OPAC: Optimal Packet over Avian Carriers

Sobhan Mohammadpour

April 24, 2022

1 Motivation

Imagine on hot summer day, you need to send a few terabytes of data to a friend that lives down the block, you somehow manage to find a service that allows you to upload that much data and start uploading. However, you realize that the upload is going to take too much time and so decided to just give an external HDD to your friend.

Inspired by the IP over Avian protocol and a story from South Africa [Abe] where the owners of a telecome company sent their data strapped to a pigeon which was much faster than uploading data.

These days 1TB micro-SD cards are fairly cheap so it's not hard to imagine that someone actually does this. But the real use-case is not necessarily with pigeons but with drones and unmanned airplanes. While normal conditions may not require such a drastic and cumbersome way of communication, in the event of disaster (man-made or otherwise), setting up a mesh network is generally preferred over total lack of networking. In such a mesh network, we want to improve the transfer rate by augmenting the network with a high capacity, high cost and very limited series of agents that can transfer the data by jumping over the nodes in the mesh network.

We note that the main difference between a "normal" communication network and a mesh one is the presence of cycles.

2 Problem Definition

The goal is to transfer a number commodities enumerated by the elements of K, all while selecting the routes pigeons have to to minimize the cost.

Let

1. K be the set of tasks and,
2. P be the set of pigeons,

Code is available at `https://github.com/SobhanMP/OPAC`

3. V be the set of nodes in the mesh network,
4. $E \subset V \times V$ be the arc connecting those nodes,
5. $N^p \subset V$ be the set of node bird p can travel between and,
6. $A^p \subset N \times N$ be the set of arcs bird p have to travel on.

$$\text{minimize} \quad \sum_{p \in P} v^p a^p + \sum_{k \in K} \left(\sum_{e \in E} c_e x_e^k + \sum_{p \in P, i \in V} g u_i^{pk} + h d_i^{pk} \right) \tag{1a}$$

subject to

$$\sum_k x_e^k \leq m_e, \qquad e \in E, \tag{1b}$$ {eqs:land_const}

$$l_e^p \leq a^p \qquad p \in P, e \in A^p, \tag{1c}$$ {eqs:air_active}

$$\sum y_e^{pk} \leq q l_e^p, \qquad p \in P, e \in A^p, \tag{1d}$$ {eqs:air_cap}

$$\sum_{e \in A^p} l_e^p \ell_e \leq L, \qquad p \in P, \tag{1e}$$ {eqs:max_length}

$$\sum_{(i,j) \in E^p} l_{ij}^p = \sum_{(j,i) \in E^p} l_{ji}^p, \qquad p \in P, i \in N^p, \tag{1f}$$ {eqs:air_cycle}

$$U_i^p - U_j^p + 1 \leq |N^p|(s_i^p + s_j^p + 1 - l_{ij}^p), \qquad p \in P, (i,j) \in E^p, \tag{1g}$$ {eqs:MTZ}

$$\sum_{i \in N^p} s_i^p = a^p \qquad p \in P, \tag{1h}$$ {eqs:smax}

$$\delta_i^k + \sum_{p \in P} u_i^{pk} - d_i^{pk} = \sum_{(i,j) \in E} x_{ij}^k - \sum_{(j,i) \in E} x_{ji}^k \quad i \in V, \tag{1i}$$ {eqs:flow_land}

$$u_i^{pk} + \sum_{(j,i) \in A^p} y_{ji}^{pk} = d_i^{pk} + \sum_{(i,j) \in A^p} y_{ij}^{pk} \qquad p \in P, i \in N^p \tag{1j}$$ {eqs:flow_air}

