

Improved PSO for TNDBDCP

Kai Shaoa, Kang Zhou^{b*}, Yingying Duanc and Wenbo Dongd
*School of Math and Computer, Wuhan Polytechnic University, Wuhan,
430023, China*
*aemail: 892771585@qq.com, bemail: zhoukang_wh@163.com, cemail:
767503653@qq.com, demail: 1623572531@qq.com*

Abstract

Traffic network distribution based on distribution center problem (TNDBDCP) is put forward, which can not be solved by traditional algorithms. In order to solve TNDBDCP, Particle Swarm Optimization (PSO) is put forward based on the idea of global and feasible searching. In the PSO, introducing check function ensures the feasibility of particle swarm, which is designed based on the neighborhood search method which only uses one loop program to meet the requirement of time constraint and radialized network structure; more reasonable fitness function is designed based on the relationship between spanning tree and its cotree; improvements for updating the position of particle make the current position of particle be beneficial to being close to the best position of particle in its neighborhood.

Keywords: PSO; TNDBDCP; Spanning tree

Introduction

Distribution center is an important part of logistics system [1], which is a circulation node of collected goods, picking and delivery [2]. Cargo allocation plan based on distribution center will directly affect transportation cost, operation cost and logistics service quality. Therefore, research on traffic network distribution based on distribution center problem (TNDBDCP) has important theory and real meaning [3, 4].

Particle Swarm Optimization (PSO) [5] is an adaptive, heuristic, collective type and probabilistic type global evolutionary algorithm [6, 7], which is put forward in 1995 by Kennedy and Eberhart. The idea of PSO comes from artificial life and evolutionary computation theory. By the study of flock flying, Reynolds found that bird just tracks its neighbors whose number is limited; however the final overall result is that the flock is in a center. That is to say, the complex global behavior is caused by the interactions among simple rules. PSO is produced getting inspiration from this model, and used to solve optimization problem. An ordinary algorithm can not solve TNDBDCP. Subsequently, PSO is paid wide attention by scholars worldwide. PSO is used in the paper to solve

TNDBDCP.

TNDBDCP

In urban traffic network $G=(V(G), E(G), W(G))$, $V(G)=\{v_i|i=1,2,\dots,m\}$ is node set at which logistics company must arrive every day, $E(G)=\{e_i|i=1,2,\dots,n\}$ is the shortest path set, $W(G)=\{w_i|e_i \in E(G)\}$ is the shortest time set. Let node v_1 be distribution center, and time from distribution center to an arbitrary node is less than p_0 . TNDBDCP is to find a traffic layout plan from node v_1 to all nodes in urban traffic network G to satisfy that time from node v_1 to other node is less than p_0 , whose sum of time reaches the minimum value.

Traffic layout whose sum of time reaches the minimum value must form a spanning tree of urban traffic network G . If the spanning tree satisfies that time from node v_1 to other node is less than p_0 , then the spanning tree is a feasible traffic layout. The feasible traffic layout whose sum of time reaches the minimum value is the optimal solution of TNDBDCP. Let $Layout(G)$ be a set of all traffic layouts of urban traffic network G , and $F-layout(G)$ be a set of all feasible traffic layouts of urban traffic network G .

Particle Swarm Optimization

The standard Particle Swarm Optimization (PSO) does not ensure the feasibility of the particle position and global optimality of solution, so PSO should be improved to solve TNDBDCP.

(1)The Standard PSO

Consider particle swarm containing S particles. For $i=1,2, \dots, S$, let $X_i=(x_{i1},x_{i2}, \dots,x_{in})$ be the position of the i th particle, $V_i=(v_{i1},v_{i2},\dots,v_{in})$ be the flight speed of the i th particle, and $P_i=(p_{i1},p_{i2},\dots,p_{in})$ be the individual extreme point of the i th particle. Let $P_g=(p_{g1},p_{g2},\dots,p_{gn})$ be global extreme point. Then for $d=1,2,\dots,n, i=1,2,\dots, S$, formulas updating flight speed are as follows:

$$v_{id}^{(t+1)}=w \times v_{id}^{(t)}+c_1 \times r_1 \times (p_{id}-x_{id}^{(t)})+c_2 \times r_2 \times (p_{gd}-x_{id}^{(t)})$$

