Progetto adatto a tesi di laurea magistrale e a progetto d'esame per Complementi di R.O.

1 Hierarchical TSP

The HTSP is defined as follows. A digraph $G = (N, A)$ is given. Weights associated with the arcs indicate their length. The node set $N = \{0, \ldots, n\}$ is partitioned into subsets, i.e. $N = \bigcup_{k=0}^{K} N_k$ and subsets N_k are disjoint and each subset N_k has a priority $k = 0, \ldots, K$. A starting depot 0 is the only node with priority 0, i.e. $N_0 = \{0\}$. A non-negative (and small) integer p is given. A vehicle must visit all nodes starting from the depot (and not necessarily returning to it) with the following constraint: all nodes with priority k must be visited before any node with priority $k + 1 + p$. The objective is to minimize the total distance traveled.

The motivation is the distribution of goods in an emergency context, where a priority is associated with each site to be visited. For this reason p is a small number because it represents the extent to which exceptions are allowed to the rule of following the priorities.

The HTSP is a special case of the ATSPPC, where the set of precedences has a special (and regular) structure. There are no customized exact optimization algorithms for this problem, at the best of my knowledge.

Reference: Pamchamgan et al. (2013).

The problem can be tackled with branch-and-bound based on additive bounds, with branch-and-cut and with branch-and-price.

Existing branch-and-bound and branch-and-cut algorithms for the ATSPPC can be used as benchmarks for a new branch-and-price algorithm, which exploits the special structure of the precedences of the HTSP.

Additive bounds. The following lower bounding procedures can be used:

- linear assignment, taking into account only the degree constraints on each node;
- shortest spanning 1-arborescence (in both directions, outgoing from the depot or incoming to it)
- variable decomposition induced by all precedences due to the hierarchical constraint, as in the ATSP with precedence constraints
- disjunctions, as in the ATSPPC

Branch-and-cut. A branch-and-cut algorithm can use the valid inequalities developed by Ascheuer et al. for the ATSPPC (and possibly others).

Branch-and-price. The problem can be reformulated as a set partitioning problem, where each row is a node and each column is a path. Each node must be visited within a path or it must be the starting node of a path and the ending node of another. So, the master problem contains a constraint on the indegree and a constraint on the outdegree for each node. Each column is a path starting from a node $i \in N_{k-1}$, complying with the precedence constraints and ending at a node $j \in N_k$, meaning that in a solution containing that path, node i is the last visited node in N_{k-1} and node j is the last visited node in N_k. Hence each column belongs to a column subset k, and a solution is made by one column from each subset.

The master problem is:

$$\text{minimize } z = \sum_{k=1}^{K} \sum_{l \in \Omega_k} c_l \lambda_l \tag{1}$$

$$\text{s.t.} \sum_{h=k-p}^{k} \sum_{l \in \Omega_h} a_{il} \lambda_l + \sum_{l \in \Omega_k} e_{il} \lambda_l = 1 \qquad \forall k = 1, \ldots, K \ \forall i \in N_k \tag{2}$$

$$\sum_{h=k-p}^{k} \sum_{l \in \Omega_h} a_{il} \lambda_l + \sum_{l \in \Omega_{k+1}} s_{il} \lambda_l + \sum_{l \in \Omega_K} e_{il} \lambda_l = 1 \qquad \forall k = 0, \ldots, K \ \forall i \in N_k \tag{3}$$

$$\sum_{l \in \Omega_k} \lambda_l = 1 \qquad \forall k = 1, \ldots, K \tag{4}$$

$$\lambda_l \in \{0, 1\} \qquad \forall k = 1, \ldots, K \ \forall l \in \Omega_k \tag{5}$$

