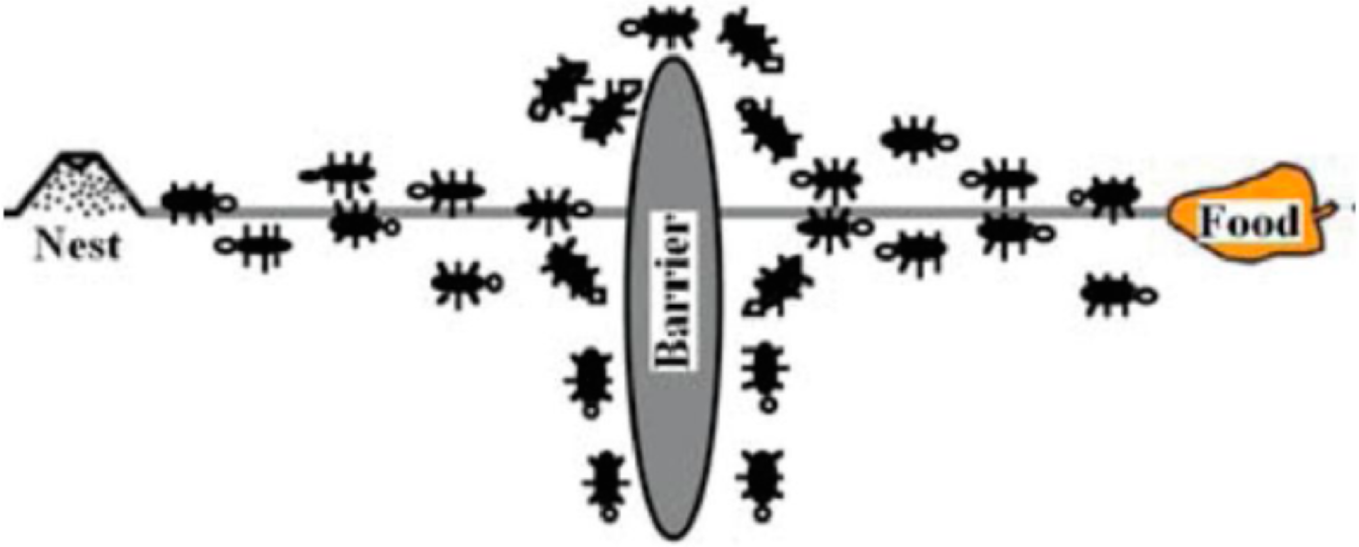
多處理機平行程式設計 2021 fall 作業六說明

題目

1. Ant Algorithm



Psuedo Code for Ant Algorithm 🐜

Serial Version

```
Initialize the pheromone matrix τ for each pair of cities
Place the m ants on n random cities
for t=1 to nc do
    for i=1 to n do
        for k= 1 to m do
            Choose next city j according to the transition rule
    for k = 1 to m do
        Calculate tour distance Lk for ant k
        if an improved tour is found then
            Update T* and L*
    Update the pheromone matrix τ
```

Parallel Version

```
Initialize TGlobal {this data is shared, everything else is private}
parallel region with nColonies threads
    Initialize the pheromone matrix τ for each pair of cities
    Place the m ants on n random cities
    for t=1 to nc do
        for i=1 to n do
            for k=1 to m do
                Choose next city j according to the transition rule
        for k=1 to m do
            Calculate tour distance Lk for ant k
            if an improved tour is found then
                Update T* and L*
            if this is an exchange cycle then
                if L* < LGlobal then
                    ***Critical section***
                        TGlobal= T*
                        LGlobal= L*
                    ***End critical section***
                ***Synchronization barrier***
                T* = TGlobal
        Update the pheromone matrix τ
```

Ant Algorithm 說明 👨‍🏫

An ant k at city i has not visited set of cities S_p then P_{ij} be the probability to visit edge k after edge i.

$$P_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{j\epsilon S_p} \tau_{ij}^{\alpha}\eta_{ij}^{\beta}} & if\ j\epsilon S_p \\ 0 & \end{cases}$$

> 💡 α 及 β 會影響收斂速度！

S_p represents the set of cities which has not been visited yet and to be visited again so that the probability of the ant visiting a city which has already visited becomes 0. Where τ_{ij} is the pheromone content on the edge joining node i to j. η_{ij} represents the heuristic value which is inverse of the distance between the city i to j, which is given by:

$$\eta_{ij} = \frac{1}{d_{ij}}$$

Where d_{ij}, is the distance between the city i to j. α and β represents the dependency of probability on the pheromone content or the heuristic value respectively. Increasing the value of α and β may vary the convergence of ACO.

