

Closed parasitic flow loops and dominated loops in networks

Michael T. Todinov

Oxford Brookes University, Department of Mechanical engineering
and Mathematical sciences, Oxford, OX33 1HX, UK
mtodinov@brookes.ac.uk

Abstract. The paper raises awareness of the presence of closed parasitic flow loops in the solutions of published algorithm for maximising the throughput flow in networks. If the rooted commodity is interchangeable commodity, a closed parasitic loop can effectively be present even if the routed commodity does not physically travel along a closed loop. The closed parasitic flow loops are highly undesirable loops of flow, which effectively never leave the network. Parasitic flow loops increase the cost of transportation of the flow unnecessarily, consume residual capacity from the edges of the network, increase the likelihood of deterioration of perishable products, increase congestion and energy wastage. Accordingly, the paper presents a theoretical framework related to parasitic flow loops in networks. By using the presented framework, it is demonstrated that the probability of existence of closed and dominated flow loops in networks is surprisingly high.

The paper also demonstrates that the successive shortest path strategy for minimising the total length of transportation routes from multiple interchangeable origins to multiple destinations fails to minimise the total length of the routes. It is demonstrated that even in a network with multiple origins and a single destination, the successive shortest path strategy could still fail to minimise the total length of the routes. By using the developed theoretical framework, it is shown that a minimum total length of the transportation routes in a network with multiple interchangeable origins, is attained if and only if no closed parasitic flow loops and dominated flow loops exist in the network. Accordingly, an algorithm for minimising the total length of the transportation routes by eliminating all dominated parasitic flow loops is proposed.

Keywords: *parasitic flow loops, routing loops, dominated flow loops, maximum throughput flow, successive shortest paths, interchangeable origins, multiple destinations*

1. Introduction

Methods and technologies based on maximising the throughput flow in networks have received popularity and significant research attention since the seminal research conducted in (Ford and Fulkerson, 1956). Soon, various other algorithms for maximising the throughput flow in networks appeared, based on different ideas (Goldberg and Rao 1998; Asano and Asano, 2000).

The first distinct category of algorithms for maximising the throughput flow will be referred to as *saturation algorithms*. The algorithms from this category start from an empty network and continue by gradually saturating the edges with flow (Ahuja and Orlin, 1991; Cherkaski, 1977; Dinic, 1970; Edmonds and Karp, 1972; Elias et al., 1956; Ford and Fulkerson, 1956; Goldberg and Tarjan, 1988; Karzanov, 1974; Shiloach and Vishkin, 1982; Sleator and Tarjan, 1980; Goldberg and Rao 1998; Hochbaum, 2008; Chandran and Hochbaum, 2009). Within this big category of algorithms, two major groups can be distinguished. The first major group of saturation algorithms are the path-augmentation algorithms which, at all steps, preserve the feasibility of the network flow until the maximum

flow is attained (Dinic, 1970; Edmonds and Karp, 1972; Elias et al., 1956; Ford and Fulkerson, 1956). Thus, the Ford-Fulkerson algorithm (Ford and Fulkerson, 1956) is based on augmenting available source-to-sink paths until no more augmentable paths can be found. An improvement of the Ford-Fulkerson algorithm was the approach proposed by Dinic (1970), based on determining the blocking flow in the network, for which every directed source-to-sink path contains an edge with zero residual capacity. Another important algorithm based on a sequential augmentation of the shortest paths was independently developed by Edmonds and Karp (1972). A further improvement of the blocking flow concept was proposed in (Goldberg and Rao, 1998). A further improvement of the average running time of the Edmonds and Karp (1972) algorithm was proposed in (Todinov, 2013a) by a two-stage augmentation, combining the blocking flow concept and augmentation along the shortest paths.

The second big group of saturation algorithms is based on the preflow concept proposed by Karzanov (1974). A preflow allows the sum of the edge flows going into a node to exceed the sum of the flows going out of the node. As a result, the preflow satisfies the capacity constraints on the edges, but the flow conservation law at the nodes may be violated, in the sense that each node other than the source may contain excess flow. This concept was later used as a basis of the preflow-push algorithms by Sleator and Tarjan (1980) and Goldberg and Tarjan (1988). The preflow-push algorithms work to convert the preflow into a feasible flow. This is done by including labels on the nodes indicating their ‘height’ with respect to the source node. Flow is pushed from an excess node with a higher height to a neighbouring node with a lower height. If this cannot be done for a particular neighbouring node, connected with the original node by an augmentable edge, a relabelling operation is initiated. This consists of increasing the height of the excess node. Continuing this process until no internal node (other than the source and the sink) has an excess flow, guarantees that the maximum throughput flow will be reached.