The objective function (1) asks for the minimization of the overall length of the selected paths; the cost c_l of each path l is formally defined later. Constraints (2) and (3) impose that each node has indegree and outdegree equal to 1, that is it is visited once in the solution. In constraints (2) a node $i \in N_k$ has indegree equal to 1 when it is visited along a path $l \in \Omega_h$ with $k - p \leq h \leq k$ ($a_{il} = 1$) or when it is the end node of a path $l \in \Omega_k$ ($e_{il} = 1$). In constraints (3) a node $i \in N_k$ has outdegree equal to 1 when it is visited along a path $l \in \Omega_h$ with $k - p \leq h \leq k$ ($a_{il} = 1$) or when it is the start node of a path $l \in \Omega_{k+1}$ ($s_{il} = 1$); the constraint must not be imposed on one of the nodes of N_K, i.e. the last node of the solution. For this reason the left hand side of constraints (3) also include the additional term $\sum_{l \in \Omega_K} e_{il} \lambda_l$, so that the constraint is satisfied also for a node of N_K with no outgoing arcs, provided it is the end node of its path. Remarkably, constraints (2) and (3), besides working as degree constraints, also forbid subtours owing to the special structure of the precedence constraints that characterize the HTSP with respect to the more general ATSPPC. Constraints (4) are convexity constratins, stating that the solution must include exactly one path for each priority class $k = 1, \ldots, K$ i.e. one column for each column subset Ω_k. Finally, integrality constraints (5) on the binary variables corresponding to the columns of the master problem are relaxed into $0 \leq \lambda_l \leq 1$ when the linear relaxation of the master problem is solved via column generation. The dual variables corresponding to constraints (2), (3) and (4) are indicated by β^-, β^+ and α respectively.

The pricing subproblem can be solved independently for each $k = 1, \ldots, K$. To define the pricing subproblem we introduce the sets $A_k = \bigcup_{h=k}^{k} +pN_h \ \forall k = 1, \ldots, K$. Each subset A_k is the subset of nodes that can be visited by a path corresponding to a column in Ω_k. Such a path must start from a node in N_{k-1} (from the depot, if $k = 1$), it must end at a node in N_k and it can visit any node in A_k. A binary variable s_i indicates whether $i \in N_{k-1}$ is the start node; a binary variable e_i indicates whether $i \in N_k$ is the end node; a binary variable a_i indicates whether $i \in A_k$ is visited along the path, not being an endpoint. We also introduce binary variables associated with the arcs: a binary variable x_{ij} indicates whether arc (i, j) is an arc of the path from an intermediate node $i \in A_k$ to an intermediate node $j \in A_k$; a binary variable u_{ij} indicates whether arc (i, j) is the first arc of the path, from the start node $i \in N_{k-1}$ to an intermediate node $j \in A_k$; a binary variable v_{ij} indicates whether arc (i, j) is the last arc of the path, from an intermediate node $i \in A_k$ to the end node $j \in N_k$; a binary variable w_{ij} indicates whether arc (i, j) is the only arc of the path, from the start node $i \in N_{k-1}$ to the end node $j \in N_k$. With these definitions, the pricing subproblem is as follows (when not needed, index k has been dropped for readability).

$$\text{minimize } r = \sum_i \sum_j c_{ij}(x_{ij} + u_{ij} + v_{ij} + w_{ij}) - \sum_j \beta_j^-(a_j + e_j) - \sum_i \beta_i^+(s_i + a_i) - \alpha \quad (6)$$

$$\text{s.t. } \sum_i s_i = 1 \quad (7)$$

$$\sum_i e_i = 1 \quad (8)$$

$$s_i = \sum_j (u_{ij} + w_{ij}) \qquad \forall i \in N_{k-1} \quad (9)$$

$$e_j = \sum_i (v_{ij} + w_{ij}) \qquad \forall j \in N_k \quad (10)$$

$$a_i = \sum_j (x_{ij} + v_{ij}) = \sum_j (x_{ji} + u_{ji}) \qquad \forall i \in A_k \quad (11)$$

subtour elimination constraints on variables x (12)

integrality constraints on all binary variables (13)

In the objective function (6) r indicates the reduced cost of the column corresponding to the path. Constraints (7) and (8) enforce the selection of a start node and an end node. Constraints (9), (10) and (11) link node variables with arc variables. The cost c_l of a path is given by $\sum_i \sum_j c_{ij}(x_{ij} + u_{ij} + v_{ij} + w_{ij})$. The problem turns out to be an Elementary Shortest Path Problem on a digraph with possibly negative cost cycles, owing to the dual variables. The problem is NP-hard and it can be solved either with a cutting planes algorithm or with dynamic programming. Since the value of p is assumed to be small, it is likely that each A_k be significantly smaller than N which should make each instance of the pricing subproblem solvable within a reasonable amount of computing time.

References

[1] F. Gandellini, *Due euristiche per HTSP a confronto*, University of Milan, 2013.

[2] K. Pamchamgam et al. *The hierarchical traveling salesman problem* Optimization Letters 7 (2013) 1517-1524