Align-BDD application: on joint UFLP.

Alexey Bochkarev

June 3, 2022

1 TBD: Summary and questions

- weak point: CPP MIP is still faster.
- weak point: feeding more info to the DD-based methods.
- weak point: not that much variety in graph topologies.
- question: not equivalent to another UFLP? I guess not (consider j10 and j16).

2 Joint UFLP formulation

We next illustrate the performance of our algorithms in the context of a specific application. We will consider the following variant of the uncapacitated facility location problem, UFLP (Owen and Mark S. Daskin, 1998; ReVelle, Eiselt, and M. S. Daskin, 2008). In this section, we introduce the problem, discuss a naive MIP formulation, and outline a key idea for an alternative CPP formulation (where our heuristic can be used to align the diagrams). The numerical results are then discussed in the next section.

We have designed the problem specifically to highlight a possible application of our heuristic: It comprises two similar UFLP subproblems of special structure (which can be represented with BDDs), linked with side constraints implying the interdependence of some decisions across two problems.

Each of the two subproblems (indexed with $t = 1$ or 2) considers a set of N points. At each point i, we can locate a facility at a cost given by c_i^t, covering all points in a set given by S_i^t, where $i \in S_i^t$. Set S_i^t might refer to customers that are sufficiently close to location i according to some specified metric, like distance or travel time. Therefore the N points represent some graph G^t and S_i^t encode the respective adjacency lists. (For convenience, we assume G^t to be connected.) We also define a cost function $f_i^t : \mathbb{N}_0 \to \mathbb{R}$, so that $f_i^t(a)$ would indicate a cost of covering point i exactly a_i times, within subproblem t. Note that we do not imply any properties of f_i, such as convexity or concavity. Connection between the subproblems is given by the condition that there is a set of pairs of points $(i^1, i^2) \in J$ such that the location decision regarding point i^1 in G_1 must coincide with corresponding decision on i_2 in G_2. The problem minimizes the total cost and can be formulated as follows:

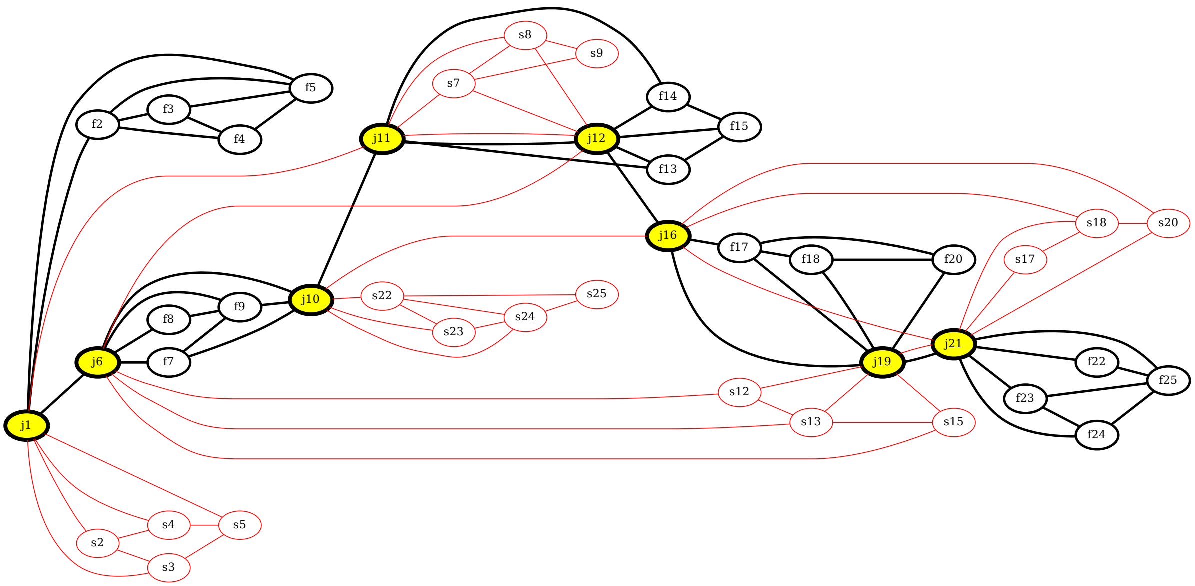

Figure 1: Sample j-UFLP instance graph.