where

1. $a^p \in \{0,1\}$ indicates whether the pigeon p is active,
2. $x_e^k \in \mathbb{R}^+$ is the flow of commodity k on arc e,
3. $u_i^{pk} \in \mathbb{R}^+$ and $d_i^{pk} \in \mathbb{R}^+$ are, respectively, the upload amount and download amount of the data from task k to and from the pigeon p,
4. $l_e^p \in \{0,1\}$ indicates whether arc $e \in A^p$ is active for pigeon $p \in P$,
5. $s_i^p \in \{0,1\}$ indicates if the loop of the pigeon p started at node i, and
6. $0 \leq U_i^p \leq |N^p|$ is an MTZ [MTZ60] like potential for each node i and pigeon p,

and

1. v is the cost of using a pigeon,
2. c_e is the cost of one unit of flow on arc e,
3. g is the cost of uploading one unit of data to a pigeon,
4. h is the cost of downloading one unit of data from a pigeon
5. m_e is the capacity of the edge e,
6. L be the maximum distance pigeons can travel,
7. ℓ_e is the length of edge e,
8. q is the maximum capacity of pigeons, and
9. δ_i^k be the flow constraint's rhs for the node i and the task k i.e. for a transfer v, it results $-v$ at the origin of the transfer, v at the destination and zero other wise such.

The

1. constraint (1b) defines the capacity of the mesh network,
2. constraint (1b) force the capacity of inactive pigeons to be zero,
3. constraint (1d) enforces the pigeon capacity of pigeons,
4. constraint (1e) forces the travel distance of each pigeon to be less than L,
5. constraint (1f) forces the paths of pigeons to consist of cycles,
6. constraint (1g) is an MTZ like constraint that forces only one simple cycle to exist, The main difference with MTZ is that the constraints are defined for all nodes but the node whose s is 1 will not have any active constraints as the right hand side will be at least $|N^p|$ thus practically disabling the constraints that contain that node,
7. constraint (1h) forces one of the s_i^p to be 1 when the pigeon is active,
8. constraint (1i) is the flow conservation constraint on the mesh, and
9. constraint (1j) is the flow conservation constraint for the pigeons.

We while we assume that the pigeons are the same, it's fairly easy to extend all pigeon properties by an index p to make them unique. Furthermore, we assume that for $a = 0$ i.e. no pigeons are active, the problem is feasible.

3 Benders' decomposition

In the problem

$$\begin{aligned} \text{minimize} \quad & c \cdot x + d \cdot y && \text{(2a)} \\ \text{subject to} \quad & Ax + By = b, && \text{(2b)} \\ & x \geq 0, && \text{(2c)} \end{aligned}$$

where x is a real valued vector in $\mathbb{R}^n$ and y is a integer vector in $\mathbb{Z}^m$, we can re-write the problem as bi-level problem such that

$$\begin{aligned} \text{minimize} \quad & \theta + d \cdot y && \text{(3a)} \\ \text{subject to} \quad & \theta \in \arg\min_x \{c \cdot x | Ax = b - By, x \geq 0\}. && \text{(3b)} \end{aligned}$$ {benders:2:sp}

or alternatively since the outer problem is a minimization,

$$\begin{aligned} \text{minimize} \quad & \theta + d \cdot y && \text{(4a)} \\ \text{subject to} \quad & \theta \geq \min_x \{c \cdot x | Ax = b - By, x \geq 0\}, && \text{(4b)} \\ & \emptyset \neq \{x \in \mathbb{R}^n | Ax = b - By, x \geq 0\}. && \text{(4c)} \end{aligned}$$

We can dualize the subproblem to get

$$\begin{aligned} \text{minimize} \quad & \theta + d \cdot y && \text{(5a)} \\ \text{subject to} \quad & \theta \geq \max_\pi \{(b - By) \cdot \pi | A^\top \pi \leq c, \}. && \text{(5b)} \end{aligned}$$ {benders:3:sp}

We now longer need the feasibility constraint because the sub-problem is never unbounded(assuming the original problem is bounded) as such the dual problem is never infeasible but is unbounded (infinity) at infeasible points. This forces the problem to be feasible

