Algorítmica II
Algoritmos Metaheurísticos

Joaquín Roiz Pagador

Abril 2018.

Contenido

Descripción del Problema

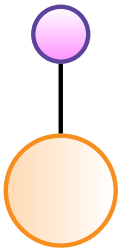
Idea principal

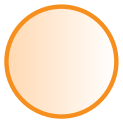
- Asignación óptima de clientes a médicos $\rightarrow$ Se desea minimizar desplazamiento.
- Contratación basada en la necesidad.
- Ahorrar costes de contratación $\rightarrow$ Cada médico tiene un coste asignado y máximo número de pacientes asignables.

Descripción del Problema

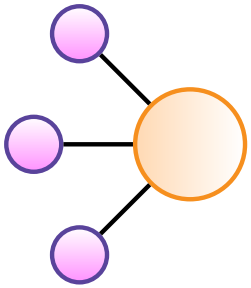
En otras palabras...

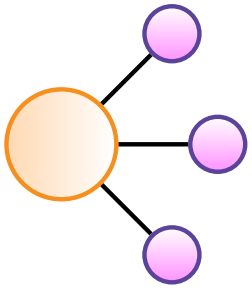
Solución Ideal Pacientes



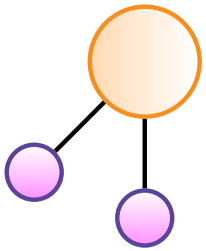


Pacientes





Médicos



Representación del problema

Teoría de Grafos

- Médicos con coste asignado, coordenadas X e Y.
- Número máximo de pacientes por médico.
- Pacientes con ubicación también: X e Y.

Idea Principal

Aplicando teoría de grafos buscaremos las aristas que interconecten el catálogo de médicos contratados con todos los pacientes de la forma más eficiente posible.

Representación del problema

Teoría de grafos

- Una matriz de adyacencias que unirá al paciente i con el médico j.
- Representación Bidimensional.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Más eficiente en una dimensión

$$\begin{bmatrix} 0 & 1 & 2 & 2 \end{bmatrix}$$

Cálculo de Fitness

$peorDist \in N \leftarrow$ Peor distancia posible para cada uno de los pacientes
$peorCost \in N \leftarrow$ Peor coste posible (contratar a todos)

$$f(x) = \frac{\sum_{i=1}^{n} dist(i, g(i))}{peorDist} \cdot \frac{\sum_{j=1}^{n} cost(g(j))}{peorCost} \quad \exists!\ g(j) \in g$$

- g $\rightarrow$ vector representativo de la solución o genotipo.

Representación del problema

Validez de una Solución

- Una solución será válida si, y sólo si, no existe un médico que supere el número de pacientes asignados.

Algoritmo

assignations[]
$i \leftarrow 0$
$valid \leftarrow true$
while $(valid \wedge i < genotypeSize)$ **do**
  assignations$[i]++$
  if *(superadoAsignaciones)* **then**
    $valid = false$
  end
  $i++$
end

Algoritmos Escogidos

Simulated Annealing

Aplicación al problema

En cada iteración se permutarán dos posiciones del vector solución con el que se esté trabajando, o bien, se invertirá el orden del mismo. Se someterá al vecino a la probabilidad de aceptación.

Temperatura Inicial

$$n \cdot \ln^2(n) \ (\forall n \in Max(nPacientes, nMedicos))$$

Enfriamiento

$$T_{k+1} = T_k - T_k \cdot 0.001$$

Algoritmos Escogidos

Genetic Algorithm

Aplicación al problema

Se tomará un número de iteraciones (generaciones) para un tamaño de población adecuado al tamaño del problema a resolver realizando selecciones, cruces y mutaciones.

Población y generaciones

- Población inicial $\rightarrow \ln(n) \cdot 10 \ \forall n \in Max(nPacientes, nMedicos)$.
- Generaciones $\rightarrow n \cdot \ln^2(n)$ $(\forall n \in Max(nPacientes, nMedicos))$ o $n \cdot \ln(n)$ iteraciones sin mejora.