(1)

$$v_{\min} \leq v_{id}^{(t+1)} \leq v_{\max}$$

(2)

where w is inertia factor (the greater w the stronger global searching ability of PSO, in general, $w=0.9$ at the start of PSO; $w=0.4$ at the end of PSO); c_1 and c_2 are learning factor (c_1 and c_2 are constants in interval $[1,2]$); r_1 and r_2 are random numbers in interval $[0,1]$.

Formula updating position of continuous PSO is as follows:

$$x_{id}^{(t+1)}=x_{id}^{(t)}+v_{id}^{(t+1)}$$

(3)

Formulas updating position of discrete PSO are as follows:

$$S(v_{id}^{(t+1)})=1/(1+\exp(-v_{id}^{(t+1)}))$$

(4)

$$\text{If } r < S(v_{id}^{(t+1)}) \text{ then } x_{id}^{(t+1)}=1, \text{ else } x_{id}^{(t+1)}=0$$

(5)

where r is random number in interval $[0,1]$.

(2) Improved PSO

For TNDBDCP where distribution center is node v_1 and time constraint is p_0 , the calculation process of improved PSO is as follows:

Step 1 Initialize particle swarm.

a. Let v_{\max} be maximum speed and v_{\min} be minimum speed; let $V_i=(v_{i1},v_{i2},\dots,v_{in})=0$ be initial speed.

b. Generate randomly S particles $X_i=(x_{i1},x_{i2},\dots,x_{in})$, which meet the requirements of feasibility.

c. For each particle, compute adaptive value.

d. Compute individual extreme point $P_i=X_i$.

e. Compute global extreme point P_g from all the individual extreme points.

f. Let $k=1$.

Step 2 Update particle swarm.

a. Update flight speed of particle swarm according to formula (1) and formula (2).

b. Update position of particle swarm according to flight speed of particle swarm and formulas updating position.

Steps 3 If the termination condition of improved PSO is satisfied, then putout the optimal solution of TNDBDCP, end.

Step 4 Update extreme points of particle swarm.

a. For each particle, compute adaptive value.

b. For each particle, update individual extreme point by compared to the original individual extreme point.

c. Update global extreme point by compared to all new individual extreme points.

Step 5 Let $k=k+1$, return to Step 2.

Key Problems for PSO and Solving Strategies

Applying properly discrete PSO to optimization problem needs to solve some problems, for example encoding and its feasibility, the design of fitness function, the improvement of formula of updating particle swarm, the choice of parameters and termination conditions, and so on. Solving strategies for these problems are as follows.

(1) Update Flight Speed

In order to meet that all particles maintain coordination and consistency in flight speed, if flight speed is beyond interval $[v_{\min},v_{\max}]$, then flight speed takes

boundary value. So the process of updating the flight speed of the i th particle is as follows:

- Step 1 Let $d=1$.
- Step 2 Generate random number r_1 and random number r_2 in interval $[0,1]$.
- Step 3 Compute $v_{id}^{(t+1)}$ according to formula (1).
- Step 4 If $v_{id}^{(t+1)} < v_{\min}$, then $v_{id}^{(t+1)} = v_{\min}$; otherwise, if $v_{id}^{(t+1)} > v_{\max}$, then $v_{id}^{(t+1)} = v_{\max}$.
- Step 5 If $d=n$, then putout $(v_{i1}^{(t+1)}, v_{i2}^{(t+1)}, \dots, v_{in}^{(t+1)})$, end; otherwise, let $d=d+1$, return to Step 2.

(2) Improved Formulas Updating Position

New position encoding of particle updated by using formula (4) and formula (5) can't meet the requirements of feasibility, therefore, formula (5) needs to be improved according to the characteristics of TNDBDCP.