After solution construction we have to update the pheromone accordingly, as follows;

$$\tau_{ij} \leftarrow (1-\rho).\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

Where ρ is the evaporation rate, m is the number of ants, and $\Delta\tau_{ij}^{k}$ is the quantity of pheromone laid on edge(i, j) by an ant k:

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \textit{if ant k uses edge (i, j) in its tour,} \\ 0 & \textit{otherwise} \end{cases}$$

Where Q is a constant and L_k is the length of the tour constructed by an ant k.

Roulette Wheel Selection 🎯

Printing the best tour 🚶

```
struct {
    int cost;
    int rank;
} loc_data, global_data;

loc_data.cost = Tour_cost(loc_best_tour);
loc_data.rank = my_rank;

MPI_Allreduce(&loc_data, &global_data, 1, MPI_2INT, MPI_MINLOC, comm);

if (global_data.rank == 0) return; /* 0 already has the best tour */
if (my_rank == 0)
    Receive best tour from process global_data.rank;
else if (my_rank == global_data. rank)
    Send best tour to process 0;
```

要求 💡

- 使用 **MPI+OpenMP** 實作，每一台電腦各啟動一個 process，每個 process 再 fork 出 multi-thread
- txt 輸入程式格式： `mpiexec -np $np ./myexe "your_txt"`

Moodle 附件說明 (最佳解) 🏷️

- GR17 is a set of 17 cities, from TSPLIB. The minimal tour has length 2085.
- FRI26 is a set of 26 cities, from TSPLIB. The minimal tour has length 937.
- DANTZIG42 is a set of 42 cities, from TSPLIB. The minimal tour has length 699.
- ATT48 is a set of 48 cities (US state capitals) from TSPLIB. The minimal tour has length **33523** HOT FIX.

參考資源 💩

- 旅行商問題 Traveling salesman problem (https://zh.wikipedia.org/wiki/%E6%97%85%E8%A1%8C%E6%8E%A8%E9%94%80%E5%91%98%E9%97%AE%E9%A2%98)
- Guide for Running an MPI Program per node (https://www.intel.com/content/www/us/en/develop/documentation/mpi-developer-guide-linux/top/running-applications/running-an-mpi-program.html)

如何交作業

1. 把作業交到你的家目錄底下 (`mv yourfile.c ~`) 並使用正確的檔名: `h6_problem1.c` (or `.cpp`) , 想用資料夾稍作整理也可以，但是請確保只有一個 `h6_problem1.c` 。 請勿抄襲。
2. 上傳你的 report (使用 `.md` 或是 `.pdf` 格式，或是包含連結至你的 HackMD 頁面 / GitHub Readme 的文字檔)(in Chinese or English) 至 moodle 並包含：
   - What have you done
   - Analysis on your result
   - Any difficulties?
   - (optional) Feedback to TAs

截止日期： 2021/1/14 23:59:59

Please report any server mis-configuration you found. TAs are new to System/Network administration. We will appreciate your report.

計分方式

- 程式 style 25%
  1. 巢狀結構需用階層式編排。
  2. 適當地使用空白列來區隔功能上無關的程式碼，使你的程式段落分明。
  3. 清楚且詳細的註解。
- 結果正確無誤 20%
  平行程式執行的結果需與循序程式執行的結果一致。
- 效能 30%
- 平行程式的觀察報告 25%
  請針對作業中提到的問題逐一回答，相對應的觀察及分析請寫在報告中，
- **Bonus 加分題 25%**
  每個 thread 都跑一整個蟻巢，此會影響到 critical section 部分，也就是課本介紹到更新的問題，若寫出來會在這個作業額外加 25 分。