# Research on Team Orienteering Problem with Dynamic Travel Times

Jin Li
Department of Management Scicence and Engineering
Zhejiang Gongshang University, Hangzhou, P.R.China
Email: jinli@mail.zjgsu.edu.cn

***Abstract*—In the team orienteering problem (TOP) a set of locations is given, each with a score. The goal is to determine a fixed number of routes, limited in length, that visit some locations and maximize the sum of the collected scores. The team orienteering problem is often used as a starting point for modeling many combinatorial optimization problems. This paper studies the dynamic team orienteering problem considering the travel cost varying with times and visiting time constraints. After a mixed integer programming model is proposed, a novel optimal dynamic labeling algorithm is designed based on the idea of network planning and dynamic programming. Finally, a numerical example is presented to show the validity and feasibility of this algorithm.**

***Index Terms*— team orienteering problem, time-dependent network, travel time, optimal algorithm**

## I. INTRODUCTION

In the orienteering problem (OP) a set of $n$ locations $i$ is given, each with a score $s_i$. The starting point (location1) and the end point (location n) are fixed. The time $t_{ij}$ needed to travel from location $i$ to $j$ be known for all location pairs. A given $T$ limits the time to visit locations. The goal of the OP is to determine a single route, limited by $T$, in order to maximize the total collected score. Each location can be visited at most once. The Team Orienteering Problem (TOP) is an OP that maximizes that total collected score of $m$ routes, each limited to $T$. The team orienteering problem (TOP) first appeared in Butt and Cavalier [1] under the name of the Multiple Tour Maximum Collection Problem. The term TOP, first introduced in Chao et al. [2], comes from a sporting activity: team orienteering. A team consists of several members who all begin at the same starting point. Each member tries to collect as many reward points as possible within a certain time before reaching the finishing point. Available points can be awarded only once. Chao et al. [2] also created a set of instances, used nowadays as standard benchmark instances for the TOP.

The TOP is an extension to multiple-vehicle of the orienteering problem (OP), also known as the selective traveling salesman problem (STSP). The TOP is also a generalization of vehicle routing problems (VRPs) where only a subset of customers can be serviced. Several researchers propose solving the OP with exact algorithms based on a branch-and-bound [3,4] and branch-and-cut approach [5,6]. With the branch-and-cut procedure, instances of up to 500 locations can be optimally solved [5], provided sufficient calculation time is available. Since the OP is NP-hard [11], obviously these exact algorithms are extremely time consuming so most research has focused on heuristic approaches[7-9].

As an extension of the OP, the TOP clearly appears to be NP-hard. Many TOP real applications are described in the literature: the sport game of orenteering[10], the home fuel delivery problem[11], athlete recruiting from high schools [12], technician routing and scheduling problem[13], TSPs with profits [14], etc. Butt et al. [15] present an exact algorithm based on column generation to solve the TOP. They deal with problems up to 100 locations, provided that the number of selected locations in each tour remains small. Boussier et al. [16] propose a branch-and-price approach to solve problems with up to 100 locations. Only problems in which the number of possible locations in a route is low, namely, up to about 15 per route, can be solved in less than 2h of calculation. The first published TOP heuristic was developed by Chao et al [17]. Tang and Miller-Hooks [13] developed a tabu search heuristic embedded in an adaptive memory procedure. Archetti et al. [18] came up with two variants of a tabu search heuristic and a slow and fast Variable Neighbourhood Search (VNS) algorithm. Ke et al. [19] developed four variants of an ant colony optimization approach for the TOP. Vansteenwegen et al. [20, 21] implemented a Guided Local Search (GLS) and a Skewed Variable Neighbourhood Search (SVNS) algorithm. Souffriau et al. [22] designed a path relinking metaheuristic approach. Bouly et al. [23] proposed a simple hybrid genetic algorithm.

However, most of the previous research for TOP in the literature seldom takes into account changes to the network over time. Clearly, the route between two locations does not depend only on the distance traveled, but on many other time dependent properties of the network such as congestion levels, incident location, and construction zone on certain road segments, which would change the travel cost on that segment. Therefore, based on above literature, this paper will consider the time-dependent team orienteering problem (TDTOP for short) which meets the needs to real-world problems. Here, the transportation cost of traveling (time cost in our terminology) varies with time and the travel cost from one location to another depends on the start time. So


doi:10.4304/jsw.7.2.249-255

decision-makers could choose the right route, locations and departure time according to their own situation.

