Matheuristics for solving the Traveling analyst problem

Alexandre Chanson[1], Nicolas Labroche[1], Patrick Marcel[1], Vincent T'Kindt[1,2]

[1] Université de Tours, Laboratoire d'Informatique Fondamentale et Appliquée (EA 6300)
{name.surname}@univ-tours.fr

[2] EMR CNRS 7002 ROOT, Tours, France

Mots-clés : *Orienteering Problem, Matheuristic*

1 Introduction

The Traveling Analyst Problem (TAP) [2] can be defined as follows : let Q be a set of n database queries labeled q_1 through q_n, defined by an interstingness score v_i and an execution time t_i . Besides between q_i and q_j is's defined a distance $c_{i,j}$ representing the suitability of finding q_j right after q_i. The goal is finding the most interesting sequence such that the overall time and distance are bounded by ϵ_t and ϵ_d respectively.

The TAP can be seen as an extension to the orienteering problem (OP) [4] which originates from the database community where it represents a data exploration problem. The TAP is NP-hard and we focus on its heuristic solution. Matheurisitcs are known for producing near optimal solutions in a reasonable amount of time by exploiting mixed integer programming solvers. [3]. To the best of our knowledge there are only two previous works describing the use of matheuristics for solving related problems : [1] focuses on the team orienteering problem and [5] on time dependent profits. These matheuristics have shown great efficiency compared to state of the art MILP solvers. We propose the first matheuristic to solve the TAP and show its efficiency by means of computational experiments.

2 A matheuristic

We propose an application of the Variable Partitioning Local Search (VPLS) framework [3]. This approach relies on a MILP solver to perform a local search around a feasible solution. At each iteration, starting from an initial feasible solution, the variables are split into two set one set free for the MILP solver to re-optimize the second fixed to their current values. We propose 4 different approaches for dividing variables between the two sets (free/fixed) . Two of the approaches rely on the solutions sequence structure were a window of consecutive queries is let free and the rest of the solution fixed. One method relies on a simple procedure to move the window at each iteration while the other is non deterministic. The two other approaches rely on adding a single constraint based on the hamming distance between to the model, in this case we don't define directly the set of free variables but rather it's size. The hamming distance is computed on either s or both s and x between the previous or initial solution and the current one, a bound h is set to allow a maximum number of "edits" per iteration. Those approaches offer the greatest flexibility as to the modifications the solver can perform on an initial solution. As previously mentioned initially a feasible solution is required so we propose a polynomial heuristic to construct it which exploits the knapsack structure of the problem.

At the conference we will present the details of the matheuristic and provide experimental evidence that at least one of the variants is very effective even on large sized instances (up to 600 queries).

2.1 Mathematical model

variables

$x_{i,j}, \forall i, j \in 1..n, x_{i,j} = 1$ if q_i comes directly before q_j in the solution, 0 otherwise
$x_{0,i}, \forall i \in 1..n, x_{i,j} = 1$ if q_i is the first query of the solution, 0 otherwise
$x_{i,n+1}, \forall i \in 1..n, x_{i,j} = 1$ if q_i is the last query of the solution, 0 otherwise
$s_i, \forall i \in 1..n$: boolean variables denoting the presence of q_i in the solution, 0otherwise
$u_i \in [2, n], \forall i \in 1..n$: integer variables used in subtour elimination constraints.

Objective

$$max \sum_{i=1}^{n} v_i s_i \tag{1}$$

Constraints

$$\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} c_{i,j} x_{i,j} \leq \epsilon_d \tag{2}$$

$$\sum_{i=1}^{n} t_i s_i \leq \epsilon_t \tag{3}$$

$$\sum_{i=0, j\neq i}^{n} (x_{i,j}) - s_j = 0, \forall j \in 1..n \tag{4}$$

$$\sum_{j=1, j\neq i}^{n+1} (x_{i,j}) - s_i = 0, \forall i \in 1..n \tag{5}$$

$$\sum_{j=1}^{n} x_{0j} = \sum_{i=1}^{n} x_{i,n+1} = 1 \tag{6}$$

$$u_i - u_j + 1 \leq (n-1)(1 - x_{ij}), \forall i, j \in 1..n \tag{7}$$

Références

[1] C. Archetti, Á. Corberán, I. Plana, J. M. Sanchis, and M. Grazia Speranza. A matheuristic for the team orienteering arc routing problem. *European Journal of Operational Research*, 2015.

[2] A. Chanson, B. Crulis, N. Labroche, P. Marcel, V. Peralta, S. Rizzi, and P. Vassiliadis. The traveling analyst problem :definition and preliminary study. 22nd International Workshop On Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP 2020).

[3] F. Della Croce, A. Grosso, and F. Salassa. Matheuristics : Embedding milp solvers into heuristic algorithms for combinatorial optimization problems. *Heuristics : Theory and Applications*, pages 53–68, 2013.

[4] T. Tsiligirides. Heuristic methods applied to orienteering. *The Journal of the Operational Research Society*, 35 :797–809, 1984.

[5] Q. Yu, K. Fang, N. Zhu, and S. Ma. A matheuristic approach to the orienteering problem with service time dependent profits. *European Journal of Operational Research*, 273 :488–503, 2019.