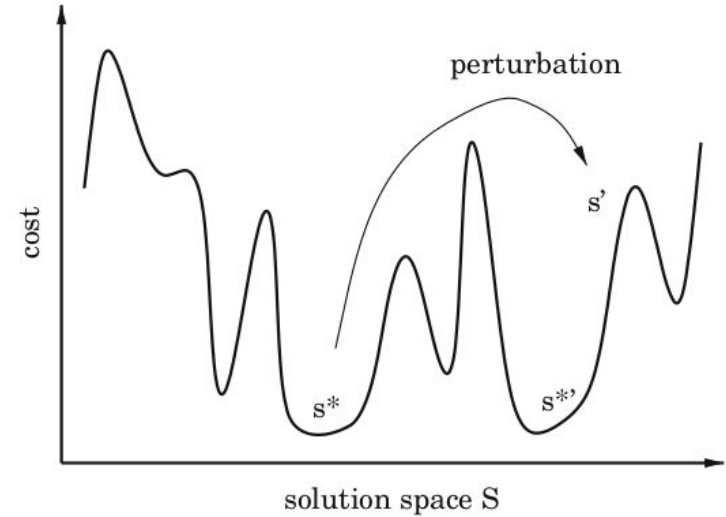
Otros algoritmos de trayectoria: ILS

- 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.
-

Otros algoritmos de trayectoria: ILS

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
```



Otros algoritmos de trayectoria: LNS

- Utiliza dos métodos: destroy y repair.
 - El método destroy destruye parte de la solución (estocásticamente), mientras el método repair reconstruye la parte destruida. Dicho esto, el vecindario es definido como las soluciones a las cuales se puede llegar a partir de dicha nueva (incompleta) configuración.
 - Para la reparación puede ser utilizado un método heurístico.
-

Otros algoritmos de trayectoria: LNS

Algorithm 1 Large neighborhood search

```
1: input: a feasible solution  $x$ 
2:  $x^b = x$ ;
3: repeat
4:    $x^t = r(d(x))$ ;
5:   if  $\text{accept}(x^t, x)$  then
6:      $x = x^t$ ;
7:   end if
8:   if  $c(x^t) < c(x^b)$  then
9:      $x^b = x^t$ ;
10:  end if
11: until stopping criterion is met
12: return  $x^b$ 
```

Metodología experimental

- Toda propuesta debe ser validada de manera cuantitativa, esto es, resolución de N instancias, mostrar resultados, comparar con el estado del arte y analizar.
 - Mencionar características de la máquina utilizada, realizar todos los experimentos en las mismas condiciones.
 - Validación a través de:
 - Benchmarks de la comunidad, generalmente de distinta dificultad.
 - Generación de benchmarks(*)
 - ¿Cuántas veces debo ejecutar cada benchmark?
 - Revisar estado del arte.
 - Tests estadísticos → t-test, wilcoxon, etc
-

Metodología experimental

- Definir cuáles serán las métricas que se usarán para la comparación.
 - Técnicas completas: Nodos, Tiempo de CPU
 - Técnicas incompletas: (Promedios/desv. estándar/...) Número de iteraciones, Tiempo de CPU, calidad de solución
 - Análisis paramétrico: ¿Cómo cambian los resultados de mis métricas al modificar los distintos parámetros (tamaño de lista tabú, temperatura, etc) de mi algoritmo?
 - Barrido paramétrico
 - Técnicas de optimización para encontrar una buena configuración
 - Tablas y gráficos: La idea no es mostrar información redundante, siempre complementaria.
