Subnet Generation Problem: A New Network Routing Problem

Student: Cheuk Fun Bede Leung
Supervisors: Barry Richards and Olli Kamarainen

IC-Parc, Imperial College London, London SW7 2AZ, UK,
cheuk.leung@icparc.ic.ac.uk

Abstract. We introduce a new type of network routing problem, the subnet generation problem (SGP). The SGP is a special case of a well-known network routing problem, the traffic placement problem (TPP). The aim of the TPP is to place a given set of point-to-point traffic demands onto a given network composed of routers and links. The SGP imposes additional constraints that restrict the combinations of demands which can be placed onto the network. We describe a structure to split the problem into a master problem and a subproblem: the task of the master problem is to select a set of demands to be placed; whereas the task of the subproblem is to compute a set of feasible paths for the demands selected by the master problem. To solve the SGP, we first apply a state-of-art hybrid tree search (HTS) algorithm, designed to solve the TPP. Then, we explore how the additional constraints in the SGP could be used to solve the problem more efficiently. In particular, we focus on balancing search between the master problem and the subproblem to maximize the solution quality within a fixed computational time.

1 Introduction

We introduce a new type of network routing problem, the subnet generation problem (SGP) which is a special case of a well-known network routing problem, the traffic placement problem (TPP). The TPP is a type of knapsack problem: *place* (and compute paths for) a set of point-to-point traffic demands onto the network such that the sum of the costs of the *unplaced* demands (the demands which were left without feasible paths) is minimized. The SGP is a TPP with an extra set of constraints that restricts the combinations of demands to be placed. The constraints arise from the fact that every node in a selected group of routers (of the SGP) is required to be a client and a server. This is a feature of emerging applications in the communication networking, including the Internet, where hosts send information to other hosts and receive information from them. An example of this is a command, control, and communication (C^3) [4] application, where the aim is to find a maximal communication group such that every host in that group shares the same information concurrently.

2 Problem description

We first describe the TPP and its search space, and then the SGP that we will focus on. Finally, we detail the key difference between the TPP and the SGP.

2.1 Traffic placement problem (TPP)

The underlying network topology consists of a set of routers $\mathbf{N}$ and a set of links $\mathbf{E}$. The link from router $i \in \mathbf{N}$ to router $j \in \mathbf{N}$ is denoted by $e(i,j) \in \mathbf{E}$. Each link $e \in \mathbf{E}$ has a bandwidth capacity c_e and a propagation delay d_e. For each router n there is a set of links $\mathbf{IN}(n) = \{e(i,n) \in \mathbf{E}\}$ entering n and a set of links $\mathbf{OUT}(n) = \{e(n,j) \in \mathbf{E}\}$ leaving n. A given set of point-to-point traffic demands $\mathbf{K}$ are to be placed onto the network. A demand from router $s \in \mathbf{N}$ to router $d \in \mathbf{N}$ has a bandwidth requirement b_{sd}, a maximum propagation delay max_d_{sd} and a cost for leaving the demand unplaced. The objective is to minimize the sum of costs of the unplaced demands subject to the routing constraints. The routing constraints include flow path constraints (ensuring a single path from the source to the destination), flow delay constraints (the delay along a path must not exceed the maximum propagation-delay threshold) and bandwidth capacity constraints (for each link, the bandwidth required by all flows must not exceed its capacity).

In order to find out the optimal set, we may have to step through all the possible combinations of demands. For example, if there are $|\mathbf{K}|$ demands, there will be $2^{|\mathbf{K}|}$ possible sets to consider. For each of the possible sets, we have to find a path for each of the demands such that the set of paths is feasible, i.e. it satisfies the routing constraints. By doing this, the search has to consider all combinations of possible paths for the selected demands. For example, if i demands are selected to be placed, and each demand can be potentially routed in $2^{|\mathbf{E}|}$ ways[1], then the total search space of placing i demands is $2^{|\mathbf{E}|i}$. Thus, the TPP involves in a huge search space even with a small number of demands in a small network: $\sum_{i=0}^{|\mathbf{K}|} \frac{|\mathbf{K}|!}{i!(|\mathbf{K}|-i)!} 2^{|\mathbf{E}|i}$.

2.2 Subnet generation problem (SGP)

In the SGP, an extra set of constraints is introduced to the TPP. These constraints, namely *demand set constraints*, will forbid certain combinations of demands to be placed. In other words, the selection of demands to be placed may be dependent on the demands that are already selected.