$$\min \sum_{i=1,t=1,2}^{N} \left(c_i^t x_i^t + f_i^t(a_i^t) \right) \tag{j-UFLP}$$

$$\text{s.t. } a_i^t = \sum_{j \in S_i^t} x_i^t \qquad \text{for all } i = 1, \ldots, N, t = 1, 2, \tag{1a}$$

$$x_i^t \in \{0, 1\} \qquad \text{for all } i = 1, \ldots, N, t = 1, 2, \tag{1b}$$

$$x_j^1 = x_j^2 \qquad \text{for all } j \in J. \tag{1c}$$

Further, to make this problem hard for an MIP solver, we induce the following special structure for the underlying graph. We assume that each sub-problem graph consists of n subgraphs, M nodes each, and that there is at most one arc between every two subgraphs. Moreover, we assume J to be the set of endpoints of such arcs between the subgraphs. An example of such instance is presented in Figure 1. A subgraph corresponding to the first subproblem is depicted with thick black lines, while the one corresponding to the second subproblem has thin red lines. Nodes that are common for the two subgraphs (corresponding to conditions (2d)) are marked with background color (nodes `j1`, `j6`, `j10`–`j12`, `j16`, `j19`, and `j21`). Note that overlaps are calculated independently: e.g., locating a facility at `j10` does contribute to a_{16}^2, but not a_{16}^1.

3 Solution methods

A naive MIP reformulation for (1) implies introducing new binary variables $y_{i,a}^t$ indicating whether point i in subproblem t was covered at least a times. Therefore, the equivalent problem is:

$$\min \sum_{i=1,t=1,2}^{N} \left(c_i^t x_i^t + \sum_{a=0}^{|S_i^t|} q_{i,a}^t y_{i,a}^t \right) + C \tag{j-UFLP-MIP}$$

$$\text{s.t.} \sum_{a=1}^{|S_i|} y_{i,a}^t = \sum_{j \in S_i^t} x_i^t \quad \text{for all } i = 1, \ldots, N, t = 1, 2, \tag{2a}$$

$$y_{i,a}^t \geq y_{i,a+1}^t \quad \text{for all } i = 1, \ldots, N, t = 1, 2, a = 0, \ldots, |S_i|, \tag{2b}$$

$$x_i^t \in \{0, 1\} \quad \text{for all } i = 1, \ldots, N, t = 1, 2, \tag{2c}$$

$$x_i^1 = x_j^2 \quad \text{for all } (i, j) \in J, \tag{2d}$$

where $q_{i,a} = f_i^t(a) - f_i^t(a-1)$ and $C = \sum_{i,t} f_i^t(0)$ are constants.

From the other hand, such problem can be represented as a CPP with two fixed-width diagrams (each corresponding to a subproblem), having different order of variables. For each subproblem, we build a BDD, where each path captures total cost stemming from this subproblem and corresponding to the respective location decisions. Interconnection between subproblems is achieved by renaming the variables in the diagrams: for each $(i, j) \in J$ labels of the BDD layers corresponding to x_i^1 and x_j^2 (in the first and the second BDD, respectively) must coincide.

I think this part will go to the appendix:

