

## Cockroach Swarm Optimization Algorithm for TSP

Le Cheng<sup>1, a</sup>, Zhibo Wang<sup>1, b</sup>, Song Yanhong<sup>1, c</sup>, Aihua Guo<sup>1, d</sup>

<sup>1</sup>Department of Computer Science and Engineering, Huaian College of Information Technology, Huaian, Jiangsu, China 223003

<sup>a</sup>CL211282@163.com, <sup>b</sup>wzbbyr@163.com, <sup>c</sup>4793173@qq.com, <sup>d</sup>65788846@qq.com

**Keywords:** cockroach swarm optimization algorithm, Traveling Salesman Problem, solution space, particle swarm optimization

**Abstract.** we propose a novel cockroach swarm optimization(CSO) algorithm for Traveling Salesman Problem(TSP) in this paper .In CSO, a series of biological behavior of cockroach are simulated such as grouping living and searching food ,moving-nest, individual equal and so on. For cockroaches crawl and search the optimal solution in the solution space, we assume that the solution which has been searched as the food can split up some new food around solution's position. The experimental results demonstrate that the CSO has better performance than particle swarm optimization in TSP.

### Introduction

Recent years many population-based optimizations are extensively studied,such as ant colony optimization(ACO) [1]and particle swarm optimization(PSO)[2-3].Cockroach has lived over 0.35 billion years,more than 0.3 billion years than dinosaur.Cockroach has bad eyesight and good smell . Social living make them exist until now time. Entomologist discovery that cockroach 's society is equal ,which is different from other social biology such as ant and bee.But they still possess swarm intelligence.When one member of family go out for seaching food,other members could choose following it.Dr. J.halloy 's experiment demonstrates that the probability of cockroach as a tailer is about 60%<sup>[4]</sup>.

By simulating cockroach's behavior of searching food,we propose the cockroach swarm optimization.The CSO possesses simple formula and fully utilizes cockroach swarm's equality and swarm intelligence.To find better solution around the local optimal solution,we use the food- splitting strategy,which enhance the local search capability.Moving-nest make CSO possesses strong global search.The remainder of the paper is organized as follows.Section 2 firstly describes CSO in detail.The overall scheme of CSO for solving the TSP is presented in section 3.Experimental comparisons of CSO are gived in section 4. Finally, some concluding remarks are given in section 5.

### Cockroach Swarm Optimization

#### A. The Step of Cockroach Crawling

To apply CSO to TSP,we define the step of cockroach crawling as  $Step(x,y)$ .In TSP ,  $Step(x,y)$  represents that x-city interchange its place with y-city in a solution.Supposing that the solution is [1,2,3,4,5],the processing of  $Step(2,4)$  is show in Fig.1.

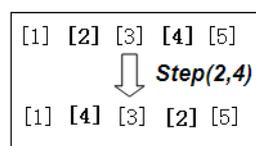


Fig.1 The step of cockroach 's crawling

In fact,we regard [1,2,3,4,5]as a position or coordinate of 6- dimensional space,then a cockroach crawls from A[1,2,3,4,5] to B[4,3,5,1,2] can be show in Fig.2.

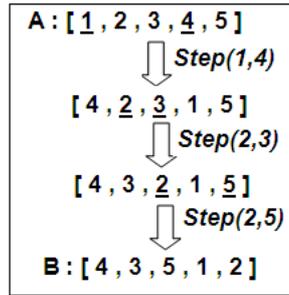


Fig.2 The road of cockroach 's crawling

The road of cockroach crawling from A to B is defined in formula 1:

$$Road(B, A) = B - A = Step(1,4) + Step(2,3) + Step(2,5) \tag{1}$$

**B. The Strategy of Food Splitting**

Supposing that in TSP the number of cities is  $D$  and  $m$  cockroaches form a swarm, we look on the whole solution of TSP as  $D$ -dimensional space and each cockroach, foods, nest of cockroach are looked as a point in the  $D$ -dimensional space, and the  $i$ -th cockroach represents a  $D$ -dimensional vector  $c_i = (c_{i1}, c_{i2} \dots c_{iD})$ . It means that each cockroach is a potential global optimum of the function  $f(x)$  over a given domain  $D$ . Here  $f(x)$  is used for evaluating the cockroach, using the cockroach's positional coordinates as input values. The output value is often named fitness value, and according to the fitness value, the cockroach is updated to move towards the better area by the corresponding operators till the best point is found. In fact, we look on all foods and cockroaches as  $D$ -dimensional vector. The local optimal is defined as  $LOF$  (Local Optimal Food).