The remaining of this paper is structured as follows: the time-dependent team orienteering problem is described in Section 2. A mixed integer programming model for TDTOP is proposed in Section 3. And an optimal dynamic labeling algorithm is designed in Section 4. Moreover, in Section 5, a numerical example is introduced to show the validity and feasibility of the algorithm. Finally, the concluding remarks and further research are included in Section 6.

## II. PROBLEM DESCRIPTION

This paper will study the team orienteering problem on transportation network with a given source and destination node, in which the travel cost is dependent on time and primarily on the start time on the edge. It is assumed that time is discredited into small units (such as 1 hour or less 10 minutes). In the TDTOP, each location has a score. It consists in visiting some of the locations in order to maximize the total collected score of multiple tours within a given time budget. And each location can be visited at most once. We don't consider the phenomenon of return and round in traveling road. The road segments may not satisfy the "first-in-first out" property. On this basis, taking into account the application in real world, this paper will consider time-dependent travel cost, location stay time and total transportation time constraint to study the team orienteering problem. Therefore, the time-dependent team orienteering problem is actually a multi-routes-selection one in the time-dependent network with the determination of the departure time from each node on the selected route.

Given a transportation network $G=(V,E)$, where $V=\{v_i \mid i=1,2,\cdots,n\}$ is the node set, $E=\{e_{ij} \mid v_i, v_j \in V\}$ is the edge set, if $v_i$ is adjacent with $v_j$, then there is one edge $e_{ij}$ between them. Time and spatial dimensions are used to represent the selected routes, where time is discrete time unit and space is expressed as $V=\{v_i \mid i\in[1,|V|]\}$. So, every selected route is a list constitute in elements $(v_i, a_i, b_i)$, the selected route $r_d$ is presented as the following.

$$P_d(v_1,v_n)=\{(v_1,a_1,b_1),(v_{d_1},a_{d_1},b_{d_1}),\cdots,(v_{d_w},a_{d_w},b_{d_w}),(v_n,a_n,b_n)\} \quad (1)$$

Where $v_1$ is the source node, $v_n$ is the destination node, $a_i$ is the arrival time at node $v_i$, $b_i$ is the departure time from node $v_i$, and $d_i$ ( $i=1,2,\cdots,w$ ) is the node subscript on route $r_d$. Let $v_{d_0}=v_1$, $v_{d_{w+1}}=v_n$, thus we have the following results.

(1) $v_{d_i}\in V\setminus\{v_1,v_n\}$, $i=1,2,\cdots,w$;

(2) $1=a_{d_0}\le b_{d_0}\le\cdots\le a_{d_{w+1}}\le b_{d_{w+1}}\le T$;

(3) $b_{d_i}-a_{d_i}=vt_{d_i}$, $i=0,1,\cdots,w+1$;

(4) $a_{d_{i+1}}-b_{d_i}=t_{d_i d_{i+1}}(b_{d_i})$, $i=0,1,\cdots,w$.

Where $T$ denote the total time budget of the route, $vt_i$ is the visitng time on node $v_i$, and $t_{ij}(t)$ is the travel time on edge $e_{ij}$ when the entry time is $t$.

The route $r_d$ is expressed as $r_d=\{v_1,v_{d_1},\cdots,v_{d_w},v_n\}$.

The collected score of route $r_d$ is donted as $U_d=\sum_{i=1}^{w}s_{d_i}$, where $s_{d_i}$ is the score of node $v_i$.