Algoritmos Escogidos

Genetic Algorithm

Selección

- Elitimos 10%
- Torneo 10%
- Ruleta 20%

Cruce y mutación

El restante 60% se genera de la población inicial aleatoriamente existiendo un 50% de probabilidad de que sea uniforme o porción. Existe un 10% de probablidades de que el individuo cruzado mute.

Algoritmos Escogidos

Memetic Algorithm

Aplicación al problema

Se aprovechará el algoritmo genético y el enfriamiento simulado. Antes de comenzar y cada 3% iteraciones se ejecutará la búsqueda local a la mitad de la población.

Iteraciones y convergencia

Se aplicarán los mismos que el algoritmo genético, reduciendo la temperatura inicial del enfriamiento simulado a $\ln^{1.7}(n)$ $(\forall n \in Max(nPacientes, nMedicos))$.

Resultados

Simulated Annealing

Instancia.	20	200	2000
Fitness	0.2164	0.2799	0.3072
Tiempo Necesitado (s:ms).	0:048	0:193	4:587
Iteración exitosa	7969	15103	16840
Coste de la contratación	186118	2347819	24904281
Número de contrataciones	9	118	1246
Distancia total de los pacientes(km)	3.94	47.95	519.16

Figure: Datos obtenidos de las instancias en el algoritmo Enfriamiento Simulado.