[Ben62] showed that this problem can be modeled as a finite number of cuts such that

$$\begin{aligned} \text{minimize} \quad & \theta + d \cdot y && \text{(6a)} \\ \text{subject to} \quad & \theta \geq (b - By)^\top \pi, \quad \pi \in \mathcal{P}, && \text{(6b)} \\ & 0 \geq (b - By)^\top \pi \quad \pi \in \mathcal{R}, && \text{(6c)} \end{aligned}$$ {benders:2:sp}

where $\mathcal{P}$ and $\mathcal{R}$ are the set of extreme points and extreme rays of (5b). We note that if $\bar{y}$ is the point where the cut was generated and θ^0 is the objective of the dual (or the dual ray if it is unfeasible), we can write $(b - By)^\top \pi$ as $\theta^0 - (By - B\bar{y})^\top \pi$ since $\theta^0 + (B\bar{y})^\top \pi = b^\top \pi$. Since the number of cuts is finite, we can dynamically add those to the problem as needed i.e. when they become invalidated. The cuts are on the extreme rays are called feasibility cuts and are only needed when the sub-problem is not feasible.The problem that has

the benders cut is called the master problem and the problem that is used to generate the cuts is called the sub-problem.

While benders original method would add these cuts at integer points, [MD77] showed that the cuts can be added even in fractional points. Furthermore, [MW81] showed that if the cuts are generated with a core point and an additional constraint it will be non-dominated. [Pap08] showed that the additional constraint is not needed and we can take a moving average of y^t as our core point. The choice of the moving average coefficient greatly changes the performance of the method. We use $l = 0.001$ as our initial core-point (we set the variable to 0.001 for each arc) and use $\eta = 0.5$ as our moving average constant.

Our problem is a good candidate for benders decomposition as the constraints (1i) and (1j) only contain real variables and the constraint (1d) is the only linking constraint between the integer and real variables. We note that we keep the U in the master problem. This does not cause any problem on the theoretical level and helps reduce the number of infeasibility cuts. When taking out x and y out of the master problem, we only need to add cuts for the constraints that have integer variables in them (otherwise they will just be zero). In the constraint(1d) constraint matrix B is the identity matrix times q. The cuts will be in the form of

$$\theta \geq \theta^0 - q \sum_{p \in P, e \in A^p} (l_e^p - \bar{l}_e^p)\pi_e^p \tag{7}$$

and

$$0 \geq \theta^0 - q \sum_{p \in P, e \in A^p} (l_e^p - \bar{l}_e^p)\pi_e^p, \tag{8}$$

where π_e^p is the dual variable associated with the constrain (1d).

The master is

$$\begin{aligned}
\text{minimize} \quad & \sum_{p \in P} v^p a^p + \theta && && (9a) && \\
\text{subject to} \quad & l_e^p \leq a^p && p \in P, e \in A^p, && (9b) && \texttt{\{bm:eqs:air_active\}} \\
& \sum_{e \in A^p} l_e^p \ell_e \leq L, && p \in P, && (9c) && \texttt{\{bm:eqs:max_length\}} \\
& \sum_{(i,j) \in E^p} l_{ij}^p = \sum_{(j,i) \in E^p} l_{ji}^p, && p \in P, i \in N^p, && (9d) && \texttt{\{bm:eqs:air_cycle\}} \\
& U_i^p - U_j^p \leq |N^p|(s_i^p + s_j^p + 1 - l_{ij}^p) - 1, && p \in P, (i,j) \in E^p, && (9e) && \texttt{\{bm:eqs:MTZ\}} \\
& \sum_{i \in N^p} s_i^p = a^p && p \in P, && (9f) && \texttt{\{bm:eqs:smax\}} \\
& \theta \geq \theta^0 - q \sum_{p \in P, e \in A^p} (l_e^p - \bar{l}_e^p)\pi_e^p && \pi, \bar{l}, \theta^0 \in \mathcal{P}, && (9g) && \\
& 0 \geq \theta^0 - q \sum_{p \in P, e \in A^p} (l_e^p - \bar{l}_e^p)\pi_e^p, && \pi, \bar{l}, \theta^0 \in \mathcal{R}, && (9h) &&
\end{aligned}$$

