ISSN 2091-1521 (Print)
2091-153X (Online)
International Journal of Operational Research/Nepal - IJORN - 2022, Vol. 10, Issue 1

The Max Static *k*-Sinks Location Problem

Hari Nandan Nath
Tribhuvan University, Bhaktapur Multiple Campus, Bhaktapur, Nepal
Email:hari672@gmail.com

Abstract

The maximum static flow problem sends a maximum amount of flow from a set of sources to a set of sinks in a network with multiple sources and multiple sinks. Because of cost or other constraints, e.g. in evacuation planning, only a limited number of sinks may have to be chosen. To address such an issue, we consider a problem of choosing a set of at most a given number of sinks from a given set of possible sinks T as a Max Static k-sinks problem. We present a non-polynomial time algorithm to solve it and show that $(T-1)$-sinks problem can be solved in a strongly polynomial time.

Keywords: network flow, static flow, shelter location, multiple sinks

1. Introduction

Network flow modeling, has been widely used to solve problems in various fields of inquiry, e.g. engineering, management, applied mathematics, and computer science [1]. Evacuation planning is one of the active research areas in which network flow modeling has been widely used. The survey of such models can be found in [2, 3]. One of the current research interests is identification of facility locations using network flow approach [4, 5]. One of the interesting problems is the shelter (sink) location problem in which appropriate sinks are chosen from among a given set of possible shelters [6, 7]. In their work, Heßler and Hamacher [6] present a model that choose shelters so as to minimize the opening cost of the shelter. Nath and Dhamala [7] present sink location models based on static and dynamic network flows to choose a single sink with the objectives of maximizing the flow or minimizing the time of evacuation.

In this work, we introduce a problem of choosing at most a given number of sinks, from among a given set of possible sinks, to maximize the static flow. The paper is organized as follows. Section 2 describes the static flow modeling on which the presented problem is based. Section 3 introduces the problem and a solution algorithm, and Section 4 concludes the work.

2. The maximum static flow problem

A network $N = (V, A)$ consists of a set V of nodes, and a set $A \subseteq V \times V$ of directed arcs. We assume that $|V| = m$ and $|A| = n$. For each $i \in V$, we define

$$A_a(i) = \{j \in V: (i,j) \in A\}$$

which denotes the set of nodes that succeed i (the nodes after i) and

$$A_b(i) = \{j \in V: (j,i) \in A\}$$

which denotes the set of nodes that precede i (the nodes before i). Associated with each arc (i,j) is a capacity u_{ij} denoting the maximum amount of flow that can flow on (i,j). Given two special node sets, a set of source nodes $S \neq \emptyset$ and a set of sink nodes $T \neq \emptyset$ such that $S \cap T = \emptyset$, a vector $x = \{x_{ij}\}$ satisfying the mass balance constraints

$$\sum_{j \in A_a(i)} x_{ij} - \sum_{j \in A_b(i)} x_{ji} = 0 \; \forall i \in V - \{S,T\} \tag{1}$$

and the capacity constraints

$$0 \leq x_{ij} \leq u_{ij} \; \forall (i,j) \in A \tag{2}$$

is known as *static flow* or simply a *flow* [8]. The value of x is defined by

$$v(x) = \sum_{s \in S} \left(\sum_{j \in A_a(s)} x_{sj} - \sum_{j \in A_b(s)} x_{js} \right) = \sum_{t \in T} \left(- \sum_{j \in A_a(t)} x_{ij} + \sum_{j \in A_b(t)} x_{jt} \right) \tag{3}$$

The flow x, that maximizes $v(x)$ is known as *maximum flow* or a *maximum static flow.*

The maximum static flow problem is a well-studied problem. Starting with the pseudo-polynomial time Ford and Fulkerson's algorithm [9], there are several improvements resulting into strongly polynomial algorithms (see [1] for such several algorithms). The algorithms solve the problem in a single-source-single-sink network ($|S| = |T| = 1$). As suggested in [9], the algorithms can be used to solve the problem in the multi-source-multi-sink network by adding two nodes, the super source s^* and the super sink t^* and the arcs (s^*, s) $\forall s \in S$,(t, t^*) $\forall t \in T$ with infinite capacity to the network. This is illustrated in Figure 1.