## III. MATHEMATICAL MODEL

### A. Notations

$G=(V,E)$: Transportation network;

$P(i)$: Set of predecessor nodes of node $v_i$;

$S(i)$: Set of successor nodes of node $v_i$;

$R$: Set of multiple routes, $R=\{r_d \mid d=1,2,\cdots,m\}$;

$\bar{T}$: Set of unvisited nodes;

$T$: The total time budget of the route;

$s_i$: Score of node $v_i$;

$vt_i$: Visiting time on node $v_i$;

$a_i$: Arrival time at node $v_i$;

$b_i$: Departure time from node $v_i$;

$t_{ij}(t)$: Travel time on edge $e_{ij}$ when the entry time is $t$;

$x_{ijd}(t)$: If edge $e_{ij}$ on route $r_d$ is entered at time $t$, then $x_{ijd}(t)$ =1, else $x_{ijd}(t)$ =0.

### B. Model

Choose node $v_1$ as the source point and node $v_n$ as the destination point. So we establish the following mixed integer programming model.

$$\max \sum_{t=1}^{T}\sum_{d=1}^{m}\sum_{i=2}^{n-1}\sum_{j\in S(i)} s_i x_{ijd}(t) \quad (2)$$

$$s.t. \sum_{t=1}^{T}\sum_{d=1}^{m}\sum_{j\in S(1)} x_{1jd}(t)=\sum_{t=1}^{T}\sum_{d=1}^{m}\sum_{i\in P(n)} x_{ind}(t)=m \quad (3)$$

$$\sum_{t=1}^{T}\sum_{i\in P(k)} x_{ikd}(t)=\sum_{t=1}^{T}\sum_{j\in S(k)} x_{kjd}(t),\ \forall k=2,\cdots,n-1,\ \forall d=1,\cdots,m \quad (4)$$

$$\sum_{t=1}^{T}\sum_{d=1}^{m}\sum_{j\in S(i)} x_{ijd}(t)\le 1,\ \forall i=2,\cdots,n-1 \quad (5)$$

$$\sum_{t=1}^{T}\sum_{d=1}^{m}\sum_{i\in P(j)} (t+t_{ij}(t))x_{ijd}(t)=a_j,\ \forall j=2,\cdots,n \quad (6)$$

$$\sum_{t=1}^{T}\sum_{d=1}^{m}\sum_{j\in S(i)} tx_{ijd}(t) = a_i + vt_i\ ,\quad \forall\ i = 1,\cdots,n-1 \tag{7}$$

$$a_1 = 1\ ,\ a_n \le T \tag{8}$$

$$a_i > 0\ ,\ \forall\ i = 1,\cdots,n \tag{9}$$

$$x_{ijd}(t) = 0,1\ ,\quad \forall\ e_{ij} \in E\ ,\quad \forall\ d = 1,\cdots,m\ ,\quad \forall\ t = 1,\cdots,T \tag{10}$$

The objective of the TDTOP is to maximize the total collected score, as shown in (2). In this formulation, constraint (3) and (4) are flow-conservation constraints. Constraint (5) ensures that every location is visited at most once. Constraint (6) and (7) guarantees that if one edge is visited in a given tour, the arrival time of the edge following node is the sum of the preceding arrival time, visiting time and the edge travel time. Constraint (8) is the start time and latest finish time constraint. Constraint (9) and (10) are the variables constraint.

## IV. Optimal Algorithm

### A. Definition and Proposition

**Definition 1:** $(i, a_j, k)$ is the label of node $v_j$, where $i$ is the subscript of node $v_i$ followed by node $v_j$, and $k$ shows which stage it is.

**Definition 2:** $L_k$ is the set of nodes belong to the $k$ stage, for $\forall v_j \in L_k$, $P_{k-1}(j)$ is the set of predecessor nodes belong to the $k-1$ stage, $P_{k-1}(j) = L_{k-1} \cap P(j)$.

**Definition 3:** $U_k(j,t)$ represents the optimal collected score from sourcing node to the node $v_j$ at time $t$ on stage $k$, $v_j \in L_k$. So, $U_k(j,t) = \max_{i\in T(i,t)} (U_{k-1}(i,a_i) + s_i)$, where $T(i,t) = \{i \mid t = b_i + t_{ij}(b_i), i \in P_{k-1}(j)\}$.

**Definition 4:** $U(i)$ represents the optimal collected score from sourcing nodes $v_1$ to node $v_i$.

According to the definition, we have the following proposition, the proof of which is straightforward, and hence omitted.

