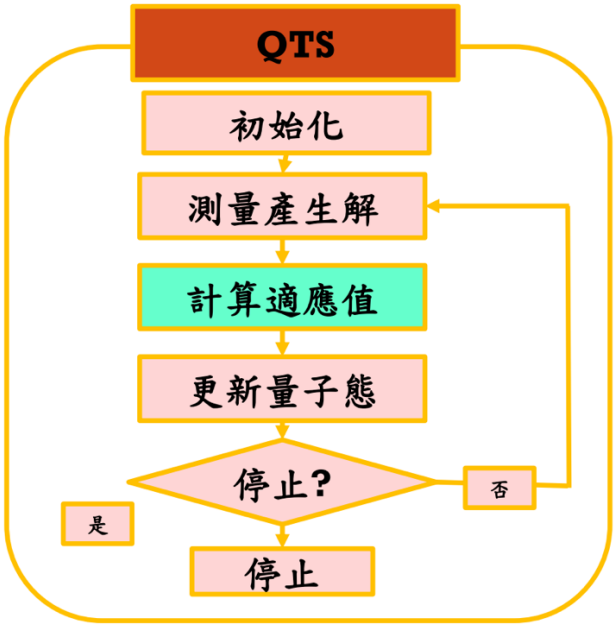


參數設定：

```
EXPERIMENT 50
ITERATION 1000
POPULATION 25
DELTA 0.02
```

初始化：

```
void init()
{
    for (int i = 0; i < ITEM_NUM; i++)
    {
        Q[i] = 0.5;
    }
}
```

測量產生解：

randNum < Q[j] ⇢ 選 ⇢ 1

randNum ≥ Q[j] ⇢ 不選 ⇢ 0

```
void measure()
{
    for (int i = 0; i < POPULATION; i++)
    {
        for (int j = 0; j < ITEM_NUM; j++)
        {
            float randNum = rand() % 101 / (float)100; //0~1

            if (randNum < Q[j])
            {
                chrom[i].gene[j] = 1;
            }
            else
            {
                chrom[i].gene[j] = 0;
            }
        }
    }
}
```

計算適應值：

item	Weight = index/10+1	Value = weight+5
A：gene[0~9]	1	6
B：gene[10~19]	2	7
C：gene[20~29]	3	8
D：gene[30~39]	4	9
E：gene[40~49]	5	10
F：gene[50~59]	6	11
G：gene[60~69]	7	12
H：gene[70~79]	8	13
I：gene[80~89]	9	14
J：gene[90~99]	10	15

```
void fitness()
{
    for (int i = 0; i < POPULATION; i++)
    {
        int weight_ind = 0, value_ind = 0;
        chrom[i].weight = 0;
        chrom[i].value = 0;
        for (int j = 0; j < ITEM_NUM; j++)
        {
            // weight fitness
            weight_ind = j / 10 + 1;
            chrom[i].weight += chrom[i].gene[j] * weight_ind;

            //value fitness
            value_ind = weight_ind + 5;
            chrom[i].value += chrom[i].gene[j] * value_ind;
        }
    }
}
```

更新量子態：

找出 Best,Worst

```
    for (int i = 0; i < POPULATION; i++)
    {
        //find the best value in POPULATION
        if (chrom[i].value >= best && chrom[i].weight <= WEIGHT_LIMIT)
        {
            best = chrom[i].value;
            best_index = i;
        }

        //find the worst value in POPULATION
        int dev = abs(chrom[i].weight - WEIGHT_LIMIT);
        if (dev > worst || (dev == worst && chrom[i].value < chrom[worst_index].value))
        {
            worst = chrom[i].value;
            worst_index = i;
        }
    }
```

更新機率矩陣

```
    //renew the propability in Q[]
    for (int i = 0; i < ITEM_NUM; i++)
    {
        if (chrom[best_index].gene[i] == 1 && chrom[worst_index].gene[i] == 0)
        {
            Q[i] += DELTA;
        }
        else if (chrom[best_index].gene[i] == 0 && chrom[worst_index].gene[i] == 1)
        {
            Q[i] -= DELTA;
        }
        else //chrom[best_index].gene[i] == chrom[worst_index].gene[i]
        {
            //probability will not renew
        }
    }
```