Recently, improved saturation algorithms for maximising the throughput flow based on the modifications of classical augmentation algorithms appeared in (Madry, 2016) and (Mallik et al. 2016). In addition, more advanced saturation algorithms, improving the running time of the flow maximisation algorithms have been proposed, based on combining ideas from existing approaches (Goldebrg et al, 2011; Orlin 2013).

Another distinct category of algorithms for maximising the flow in a network also appeared which will be referred to as *draining algorithms*. The draining algorithms start from a fully saturated network with flow edges and work backwards, by redistributing and draining flow from the network, until a maximum throughput flow is reached (Dong et al., 2009; Todinov 2013b).

Despite the significant progress made in maximising the throughput flow in networks, surprisingly, the algorithms for maximising the throughput flow published since 1956, leave highly undesirable *closed parasitic flow loops* in the optimised networks. A closed parasitic flow loop is a cyclic path whose entire length is occupied with flow along the same direction of traversal of the cyclic path. Despite that no commodity may be physically circulated around a closed loop, effectively, commodity is being circulated around a closed loop.

This paper also demonstrates the appearance of parasitic flow loops in common successive shortest-path solutions from interchangeable origins. A shortest-path from an origin to a destination is usually built by one of the well-known classical algorithms: the *Dijkstra algorithm* or the *Bellman-Ford algorithm* or their modifications. To minimise the total cost of a maximum flow between a single origin and a single destination, Goodrich and Tamassia (2002) proposed building repeated successive shortest paths. Repeated building of shortest paths has, for example, been adopted in (Wang et al, 2005) to minimise the total length of paths between multiple origins and multiple destinations, in solving the multiple-pairs

shortest-path problem. Repeated building of shortest paths from multiple origins to a common destination to minimise the total length of routes has been recently proposed in Zhao and Wang (2014).

Apart from maximising the throughput flow in networks, the performance of many route planners also depends exclusively on directing flow along successive shortest paths on a road map (Zhao and Wang, 2014). Unfortunately, as it will be demonstrated later, the successive shortest path strategy for minimising the total length of routes from origins to destinations results in solutions which are far from optimal.

Closed routing loops occur in perfectly engineered computer networks, using a shortest path algorithm to direct packets (Hengartner et al., 2002; Fu et al., 2008; Bohdanowicz et al., 2009, Li et al. 2015; Yousef et al, 2015). In computer networks, the routing loops increase the packet loss and have a detrimental effect on performance. An important feature of routing loops is that the packet physically travels around a closed loop. In this paper it is shown that a closed routing loop can effectively be present *even if the routed commodity does not physically travel along a closed loop*, as long as the routed commodity is interchangeable commodity.

Building optimal routes from multiple interchangeable origins to destinations has numerous important practical applications. It covers transportation and delivery services from interchangeable supply centres, gas supply to consumers from different storage locations, fuel supply from interchangeable terminals to petrol stations, vehicle routing with real-time traffic information (taxi car fleets servicing customers placing calls from different locations; ambulance cars from different hospitals servicing calls from different locations etc. Common services provided by professionals living at different locations are also covered by this case.

A related problem is the problem of optimal matching of sources and destinations on a weighted bipartite graph which minimises the total transportation cost. Various algorithms for solving this problem have recently been reviewed in (Sharathkumar and Agarwal, 2012). The delivery cost from a source to a destination however, is fixed and cannot be altered in the process of finding an optimal solution. Moreover, as it will be demonstrated later, fixed costs based on the shortest paths from origins to destinations fail to deliver the optimal solution because of the appearance of parasitic flow loops.

Parasitic flow loops are highly undesirable in transportation networks because they increase the cost of transportation of the flow, consume residual capacity from the edges of the network and increase congestion and energy consumption. In production processes, parasitic flow loops increase the likelihood of deterioration of perishable products. In computer networks, parasitic loops of flow consume bandwidth capacity unnecessarily, increase data traffic and ultimately lead to a loss of packets and delayed data transmission which affects negatively the quality of service.