To search better solution around the local optimal solution, we introduce a food-splitting strategy. It can be shown in fig.3.

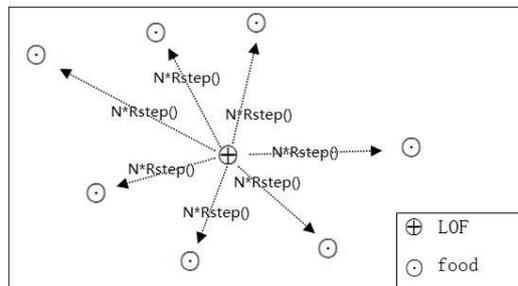


Fig.3 Food-splitting

In fig.3 the  $N^*Rstep(r_x, r_y)$  represents the distance from  $LOF$  to its splitting foods.  $N$  is a positive integer, we can define its value according to the need. If in TSP the city number is  $D$  ( $D$ -TSP),  $r_x$  and  $r_y$  are the stochastic number selected from a uniform distribution in  $[1, D]$ . So  $Rstep()$  represents the one step distance of cockroach crawling and  $N^*Rstep(r_x, r_y)$  is  $N$  step distance in stochastic direction.  $N^*Rstep(r_x, r_y)$  and the strategy of food splitting can be defined by following formula:

$$LOF + N^*Rstep(r_x, r_y) (r_x, r_y \in [1, D]; r_x \neq r_y) \tag{2}$$

**C. Moving-nest**

In CSO the distance is abstract and different from real world's. It can be measured and recorded by the amount of  $Step()$  which cockroach crawls. We obtained an interesting result that in  $D$ -TSP the farthest distance between any two solutions is  $D \cdot Step()$ . In other words, the maximum quantity of steps which is needed by formula 1 is  $D$ . To increase the diversity of solutions, in every searching all cockroaches start to crawl from a new nest (stochastic solution) to the  $LOF$  or a food, and in this process CSO will evaluate the fitness value after every step of cockroach moving, and record the solution better than  $LOF$ . Symbol  $c$  represents cockroach's vector and symbol  $f$  represents food's vector. The process of cockroach crawling to food and  $LOF$  can be shown by following formula:

$$f_i - c_j = \text{Road}(c_j, f_i) \quad (3)$$

$$\text{LOF} - c_j = \text{Road}(c_j, \text{LOF}) \quad (4)$$

$c_i(i=1 \dots m)$  is the  $i$ -th cockroach,  $f_j(j=1 \dots n)$  is the  $j$ -th food. This process can be show in fig.4.

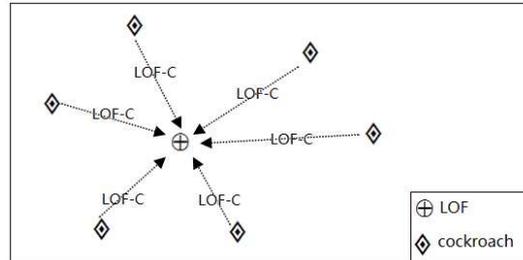


Fig.4 Cockroaches crawl to LOF

### Procedure of CSO Optimizing TSP

When we use CSO algorithm to optimize the TSP problem, firstly, all the cockroaches are initialized with the random solutions, then take the formulas for evolution until the terminal rule coming. The whole procedure of CSO optimizing the TSP problem can be described as follows:

**Step 1:** Initialize the swarm and parameters of CSO; the population size is set as  $m$ ; the food size is set as  $n$  and choose the optimal food as LOF.