-

Ejemplo

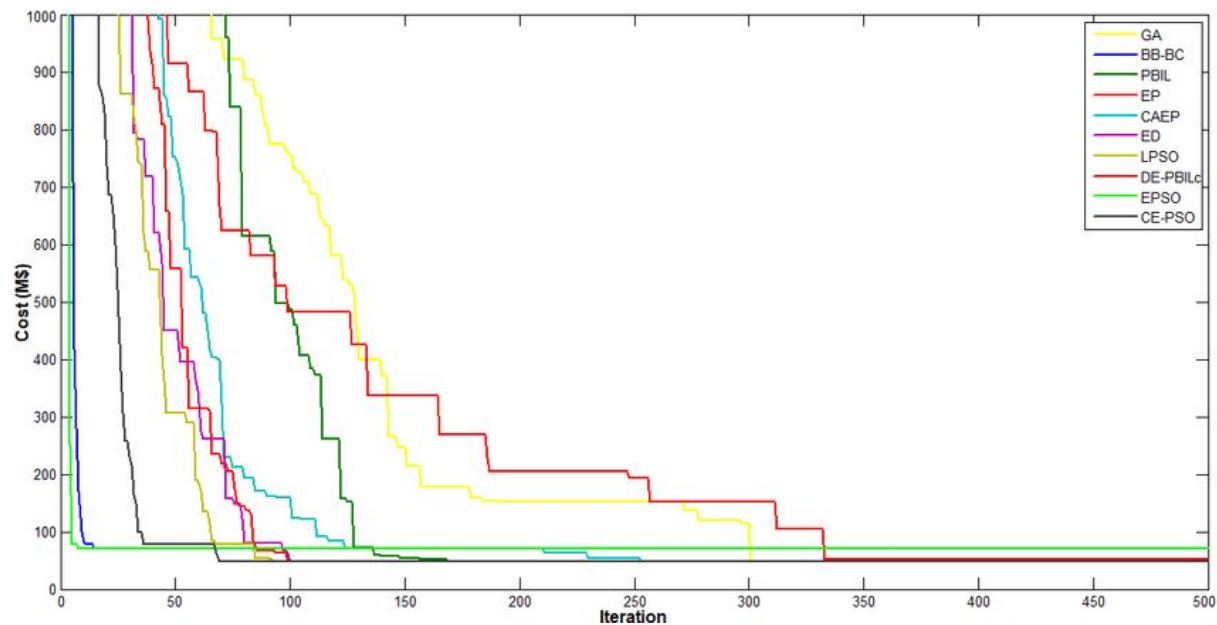
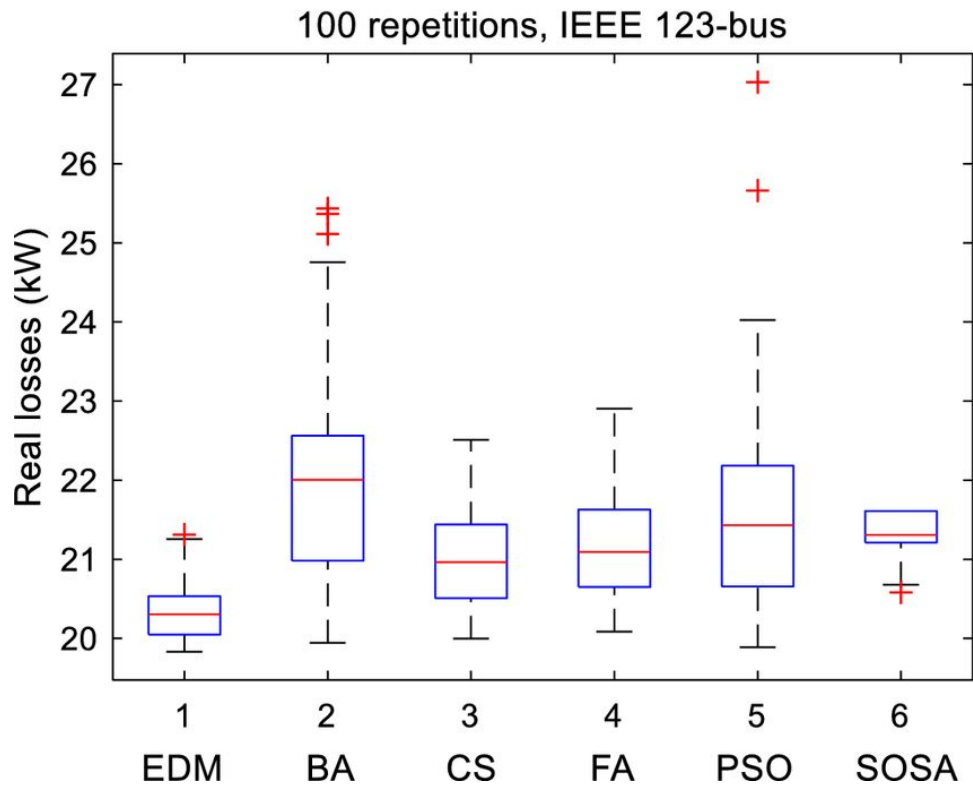


Table 9

Comparison of the results of Abdulkader et al. [1] on instances with division strategy S3 to ITS with single and multiple visits and 10^4 iterations.