Resultados

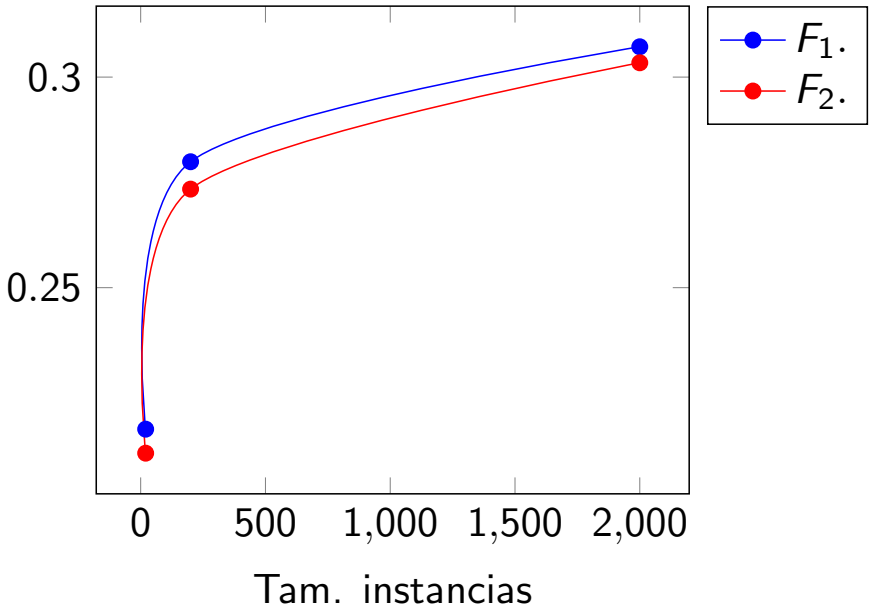
Simulated Annealing

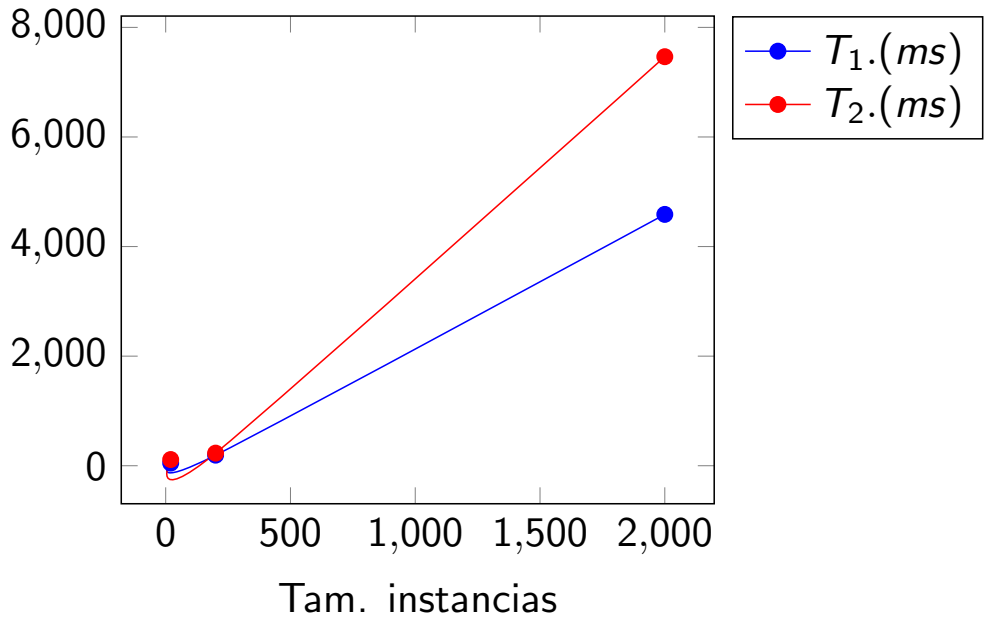
Instancia.	20	200	2000
Fitness	0.2107	0.2734	0.3034
Tiempo Necesitado (s:ms).	0:113	0:230	7:465
Iteración exitosa	3049	9475	308
Coste de la contratación	159497	2273990	25608893
Número de contrataciones	8	114	1253
Distancia total de los pacientes(km)	4.466	48.34	510.55

Figure: Datos obtenidos de la segunda ejecución de las instancias en el algoritmo Enfriamiento Simulado.

Resultados

Simulated Annealing





Resultados

Genetic Algorithm

Instancia.	20	200	2000
Fitness	0.1015	0.03769	0.0184
Tiempo Necesitado (m:s:ms).	0:0:062	0:2:750	15:09:109
Iteración exitosa	179	5613	115437
Coste de la contratación	96975	737228	6872659
Número de contrataciones	5	37	348
Distancia total de los pacientes(km)	3.54	20.56	112.70

Figure: Datos obtenidos de las instancias en el Algoritmo Genético.