**Proposition 1:** $U(i) = \max\{U_k(i,a_i) \mid k = 1,2,\cdots,K\}$, where $K$ is the maximum of stages division.

**Proposition 2:** The sub-route of the maximum colleted score route may not be optimal route from the source node $v_1$ to node $v_i$.

### B. Optimal Algorithm and Steps

According to the constraints in the mathematical model and above proposition, this paper will study an optimization problem without loop. Considering the travel time is time-dependent, a novel dynamic node labeling algorithm is presented based on the idea of network planning and dynamic programming. The basic idea of the algorithm is : for a given time-dependent network, firstly, the stages are divided according to the number of edges on the route. Then calculating the collected score to every node and labeling using strategy space iteration method for uncertainty multi-stage decision problem. Here, $Q_k(x_{ij}(t))$ is the decision set, $S_k(a_i)$ is state set, and state variables satisfy the constraints (6)-(9). $U_k(Q_k, S_k)$ is the total collected score funciton. So, the iteration is carrired out to the last step. Finally, we get the maximum total collected score to the destination node. According to the label, the optimal route is selected by reverse back. The algorithm is finished until every route is selected. The solution of the algorithm is optimal based on the optimization theory of dynamic programming.

Optimal algorithm is presented in the following:

**Step 1 (Initialization):**
Given the score $s_i$ and visiting time $vt_i$ on node $v_i$, $i = 2,\cdots,n-1$. Let $s_1 = s_n = 0$, $vt_1 = vt_n = 0$, $d = 1$, $\overline{V} = V$;

**Step 2 (Stages division):**
According to the arcs on the route, the time dependent network $G = (\overline{V}, E)$ is divided into $K$ stages; $v_1$ is the sourcing node, let $a_1 = 1$, $k = 0$, $U_0(1,1) = 0$;

**Step 3 (Calculation of arrival time on stage $k$):**
For each node $v_j$ in $L_k$, we find the $P_{k-1}(j)$ which is the predecessor nodes set of $v_j$; for $v_i \in P_{k-1}(j)$, calculating $a_j = a_i + vt_i + t_{ij}(b_i)$, if $a_j > T$, then $U_k(j, a_j) = 0$, else go to step 4.

**Step 4 (Calculation of the collected score and labeling on stage $k$):**
Calculating the total collected score to node $v_j$ at $a_j$, $U_k(j,t) = \max_{i\in T(i,t)} (U_{k-1}(i,a_i) + s_i)$, labeling $(i, a_j, k)$;

**Step 5 (Judgment of the iteration on stage $k$):**
Let $L_k = L_k \setminus \{v_j\}$, if $L_k = \phi$, then the iteration is finished on stage $k$; if $k = K$, then go to step 6, else $k = k+1$, go to step 3;

**Step 6 (Calculation of the total collected score):**
Calculating $U(n) = \max_k U_k(n, a_n)$;

**Step7 (Backward the maximum collected score route):**
According to the label, reverse deduction and find out the maximum collected score route: $P_d(v_1, v_n) = \{(v_1, a_1, b_1), (v_{d_1}, a_{d_1}, b_{d_1}), \cdots, (v_{d_w}, a_{d_w}, b_{d_w}), (v_n, a_n, b_n)\}$, the visiting route is $r_d = \{v_1, v_{d_1}, \cdots, v_{d_w}, v_n\}$;

**Step 8 (Judgment of the iteration in route set $R$):**
Update $\overline{V} = \overline{V} \setminus \{v_{d_1}, \cdots, v_{d_w}\}$, if $\overline{V} = \{v_1, v_n\}$, then the iteration is finished, else if $d = m$, then the iteration is finished, else $d = d+1$,go to step 2.