Inst.	BKV	HAC		ITS							
		R.d.	t(s)	Single			Multiple				t(s)
				Avg.	Best	t(s)	Avg.	Best	Vis.		
CMT1	550.7	0.0	5	0.1	0.0	4.7	0.0	0.0	0.0	6.4	
CMT2	890.7	0.0	15	-1.8	-2.1	6.5	-1.0	-2.1	1.2	9.2	
CMT3	874.1	0.0	40	-0.5	-0.8	9.9	-0.1	-0.6	0.3	15.0	
CMT4	1126.1	0.0	146	-2.2	-3.0	15.5	-1.7	-2.4	0.7	22.8	
CMT5	1444.3	0.0	257	-3.3	-3.9	23.0	-2.9	-3.5	1.4	32.6	
CMT6	557.5	0.0	11	0.0	0.0	3.8	0.0	0.0	0.0	4.8	
CMT7	928.2	0.0	28	0.5	-0.2	6.8	0.0	-0.2	0.0	9.9	
CMT8	883.0	0.0	93	0.1	-0.7	7.9	1.4	-0.7	0.4	10.8	
CMT9	1228.9	0.0	326	-1.7	-3.0	14.6	-1.2	-2.1	0.5	17.0	
CMT10	1511.7	0.0	624	-2.1	-3.1	15.8	-2.5	-3.8	0.9	25.0	
CMT11	1110.5	0.0	75	-0.2	-0.4	11.8	-0.1	-0.4	1.1	14.7	
CMT12	911.9	0.1	15	-0.5	-0.7	12.0	-0.2	-0.6	0.7	15.4	
CMT13	1556.5	0.0	117	-0.5	-0.8	7.6	1.5	-0.7	0.0	11.4	
CMT14	911.4	0.0	34	0.1	-0.0	11.9	0.1	0.0	0.7	8.4	
Avg.	1034.7	0.0	128	-0.9	-1.3	10.8	-0.5	-1.2	0.6	14.5	

Ejemplo



Metodología experimental

- Análisis de resultados:
 - Resumir y destacar resultados más importantes.
 - Explicar casos en donde su propuesta mejora los resultados, y también explicar aquellos en donde pierde.
 - Esto es de suma importancia para validar la contribución.
-

“El repaso”

- CSP/COP → ¿Características?
 - Técnicas completas v/s incompletas: ¿Cuándo usar cada una? ¿Limitantes?
 - Técnicas completas: Ejemplos y cómo funcionan.
 - Técnicas incompletas: Constructivas v/s Perturbación, Trayectoria v/s Población
 - Técnicas incompletas (trayectoria): Ejemplos y cómo funcionan.
-

Resumen

- Y con esto terminamos la parte de MH de solución única.
 - ¿Qué viene? → Algoritmos basados en poblaciones: Evolutivos, Swarm y sintonización de parámetros.
-