Heuristic and exact solution strategies for the Team Orienteering Problem

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

Final Presentation
Research Practise 2
Universidad EAFIT, Medellín
Colombia

June 3, 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].

Example

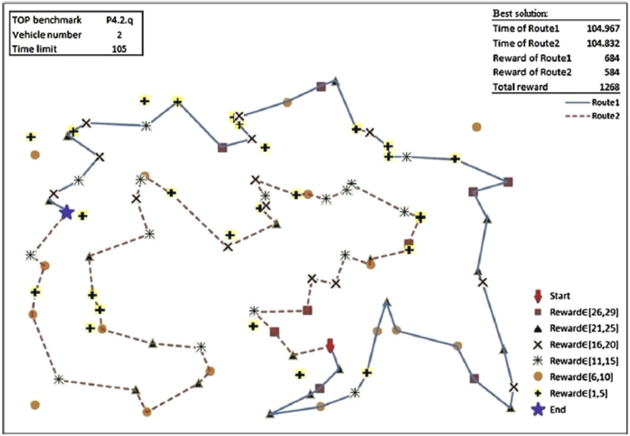

Figure: A TOP problem (instance p4.2.17 and a solution with total reward of 1268). Taken from [Kim et al., 2013] p. 3066.

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 matheuristic algorithm based on the hybridization of mathematical programming formulations and a large neighborhood search heuristic (LNS).

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] and [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.

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

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

Constraint Programming Model I

Parameters, sets and decision variables:

n : number of nodes

m : number of vehicles

H : set of all nodes

H^1 : $H \setminus H^s$

H^s : set of initial nodes

H^f : set of final nodes

H' : set of required nodes $H \setminus (H^s \cup H^f)$

K' : set of available vehicles K and the additional dummy vehicle

a_i : antecedent node

s_i : subsequent node

t_i : team which visits i

L_i : traversed distance from initial depot to i

p_i : profit of node i

L_{max} : maximal length tour

d_{ij} : distance between nodes i and j

Constraint Programming Model II

The mathematical model is:

$$\max Z = \sum_{j \in H'} p_j \cdot (1 - (t_j = m + 1))$$

$$\begin{aligned}
t_{i-1} &= i && \forall\ i \in K' \\
t_{n+m+i-2} &= i && \forall\ i \in K' \\
t_i &= t_{a_i} && \forall\ i \in H \\
t_i &= t_{s_i} && \forall\ i \in H \\
a_{i-1} &= n + m + i - 2 && \forall\ i \in K' \\
s_{n+m+i-2} &= i - 1 && \forall\ i \in K' \\
a_i &\neq i && \forall\ i \in H
\end{aligned}$$

Constraint Programming Model III

$$
\begin{aligned}
a_{s_i} &= i && \forall\ i \in H \\
a_i &\neq a_j && \forall\ i,j \in H,\ \ i \neq j \\
L_{i-1} &= 0 && \forall\ i \in K' \\
L_i = (L_{a_i} + d_{a_i i}) \cdot (1 - (t_i &= m+1)) && \forall\ i \in H^1 \\
L_i * (1 - (t_i = m+1)) &\leq L_{max} && \forall\ i \in H^1 \\
a_i &\in \mathbb{Z}^+ && \forall\ i \in H \\
s_i &\in \mathbb{Z}^+ && \forall\ i \in H \\
t_i &\in \mathbb{Z}^+ && \forall\ i \in H \\
L_i &\geq 0 && \forall\ i \in H
\end{aligned}
$$

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].

Comparison of Model Computational Time

Table: Comparison of best MILP for an specific data set

Data	$MILP_3$	$MILP_4$
1.2	**129.4**	106.9
1.3	**109.2**	99.2
1.4	**84.2**	81.1
Time (s)	**11583**	13982

Matheuristic

The matheuristic approach is based on a LNS (Large Neighborhood Search) structure in which every iteration consists on two basic procedures: destroy and rebuilt. In addition, a post-optimization phase based on a set partitioning model selects the best routes founded by different LNS iterations.

Matheuristic: Destroy Procedures

The different ways to remove nodes from a route are as follows:

- *Destroy 1*: In this procedure the removed nodes are chosen randomly.
- *Destroy 2*: This procedure removes the set of nodes with the least profit and, in case of ties, it removes those which save more time.
- *Destroy 3*: This procedure removes sequences of 3 consecutive nodes.
- *Destroy 4*: This procedure starts by inserting one node randomly chosen in the route, and then removes as less nodes as possible to have a feasible route.

Matheuristic: Rebuilt Procedures

The rebuilt route is the solution of a mathematical model based on Model 4 ($MILP_3$), which has the best performance, that seeks for the optimal way to integrate the nodes in sets R (removed nodes) and N (new nodes) into the route M which is already a feasible route.