## V. Numerical Example

The efficiency and feasibility of algorithm would be demonstrated by the following numerical example in this section. Given a TDTOP directed graph, where there are 20 nodes and 189 edges, node 1 is the source vertex, node

20 is the destination vertex, and there are 18 valid vertices. To simplify the problem, we only consider the one-way travel. The time unit is assumed to be 6 minutes. We consider two routes selection, and the duration of the trip is 10 hours(e.g. from 9:00 am to 5:00 pm). So, the time budget is T=80 minutes. The graph parameters are set as follows: The average travel times on edges are generated with random integer on the interval [3, 20]; The matrix of average travel times is shown in table 1, where the blank spaces represent both nodes can't reach each other; The travel times on edges at each time are selected with random integer of the fluctuation range $\alpha$ =30% to average travel times; The average visiting times are generated with random integer on the interval [5,15] as shown in table 2, where the average visiting times for node 1 and node 20 are set 0; The scores for every node are generated with random integer on the interval [1,10] as shown in table 3, where the scores for node 1 and node 2 are set 0.

TABLE I.
THE MATRIX OF AVERAGE TRAVEL TIMES

| Nodes | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 5 | 10 | 10 | 5 | 11 | 4 | 8 | 15 | 20 | 17 | 6 | 12 | 16 | 13 | 14 | 8 | 5 | 15 | |
| 2 | | | 8 | 17 | 14 | 15 | 12 | 14 | 11 | 5 | 3 | 16 | 3 | 9 | 18 | 14 | 8 | 15 | 13 | 13 |
| 3 | | | | 6 | 10 | 18 | 10 | 15 | 17 | 3 | 13 | 15 | 4 | 12 | 9 | 18 | 18 | 16 | 16 | 9 |
| 4 | | | | | 8 | 12 | 11 | 17 | 4 | 14 | 11 | 7 | 4 | 3 | 7 | 12 | 16 | 6 | 6 | 7 |
| 5 | | | | | | 6 | 5 | 11 | 14 | 8 | 6 | 19 | 11 | 6 | 13 | 20 | 15 | 14 | 3 | 9 |
| 6 | | | | | | | 9 | 10 | 15 | 11 | 12 | 8 | 9 | 12 | 17 | 4 | 15 | 3 | 3 | 18 |
| 7 | | | | | | | | 7 | 9 | 4 | 5 | 10 | 12 | 18 | 19 | 10 | 5 | 18 | 14 | 16 |
| 8 | | | | | | | | | 7 | 8 | 17 | 4 | 10 | 7 | 15 | 16 | 16 | 5 | 4 | 5 |
| 9 | | | | | | | | | | 6 | 10 | 18 | 17 | 3 | 5 | 18 | 9 | 14 | 20 | 9 |
| 10 | | | | | | | | | | | 3 | 19 | 11 | 3 | 20 | 9 | 11 | 7 | 9 | 10 |
| 11 | | | | | | | | | | | | 3 | 19 | 6 | 17 | 20 | 9 | 6 | 4 | 16 |
| 12 | | | | | | | | | | | | | 11 | 6 | 8 | 12 | 6 | 9 | 17 | 16 |
| 13 | | | | | | | | | | | | | | 19 | 16 | 19 | 13 | 13 | 14 | 20 |
| 14 | | | | | | | | | | | | | | | 7 | 11 | 20 | 5 | 9 | 19 |
| 15 | | | | | | | | | | | | | | | | 10 | 12 | 8 | 9 | 11 |
| 16 | | | | | | | | | | | | | | | | | 18 | 9 | 4 | 8 |
| 17 | | | | | | | | | | | | | | | | | | 19 | 10 | 10 |
| 18 | | | | | | | | | | | | | | | | | | | 19 | 5 |
| 19 | | | | | | | | | | | | | | | | | | | | 5 |
| 20 | | | | | | | | | | | | | | | | | | | | |

TABLE II.
THE AVERAGE VISITING TIMES FOR EVERY NODE

| Nodes | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $vt_i$ | 0 | 8 | 10 | 8 | 12 | 15 | 8 | 6 | 7 | 14 |
| Nodes | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| $vt_i$ | 13 | 5 | 8 | 6 | 9 | 5 | 5 | 8 | 9 | 0 |

TABLE III.
THE SCORES FOR EVERY NODE

| Nodes | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $s_i$ | 0 | 6 | 5 | 3 | 4 | 1 | 3 | 4 | 10 | 7 |
| Nodes | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| $s_i$ | 5 | 5 | 10 | 6 | 5 | 9 | 6 | 7 | 10 | 0 |

TABLE IV.
STAGES DIVISION WHEN $d = 1$

| Stages | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Node Set | {1} | {2-19} | {3-20} | {4-20} | {5-20} | {6-20} | {7-20} | {8-20} | {9-20} | {10-20} |
| Stages | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| Node Set | {11-20} | {12-20} | {13-20} | {14-20} | {15-20} | {16-20} | {17-20} | {18-20} | {19,20} | {20} |

According to the dynamic node labeling algorithm, a simulation program is developed with MATLAB 7.0 and conducted on a server with 1.6 GHZ CUP, 1GB RAM and MS Windows XP OS. The optimal routes for TDTOP are designed as shown in the following.

