

Inteligencia Artificial

Tarea Práctica 1

Profesor: Nicolás Rojas Morales

Universidad Técnica Federico Santa María,
Departamento de Informática.

1. Ejecutando un Algoritmo

La primera tarea busca ejercitar la creación, diseño y ejecución de un escenario experimental en su máquina. El objetivo de este escenario sera visualizar el desempeño de un algoritmo (Ant Knapsack) en un set de instancias, tanto en la calidad de las soluciones que puede obtener como en su tiempo de ejecución.

Ant Knapsack (AK) es un algoritmo de hormigas diseñado para la resolución del Multidimensional Knapsack Problem. AK requiere los siguientes argumentos para su ejecución (puede observarlos en Initialize.cpp):

```
InstanciaFile = argv[1]; \\ Path a la instancia
Seed = atoi(argv[2]); \\ Semilla
TotalAnts = atoi(argv[3]); \\ Cantidad de hormigas
TotalEvaluations = atoi(argv[4]); \\ Cantidad de Evaluaciones (Recursos)
alpha = atof(argv[5]); \\ Importancia de la Feromona
beta = atof(argv[6]); \\ Importancia del Conocimiento Heuristico
ph_max = atof(argv[7]); \\ Feromona maxima en la matriz
ph_min = atof(argv[8]); \\ Feromona minima en la matriz
rho = atof(argv[9]); \\ Tasa de Evaporacion de la Feromona
```

Un ejemplo de ejecución de AK:

```
./AK ../instances/mknap1_7.txt 1 20 10000 9 1 50 0.01 0.99
```

considerando el orden de los argumentos y parámetros señalado, AK obtiene como resultado:

0.16327

Esto significa que la calidad de la mejor solución encontrada por AK se encuentra a un 0.16 % de distancia al óptimo global de la instancia MKNAP1.7. El gap porcentual se calcula (observar en Main.cpp):

```
cout << 100*(Opt - BestFitnessFound)/Opt << endl;
```

1.1. Instancias

Para esta tarea, utilizaremos las siguientes instancias benchmark:

```
OR10x500-0.25_8.dat
OR5x500-0.25_7.dat
mknnap1_5.txt
mknnap1_7.txt
OR30x500-0.25_3.dat
gk01.dat
mknnap1_6.txt
sento1.txt
```

Mas detalles pueden ser obtenidos en https://www.researchgate.net/publication/271198281_Benchmark_instances_for_the_Multidimensional_Knapsack_Problem

2. Detalles Entrega

Se requiere crear un script que realice ejecuciones independientes de AK en todas las instancias. Considere:

- Se deben realizar 20 ejecuciones independientes (semillas distintas) por instancia.
- Los valores a los parámetros de AK serán leídos de un archivo de texto llamado params.txt:

```
20 10000 9 1 50 0.01 0.99
```

- Mientras se ejecuta AK, el script debe crear un archivo de salida por instancia, donde se almacene la calidad de la mejor solución obtenida por AK (por semilla).
- Además, el script debe almacenar el tiempo de ejecución de AK (en segundos) en otro archivo de texto.
- Para agilizar la ejecución de los experimentos, el script puede controlar la cantidad de ejecuciones simultáneas de AK con un parámetro (cantidad de hebras disponibles para ejecutar).

- Una vez terminada la ejecución de AK, el script debe obtener métricas de desempeño (mejor obtenido, μ , σ , tiempo promedio de ejecución, mínimo y máximo tiempo de ejecución).
- El script debe generar un archivo que contenga una tabla con las métricas de desempeño (por instancia) en formato \LaTeX .

Además, es requisito entregar un escrito en inglés que contenga:

- Una sección que presente características sobre las instancias utilizadas.
- Explicación del escenario experimental y hardware utilizado. Detallar valores de parámetros utilizados, cantidad de recursos (número de evaluaciones).
- Resultados obtenidos por AK en cada instancia y análisis respectivo.
- El escrito debe ser realizado en \LaTeX .

Evaluación de su script:

- Su script será revisado en una máquina Linux/Unix.
- Considere que su script se ejecutará en un directorio que además contendrá el archivo de parámetros, el directorio AntKnapsack y el directorio de instancias (disponibles en Aula).
- Se sugiere realizar make clean y make antes de ejecutar sus experimentos.

Fecha de Entrega: 31 de Agosto - Informe en Clases - Script en Aula (Nombre_Apellido.zip).