Matheuristic: Rebuilt Procedures

$$\max z = \sum_{i \in R \cup N} p_i \cdot y_i$$

$$w_{ij} = 1 \qquad \forall\ i, j \in M$$

$$\sum_{i \in M \cup R \cup N} w_{ij} \geq y_j \qquad \forall\ j \in R \cup N$$

$$\sum_{j \in M \cup R \cup N} w_{ij} \geq y_i \qquad \forall\ i \in R \cup N$$

$$w_{ij} + w_{ji} \leq y_j \qquad \forall\ i \in M, j \in R \cup N$$

$$w_{ij} + w_{ji} \geq y_i + y_j - 1 \qquad \forall\ i \in R \cup N, j \in M \cup R \cup N$$

$$L_j \geq L_i + d_{ij} - L_{max} \cdot (1 - w_{ij}) \qquad \forall\ i, j \in M \cup R \cup N$$

$$L_j \geq L_i + d_{ij} - L_{max} \cdot w_{ji} \qquad \forall\ i, j \in M \cup R \cup N$$

$$L_i + d_{i,n-1} \leq L_{max} \qquad \forall\ i \in M \cup R \cup N$$

$$w_{ij} \in \{0, 1\} \qquad \forall\ i, j \in M \cup R \cup N$$

$$y_j \in \{0, 1\} \qquad \forall\ j \in R \cup N$$

$$L_j \geq 0 \qquad \forall\ j \in M \cup R \cup N$$

Matheuristic: Post-optimization Process

The post-optimization process selects a set of m routes from a set Ω, where all the routes are stored, which reach the maximum total collected profit and visit every node at most once. That selection is given by solving a mathematical model. Each route $k \in \Omega$ has associated a total collected profit P_k and a parameter γ_{ki} which indicates if the node $i \in V'$ is visited ($\gamma_{ki} = 1$) or not ($\gamma_{ki} = 0$) by that route k where V' is the set of requiered nodes.

Matheuristic: Post-optimization Process

$$\max z = \sum_{k \in \Omega} P_k \cdot \chi_k \tag{1}$$

$$\sum_{k \in \Omega} \gamma_{ki} \cdot \chi_k \leq 1 \qquad \forall\ i \in V' \tag{2}$$

$$\sum_{k \in \Omega} \chi_k = m \tag{3}$$

$$\chi_k \in \{0, 1\} \qquad \forall\ k \in \Omega \tag{4}$$

Results

Table: Comparing final results

Data	Nodes	Problems	$MILP_3$	CP	Mh	Best[2]
1.2	32	18	116.7	75.8	**139.2**	**149.1**
2.2	21	11	**190.5**	166.4	187.7	**190.5**
3.2	33	20	430.5	276.5	**480.0**	**496.0**
4.2	100	20	98.8	31.2	**791.4**	**917.1**
5.2	66	26	272.5	249.2	**797.5**	**897.8**
1.3	32	18	100.3	68.1	**109.4**	**125.0**
2.3	21	11	**136.4**	131.4	135.0	**136.4**
3.3	33	20	364.0	266.5	**401.0**	**411.5**
4.3	100	20	172.4	1.9	**700.7**	**856.2**
6.3	64	14	306.0	86.6	**435.0**	**454.4**
1.4	32	18	81.1	57.5	**83.3**	**101.0**
2.4	21	11	**94.5**	92.7	92.7	**94.5**
4.4	100	20	191.5	0.0	**583.4**	**804.1**
5.4	66	26	278.7	171.3	**619.6**	**708.8**
6.4	64	14	199.7	15.3	**247.3**	**255.0**
		Time (s)	24022.7	17755.6	23420.0	**5204.2**

[2] Taken from [Kim et al., 2013]

Future Work

- To define new subproblems and methods to solve them. For instance, the method split for vehicle routing problems [Prins, 2004] can be used to solve the TOP as shown by [Bouly et al., 2010].
- To propose a different MILP model based on set partitioning models as the post-optimization procedure.
- The proposed matheuristic can be improved by adding the use of some memory structures and precomputations in order to speed up the solution procedure.

Questions?

References I

Bouly, H., Dang, D.-C., and Moukrim, A. (2010).
A memetic algorithm for the team orienteering problem.
4OR, 8(1):49–70.

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.

References II

Kim, B.-I., Li, H., and Johnson, A. L. (2013).
An augmented large neighborhood search method for solving the team orienteering problem.
Expert Systems with Applications, 40(8):3065–3072.

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 III

Prins, C. (2004).
A simple and effective evolutionary algorithm for the vehicle routing problem.
Computers & Operations Research, 31(12):1985–2002.

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.