where $\mathcal{P}$ and $\mathcal{R}$ are the extreme cones and rays with their corresponding objective and evaluation point, and the Benders sub-problem is

$$
\begin{aligned}
SP(\bar{l}) = \quad & \text{minimize} && \sum_{k\in K}\sum_{e\in E} c_e x_e^k + \sum_{p\in P, i\in V} g u_i^{pk} + h d_i^{pk} && && \text{(10a)} \\
& \text{subject to} \\
& \sum_k x_e^k \le m_e, && && e \in E, && \text{(10b)} && \texttt{\{sp:eqs:land\_const\}} \\
& -\sum y_e^{pk} \ge -q\bar{l}_e^p, && && p \in P, e \in A^p, && \text{(10c)} && \texttt{\{sp:eqs:air\_cap\}} \\
& \delta_i^k + \sum_{p\in P} u_i^{pk} - d_i^{pk} = \sum_{(i,j)\in E} x_{ij}^k - \sum_{(j,i)\in E} x_{ji}^k && && i \in V, && \text{(10d)} && \texttt{\{sp:eqs:flow\_land\}} \\
& u_i^{pk} + \sum_{(j,i)\in A^p} y_{ji}^{pk} = d_i^{pk} + \sum_{(i,j)\in A^p} y_{ij}^{pk} && && p \in P, i \in N^p. && \text{(10e)} && \texttt{\{sp:eqs:flow\_air\}}
\end{aligned}
$$

The subproblem is called the multi-commodity min cost flow problem.

4 Dantzig-Wolfe decomposition and Column generation

For any linear problem in the form

$$
\begin{aligned}
& \text{minimize} && c \cdot x && \text{(11a)} \\
& \text{subject to} && Ax \le b, && \text{(11b)} \\
& && Bx \le c. && \text{(11c)}
\end{aligned}
$$

We can rewrite it as

$$
\begin{aligned}
& \text{minimize} && c \cdot x && \text{(12a)} \\
& \text{subject to} && x = \sum_{i=1,\dots,n} \lambda^i y^i + \sum_{i=1,\dots,m} \mu^i r^i, && \text{(12b)} \\
& && \sum_{i=1,\dots,n} \lambda^i = 1, && \text{(12c)} \\
& && \lambda, \mu \ge 0, && \text{(12d)} \\
& && Bx \le c, && \text{(12e)}
\end{aligned}
$$

where $y_1, \dots, y_n$ are the extreme points of

$$P = \{x | Ax \le b\} \tag{13}$$

and $\mu_1, \dots, \mu_m$ are extreme rays of P. This is called the Dantzig-Wolfe decomposition [DW60] and has been shown (by whom?) that the new problem is

equivalent to the old one. The interest of this process is when a small subset of extreme points are active or convexifying the constraint $Ax \leq b$ leads to a special problem that breaks down into small problems.

Column generation is the process of dynamically generating variables as needed. Let an LP like

$$\text{minimize} \quad \sum_{i=1,\ldots,n} c_i x_i \tag{14a}$$

$$\text{subject to} \quad \sum_{i=1,\ldots,n} a_i^j x_i \leq b^j \quad j \in 1,\ldots,m, \tag{14b}$$

assume that the problem has solution if we set x_i $i = k,\ldots,n$ to zero. We start with the reduced problem with the variables k to n set to zero, solve the problem and add a new variable that can improve the objective. We can find variables that improve the reduced cost by looking at their reduced cost, we add a variables that has a negative reduced cost (what about zero?). It is possible to either enumerate all x (if it is finite) or if the columns a_i have special structure, we can reformulate the problem as an optimization problem over that special structure.