Let us briefly illustrate the BDD construction procedure for a single subproblem with Figure 2. Assume we have four clusters of points, denoted `C1`, …, `C4` and points `1`, …, `6` are endpoints of the edges connecting these clusters. First, observe that when we fix values for x_1 and x_2, the costs contribution to the objective stemming from cluster `C1` can be obtained by solving a smaller mixed-integer program. The formulation is similar to (2), but with index t and linking condition (2d) dropped and the set of nodes restricted to `C1`. Such MIP would have only M variables x_i, corresponding to `C1`. Therefore, we introduce x_1 and x_2 to the BDD and capture the costs stemming from cluster `C1` in BDD arc weights pointing towards the $2^2 = 4$ nodes in the last layer, for (x_1, x_2) being equal $(0, 0)$, $(1, 0)$, $(0, 1)$, and $(1, 1)$, respectively. If we further add x_3 and x_4 to the diagram, we can fix costs stemming from cluster `C2` in BDD arc costs as well. After we add x_3 the diagram width increases to $2^3 = 8$ (we are keeping track of x_1, x_2, and x_3), but after we introduce x_4 we do not need to have $2^4 = 16$ nodes in the last layer. Note that x_1 will not affect any costs stemming from `C3` and `C4`. Therefore, BDD nodes corresponding to different values of these two variables in the last layer can be joined, which would make it sufficient to have x_4 layer with 4 nodes only. Continuing this process, we build a diagram of width $2^2 = 8$ at most (regardless of n and M), that encodes the subproblem. Note that depending on costs, it may or may not be quasi-reduced. The resulting BDD is presented in Figure 3. We indicate selected arc costs in the figure, denoting the objective for the cluster subproblem `C1` as $C1(x_1, x_2)$, for `C2` as $C2(x_1, x_2, x_3, x_4)$, for `C3` as $C3(x_3, x_4, x_5, x_6)$, and for `C4` as $C4(x_5, x_6)$. Note that because we drop the information regarding specific choices of x_1 and x_2 the highlighted nodes in Layer 4 will yield only two child nodes in Layer 5. Note that the natural order of variables that allows such compact BDD representation is determined by the sequence of clusters in the figure. Finally, if we have another subproblem encoded by a BDD with variables $x'_1, \ldots, x'_6$, and $J = \{(1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1)$, we could reformulate the j-UFLP as a Consistent Path problem instance by renaming the variables in the diagram depicted in Figure 3 from $(x_1, x_2, x_3, x_4, x_5, x_6)$ to $(x'_6, x'_5, x'_4, x'_3, x'_2, x'_1)$.

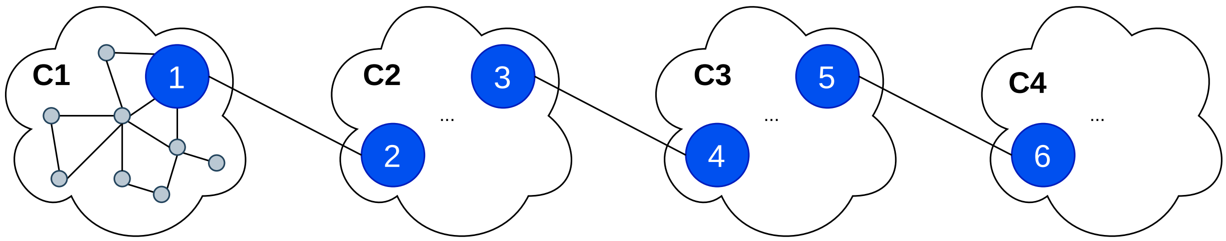

Figure 2: Sample j-UFLP subproblem graph.