**Step 2:** All of the cockroaches crawl to foods

```
FOR(int i=1;i<=n;i++)
FOR(int i=1;i<=m;i++)
{
    {evaluate the solution which is generated when the cockroach crawl to food.
    If(solution is better than LOF)
    {update LOF with the solution}
    }
    The cockroach are set a new solution(new nest)
}
```

**Step 3:** All of the cockroaches crawl to LOF

```
FOR(int i=1;i<=n;i++)
{
    {evaluate the solution which is generated when cockroach crawl to LOF.
    If(solution is better than LOF)
    {update LOF with the solution;}
    }
    The cockroach are set a new solution(new nest)
}
```

**Step 4:** If the terminal rule is satisfied, stop the iteration and output the results, otherwise go to step2.

### Experiment Results and Discussion

To know the CSO well, we use Oliver30 problem to test the CSO algorithm. Oliver30 is a TSP problem with 30 cities. In reference [5], the author use ACO algorithm for Oliver30 and gets the shortest path is 423.74. We get the same result by CSO. The circumstance of experiment is Pentium R 4 -2.93 GHz CPU, 256M RAM, Win XP OS, VC++ 6.0. Table-1 show the convergence processes and results of eight consecutive experiments. We set the population size is 200, the number of food is 20. the max iterations is 1000.

Table-1  
Optimization results of CSO on Oliver30

Index of experiment	Value of LOF
Run1	668.58 659.65 621.75 585.52 550.70 525.89 508.31 494.27 482.21 478.87 473.71 471.94 471.07 458.63 457.79 457.63 454.62 425.65 <b>423.7</b>
Run 2	678.00 636.03 612.11 569.94 553.29 538.60 475.12 427.88 424.69 <b>423.74</b>
Run 3	657.12 622.12 595.03 552.06 536.91 525.50 515.26 506.41 501.69 470.92 466.86 462.13 458.39 438.85 429.74 <b>424.90</b>
Run 4	630.18 587.33 566.03 489.68 467.21 455.82 448.55 428.42 <b>423.74</b>
Run 5	679.95 582.19 533.29 447.09 447.09 432.12 424.74 423.91 <b>423.74</b>
Run 6	601.48 499.14 499.14 476.94 441.13 435.36 430.68 425.99 <b>423.74</b>
Run 7	601.48 499.14 499.14 476.94 441.13 435.36 430.68 425.99 <b>423.74</b>
Run 8	551.77 551.77 526.47 515.70 507.50 490.48 425.94 425.10 423.91 423.91 <b>423.74</b>

In table 1, the “value of LOF” is the result of CSO getting after all cockroach crawl to all foods and LOF.

The result of CSO and PSO is showed in Table 2. The algorithm of PSO comes from reference[6].

Table -2  
Optimization results of CSO and PSO on Oliver30

Algorithm	populati on size	The Length of Oliver30								Averave	Error%
		Run1	Run2	Run3	Run4	Run5	Run6	Run7	Run8		
CSO	200	423.74	423.74	423.74	423.74	423.74	423.74	425.48	423.74	423.96	0.05%
PSO	800	432.66	423.74	425.27	425.48	423.95	423.74	426.31	423.74	424.49	0.28%

In table -2 the averave is the averave length of eight experiments. Error% is according formula 5:

$$Error = \frac{Average - 423.74}{423.74} \times 100\% \quad (5)$$

## Conclusions

This paper proposes a novel cockroaches warm optimization algorithm. The performance of CSO algorithm is evaluated and compared with the well-known PSO algorithm on TSP problem. The experimental results support the claim that the proposed CSO algorithm exhibits better optimization performance in terms of global search ability. Specially, CSO further simplify the updating computation.

## References

- [1] WANG Chuan-xin. Wavelet de-noising double threshold optimization method based on ant colony algorithm[J]. Control and Decision, 2011(1): 121-125
- [2] Cao P, Chen P, Liu S H. Application of improved particle swarm optimization in TSP[J]. Computer Engineering, 2008, 34(11): 217-218
- [3] M. Meissner, M. Schmuker, G. Schneider, “Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training,” Bmc Bioinformatics, England, vol. 7, pp. 125, March 2006.
- [4] Halloy J. Individual discrimination capability and collective decision-making[J]. Journal of Theoretical Biology, 2006, 239: 313-323.
- [5] WU Q ing-Hong. An ant colony algorithm with mutation features[J]. Journal of computer research & development, 1999, 36(10): 1240-1243
- [6] HUANG Lan. Particle Swarm Optimization for Traveling Salesman Problems[J]. Journal of Jilin university (science edition), 41(4): 477-480