

# SIMULATED ANNEALING TEMPLATE

## DESCRIPCIÓN

Leslie Pérez Cáceres

---

El template de Simulated Annealing que utilizará en este curso le permitirá aplicar el algoritmo para resolver instancias del problema del vendedor viajero (TSP) en formato TSPLib. La carpeta **sa\_solver** contiene:

- código: clases Java en la carpeta **src/**
- instancias: instancias en formato TSPLib en la carpeta **instances**

El código posee las siguientes clases:

- **TSPLibReader.java**: clase que implementa funciones para leer las instancias de TSP en el formato de la librería de instancias TSPLib.
- **TSP.java**: clase que representa la instancia de TSP. Contiene los siguientes métodos:
  - **compute\_tour\_length**: calcula el costo de un tour  $t$
  - **tsp\_check\_tour**: revisa si un tour  $t$  es correcto
  - **random\_tour**: genera un tour aleatorio
  - **greedy\_nearest\_n**: genera un tour utilizando la heurística del vecino más cercano
- **Tour.java**: clase que implementa una solución del TSP. El constructor de clase permite definir si la solución inicial es construida aleatoriamente o utilizando la heurística del vecino más cercano. Contiene las siguientes variables y métodos:
  - **current**: solución del TSP que se representa con un arreglo de enteros de tamaño  $n + 1$ , donde  $n$  son los nodos (ciudades) y la última ciudad del tour corresponde siempre a la primera ciudad. Ejemplo para TSP  $n = 5$ : 243102
  - **cost**: costo del tour de la solución
  - **swap**: método que aplica el movimiento swap a dos nodos en el tour actual
  - **twoOptSwap**: método que aplica el movimiento 2-opt a dos nodos en el tour actual
  - **randomNeighbor**: método que aplica aleatoriamente a dos nodos el movimiento definido en **move\_type** (**TSPMove.TWO\_OPT** o **TSPMove.SWAP**)
- **SimulatedAnnealing.java**: clase que implementa el método de búsqueda de Simulated Annealing
  - **search**: método que inicia la búsqueda de Simulated Annealing comenzando de una solución inicial
  - **terminationCondition**: método que revisa si la condición de término (temperatura mínima o número de evaluaciones) se ha cumplido
  - **getAcceptanceProbability**: método que calcula la probabilidad de aceptar una solución

- **reduceTemperature**: método que reduce el valor de la temperatura siguiendo un esquema de enfriamiento seleccionado
- **Runner.java**: clase que ejecuta Simulated Annealing
- **AlgorithmOptions.java**: clase que permite pasar opciones (parámetros) al algoritmo a través de argumentos

## Parámetros del algoritmo

Los parámetros del algoritmo se pueden cambiar en la clase **AlgorithmOptions.java** o si el algoritmo es ejecutado directamente desde la línea de comando utilizando pasándolos como argumentos

- **filename**: ruta al archivo de la instancia de TSP a resolver, por defecto **instances/burma14.tsp**.  
(**--instance <instance\_path>**)
- **seed**: semilla para el generador de números aleatorios.  
(**--seed <int>**)
- **mh\_move**: variable del tipo **TSPMove** que indica que tipo de movimiento se utilizará para la ejecución de Simulated Annealing. Los valores posibles son **TSPMove.SWAP** y **TSPMove.TWO\_OPT**.  
(**--move [ swap | 2opt ]**)
- **cooling**: variable del tipo **CoolingType** que indica el tipo de esquema de enfriamiento que se utilizará para la ejecución de Simulated Annealing. Los valores posibles son **CoolingType.GEOMETRIC**, **CoolingType.LOG** y **CoolingType.LINEAR**.  
(**--cooling [ linear | geometric | log ]**)
- **alpha**: valor del parámetro alfa para el enfriamiento lineal.  
(**--alpha [0,1]**)
- **t0**: valor de la temperatura inicial.  
(**--t0 [0,DOUBLE\_MAX]**)
- **t\_min**: valor de la temperatura mínima.  
(**--tmin [0,DOUBLE\_MAX]**)
- **max\_evaluations**: número máximo de funciones de evaluación calculadas.  
(**--evaluations <int>**)

## Compilación y ejecución

Si utiliza Eclipse IDE, revise este video

- Para compilar en la consola:  
`javac -d bin/ -sourcepath src/ -cp lib/commons-cli-1.2.jar src/algorithms/*.java`
- Para mostrar la ayuda en la consola:  
`java -cp bin/:lib/commons-cli-1.2.jar algorithms.Runner --help`
- Para ejecutar en la consola:  
`java -cp bin/:lib/commons-cli-1.2.jar algorithms.Runner --instance instances/burma14.tsp`