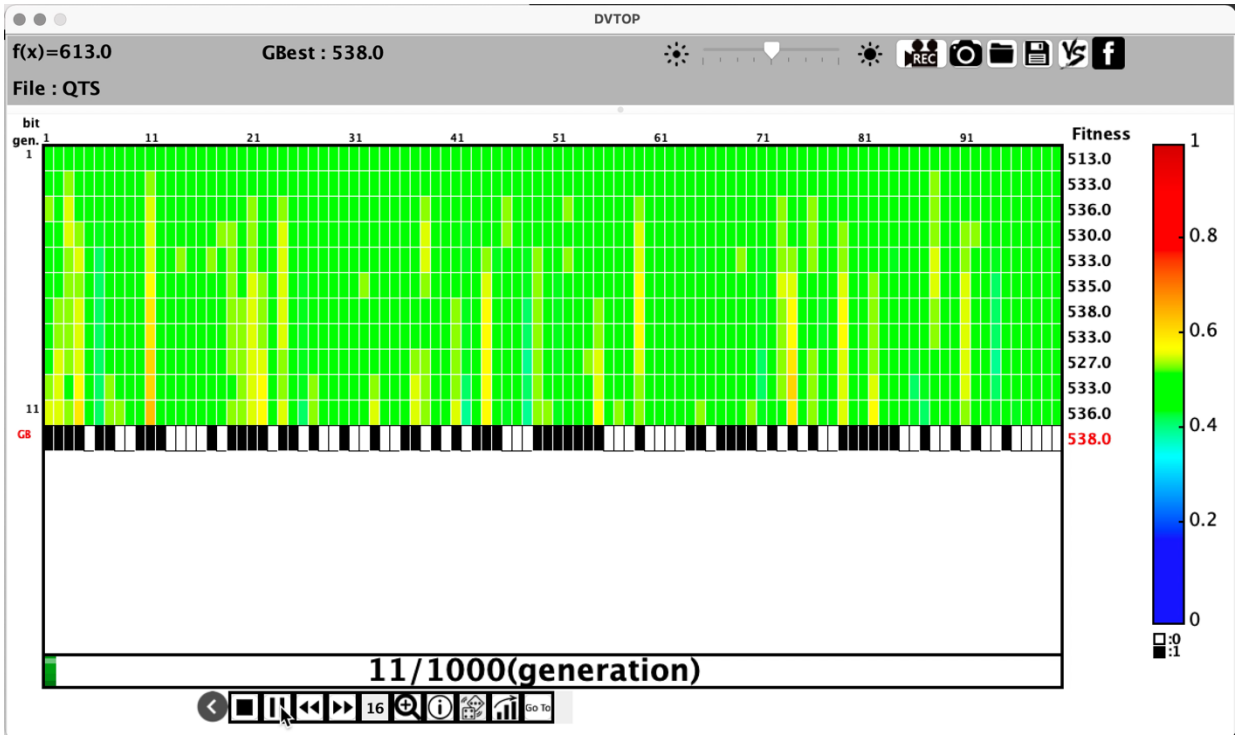
更新全域最佳解

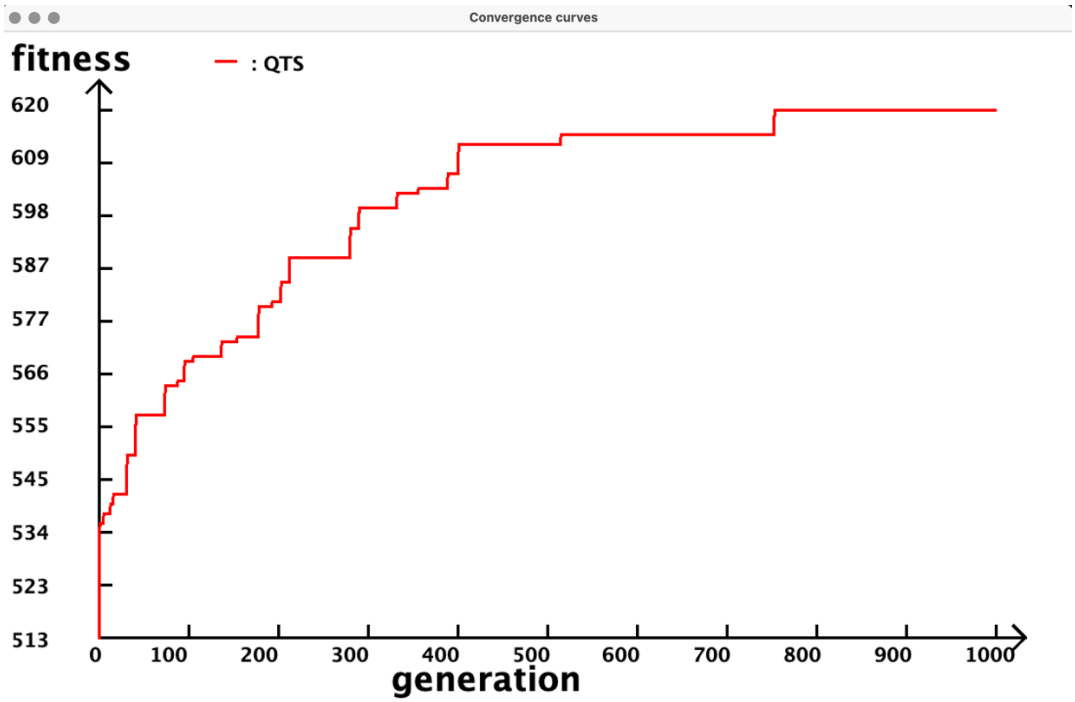
```
    //renew the optimal solution (global best)
    if (chrom[best_index].value > opt_sol.value)
    {
        opt_sol = chrom[best_index];
        opt_index = best_index;
        found = gen; //the generation which find the optimal solution