We fist decompose the multi-comodity flow problem with Dantzig-Wolfe into the arc-cycle formulations. Let the multi-comodity flow problem be

$$\text{minimize} \quad \sum_{k\in K, e\in E} c_e x_e^k \tag{15a}$$

$$\text{subject to} \quad Nx^k = b^k \quad k \in K, \tag{15b}$$

$$\sum_{k\in K} x_e^k \leq u_e \quad e \in E, \tag{15c}$$

where $Nx = b^k$ is the flow conservation constraint for kth commodity, $x_e^k \leq u_e$ is the arc capacity constraint and $c \geq 0$. The extreme points of $Nx = b^k$ are the paths from the source the destination of the commodity (with the flow equal to the demand), the extreme rays are cycles. We know this is true because any valid flow matrix can be broken into simple paths and cycles (elaborate?). Thus we can reformulate the problem in term of paths and cycles. We also note that we considers paths with unite flow instead of paths with demand flow to simplify the formulation. Let v^k be the demand of the kth commodity. However, the set of cycles is very large and has non negative cost as $c \geq 0$ thus we don't need to consider them. Let P^k be the set of possible paths for the commodity k, for path p, let $f(p)$ be the flow, $c(p)$ the cost and $\delta_e(p)$ an indicator of the presence of the arc e in the path p. We can rewrite the problem in term of paths (or

extreme points) as

$$\text{minimize} \quad \sum_{k\in K, p\in P^k} c(p)f(p), \tag{16a}$$

$$\text{subject to} \quad \sum_{k\in K, p\in P(k)} -\delta_e(p)f(p) \geq -u_e, \quad e \in E, \tag{16b}$$ {path:cap}

$$\sum_{p\in P^k} f(p) = d^k, \qquad k \in K., \tag{16c}$$ {path:cost}

$$f(p) \geq 0, \qquad k \in K, p \in P^k \tag{16d}$$

Due to large number of paths and many extremely useless paths, we use column generation to generate a subset of all the paths. Let w_e be the dual variable of constraint (16b) and σ be the dual of (16c), we can see that in this form the dual w_{ij} and σ_k are positive are positive. The second dual variable σ will be the least expensive used path ($\max \sum \sigma$ s.t. $\sigma^k \leq c(p)$). Assuming we can split the rhs matrix into A and B, the reduced cost of a path $p \in P^k$ is be

$$c(p) + A^\top w - B^\top \sigma, \tag{17}$$

where $A^\top w$ is $\sum_{e\in p} w_e$ and $B^\top \sigma$ is σ^k. Thus the reduced of a path cost is

$$\sum_{e\in E} \delta_e(p)(c_e + w_e) - \sigma^k. \tag{18}$$

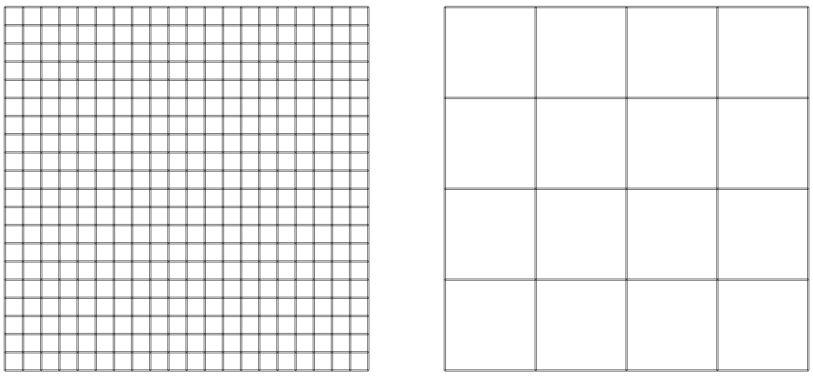
For any commodity k, this amounts to finding a shortest path with adjusted costs since σ_k is constant. Since w is positive, the cost will stay positive and we can use Djikstra's algorithm [Dij59] to find the shortest path. At each iteration we'll a shortest path to the set of basis. While in theory we only need to add one path who has a negative reduced cost, this helps accelerate things.

While the usual approach is to use the revised simplex method and to discard the generated columns when they go out of the basis, we use a solver and keep those in case they become useful later on.