Resultados

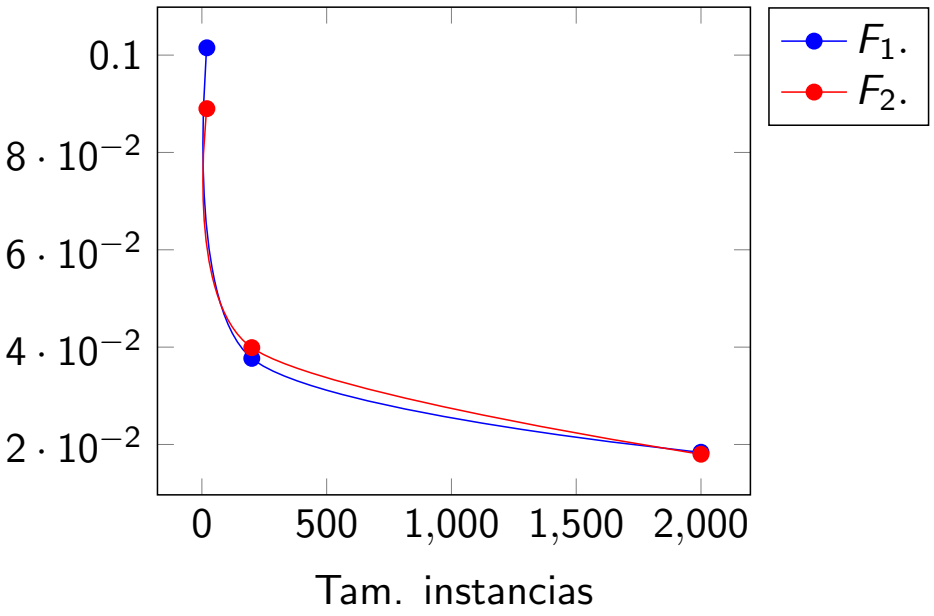
Genetic Algorithm

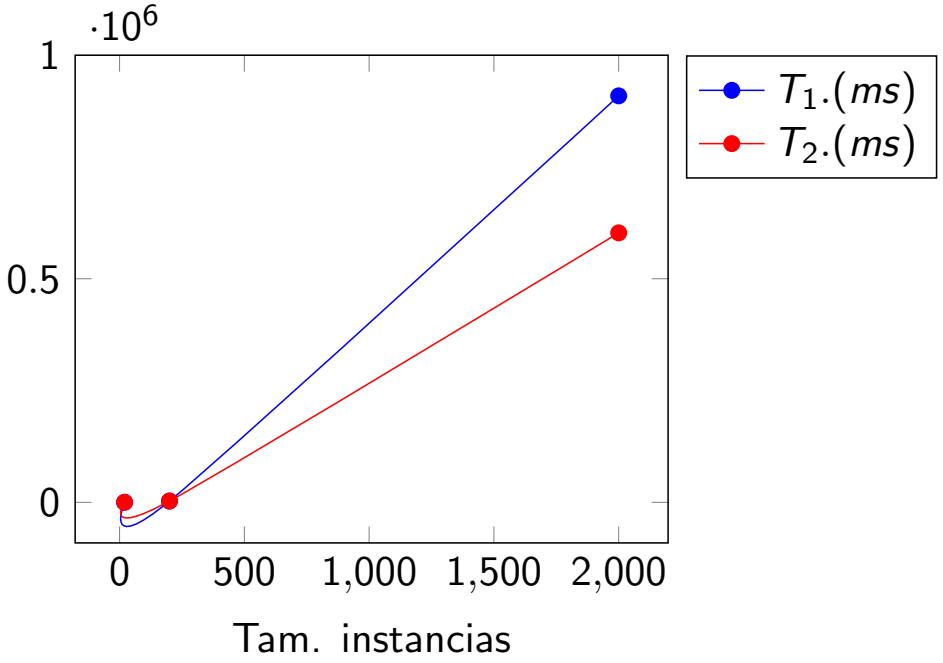
Instancia.	20	200	2000
Fitness	0.089	0.0399	0.0180
Tiempo Necesitado (m:s:ms).	0:0:117	0:2:916	10:0.2:671
Iteración exitosa	103	5491	115539
Coste de la contratación	88000	726260	7027870
Número de contrataciones	5	35	358
Distancia total de los pacientes(km)	3.44	22.14	107.94

Figure: Datos obtenidos de la segunda ejecución de las instancias en el Algoritmo Genético.

Resultados

Genetic Algorithm





Resultados

Memetic Algorithm

Instancia.	20	200	2000
Fitness	0.0654	0.034	0.0183
Tiempo Necesitado (m:s:ms).	0:0:174	0:07:823	14:27:747
Iteración exitosa	77	5606	115538
Coste de la contratación	73421	749875	6093097
Número de contrataciones	4	39	337
Distancia total de los pacientes(km)	3.0206	18.62	116.344

Figure: Datos obtenidos de las instancias en el Algoritmo Genético.