Given $s_1 = s_{20} = 0$, $vt_1 = vt_{20} = 0$, $m$=2, $R = \{r_d \mid d = 1,2\}$, $\overline{V} = V = \{i \mid i = 2,\cdots,19\}$, $a_1 = 1$;

When $d = 1$, the stages are divided as shown in table 4. According to the algorithm iteration steps, the route is: $P_1(1,20)$ ={(1,1,1),(7,3,11),(8,15,21),(9,25,32), (14,35,41), (16,54,59),(19,62,71),(20,74,74)}. The route travel time is 74 minutes and the total collected score is 42.

When $d = 2$, the unvisited node set is $\overline{V} = \{i \mid i = 2,\cdots, 19\} \setminus \{7,8,9,14,16,19\}$ and the stages are divided as shown in table 5. So, the route is $P_2(1,20)$ ={(1,1,1), (2,6,14), (11,17,30),(12,33,38),(13,48,56),(18,66,74),(20,79,79)}. The route travel time is 79 minutes and the total collected score is 33.

To illustrate the influence of time-dependent travel time, the average values run 100 times with $\alpha$ =30% in time-dependent network are compared with the values ($\alpha$ =0) in static network as shown in table 6. From the table, we can obtain the following conclusion.

- The routes designed with $\alpha$ =30% in time-dependent network vary with changing travel time. It indicates that the results of static network can't be applied to the time-dependent network.
- The average route cumulative travel time in time-dependent network with $\alpha$ =30% decreases by 0.59% compared with the correspongding value in static network. That indicates dynamic labeling algorithm can improve the route travel time, but it is not significant.
- The average route cumulative collected score in time-dependent network with $\alpha$ =30% increased by 5.86% over the results in static network. This shows that dynamic labeling algorithm can optimize the route according to time-dependent travel time and improve the route collected score significantly.

TABLE V.
STAGES DIVISION WHEN $d = 2$

| Stages | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Node Set | {1} | {2-6,10-13, 15,17,18} | {3-6,10-13, 15,17,18,20} | {4-6,10-13, 15,17,18,20} | {5,6,10-13, 15,17,18,20} | {6,10-13,15, 17,18,20} | {10-13,15, 17,18,20} |
| Stages | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Node Set | {11-13,15, 1718,20} | {12,13,15, 17,18,20} | {13,15,17, 18,20} | {15,17,18,20} | {17,18,20} | {18,20} | {20} |

TABLE VI.
COMPARISON OF RESULTS WITH STATIC NETWORK AND TIME-DEPENDENT NETWORK

| | Static network | | | Time-dependent network with $\alpha$ =30% | | |
|---|---|---|---|---|---|---|
| | Routes | Travel time | Collected score | Routes | Average travel time | Average collected score |
| d=1 | 1-2-9-14 -16-19-20 | 75 | 41 | Variation | 77.44 | 42.72 |
| d=2 | 1-7-8-12 -13-18-20 | 80 | 29 | Variation | 76.64 | 31.38 |
| Cumulative value | | 155 | 70 | | 154.08 | 74.1 |

Sensitivity analysis is conducted furtherly. Let $\alpha$ =10%，30%，50%，70%，90% respectively, which represents the fluctuation range of travel time in the time-dependent network. For every $\alpha$, we obtain the average values of 100 times simulation. The results are shown in figure 1 and figure 2. Figure 1 represents the average route travel time of different $\alpha$ value. Figure 2 shows the average route collected score of different $\alpha$ value.

