Submitted to *INFORMS Journal on Computing*
manuscript (Please, provide the manuscript number!)

Appendix to Computing Bi-Path Multi-Commodity Flows with Constraint Programming-based Branch-and-Price-and-Cut

Jiachen Zhang
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada, jasonzjc@mie.utoronto.ca

Youcef Magnouche, Pierre Bauguion, Sebastien Martin
Huawei Technologies, Paris Research Center, Boulogne-Billancourt, France, {firstname.lastname}@huawei.com

J. Christopher Beck
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada, jcb@mie.utoronto.ca

1. Instance Generation

The majority of problem instances for experiments are generated from two well-known libraries: *SNDlib*, a library for telecommunication network design (`http://sndlib.zib.de/home.action0`; and *The Internet Topology Zoo*, a collection of real networks (`http://www.topology-zoo.org/index.html`). The rest of the instances are generated randomly. The three sets of instances follow a similar generation scheme. We summarize the similar aspects first and then the different aspects of the three generation processes.

(i) The delay of each arc is related to the arc capacity and the distance between the origin and destination of the arc, according to the coordinates of the two nodes. Specifically, the delay is the sum of the queuing delay d_a^q and the propagation delay d_a^p. Assuming an

M/M/1 queue (Kuhn and Mostafavi 2008) and at most 70% link utilization, we use the following formula to generate the delay of arc a:

$$d_a = \lambda_0 \times (d_a^q + d_a^p) = \lambda_0 \times (\frac{1}{0.3 \times c_a} + \frac{D_{a^-,a^+}}{v_p}) \tag{1}$$

where c_a is the capacity, D_{a^-,a^+} is the distance, and v_p is the propagation speed of demands along a link. Assuming the same material for physical links, we use the constant propagation speed $v_p = 2 \times 10^8$ m/s. λ_0 is a normalization factor that regulates d_a to be in the range of $[1, 100]$ in order to ease handling numerical issues of optimization solvers.

(ii) To make sure the number of feasible demands (commodities) is not trivially small, a reverse link is created for each link if it does not directly exist in the network to implicitly enable multiple elementary paths without modifying the topology too much. The capacity, primary cost, and secondary cost of the reverse link are the same as the original link.

(iii) One demand is generated in advance for each pair of different vertices in the network, with the bandwidth randomly selected from a pool of demand bandwidth. The pool is constructed differently according to the data sources as described below.

(iv) Not all demands are feasible (e.g., some infeasibility is due to impossible arc-disjointness). With the total number of feasible demands being K_f for each instance, three configurations $\{K_f/1, K_f/2, K_f/3\}$ are used in different instances. By using a simple greedy heuristic that routes demands one by one, the total number of feasible demands K_f for each delay configuration is determined.

(v) For the threshold of the delay differences of each demand, Yen's algorithm (Yen 1971) is used to find the top 40 delay-shortest paths. Then the delay difference between the 10^{th} and the $\{20^{th}, 30^{th}, 40^{th}\}$ paths are used as the delay threshold. We adopt this setting to guarantee that the delay difference threshold allows multiple paths for the demand. The demand is discarded if there are fewer than 40 paths from its origin to its destination. Finally, to ease the comparison of total routing costs between delay configurations, instances obtained from the same topology but different values of maximum delays have exactly the same demands.

1.1. SNDlib Instances.

SNDlib originates from real telecommunication scenarios and contains network topologies with multiple arc capacities, arc costs, and demands. We select 10 large topologies from

SNDlib to avoid trivially small instances. They are {france, geant, germany50, giul39, janos-us-ca, nobel-eu, norway, sun, ta1, zib54}. The first capacity candidate and the cost of each arc provided in SNDlib are directly used as the capacity and primary arc cost in bi-path MCF. The secondary arc costs are generated based on two different schemes. The first scheme adapts classical IGP costs (Wijaya 2011) in a telecommunication context with the following formula:

$$\beta_a = \frac{\max_a\{c_a\}}{c_a}. \tag{2}$$

Since the arc capacities in each selected instance from SNDlib are the same, the IGP cost of an arc is always 1, i.e., $\beta_a = 1, \forall a \in A$. The second scheme omits the secondary cost of arcs when the costs are not really important by setting them to 0, i.e., $\beta_a = 0, \forall a \in A$. Since SNDlib provides demands with bandwidth, the pool of bandwidths is the union of the demand bandwidths in the problem instance.

The coordinates provided by SNDlib are either geographical coordinates in longitude and latitude or pixel coordinates in pure values.

1.2. The Internet Topology Zoo Instances.

The Internet Topology Zoo provides networks from all over the world and we also select 10 topologies. They are {Abvt, Bandcon, Canerie, Evolink, Goodnet, HurricaneElectric, Ibm, Quest, Rediris, Xeex}. These networks do not have capacity or costs and so we have to generate them. Specifically, the capacity of each arc is determined by the arc's connectivity. For an arc a from u to v, the sum of the in-degree of u and the out-degree of v is used as a reference, i.e.,

$$c_a = \begin{cases} 5000 & 0 \le |\delta^{\text{in}}(u)| + |\delta^{\text{out}}(v)| < 3, \\ 10000 & 3 \le |\delta^{\text{in}}(u)| + |\delta^{\text{out}}(v)| < 5, \\ 20000 & 5 \le |\delta^{\text{in}}(u)| + |\delta^{\text{out}}(v)| < 9, \\ 50000 & 9 \le |\delta^{\text{in}}(u)| + |\delta^{\text{out}}(v)|. \end{cases} \tag{3}$$

The pool of demand bandwidths is $\{50, 100, 500, 1000\}$. The interpretation of this setting is that demands in MBs are routed in the network with capacities in GBs. The IGP costs are used as primary arc costs and the hop numbers are used as secondary costs, i.e., $\beta_a = 1, \forall a \in A$. Another configuration of the secondary costs is the all-zero costs, i.e.,

$\beta_a = 0, \forall a \in A$. Note that some of the nodes do not have coordinates. We generate their coordinates according to their neighbors by the following formulas.

$$x_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} x_j; \quad y_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} y_j. \tag{4}$$

where $\mathcal{N}(i)$ is the set of node neighbors (i.e., $\exists j, (i,j) \in E$ or $(j,i) \in E$) of node i.

1.3. Random Instances.

The random instances follow the same generating scheme as the Internet Topology Zoo instances for capacities, bandwidth, primary costs, and secondary costs. However, the vertices, coordinates, and links are randomly generated. The ranges of $x-$coordinates and $y-$coordinates are $[0, 100]$ and $[0, 100]$, respectively. The number of nodes in generated instances varies in $\{20, 25, 30\}$ and the density of graph varies in $\{0.3, 0.4, 0.5\}$.

In summary, we generate $10 \times 2 \times 3 \times 3 = 180$ instances from SNDlib, $10 \times 2 \times 3 \times 3 = 180$ instances from the Internet Topology Zoo, and $3 \times 3 \times 2 \times 3 \times 3 = 162$ instances randomly. All the 522 instances can be found online (`https://github.com/jasonzjc1995/Instances-of-Bi-Path-MCF`).

References

Kuhn R, Mostafavi SM (2008) Optimal routing policy. *IEEE Communications letters* 12(3):222–224.

Wijaya C (2011) Performance analysis of dynamic routing protocol EIGRP and OSPF in IPv4 and IPv6 network. *2011 First International Conference on Informatics and Computational Intelligence*, 355–360 (IEEE).

Yen JY (1971) Finding the k shortest loopless paths in a network. *Management Science* 17(11):712–716.