Joint Function Placement and Assignment

Abstract: Function-oriented networks aim at abstracting functional distribution by processing demands formulated as service descriptors instead of being directed to destination locators (corresponding to their assignment). Service requests when entering the network are translated into unsplittable sequences of ordered operations/functions; each individual node executing at most one out of n functions. More formally, a service request $F(\bar{x})$ where $\bar{x}$ is the input vector, denotes the ordered sequence $f_n(f_{n-1}(\ldots(f_2(f_1(\bar{x})))))$ where f_1 denotes the first and f_n the last function applied to the input vector.

We are given a network graph and a set of customer service demands whose functional decomposition into is known by network. Each of these demands must be serviced by an ordered sequence of at most n different functions. The problem consists thus of selecting the subset of locations where to jointly place functions and assigning demands to paths crossing these locations without exceeding both arc nominal capacity and node processing capacity (no computing node is assigned to more clients than its processing capacity). The objective is to find the set of locations and paths that minimize the sum of the location cost (function placement), the demand allocation cost (processing cost), and the routing cost; thus, obtain the least cost network answer as fast as possible.

This problem shares similarity with the multi-stage multi-product (capacitated) facility location problem with the following particularities:

- Multi-stage (or non-coherent multi-level): as each of these demands must be serviced by an ordered sequence of at most n different functions one could assume that each of them corresponds to a given stage; however, as each sequence may comprise a different number of functions their joint placement is less constraining on the spatial configuration of levels.
- Multi-product: demands can be seen as requests for ordered product mix each of them consuming a different amount of capacity units; in the present case, servicing demands involves their decomposition into different functions each of them having possibly different execution/processing capacity requirement.
- The combination with routing removes the allocation independence property (demands originated by a given customer are allocated independently of other demands), leading to strongly interrelated location and routing decisions as the former aggregate demands whereas demands sharing common sub-sequence may or not be routed along common (sub-)path depending on the corresponding arc and node load.

Specific problems (each of them may be the subject of an internship):

- Using off-the-shelve programming solver, the exact mixed-integer formulation provides optimal solutions for (relatively) small instances. To overcome this problem and cover more complex demand sets over larger network instances, one approach would be to decompose the initial formulation such that solving the decomposed problems is much easier than the original one. The working task consists of tuning the decomposition methods, designing the decomposition algorithm as well as evaluating its computational performance together with the solution quality it produces.
- As functional sequences may comprise repetitions (the same function may appear non-consecutively multiple times), the resulting loops may lead to consider a particular form of next-hop/routing decision process. The working task consists in formulating the corresponding mixed-integer program, performing numeric experiments (using mixed-integer programming solver) to evaluate the performance tradeoffs of this extended routing functionality and determining the computational properties of the mixed-integer program against its original formulation.
- As solutions to optimization problems can exhibit high sensitivity to perturbations in the problem parameters (in the present case, never-seen-before demands) and in turn rendering infeasible or suboptimal solution, modeling the optimization problem by means of the set-induced robust optimization method immunizes solutions against bounded uncertainty in the parameters of the problem. This technique intrinsically depends on the features of the uncertainty sets. The working task consists in detecting and extracting useful features out of known/prior decompositions such as to automatically construct polyhedral uncertainty sets, propose a robust-formulation of the original problem and experiment numerically the properties of the (robust) solution.

Qualifications (Skills/Knowledge):

- Computer science, applied mathematics, operational research (min.level: last year of MSc)
- Orientation: math.programming/optimization (combinatorial, continuous, robust), machine learning/AI algorithmic
- Experience with mixed-integer programming solvers (CPLEX, COIN-OR, etc.), machine learning toolbox/libraries (shogun, weka, etc.)

Practical information:

- Duration/period: academic year 2016-17
- Position: student internship (standard internship agreement)
- Location: Bell Labs, Antwerp (contact: dimitri.papadimitriou@nokia.com)