Heuristic and exact solution strategies for the Team Orienteering Problem

Miguel Tamayo Jaramillo
Camila Mejía Quintero
Tutor: Juan Carlos Rivera Agudelo

Progress Presentation

Research Practise 2

Universidad EAFIT, Medellín
Colombia

April 27, 2016

Team Orienteering Problem (TOP)

The team orienteering problem is a generalization of the orienteering problem (OP), where *m* teams or vehicles are available to visit *n* nodes and the goal is to determine *m* routes, without exceeding given thresholds, that maximize the total collected prize. No node can be visited more than once by one or several routes and there is the possibility of not visiting all nodes [Chao et al., 1996].

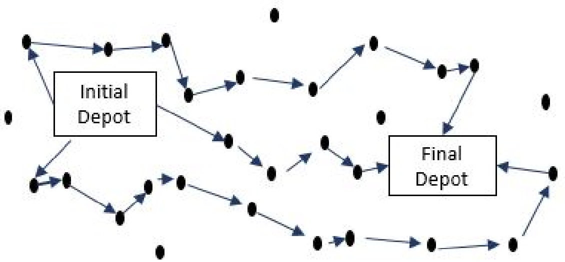

Figure: TOP Example

Objectives

- To compare exact solution approaches for TOP based on constraint programming (CP) and mixed integer linear programming (MILP) by using CPLEX.
- To propose a heuristic algorithm based on the hybridization of mathematical programming formulations and adaptive large neighborhood search heuristics (ALNS).

MILP Models

1. Model based on flow models for vehicle routing problem, where $x_{ij} = 1$ if the arc from node i to node j is crossed.
2. Model based on Model 1, but a matrix y_i is added to it, where $y_i = 1$ if node i is visited.
3. Model based on [Rivera, 2014], where $w_{ij} = 1$ if node i is visited before node j.
4. Model based on replenishment arcs [Mak and Boland, 2000, Rivera et al., 2015], where x_{ij} is used and x'_{ij} represents one route composed by m routes.

MILP Matrix Examples

For instance, take the routes 1-2-5-6 and 1-3-6, then the matrix explained above are. For x'_{ij} the route would be 1-2-5-6-1-3-6.

$$x_{ij} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \qquad w_{ij} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$y_i = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad x'_{ij} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Constraint Programming Model (1/2)

$$
\begin{aligned}
n &: \text{number of nodes} \\
V &: \text{set of all nodes} \\
V' &: \text{set of required nodes} V \setminus \{0, n-1\} \\
V^1 &: V \setminus \{0\} \\
V^2 &: V \setminus \{n-1\} \\
p_j &: \text{profit of node } j \\
K &: \text{set of all vehicles} \\
x_i^k = j &: \text{if vehicle } k \text{ visits node } j \text{ after node } i \\
y_i = 1 &: \text{if node } i \text{ is visited}
\end{aligned}
$$

Constraint Programming Model (2/2)

The mathematical model is:

$$\max z = \sum_{j \in V'} p_j \cdot y_j \tag{1}$$

$$\left(\sum_{i \in V^2} \sum_{k \in K} x_i^k = j \right) = y_j \qquad \forall j \in V' \tag{2}$$

$$\left(\sum_{i \in V^2} x_i^k = n + 1 \right) = 1 \qquad \forall k \in K \tag{3}$$

$$\left(\sum_{j \in V^1} x_0^k = j \right) = 1 \qquad \forall k \in K \tag{4}$$

$$\sum_{i \in V^2} d_{i x_i^k} \leq L_{max} \qquad \forall k \in K \tag{5}$$

$$x_i^k \in Z^+ \qquad \forall i \in V^2, k \in K \tag{6}$$

$$y_i \in \{0, 1\} \qquad \forall i \in V' \tag{7}$$

Comparing Models

Table: Comparison of model results for some instances

Set[1]	**Nodes**	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**CP**	**Best**[2]
1.4.8	32	**45**	**45**	**45**	**45**	**45**	**45**
2.3.5	21	**120**	**120**	**120**	**120**	**120**	**120**
4.2.1	100	81	43	**206**	**206**	98	**206**
6.2.4	64	132	114	**192**	**192**	168	**192**

Table: Comparison of model computational time(s) for some instances

Set[1]	**Nodes**	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**CP**	**Best**[2]
1.4.8	32	79.4	233.7	0.7	**0.5**	9.8	**0.0**
2.3.5	21	7.6	4.6	**0.5**	1.1	4.5	**0.0**
4.2.1	100	600.0	600.0	**28.6**	299.2	600.0	**0.0**
6.2.4	64	600.0	600.0	309.0	**15.9**	600.0	**0.0**

[1] The first number represents the data set, which changes the distance between nodes and its profits. The second one represents the number of vehicles used and the last one represents the file chose, which changes the time limit.

[2] Taken from [Boussier et al., 2007].

Problems

- Data preprocessing.
- Propose a model adapted to constraint programming.
- Lack of tutorials of CPLEX.
- Write CP model on CPLEX language.

Future Work

- Combine CPLEX with Matlab or C++.
- Design the matheuristic based on ALNS.
- Analyze the effect of the instance parameters on the complexity.

Questions?

References I

Boussier, S., Feillet, D., and Gendreau, M. (2007).
An exact algorithm for team orienteering problems.
4OR, 5(3):211–230.

Chao, I.-M., Golden, B. L., and Wasil, E. A. (1996).
The team orienteering problem.
European Journal of Operational Research, 88(3):464–474.

Mak, V. and Boland, N. (2000).
Heuristic approaches to the asymmetric travelling salesman problem with replenishment arcs.
International Transactions in Operational Research, 7(4-5):431 – 447.

References II

Rivera, J. C. (2014).
Logistic Optimization in Disaster Response Operations.
PhD thesis, Université de Technologie de Troyes.

Rivera, J. C., Afsar, H. M., and Prins, C. (2015).
A multistart iterated local search for the multitrip cumulative capacitated vehicle routing problem.
Computational Optimization and Applications, 61(1):159–187.