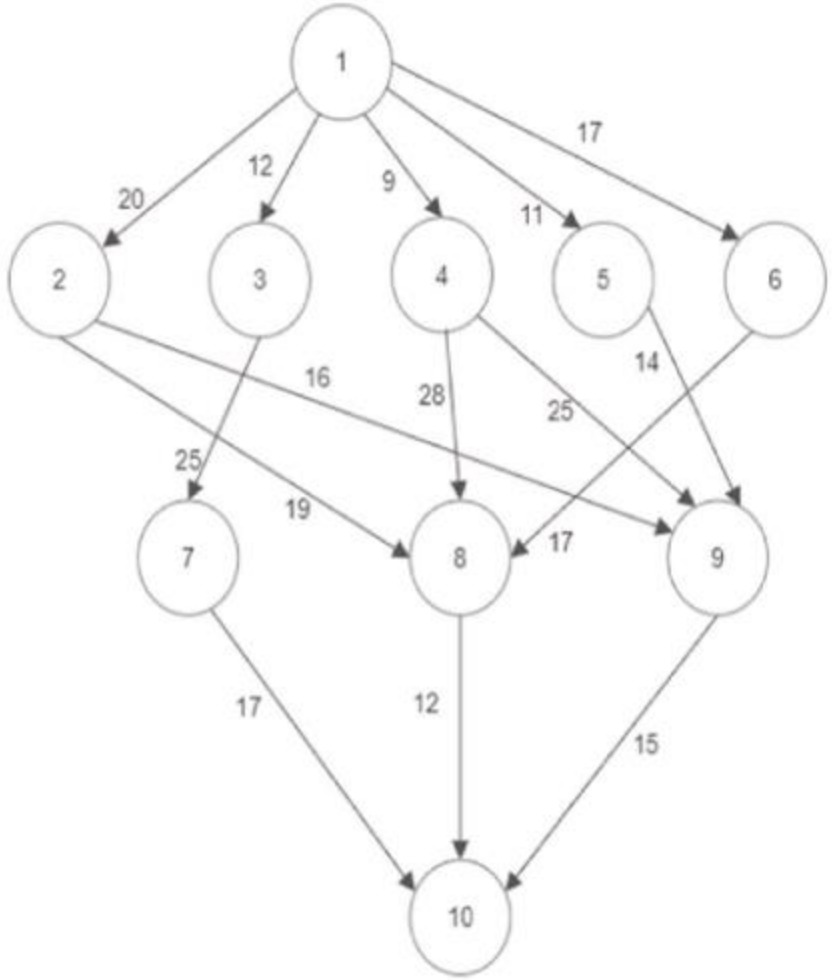
Implementation of
ACO-based algorithm for task scheduling in heterogeneous multiprocessing environments

Problem

- Scheduling in multiprocessing environment
  - GPU, CPU, TPU
  - Different speeds and communication times
- NP-hard



Ant Colony Optimization

An efficient ACO-based algorithm for task scheduling in heterogeneous multiprocessing environments

Jeffrey Elcock *, Nekiesha Edward

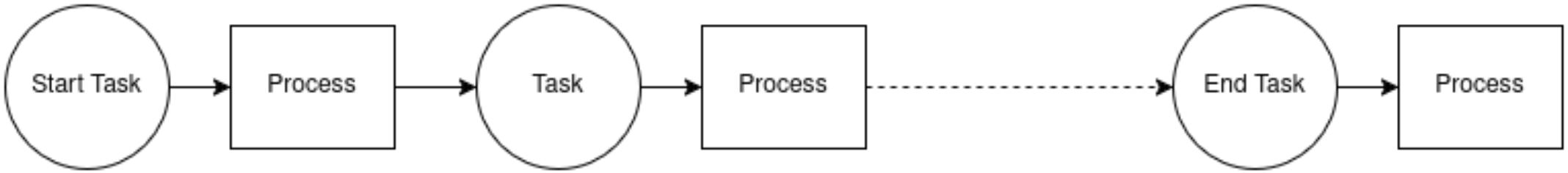
Department of Computer Science, Mathematics and Physics, University of the West Indies, Cave Hill Campus, Bridgetown, Barbados

The ACO Metaheuristic

Set parameters, initialize pheromone trails
while termination condition not satisfied **do**
    Construct Ant Solutions
    Apply Local Optimization (optional)
    Pheromone Update
end while

Fig. 1. The aco metaheuristic.

An ant solution



While there are tasks remaining

- Ant decides the next ready task
    - task where all its dependencies are assigned a process
- Ant decides the process for the task

Pheromone Matrices

- 2 pheromone matrices
- Contains pheromone for edges from Process to Task
- Contains pheromone for edges from Task to Process

Deciding Task and Process

- First iteration, decides at random task and process
- Subsequent iterations
  - Decides task based on STaskRule
  - Decides task based on SProcRule

STaskRule

$$\text{Task Heuristic} = \frac{\text{Phermone[Process,Task]}}{\text{Longest Path from Task to End}}$$

$$\text{Probability Mass Density(Task)} = \frac{\text{Task Heuristic}}{\text{Sum of All Next Tasks Heuristic}}$$

SProcRule

$$\text{Process Heuristic} = \frac{\text{Pheromone[Task,Process]}}{\text{Average Computation Cost of Task} \times \text{Estimate Start Time}}$$

$$\text{Probability Mass Density(Process)} = \frac{\text{Process Heuristic}}{\text{Sum of All Process Heuristics}}$$

Pheromone Update

- Local Pheromone Update: Pick some best ant solutions

$$\text{Gained Pheromone} = \frac{1}{\text{Completed Time}}$$

$$\text{New Pheromone} = (1 - \text{Decay}) \times \text{Old Pheromone} + \text{Gained Pheromone}$$

- Global Pheromone Update: Pick 1 best ant solution

$$\text{New Pheromone} = (1 - \text{Decay}) \times \text{Old Pheromone} + \text{Decay} \times \text{Gained Pheromone}$$

Aging Pheromone

- If the best solution is not changing (Stagnation)
  - Randomize the pheromones in the matrices, so that it is less or equal to the old pheromones
  - All ants begin to pick processes and tasks randomly as if it is the first iteration.

Limitation

- Many parameters
  - Number of ants, number of best ants, number of iterations, process heuristic, task length heuristics, decay parameter
- No guarantee of optimality

Demo