Resultados

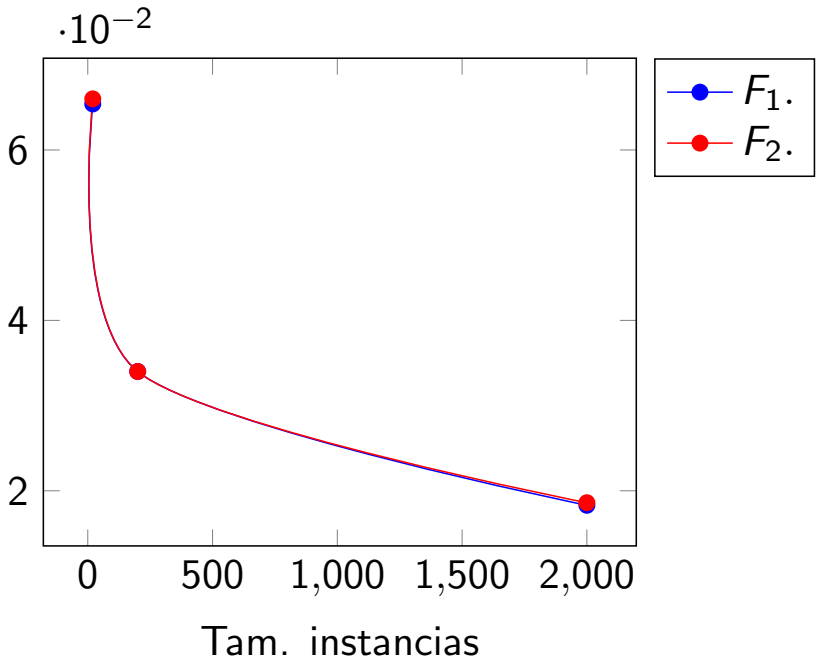
Memetic Algorithm

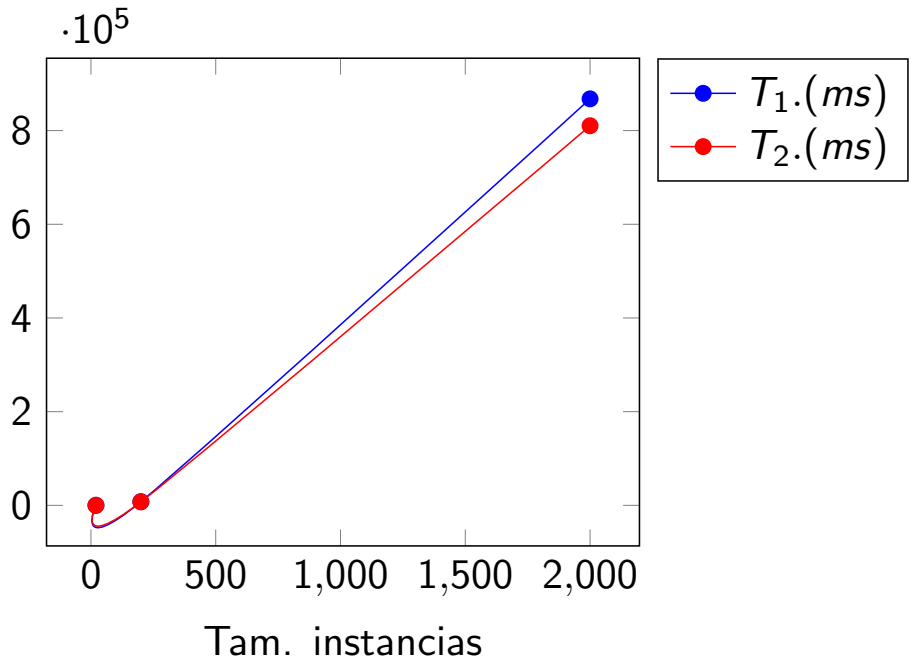
Instancia.	20	200	2000
Fitness	0.066	0.034	0.0186
Tiempo Necesitado (m:s:ms).	0:0:152	0:7:464	14:10:313
Iteración exitosa	110	5612	115479
Coste de la contratación	57531	692780	6741894
Número de contrataciones	3	36	340
Distancia total de los pacientes(km)	3.94	20.15	116.703

Figure: Datos obtenidos de la segunda ejecución de las instancias en el Algoritmo Genético.