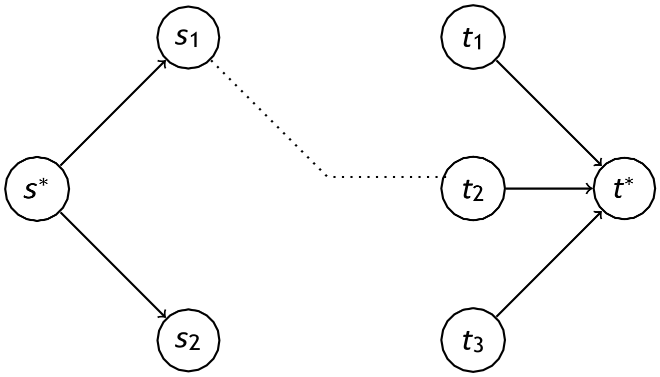

Figure 1: The super sink s^*, and the super sink t^*

3. The Max Static *k*-sinks location problem

Given a set of possible sinks, choosing a sink that maximizes the static flow is known as the Max Static Sink location problem. The problem is introduced in [7] and shown to be solved in a strongly polynomial time using a simple iterative procedure.

Sometimes, we may have to choose more than one sink from a given set of possible sinks. The optimal choice is the subset of the sinks that gives the highest value of the maximum flow. The following example illustrates the fact.

Consider a network given in Figure 2(a). Each arc-label represents the capacity of the arc. Let $S = \{s\}$. Let $\{t_1, t_2, t_3\}$ be the set of possible sinks. Suppose that we have to choose two sinks out of the three sinks. The values of the maximum static flow with the possible choices of the sinks is given in Table 1. The value of the maximum flow value is maximum when $\{t_1, t_2\}$ is considered as the sink set. For the network given in figure 2(b).

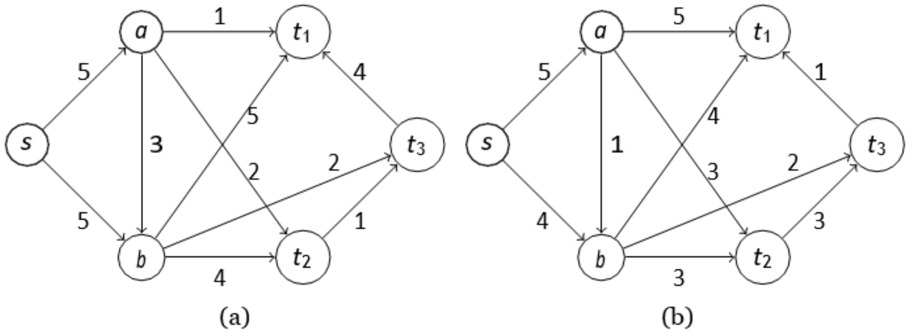

Figure 2: Example Networks

Table 1: Value of the maximum flow in Figure 2(a)

Sink set	Value of the maximum flow
$\{t_1, t_2\}$	10
$\{t_2, t_3\}$	8
$\{t_1, t_3\}$	9

The values of maximum flow are given in Table 2. The optimal sink set is $\{t_1,t_2\}$ or $\{t_2,t_3\}$. However, even if a single sink t_1 is chosen, the value of the maximum flow is 9. So, only the singleton set $\{t_1\}$ may be taken as optimal sink set.

Table 2: Value of the maximum flow in Figure 2(b)

Sink set	Value of the maximum flow
$\{t_1, t_2\}$	9
$\{t_2, t_3\}$	8

$\{t_1, t_3\}$	9

Problem: (The Max Static k-sinks location problem). Given a network $N = (V, A)$ with a set of sources S, and a set of possible sinks T. Suppose that v_X represents the value of the maximum flow from S to X for $X \subseteq T$. Let $\mathcal{T}$ be the collection of subsets of T with k sinks. The Max Static k-sinks problem is to find $T^* \subset T$ such that $v_{T^*} \geq v_X$ for each $X \in \mathcal{T}$ and $|T^*| \leq k$.

A simple procedure to solve the Max Static k-sinks location problem is given in Algorithm 1. The algorithm iterates overall all possible subsets with k sinks, finds the maximum static flow introducing the super source and the super sink. Since the maximum flow cannot be more than the sum $\sum_{t\in X}\sum_{i\in A_b(t)} u_{it}$ of the capacities of the arcs terminating in the set of sink nodes X, we calculate the maximum static flow only if the current value of the maximum flow is less than $\sum_{t\in X}\sum_{i\in A_b(t)} u_{it}$. The subset of sinks which receives the largest value of the maximum flow is chosen as the optimal set of the sinks. In the process, some sink nodes may not receive a nonzero flow as illustrated in Example 1. The optimal set of sinks is updated by removing such sinks.