To initialize the problem let M be a number strictly greater than the objective of the problem, we add an auxiliary variable for each commodity such that the cost of this commodity is M/d^k and the coef. in the constraint (16c) is 1. This allows to not start from a set of feasible columns. This is an adaptation of the so called two phase method [Wag56]. Furthermore, the initial set of column is generated via the Ford-Fulkerson algorithm [FFed].

5 Computational Study

We generate a series of instances where to mesh network is an n by n grid and the pigeons can move in another grid where to the nodes of this other grid is the nodes $s(i, j) + 1$ of the original network for a number s. So for instance the instance $(n, s) = (5, 2)$ will have a five by five mesh and the pigeons can visit the node $(1, 3, 5) \times (1, 3, 5)$. each node has distance of 10 meter.



(a) mesh network (b) pigeon paths

Figure 1: network for $(n, s) = (21, 5)$, note that the nodes with same relative position as the top right node are connected so for instance the middle node of both graph are connected.

For each of these instances we draw the arc cost from a uniform distirbution on the integer between 100 and 200. The paths are sampled using random walks. At each node, we consider all transition that have the potential to reach the destination w/o passing over already passed nodes. Out of all the possible transitions, we sample transitions with a probability proportional to the distance of that node to the destination. On this network the origin destination uniformly uniformly. The demand is sampled from a Poisson with $\lambda = 10$ truncated between 1 and infinity. We then add the the demand to the capacity of all arcs that the path pass through. We repeat this process 5 times for each pair of node. At the end we multiply the capacity with an exponential truncated between 1 et 1.05.

While it may not have been clear, we assume that all of the data can flow in the mesh network, this seems to make the problem much more interesting as there is a very large gap between the optimal solution and the worst feasible solution. This also implies that there are no Benders feasibility cuts.

We set the upload cost to 1, the download cost to 0, the pigeon data capacity to 1e4, pigeon cost to 1e5 and the pigeon range to 5e3 meters. We run everything one time with a warm up phase on a small problem. The warm up phase is needed as Julia specializes (recompile the function for that specific type). We compare our method with gurobi and cplex(to come) on a i7-11700K with turbo and 32GB of ram with Linux.

6 Conclusion

We proposed a new benchmark that solvers don't seem to handle very well by default and solved it with Benders and Dantzig-Wolfe decomposition.

{t:3}

n	m	col-Gurobi	col-CPLEX	Gurobi	CPLEX
6	10	0.01	0.00	0.01	0.01
6	100	0.01	0.01	0.06	0.06
6	500	0.03	0.02	0.36	0.31
6	1000	0.08	0.04	0.73	1.64
11	10	0.02	0.02	0.03	0.02
11	100	0.02	0.01	0.21	0.22
11	500	0.04	0.03	1.54	2.15
11	1000	0.07	0.05	3.48	5.10
21	10	0.08	0.06	0.10	0.07
21	100	0.03	0.03	1.27	1.11
21	500	0.17	0.17	9.76	16.67
21	1000	0.58	0.12	24.53	42.58
31	10	0.16	0.18	0.28	0.29
31	100	0.11	0.11	4.17	9.38
31	500	0.27	0.28	33.35	71.22
31	1000	0.54	0.48	81.68	164.64
41	10	1.54	2.19	0.81	0.90
41	100	0.33	0.34	9.47	48.59
41	500	0.29	0.32	91.74	135.12
41	1000	0.65	0.54	238.36	318.68
51	10	6.62	4.85	4.31	1.71
51	100	0.40	0.40	28.12	74.64
51	500	1.10	1.04	188.53	347.60

Table 1: Time comparison between different multi-commodity solver (zero means less than one tenth of a second), time is in second in an n by n grid with m commodity.

References

[Abe] John C. Abell. "In Africa, A Pigeon Transfers Data Faster Than The Internet". In: *Wired* (). ISSN: 1059-1028. URL: https://www.wired.com/2009/09/in-africa-a-pigeon-transfers-data-faster-than-the-internet/ (visited on 04/16/2022).