According to formula (4), function $S(v_{id}^{(t+1)})$ is a monotone increasing function of $v_{id}^{(t+1)}$; according to formula (1), the larger the value of $v_{id}^{(t)}$ is, the larger the value of $v_{id}^{(t+1)}$ is; and the greater the chance of $p_{id}=1$ and $p_{gd}=1$ is, the larger the value of $v_{id}^{(t+1)}$ is. Therefore, function $S(v_{id}^{(t+1)})$ is consistent with the value of $v_{id}^{(t)}$ and the chance of $p_{id}=1$ and $p_{gd}=1$. According to formula (5), if function $S(v_{id}^{(t+1)})$ is large enough to meet the formula $r < S(v_{id}^{(t+1)})$, where r is random number in interval $[0,1]$, then let $x_{id}^{(t+1)}=1$ can improve position of particle. In any circle of a spanning tree, position encoding component value of only one edge is 1, and position encoding component value of other edge is 0, therefore, in the circle, we can choose position encoding component whose value is 1 by comparison function $S(v_{id}^{(t+1)})$. In the meantime, in order to increase the randomness of updating position, we have function $S(v_{id}^{(t+1)})-r$ instead of function $S(v_{id}^{(t+1)})$. For position encoding $y=(x_1, x_2, \dots, x_n)$ ($T(y) \in \text{Layout}(G)$), let set M be the index set for a circle in $T(y) \cup \{e_k | x_k=1\}$. For index set M , improved formula updating position of formula (5) is as follows:

$$(6) \quad \begin{aligned} x_{id}^{(t+1)} &= 1, \text{ for } S(v_{id}^{(t+1)})-r_d = \text{Max} (S(v_{ij}^{(t+1)})-r_j | j \in M) \\ x_{id}^{(t+1)} &= 0, \text{ for } d \notin M \end{aligned}$$

where r_j is random number in interval $[0,1]$.

The process of updating the position of particle $y=(x_1, x_2, \dots, x_n)$ is as follows:

Step 1 Compute position encoding component $x_{i(1)}, x_{i(2)}, \dots, x_{i(n-m+1)}$ whose value is 1. Let $k=1, y_0=y$.

Step 2 Let $x_{i(k)}=0$ in position encoding y_{k-1} , and obtain position encoding y' . And compute the index set M of y' .

Step 3 According to formula (7), for index set M obtain position encoding y_k .

Step 4 If $k=n-m+1$, then go to Step 5; otherwise, let $k=k+1$, go to Step 2.

Step 5 If $\text{Check}(y_{n-m+1})=0$, then let $k=1, y_0=y$, go to Step 2; otherwise, let $y=y_{n-m+1}$, y is the new position of particle, end.

Computational method of index set M of position encoding y' is as follows:

Step 1 Compute adjacency matrix $G=(g_{ij})$ of position encoding y' .

Step 2 If $\exists i (1 \leq i \leq m), g_{ij}=1 \wedge g_{ik}=0 (k \neq j)$, then go to Step 3; otherwise, go to

Step 4.

Step 3 Let $g_{ij}=0$, $g_{ji}=0$, go to Step 2.

Step 4 Obtain new adjacency matrix G . Compute edge set E of matrix G . And compute edge index set M of edge set E , end.

(3) Parameter Selection

Parameters of improved PSO includes the flight speed limit v_{\min} and v_{\max} , inertia factor w , learning factor c_1 and learning factor c_2 .

According to the characteristics of function $S(x)=1/(1+\exp(-x))$, when $-5 \leq v_{id}^{(t+1)} \leq 5$, $0.0067 \leq S(v_{id}^{(t+1)}) \leq 0.9933$. Therefore, let $v_{\min}=-5$, $v_{\max}=5$.

We have that $-5 \leq v_{id}^{(t)} \leq 5$, $-1 \leq (p_{id}-x_{id}^{(t)}) \leq 1$, $-1 \leq (p_{gd}-x_{id}^{(t)}) \leq 1$, and large inertia factor is beneficial to jump out the optimal state to achieve the global optimal solution. Therefore, let $w \in [0.6, 0.85]$, $c_1 \in [1, 2]$, $c_2 \in [1, 2]$.