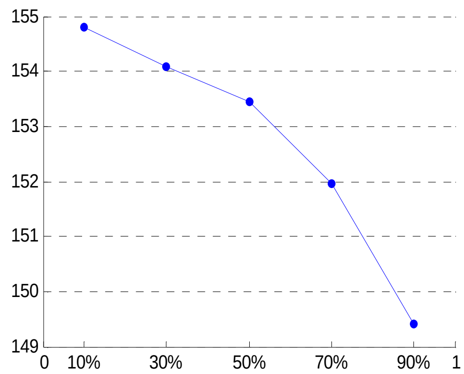


Figure 1. Average route trave time of different $\alpha$ value.

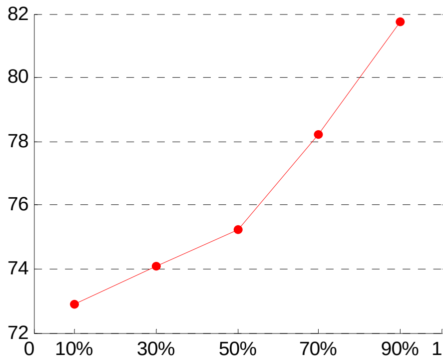


Figure 2. Average route collected score of different $\alpha$ value.

From figure 1 and figure 2, we can have the following conclusion.

- As the increase of $\alpha$ value, the average route travel time decreases, which indicates that dynamic labeling algorithm can improve route travel time. It is worth noting that when $\alpha \leq 50\%$, the reduction rate of $\alpha$ =30%, 50% is 0.46% and 0.41% respecitvely, which means the improvement is not significant, and when $\alpha > 50\%$, the reduction rate of $\alpha$ =70%, 90% is 0.96% and 1.68% respectively, which menas the improvement is effective.
- As the increase of $\alpha$ value, the average route collected score ascendes, which indicates that dynamic labeling algorithm can increase the route collected score. When $\alpha \leq 50\%$, the increase rate of $\alpha$ =30%, 50% is 1.67% and 1.55% respectively, which indicates the improvement is small, but when $\alpha > 50\%$, the increase rate of $\alpha$ =70%, 90% is 3.95% and 4.54% respectively, which shows the improvement is significant.

Based on above analysis, it shows that the dynamic labeling algorithm is effective to solve the time-dependent team orienteering problem. This algorithm can reduce the route travel time and increase the route collected score at the same time, especially when $\alpha > 50\%$, the algorithm is very effective to optimize the visiting route.

## VI. Conclusion

The team orienteering problem is often used as a starting point for modeling many combinatorial optimization problems. Study on the time-dependent team orienteering problem is rarely performed at present, especially the travel cost varying with time, which makes it rather difficult to solve the problem. Based on previous literatures, a time-dependent team orienteering problem considering the time-varying travel cost and location visiting time is studied. A corresponding mixed integer programming model is presented and an optimal dynamic labeling algorithm is also designed based on the idea of network planning and dynamic programming. Then the validity and feasibility of the algorithm is demonstrated by a numerical example. The study of this problem is beneficial for decision-makers to choose the right route, locations and departure time according to their own situation in limited time budget. Further study will consider more realistic factors such as the time windows and capacity constraints of locations.

## Acknowledgment

The author wish to thank the reviewers for their valuable comments. This work was supported in part by National Natural Science Foundation of China (Grant No. 70432001,71071035), Ministry of Education, Humanities and Social Sciences project of China (Grant No. 10YJCZH217) and Zhejiang Provincial Natural Science Foundation of China (Gant NO. Y7100556).

