

GEE9018 (5095) F'15: Midterm

**TIME: 2015/11/27 13:20 – 16:20**

**Problem01** (50%) The program uses brute-force attempt to find the minimum value of  $f(x, y)$  in the given range  $x = [-5, 10]$  and  $y = [0, 15]$ ,

$$f(x, y) = a(y - bx^2 + cx - r)^2 + s(1 - t)\cos(x) + s$$

where  $a = 1$ ,  $b = 5.1/(4\pi^2)$ ,  $c = 5/\pi$ ,  $r = 6$ ,  $s = 10$ , and  $t = 1/(8\pi)$ . The answer of the minimum value under step =  $10^{-3}$ , about ~225 million iterations, is

$$x = 9.425$$
$$y = 2.475$$

You have the serial version of source code of this program, pb01.cpp

- (a) Parallel the program using OpenMP (25%)
- (b) Parallel the program using Pthreads (25%)

Please complete the program (with thread count), run the program as follow:

```
> ./pb01_omp 2    (thread count = 2)
Max fit = fit(9.425,2.475) = 0.39789
Time = 6.2365 sec
```

```
> ./pb01_ptd 2    (thread count = 2)
Max fit = fit(9.425,2.475) = 0.39789
Time = 4.7877 sec
```

**Problem02** (50%) Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to randomly move particles around in the search space according to simple mathematical formulae over the particle's position and velocity.

The problem PSO trying to solve here is

$$| x^3 - 0.8x^2 - 10000x + 8000 |$$

It is the double `fit(double x) {...}` function in the source code. The search space is  $[-100, 100]$ , and they are specified in `min_pos` and `max_pos` in the source

code.

You have the serial version of source code of this program, pb02.c

- (1) Parallel the program using OpenMP (25%)
- (2) Parallel the program using Pthreads (25%)

Please complete the program (enter thread count), run the program as follow:

```
> ./pb02_omp 2    (thread count = 2)
PSO    solution : fit(-57.469) = 390245.79174
Optimal solution : fit(-57.469) = 390245.79174
Time = 1.4694 sec

> ./pb02_ptd 2    (thread count = 2)
PSO    solution : fit(-57.469) = 390245.79174
Optimal solution : fit(-57.469) = 390245.79174
Time = 0.9337 sec
```

**Problem03** (20%) Please use the task structure in OpenMP to parallelize the source program (pb03.c) that solves a target second-order linear recursive function, expressed as

$$a_{n+3} = 3a_{n+2} - 2a_{n+1} + a_n - 5$$

where  $a_0 = 0$ ,  $a_1 = 1$  and  $a_2 = 2$ .

run the program as follow:

```
>./pb03
0 0 0
1 1 1
2 2 2
3 -4 -4
4 -9 -9
5 1 1
6 3 3
...
```