Algorithm 1: Max Static k-sinks location problem

```
1.  Add $s^*$ to $V$ and $(i, s)$ to $A$ such that $u_{is^*} = \infty$ for each $i \in S$.
2.  Add $t^*$ to $V$.
3.  Let $\max\_v = -1$
4.  for $X \in \mathcal{T}$ do
5.      if $max_v < \sum_{t\in X}\sum_{t\in A_b(t)} u_{it}$ then
6.          add $(i, t^*)$ to A with $u_{it^*} = \infty$ for each $i \in X$
7.          let $x$ = maximum static $s^* - t^*$ flow.
8.          if value of $x > \max\_v$ then
9.              $\max\_v = value\ \ of\ x^*$
10.             $T^* = X$
11.         Remove $(i, t^*)$ from A for each $i \in X$
12. for $t \in T^*$ do
13.     if $x(t, t^*) = 0$ then
14.         Remove $t$ from $T^*$.
15.     Return $T^*$.
```

Theorem 1. Algorithm 1 solves the *Max static k-sinks* location problem in $O\binom{|T|}{k}(M + k)$ where $O(M)$ is the worst-case complexity of *a maximum static flow computation.*

Proof. It is clear that the algorithm runs over all the subsets of T with k sinks. The number of such subsets is $\binom{|T|}{k}$. So, in the worst case, it performs $\binom{|T|}{k}$ maximum static flow computations. In each maximum static flow computation, there are $O(k)$ addition, deletion of arcs. So, if $O(M)$ is the worst-case complexity of a maximum static flow computation, the worst-case complexity of Algorithm 1 is $\binom{|T|}{k}(M + k)$.

As shown in Theorem 1, Algorithm 1 is not a polynomial time algorithm. However, if $k = |T| - 1$, the procedure becomes strongly polynomial time.

Theorem 2. The Max-static $(|T| - 1)$-sinks problem can be solved in a strongly polynomial time.

Proof. If $k = |T| - 1$, the worst-case complexity of Algorithm 1, according to Theorem 1, is

$$O\left(\binom{|T|}{|T|-1}(M + |T|)\right) = O\big(|T|(M + T)\big).$$

A maximum static flow can be computed in a strongly polynomial time, e.g., an algorithm by Goldberg and Tarjan [10] solves the problem in $O(nm\,log(n^2/m))$ time. Since $|T| < n$, the result follows.

4. Conclusion

Motivated by the applications in the evacuation planning and the related fields, we consider a problem of choosing a set of sinks from among a given set of possible sinks. The problem is important because it may not be possible to use all the sinks because of cost or other constraints. We introduce a Max Static k-sinks location problem and an algorithm is constructed to find at most k sinks out of a set of possible sinks T. The procedure does not have a polynomial time complexity. However, we prove that for $k = |T| - 1$, the problem can be solved in a strongly polynomial time.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: theory, algorithms, and applications*. Prentice hall, 1993.

[2] H. W. Hamacher and S. A. Tjandra, *Mathematical modelling of evacuation problems: A state of art*. Fraunhofer-Institut für Techno-und Wirtschafts mathematik, Fraunhofer (ITWM), 2001.

[3] T. N. Dhamala, U. Pyakurel, and S. Dempe, “A critical survey on the network optimization algorithms for evacuation planning problems,” *International Journal of Operations Research*, vol. 15, no. 3, pp. 101–133, 2018.

[4] H.W. Hamacher, S. Heller, and B. Rupp, “Flow location (FlowLoc) problems: dynamic network flow and location models for evacuation planning,” *Annals of Operations Research*, vol. 207, no. 1, pp. 161–180, 2013.

[5] H.N. Nath, U. Pyakurel, T.N. Dhamala, and S. Dempe, “Dynamic network flow location models and algorithms for quickest evacuation planning,” *Journal of Industrial and Management Optimization*, vol. 17, no.5, pp. 2925–2941, 2021.

[6] P.Heßler and H.W. Hamacher, “Sink location to find optimal shelters in evacuation planning,” *EURO Journal on Computational Optimization*, vol. 4, no. 3-4, pp.325–347, 2016.

[7] H. N. Nath and T. N. Dhamala, “Network flow approach for locating optimal sink in evacuation planning,” *International Journal of Operations Research*, vol. 15, no. 4, pp.175–185, 2018.

[8] E. Minieka, “Maximal, lexicographic, and dynamic network flows,” *Operations Research*, vol. 21, no. 2, pp. 517–527, 1973.

[9] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. New Jersey: Princeton University Press, 1962.

[10] A. V. Goldberg and R. E. Tarjan, “A new approach to the maximum-flow problem,” *Journal of the ACM (JACM)*, vol. 35, no. 4, pp. 921–940, 1988.

——000——