

Task

La tarea se compone de:

- longMin: Distancia mínima conocida por el sistema, se utiliza para podar soluciones que excedan de manera parcial una solución mínima conocida.
- cityToVisit: Es la ciudad a visitar, es la tarea misma
- partialLong: Es la distancia que se ha recorrido hasta el momento (antes de visitar cityToVisit)
- visitedCount: Ciudades visitadas hasta el momento en la ruta
- visited: Arreglo de tamaño N que contiene la secuencia de las ciudades visitadas, siempre parte con 0 y solo se deberían tomar en cuenta los primeros visitedCount elementos del arreglo.

Tipos de mensajes

Mensajes de Worker a Master:

- Tipo TASK_TAG: El worker puede enviar una o mas tareas al master dependiendo del caso:
 - La tarea se subdivide: Envía todos los nodos visitables desde el nodo actual como subtarea
 - La tarea alcanza una solución: Envía una sola tarea con la solución, luego el Master la evalúa para actualizar longMin y bestRouteCabe destacar que cuando el Master recibe una Task la despacha inmediatamente si hay un worker disponible en availableWorkers, de lo contrario la encola en taskBag.
- Tipo AVAILABLE_TAG: Este mensaje se envía cuando el worker esta disponible y quiere recibir una tarea. Cuando llega al Master puede suceder dos cosas:
 - El master envía inmediatamente una tarea que estaba en la TaskBag, o
 - El master encola al worker en availableWorkers

Mensajes de Master a Worker:

- Tipo TASK_TAG: Es el mensaje que envía el Master luego de que el worker aviso que estaba disponible, este mensaje contiene una task.

Lista enlazada doble genérica

Para manejar la lista de tasks y lista de workers disponibles implementamos una estructura de datos genérica. Sobre esta creamos dos métodos convenientes: listEnqueue y listDequeue.

TaskBag

Es una lista que contiene solo elementos de tipo Task, esta lista yace en el proceso Master

AvailableWorkers

Es una lista que contiene Ids (ranks) de procesos que están disponibles y esperando una Task.

Flujo del Master

- Envía a todos los workers la matriz de adyacencia mediante Broadcasting (MPI_Bcast)
- Loop(termina si taskBag esta vacia y hay 3 workers desocupados)
 - Espera un mensaje (TASK_TAG o AVAILABLE_TAG)
 - Si es de tipo TASK_TAG, lo envía a un worker, lo agrega a TaskBag o lo procesa como ruta terminada
 - Si es de tipo AVAILABLE_TAG, envía una task al origen del mensaje o lo encola en AvailableWorkers
- Envía un mensaje FINALIZE_TAG a todos los workers
- Imprime mejor ruta y su costo

Flujo de Esclavo

- Espera una matriz de adyacencia por Broadcasting (MPI_Bcast)
- Loop (indefinido)
 - Espera un mensaje (TASK_TAG o FINALIZE_TAG)
 - Si el mensaje es de tipo TASK_TAG, procesa la tarea adjunta
 - Si la tarea se puede subdividir, lo hace y las envía al Master
 - Si la tarea sobrepasa la longitud mínima conocida, se descarta
 - Si la tarea completa una ruta, se vuelve a enviar al Master
 - Si el mensaje es de tipo FINALIZE_TAG, termina el loop