Now, we define the SGP we are going to focus on. In the SGP, each router $s \in \mathbf{N}$ has a given, fixed amount of information-gain g_s that is to be transmitted to every other router in the subnet. A *subnet* is defined as any subset of the routers in the network. This means that every router in the subnet will have exactly the same total information-gain (its own information-gain plus the information-gain received from all other routers in the subnet). The objective of the SGP is to

[1] This is the potential search space without considering any of the routing constraints.

find a subnet that maximizes the total information-gain such that the demands among the routers in the subnet get paths that satisfy the routing constraints. The SGP can be modelled as follows.

$$\max_{\{X_s, F^{sd}, P_e^{sd}\}} \sum_{s \in \mathbf{N}} g_s X_s$$

$$st. \begin{cases} \forall s \in \mathbf{N}, \forall d \in \mathbf{N}, s \neq d: \quad F^{sd} = X_s X_d & (1) \\ \forall n \in \mathbf{N}, \forall s \in \mathbf{N}, \forall d \in \mathbf{N}, s \neq d: & \\ \quad \sum_{e \in \mathbf{OUT}(n)} P_e^{sd} - \sum_{e \in \mathbf{IN}(n)} P_e^{sd} = \begin{cases} -F^{sd} & n = d \\ F^{sd} & n = s \\ 0 & \text{otherwise} \end{cases} & (2) \\ \forall s \in \mathbf{N}, \forall d \in \mathbf{N}, s \neq d: \quad \sum_{e \in \mathbf{E}} d_e P_e^{sd} \leq max\_d_{sd} & (3) \\ \forall e \in \mathbf{E}: \quad \sum_{s \in \mathbf{N}} \sum_{d \in \mathbf{N}, d \neq s} b_{sd} P_e^{sd} \leq c_e & (4) \end{cases}$$

In the above model, the 0/1 integer variables X_s, F^{sd} and P_e^{sd} state whether s is a member of the subnet, whether the demand from s to d exists within the subnet, and whether the link $e \in \mathbf{E}$ is used for the path of the demand from s to d respectively. Constraints (1) denote demand set constraints, (2) flow path constraints, (3) flow delay constraints, and (4) bandwidth capacity constraints.

For example, Fig. 1(a) shows a given network; Fig. 1(b) indicates a subnet with 3 routers resulting in 6 demands; and Fig. 1(c) shows a path solution statisfying the flow path constraints.

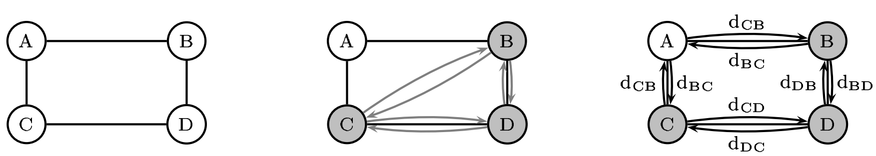

Fig. 1. (a) a network; (b) a subnet and the created demands; (c) a path solution

In the SGP, the information-gain at each router creates a potential demand to every other router in the network. If there are $|\mathbf{N}|$ routers in the network, then there are $|\mathbf{N}|(|\mathbf{N}| - 1)$ possible demands. This constitutes the given set of demands, corresponding to $|\mathbf{K}|$ in the TPP. This means that in the SGP, there are $2^{|\mathbf{N}|(|\mathbf{N}|-1)}$ possible subsets of demands to consider. Although there are $2^{|\mathbf{N}|(|\mathbf{N}|-1)}$ subsets, only $2^{|\mathbf{N}|}$ of these subsets correspond to a subnet. Thus, there are only $2^{|\mathbf{N}|}$ possible subsets that need to be considered. The size of placing i demands will remain $2^{|\mathbf{E}|i}$ as in the TPP, but comparing to the TPP, the entire search space of the SGP is reduced radically from double exponential ($\sum_{i=0}^{|\mathbf{N}|(|\mathbf{N}|-1)} \frac{(|\mathbf{N}|(|\mathbf{N}|-1))!}{i!(|\mathbf{N}|(|\mathbf{N}|-1)-i)!} 2^{|\mathbf{E}|i}$) to exponential ($\sum_{i=0}^{|\mathbf{N}|} \frac{|\mathbf{N}|!}{i!(|\mathbf{N}|-i)!} 2^{|\mathbf{E}|i(i-1)}$). However, in practice, a full mesh of demands between every pair of routers given to the TPP is rare, thus a real-life SGP may have a larger search space.

3 Problem decomposition

By the nature of the TPP and the SGP, we can decompose them into a master problem and a series of subproblems. Here, we will explain how we will do this.

Master problem. The master problem consists of the original objective function, and its purpose is to select a set of demands to be placed. In the TPP, the selected set of demands can be in any combination; whereas in the SGP, it has to respect the demand set constraints. Then, the selected set of demands will be sent to the subproblem.

Subproblem. The subproblem is an integer multicommodity flow problem (IMCFP) [1] where *all the demands must be placed*. The IMCFP with bandwidth capacity constraints is known to be NP-hard [3], and thus both the TPP and the SGP are also NP-hard. In our case, the subproblem is simply a feasibility problem aiming at computing a path for each of the demands supplied by the master problem while satisfying all the routing constraints.

4 Algorithm

In the previous section, we showed that the SGP is a special case of the TPP. Thus applying an existing TPP algorithm to solve the SGP may be a good starting point. Among the existing TPP algorithms, we select hybrid tree search (HTS) [5] for our investigation. The HTS efficiently combines local search and tree search, and it is very fast and scaleable due to its powerful use of constrained shortest path first algorithm (CSPF) which is an extension of Dijkstra's shortest path algorithm [2] such that it satisfies the routing constraints. HTS also uses the decomposition structure above. Here, we describe the HTS designed to solve the TPP and then we will modify it with minimal changes to tackle the SGP.