Simulation Experiment

In urban traffic network $G=(V(G), E(G), W(G))$ which is shown in Fig.1, where $V(G)=\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$, $E(G)=\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}\}$, $W(G)=\{9, 3, 6, 5, 7, 5, 12, 4, 4, 15, 3, 5, 7, 6, 9, 8, 4, 12, 5, 13\}$, we solve TNDBDCP where distribution center is node v_1 and time constraint is 20.

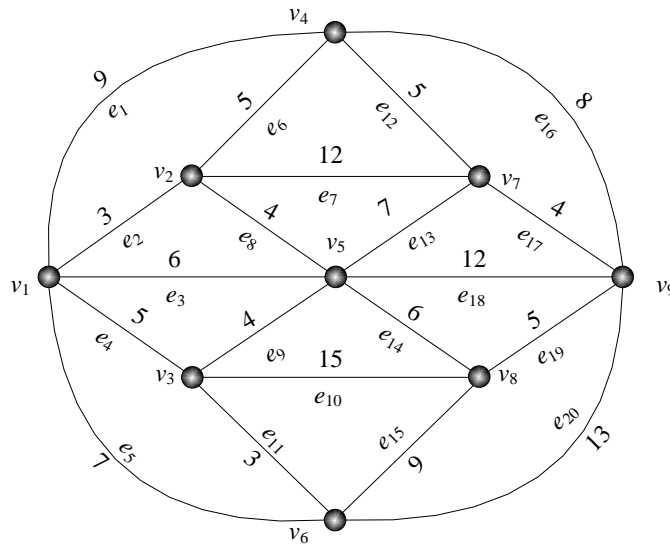


Fig.1. An example of TNDBDCP

Parameters of improved PSO to solve TNDBDCP are as follows: particle swarm scale $S=20$, iteration number $D=100$, encoding length $N=20$, inertia factor $W=0.6$, learning factor $C1=2$, learning factor $C2=2$, minimum speed $V_{\min}=-5$, maximum speed $V_{\max}=5$.

Improved PSO on TNDBDCP is realized by C language. Program is run 10 times, Optimal solution is obtained as follows: (2, 6, 8, 9,11,12,14,17). And value of optimal solution is 34.

Acknowledgement

Kang Zhou is corresponding author. The work is supported by the National Natural Science Foundation of China (61179032, 11301405).

References

- [1] Y. Wang, B. Zhao, M. Y. Liu, Y. Y. Ma. Method for Solving Double Distribution Centers Locating Problem with Capacity Constraint [J]. Logistics Management, 2011 34 11-14.
- [2] H. J. Shen, X. L. Han. Study on Logistics Distribution Center Location Based on Tabu Search [J]. Logistics Management, 2011 34 39-41.
- [3] K. Zhou, J. Chen. Simulation DNA algorithm of set covering problem[J]. Applied Mathematics and Information Sciences, 2014 8 139-144.
- [4] G. Duan, L. Chen, Y. Z. Li. Bi-level Programming Model and Algorithm on Location of Logistics Distribution Center [J]. Journal of Transportation Systems Engineering and Information Technology, 2011 11 126-129.
- [5] S. L. Li, N. Chen, X. Y. Zhao. Planning of Feeder Bus to the Urban Rail Transit Based on Particle Swarm Optimization [J]. Journal of WuHan University of Technology (Transportation Science & Engineering), 2010 34780-783.
- [6] Y. Wu, C. M. Ye, H. M. Ma, M. Y. Xia. Parallel particle swarm optimization algorithm for vehicle routing problems with time windows [J]. Computer Engineering and Applications, 2007 43 223-226.
- [7] L. P. Zhang, H. J. Yu, D. Z. Chen, S. X. Hu. Analysis and Improvement of Particle Swarm Optimization Algorithm [J]. Information and Control, 2004 33 513-517.