## References

[1] S. Butt, T. Cavalier, “A heuristic for the multiple tour maximum collection problem,” *Comput. Oper. Res.* vol.21, pp.101–111, 1994.
[2] I.M. Chao, B. Golden, E. Wasil, “The team orienteering problem,” *Eur. J. Oper. Res*, vol. 88, pp.464—474, 1996.
[3] G. Laporte, S. Martello, “The selective travelling salesman problem,” *Discrete Applied Mathematics*, vol.26, pp.193–207, 1990.
[4] R. Ramesh, Y. Yoon, M. Karwan, “An optimal algorithm for the orienteering tour problem,” *ORSA Journal on Computing*, vol.4, pp.155–65, 1992.
[5] M. Fischetti, J. Salazar, P. Toth, “Solving the orienteering problem through branch-and-cut,” *INFORMS Journal on Computing*, vol.10, pp.133–48, 1998.
[6] M. Gendreau, G. Laporte, F. Semet, “A branch-and-cut algorithm for the undirected selective travelling salesman problem,” *Networks*,vol.32, pp.263–73, 1998.
[7] Y. Liang, S. Kulturel-Konak, A. Smith, “Meta heuristics for the orienteering problem,” In: *Proceedings of the 2002 congress on evolutionary computation, Honolulu, Hawai*, pp.384–389, 2002.
[8] M. Tasgetiren, “A genetic algorithm with an adaptive penalty function for the orienteering problem,” *Journal of Economic and Social Research*, vol.4-2, pp.1–26, 2001.
[9] Q. Wang, X. Sun, B. Golden, J. Jia, “Using artificial neural networks to solve the orienteering problem,” *Annals of Operations Research*, vol.61, pp.111–120, 1995.

[10] I.M. Chao, B. Golden, E. Wasil, "A fast and effective heuristic for the orienteering problem," *European Journal of Operational Research*, vol.88-3, pp.475–489, 1996.
[11] B. Golden, L. Levy, R. Vohra, "The orienteering problem," *Naval Research Logistics*, vol.34, pp.307–18, 1987.
[12] S. Butt, T. Cavalier, "A heuristic for the multiple tour maximum collection problems," *Computers and Operations Research*, vol.21, pp.101–11, 1994.
[13] H. Tang, E. Miller-Hooks, "A tabu search heuristic for the team orienteering problem," *Comput. Oper. Res.* vol. 32, pp.1379–1407, 2005.
[14] D. Feillet, P. Dejax, M. Gendreau, "Traveling salesman problems with profits," *Transp. Sci*. vol.39-2, pp.188—205, 2005.
[15] S. Butt, D. Ryan, "An optimal solution procedure for the multiple tour maximum collection problem using column generation," *Computer and Operations Research*, vol.26, pp.427—41, 1999.
[16] S. Boussier, D. Feillet, M. Gendreau, "An exact algorithm for the team orienteering problem," *4OR*. Vol.5, pp.211—30, 2007.
[17] I.Chao, B. Golden, , E. Wasil, "Theory and methodology—the team orienteering problem," *European Journal of Operational Research*, vol.88, pp.464—474, 1996.
[18] C. Archetti, A. Hertz, M. Speranza, "Metaheuristics for the team orienteering problem," *Journal of Heuristics*, vol.13, pp.49—76, 2007.
[19] L. Ke, C. Archetti, Z. Feng, "Ants can solve the team orienteering problem," *Computers & Industrial Engineering*, vol.54, pp.648—65, 2008.
[20] P. Vansteenwegen, W. Souffriau, et al. "Metaheuristics for tourist trip planning," Metaheuristics in the Service Industry. In: *Lecture notes in economics and mathematical systems*, vol. 624. Berlin: Springer, to appear (ISBN: 978-3-642-00938-9) , 2009.
[21] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, D. Van Oudheusden, "A guided local search metaheuristic for the team orienteering problem," *European Journal of Operational Research*, vol.196-1, pp.118—27, 2009.
[22] W. Souffriau, P. Vansteenwegen, et al. "A path relinking approach for the team orienteering problem," *Computers & Operations Research*, vol.37-11, pp.1853—1859, 2010.
[23] H. Bouly, D.C. Dang, A. Moukrim, "A memetic algorithm for the team orienteering problem," *4OR-QJ Oper. Res*.vol. 8, pp.49—70, 2010.

**Jin Li** received the B.S. and M.S. degrees from Nanchang University, Nanchang, Jiangxi Province, P.R.China in 2004 and 2007 respectively, and the Ph.D. degree in management science and engineering from Fudan University, Shanghai, P.R.China, in 2010. Currently he is an assistant professor in the Department of Management Science and Engineering, School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, Zhejiang Province, P.R.China. His research interests include logistics and supply chain management, system modeling and simulation, network optimization, and emergency management.