4.1 HTS for the TPP

The HTS from [5] can be briefly described as follows.

Phase 1: Initial CSPF solution. The purpose of this phase is to obtain quickly a good bound for the objective function. The demands will be ordered according to the cost, largest first. Then they will be placed one by one, using CSPF, and the link bandwidth availabilities will be updated after each successful placement. If a feasible path cannot be found, the demand will remain unplaced and will be passed to the later phases for other attempts.

Phase 2: Incremental improvement phase. The purpose of this phase is to obtain a better bound. This phase will reconsider the unplaced demands from the previous phase. Now CSPF computes paths such that capacity violations are allowed and *local search* will be called for restoring consistency: it performs a series of "reroute calls" to recompute paths using CSPF for problematic demands - trying to divert the traffic to less congested links. If consistency cannot be restored, the demand will remain unplaced.

Phase 3: Hybrid tree search. A branch-and-bound binary search tree will be constructed aiming at minimizing the sum of the total cost of the unplaced demands. This is achieved by potentially considering all the combinations of placed demands so that the optimal combination will not be missed out. At each node of the tree, a demand is selected (and placed using CSPF) and a backtrackable choice point (when backtracking, the demand will remain unplaced) is created. When placing a demand, capacity violations are allowed. If there are capacity violations, local search will be called as in phase 2. If consistency is restored, search continues on that branch; otherwise, backtrack follows. Note that the HTS is incomplete because the local search procedure may miss a feasible solution.

4.2 HTS for the SGP

As the first attempt to solve the SGP, we introduce a modified version of the HTS.

Phase 1: Initial CSPF solution. We start from an empty subnet. All routers will be ordered according to the information-gain, largest first. At each step, a router is selected and a set of additional demands between the selected router and the existing routers in the subnet are set up and ordered by the bandwidth requirement, largest first. Then, each of them will be placed using CSPF. Once a feasible path cannot be found, the router will remain outside the subnet and all the additional demands associated to it must remain unplaced.

Phase 2: Incremental improvement phase. This phase will reconsider the unplaced routers. We start from a solution obtained at phase 1. After an unplaced router is selected, a set of router-related new demands are computed[2] and ordered as in phase 1. Then they will be placed one by one using CSPF. When placing a demand, capacity violations are allowed. If there are capacity violations, consistency will be tried to get restored using local search as in the HTS for the TPP. If consistency is not restored, all the newly placed paths will be removed from the network, and the router will remain outside the subnet.

Phase 3: Hybrid tree search. Now the branch-and-bound binary search tree will try to compute the maximum total information-gain by considering all the combinations of *members in the subnet*. At each node of the tree, a router is selected (and added to the subnet) and a backtrackable choice point (the router will remain outside the subnet) is created. When a router is selected, a set of router-related new demands are computed and ordered as in phase 1. Then they will be placed one by one using CSPF. Local search will be called if there are capacity violations. Search continues to select the next router only if consistency is restored; otherwise, backtrack follows.

[2] The demands we consider may be different from the set when the same router was selected in phase 1. It is because the current subnet may be different from the one when considering this router in phase 1.

5 Ongoing work

So far, we have defined the SGP, related it to the literature and solved it by a recent state-of-art network routing algorithm, the HTS. However, the HTS for the SGP has weaknesses: (a) it does not necessarily utilize well the special structure of the SGP, namely the demand set constraints, (b) the tree search in the master problem scales badly in large problems, and (c) the local search procedure at tree search nodes may not be able to find a feasible set of paths even if one exists.

In the ongoing work, we address the weaknesses in the following way: for (a) we investigate better router selection heuristics and router-based filtering techniques to rule out capacity-infeasible subnets, for (b) we replace the tree search phase with local search that non-systemically changes the members of the subnet, and for (c) we investigate whether more effort should be put on solving the subproblems. In the SGP, there are much fewer possible combinations of demand sets than in the TPP. Thus, more computational effort could be spent on (1) heavier filtering techniques, (2) longer runs of local search for resolving the paths, or (3) executing a complete IMCFP algorithm instead of local search in the subproblem. If the time frame is tight, finding a good computational balance becomes important between the master problem (affects reaching the optimal subnet) and the subproblem (affects finding a feasible set of paths).

Acknowledgements

I would like to thank Quanshi Xia and Christophe Guettier for their participation on this work.

References

1. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, algorithms and applications.* Prentice-Hall, 1993.
2. E. W. Dijkstra. A note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269 – 271, 1959.
3. M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Company, 1979.
4. T. S. Lee, S. Ghosh, and A. Nerode. Asynchronous, Distributed, Decision-Making Systems with Semi-Autonomous Entities: A Mathematical Framework. *Systems, Man and Cybernetics, Part B*, 30(1):229–239, 2000.
5. J. Lever. A Local Search/Constraint Propagation Hybrid for a Network Routing Problem. *International Journal of Artificial Intelligence Tools*, 14(1):43–60, 2005.