[Ben62] J. F. Benders. "Partitioning Procedures for Solving Mixed-Variables Programming Problems". In: *Numerische Mathematik* 4.1 (Dec. 1, 1962), pp. 238–252. ISSN: 0945-3245. DOI: 10.1007/BF01386316. URL: https://doi.org/10.1007/BF01386316 (visited on 04/24/2022).

[Dij59] E. W. Dijkstra. "A Note on Two Problems in Connexion with Graphs". In: *Numerische Mathematik* 1.1 (Dec. 1, 1959), pp. 269–271. ISSN: 0945-3245. DOI: 10.1007/BF01386390. URL: https://doi.org/10.1007/BF01386390 (visited on 04/24/2022).

[DW60] George B. Dantzig and Philip Wolfe. “Decomposition Principle for Linear Programs”. In: *Operations Research* 8.1 (Feb. 1960), pp. 101–111. ISSN: 0030-364X. DOI: 10.1287/opre.8.1.101. URL: https://pubsonline.informs.org/doi/abs/10.1287/opre.8.1.101 (visited on 04/24/2022).

[FFed] L. R. Ford and D. R. Fulkerson. “Maximal Flow Through a Network”. In: *Canadian Journal of Mathematics* 8 (1956/ed), pp. 399–404. ISSN: 0008-414X, 1496-4279. DOI: 10.4153/CJM-1956-045-5. URL: https://www.cambridge.org/core/journals/canadian-journal-of-mathematics/article/maximal-flow-through-a-network/5D6E55D3B06C4F7B1043BC1D82D40764 (visited on 04/24/2022).

[MD77] Dale McDaniel and Mike Devine. “A Modified Benders’ Partitioning Algorithm for Mixed Integer Programming”. In: *Management Science* 24.3 (Nov. 1977), pp. 312–319. ISSN: 0025-1909. DOI: 10.1287/mnsc.24.3.312. URL: https://pubsonline.informs.org/doi/abs/10.1287/mnsc.24.3.312 (visited on 04/24/2022).

[MTZ60] C. E. Miller, A. W. Tucker, and R. A. Zemlin. “Integer Programming Formulation of Traveling Salesman Problems”. In: *Journal of the ACM* 7.4 (Oct. 1, 1960), pp. 326–329. ISSN: 0004-5411. DOI: 10.1145/321043.321046. URL: https://doi.org/10.1145/321043.321046 (visited on 04/24/2022).

[MW81] T. L. Magnanti and R. T. Wong. “Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria”. In: *Operations Research* 29.3 (June 1981), pp. 464–484. ISSN: 0030-364X. DOI: 10.1287/opre.29.3.464. URL: https://pubsonline.informs.org/doi/abs/10.1287/opre.29.3.464 (visited on 04/24/2022).

[Pap08] Nikolaos Papadakos. “Practical Enhancements to the Magnanti–Wong Method”. In: *Operations Research Letters* 36.4 (July 1, 2008), pp. 444–449. ISSN: 0167-6377. DOI: 10.1016/j.orl.2008.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0167637708000102 (visited on 04/24/2022).

[Wag56] Harvey M. Wagner. “A Two-Phase Method for the Simplex Tableau”. In: *Operations Research* 4.4 (1956), pp. 443–447. ISSN: 0030-364X. JSTOR: 167312.