Resultados

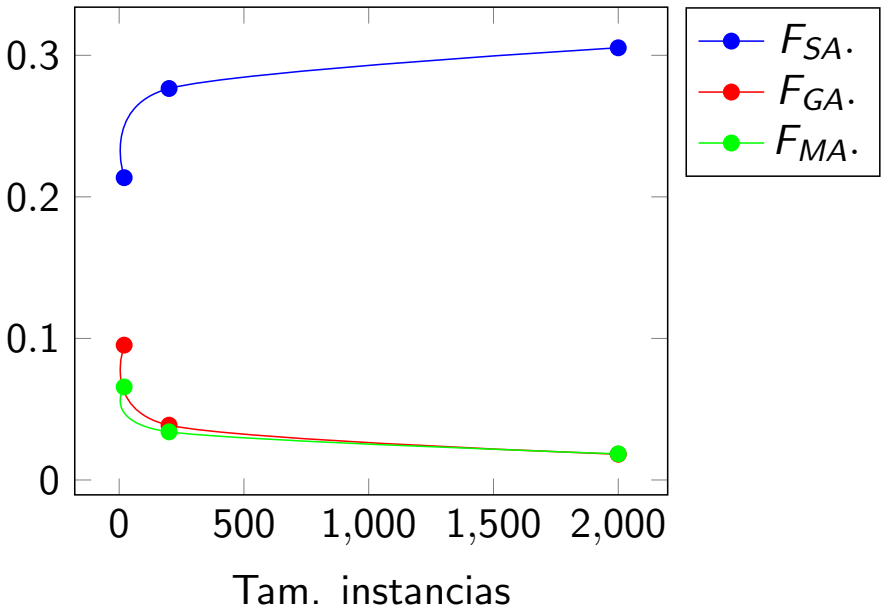
Memetic Algorithm

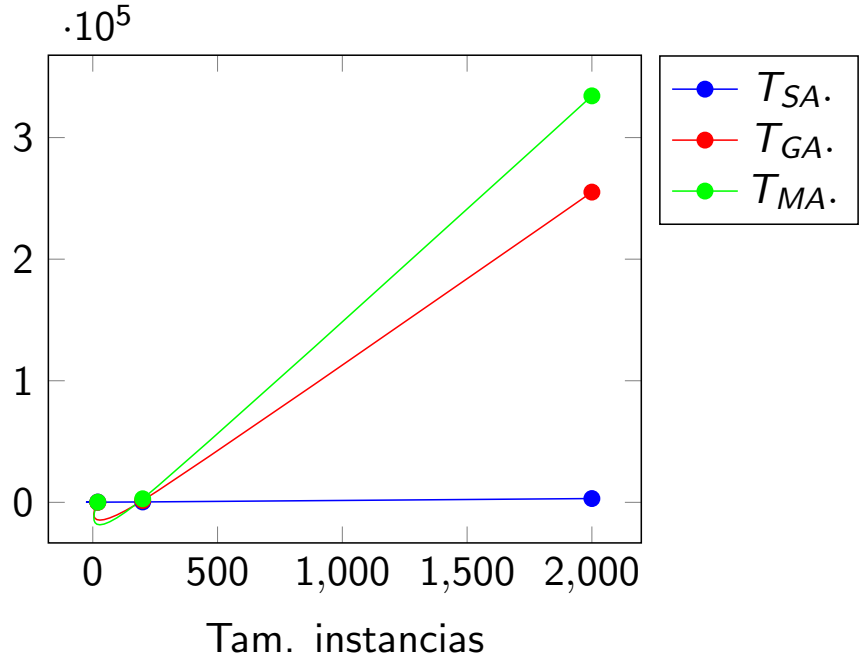




Resultados

Memetic Algorithm





Investigación

Metaheurísticas Paralelas

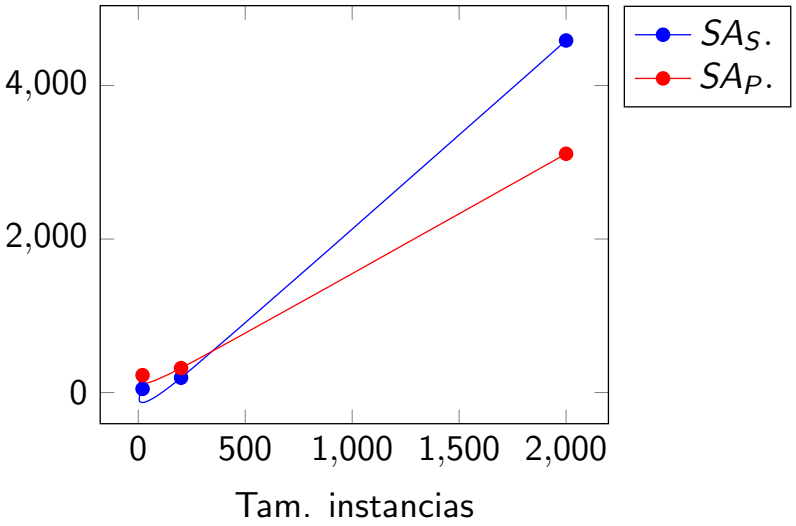
Descripción

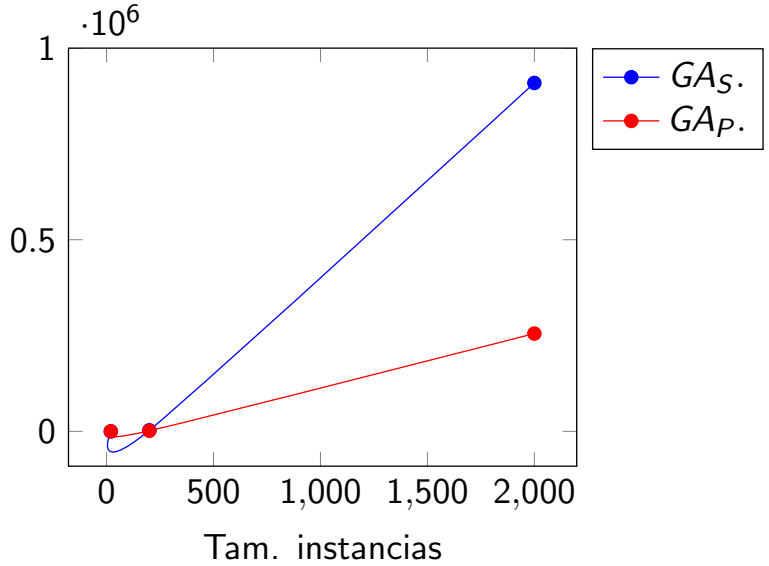
Posibilidad de lanzar hilos de ejecución para que se ejecuten en paralelo por cada uno de los núcleos del sistema, agilizando los procesos costosos.

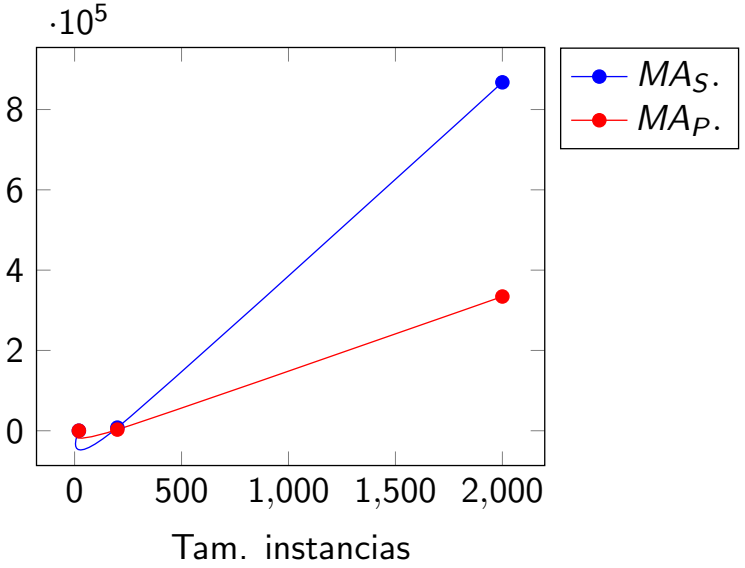
Dónde

Se utiliza en cálculos costosos, iteraciones elevadas, como comprobación de validez, generación de vecino, cálculo de bondad o fitness, operadores de cruce, etc.

- Programación funcional $\rightarrow$ Pradigma de programación declarativa basado en funciones mateáticas y cálculo lambda.
- Lambda: $(parametros) \rightarrow cuerpo$
- ParallelStream: Abstracción implementada en Java 8. Permite procesar datos de modo declarativo, aprovechando la arquitectura de núcleos múltiples sin necesidad de programar líneas de multiproceso.


4,000
2,000
0
0
500
1,000
1,500
2,000
Tam. instancias
SA_S.
SA_P.


$\cdot 10^6$
1
0.5
0
0
500
1,000
1,500
2,000
Tam. instancias
GA_S.
GA_P.


$\cdot 10^5$
8
6
4
2
0
0
500
1,000
1,500
2,000
Tam. instancias
MA_S.
MA_P.

Bibliografías

Metaheuristics in Combinational Optimization.

C. Blum, A. Roli

Inteligencia Arti

cial para desarrolladores. Conceptos e implementación en Java.Ediciones ENI, 2017

Oracle