    }

}
```

（一）實驗 50 次 看平均

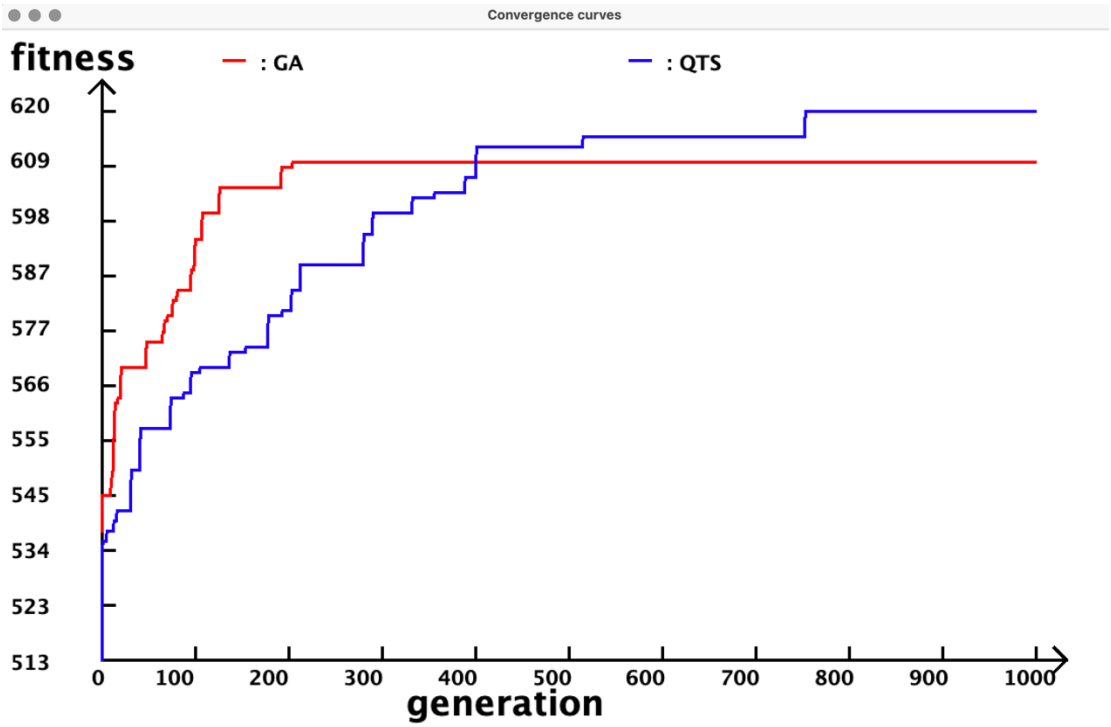
```
Average Found Generation: 729
Average Value: 620
```

（二）DVTOP





GA VS. QTS


Convergence curves
fitness
: GA
: QTS
620
609
598
587
577
566
555
545
534
523
513
0
100
200
300
400
500
600
700
800
900
1000
generation