{t:4}

n	m	s	p	BDW	BGG	BGC	Gurobi	CPLEX
6	10	10	1	0.0	0.1	0.1	0.0	0.1
6	10	10	5	0.1	0.4	0.3	0.1	0.1
6	10	10	10	0.1	0.3	0.3	0.1	0.1
6	10	5	1	0.1	0.1	0.1	0.0	0.0
6	10	5	5	0.1	0.4	0.3	0.0	0.1
6	10	5	10	0.1	0.4	0.4	0.1	0.1
6	100	10	1	0.1	0.8	0.7	0.2	0.2
6	100	10	5	0.2	1.8	1.7	0.4	0.6
6	100	10	10	0.2	2.3	2.2	0.8	1.7
6	100	5	1	0.1	0.9	0.7	0.2	0.2
6	100	5	5	0.2	1.8	1.7	0.4	0.7
6	100	5	10	0.2	2.4	2.1	0.8	1.6
6	500	10	1	0.3	4.3	6.3	1.1	1.4
6	500	10	5	0.5	18.4	11.1	10.1	9.7
6	500	10	10	0.6	10.2	12.8	86.6	11.4
6	500	5	1	0.3	4.2	6.3	1.2	1.4
6	500	5	5	0.5	18.4	11.2	10.4	9.8
6	500	5	10	0.5	10.0	15.5	114.7	13.6
6	1000	10	1	0.2	18.5	23.9	16.6	6.4
6	1000	10	5	0.3	74.9	77.0	596.3	35.8
6	1000	10	10	0.4	196.0	229.1	0.0%	114.7
6	1000	5	1	0.2	17.6	23.6	16.7	6.4
6	1000	5	5	0.3	74.0	76.9	598.7	36.3
6	1000	5	10	0.4	196.0	228.7	0.0%	116.3
11	10	10	1	0.1	0.8	0.5	0.1	0.1
11	10	10	5	1.2	1.1	0.8	0.5	0.7
11	10	10	10	4.2	0.8	1.4	1.3	1.3
11	10	5	1	0.2	0.9	0.6	0.1	0.1
11	10	5	5	1.2	1.1	0.8	0.6	0.7
11	10	5	10	4.3	0.9	1.4	1.3	1.3
11	100	10	1	0.4	7.6	14.7	1.1	2.1
11	100	10	5	1.7	18.3	55.9	111.1	22.3
11	100	10	10	5.9	24.2	18.9	75.3	223.5
11	100	5	1	0.3	7.6	14.1	1.1	2.1
11	100	5	5	1.7	18.0	54.8	111.0	21.8
11	100	5	10	6.0	23.7	19.0	76.6	222.3

Table 2: Time/duality gap comparison between different approaches with a time limit of 10 minutes. First 3 columns are Benders decomposition, the subproblem was solved by Column Generation (DW), Gurobi (G), or CPLEX (C). If the solver reached optimality, only the time was reported otherwise the GAP, in percentage, was reported. Empty means no primal solution or very large optimality gap by the solver.

{t:4}

n	m	s	p	BDW	BGG	BGC	Gurobi	CPLEX
11	500	10	1	1.8	329.1	189.4	153.8	68.4
11	500	10	5	4.4	361.1	405.0	44.4 %	0.0 %
11	500	10	10	14.1	33.7 %	21.1 %		27.4 %
11	500	5	1	1.8	332.7	189.5	159.9	68.9
11	500	5	5	4.4	371.3	403.1	44.4 %	0.0 %
11	500	5	10	14.1	33.7 %	21.1 %		27.4 %
11	1000	10	1	15.6	48.8 %	259.7	7.9 %	413.3
11	1000	10	5	23.2	726.8	37.8 %		26.8 %
11	1000	10	10	120.3	100.0 %	100.0 %		44.7 %
11	1000	5	1	15.3	48.8 %	257.8	7.9 %	405.3
11	1000	5	5	22.4	719.8	37.8 %		26.8 %
11	1000	5	10	118.8	100.0 %	100.0 %		44.7 %
21	10	10	1	2.3	4.9	3.6	0.6	1.0
21	10	10	5	27.7	11.4	5.6	11.6	8.3
21	10	10	10	144.3	13.9	5.9	26.3	17.4
21	10	5	1	2.3	5.0	3.5	0.5	1.0
21	10	5	5	27.5	11.3	5.4	11.7	8.3
21	10	5	10	144.8	14.2	5.9	26.0	17.4

Table 3: Time/duality gap comparison, part 2.