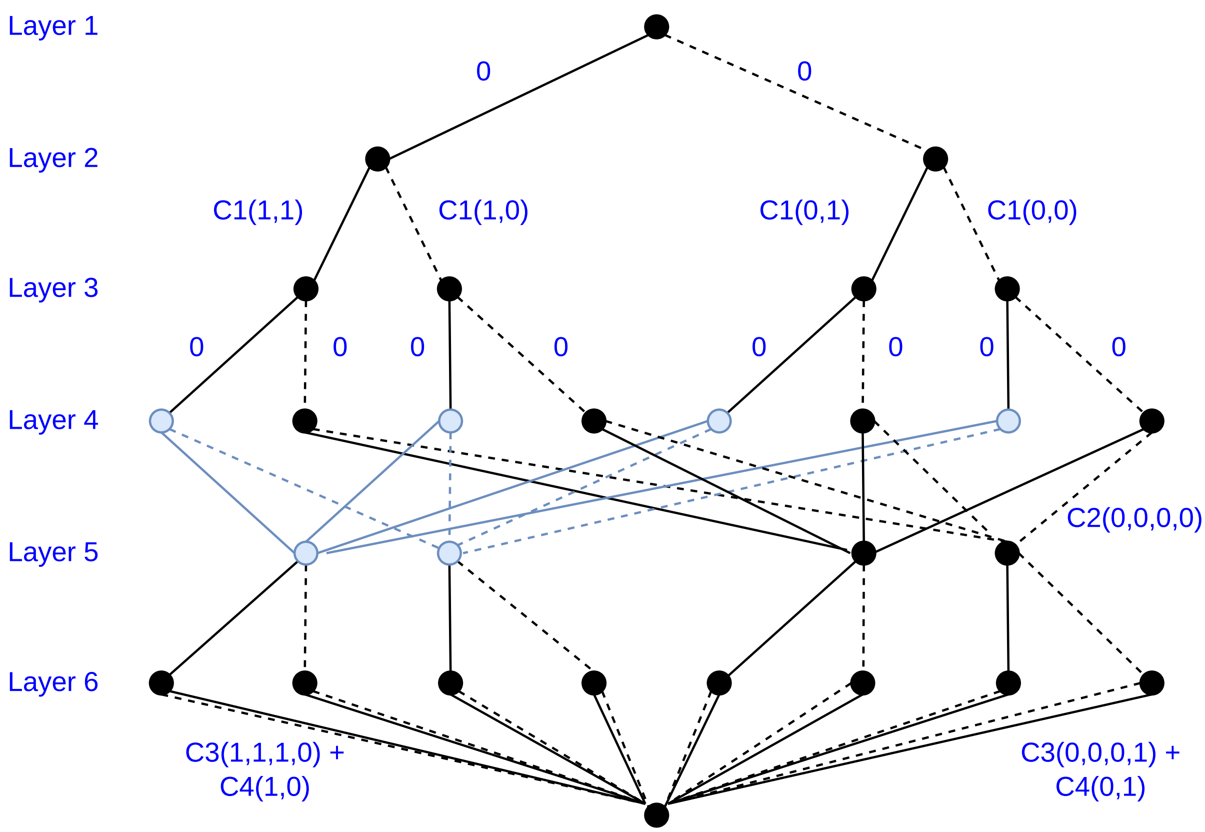

Figure 3: Resulting BDD for the subproblem in Figure 2.

4 Numerical performance

We generated approximately 400 instances for each value of n (number of clusters), ranging from 5 to 14. Each cluster contained $M = 5$ nodes, with sparsity parameter $L = 0.25$. The latter implies that the number of randomly generated edges $|E_k|$ in the cluster satisfies:

$$L = 1 - \frac{|E_k|}{M(M-1)/2},$$

assuming one quarter of all possible $M(M-1)/2$ edges are not present. We then solved each of the instances with each of the two methods.

- First, we used a naive MIP formulation (2) (denoted `MIP` in the figures below).
- Second, we used the pre-defined data regarding the composition of the clusters to build two BDDs as discussed above. The resulting CPP instance was solved by aligning the diagrams and finding a shortest path between root and terminal nodes. The diagrams were aligned using the proposed variable-sequence based heuristic (denoted `VS-heuristic`) and a simple baseline of aligning the second diagram to match the order of the first one (denoted `to A`).

The results are presented in Figure 4. On the left panel we present runtimes for each of the solved instances, one point per instance. Solution time in seconds (logarithmic scale) is along the vertical axis, number of clusters n is along the horizontal axis. We see that since we are leveraging the additional information regarding the composition of the clusters in the BDD-based approach, VS-based heuristic tends to perform relatively better than the naive MIP, and this gap increases as the instance size grows. (Numbers in the rectangles above the graph indicate the share of instances where the proposed heuristic were faster at least by 10%.) On the right panel we present histograms of the runtimes for `MIP` and `to A` heuristics, relative to the proposed `VS-heuristic`. The value of 1.0 (vertical line) implies that the heuristic takes the same time (for a particular instance) as `VS-heuristic`. Values to the left of the vertical line imply the heuristic outperforms the proposed one. We see that while for $n = 7$ clusters `VS-heuristic` is almost always slower than MIP, for larger instances it starts to outperform the baseline on more than two thirds of the instances.

References

Owen, Susan Hesse and Mark S. Daskin (Dec. 1998). "Strategic Facility Location: A Review". In: *European Journal of Operational Research* 111.3, pp. 423–447. ISSN: 03772217. DOI: `10.1016/S0377-2217(98)00186-6` (cit. on p. 1).

ReVelle, C. S., H. A. Eiselt, and M. S. Daskin (Feb. 2008). "A Bibliography for Some Fundamental Problem Categories in Discrete Location Science". In: *European Journal of Operational Research* 184.3, pp. 817–848. ISSN: 0377-2217. DOI: `10.1016/j.ejor.2006.12.044` (cit. on p. 1).

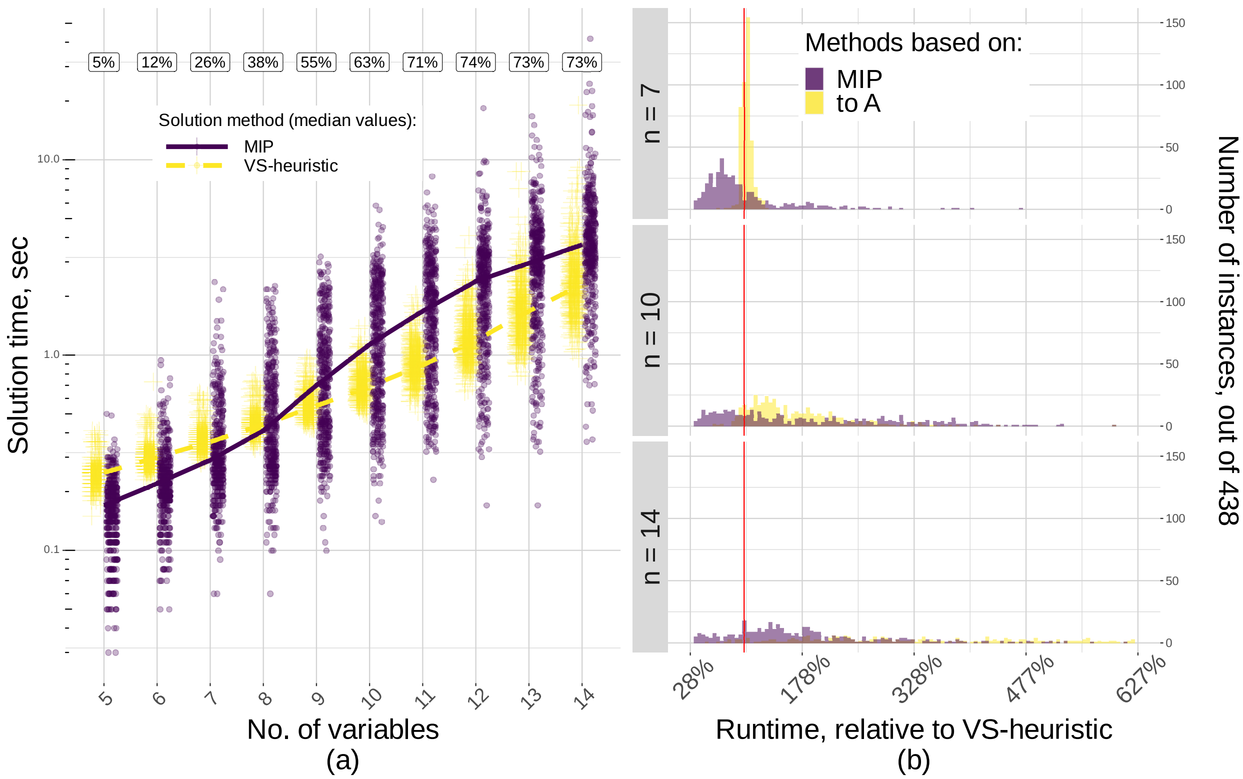

Figure 4: Numerical performance of various heuristics for j-UFLP.