The significant detrimental effects associated with parasitic flow loops in networks determine the objectives of this paper:

- to demonstrate the appearance of parasitic flow loops in networks and in classical solutions of the maximum throughput flow problem;
- to expose the detrimental effects from closed parasitic flow loops in networks and estimate the likelihood of their appearance;
- to derive a necessary and sufficient condition for a minimal transportation cost in a network with multiple interchangeable sources;
- to demonstrate the shortcomings of the successive-shortest path strategy for minimising the transportation costs from interchangeable origins to destinations;
- to present a method for eliminating all closed and dominated parasitic loops in networks and minimising the transportation costs in a network with multiple interchangeable sources.

2. Closed parasitic flow loops in networks whose throughput flow has been maximised by classical algorithms

Consider the network in Fig.1a where all edges have capacity equal to 10 flow units per unit time. The classical successive shortest-path algorithms (e.g. Edmonds and Karp, 1972) for maximising the throughput flow proceed with saturating the shortest-path (1,2,3,13) with 10 units of flow, followed by saturating the next shortest path (1,5,6,3,4,13) with 10 units of flow and finally, with saturating the path (1,8,9,10,11,4,2,7,12,13) with 10 units of flow. As a result, the throughput flow in the network is maximised (30 units of flow) but a closed flow loop (2,3,4,2) appears, carrying 10 units of flow. This flow loop *effectively circulates commodity along a closed contour, which never leaves the network*. As can be verified from Fig.1, closed parasitic flow loops appear *even if the transported commodity does not physically travel along a closed contour*. In this case, despite that no commodity has been deliberately moved around a closed loop effectively, the commodity is being circulated around a closed loop.

This feature distinguishes the closed parasitic loops considered in this paper from the closed routing loops observed in computer networks.

Indeed, consider network flow I where the flows follow path (1,2,3,13), path (1,5,6,3,4,13), and path (1,8,9,10,11,4,2,7,12,13) from the source s to the sink t and network flow II, where the flows follow path (1,2,7,12,13), path (1,5,6,3,13), path (1,8,9,10,11,4,13) from the source s to the sink t plus the additional flow of 10 units circulated endlessly around the loop (2,4,3,2) (Fig.1a). Network flow I is identical to network flow II.

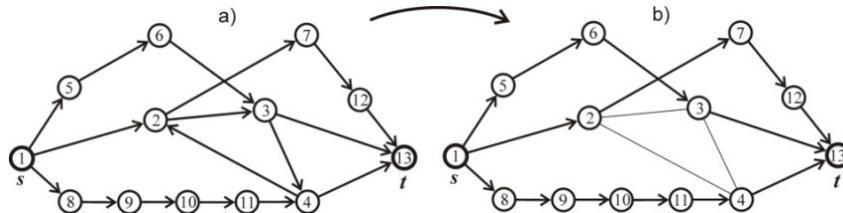


Figure 1. A counterexample network, demonstrating that the classical shortest-path algorithms for maximising the throughput flow leaves closed parasitic loops of flow in the optimised networks. All edges have a flow capacity of 10 units.

The parasitic flow loop (2,3,4,2) occupies unnecessarily edge capacity, congests the network sections (2,3), (3,4) and (4,2), and consumes energy for maintaining the circulation of flow. The parasitic flow loop (2,3,4,2) can be removed safely, without affecting the total throughput flow of 30 units from the source s to the sink t (Fig.1b).

It needs to be pointed out that it is possible that the commodity is physically circulated around a closed loop, which is particularly detrimental if perishable goods are transported between intermediate stops of a production process.

Indeed, suppose that perishable product (e.g. a food product) is being transported from node 1 towards node 13, with intermediate storage at nodes 2,4, and 3. At node 13, the perishable product is to be processed. If the perishable product is independently distributed from node 4 to node 2, from node 2 to node 3 and from node 3 to node 4, there is a real possibility that the same units of the product will be circulated around the loop (2,3,4,2) for an unacceptable amount of time. During this time, the perishable product could deteriorate due to bacteria or other causes and enter a state which makes it unsuitable/unsafe for further processing.

Parasitic flow loops may also appear after an optimisation performed by using preflow-push algorithms. Consider the network in Figure 2a. The network includes a single source s

with capacity of 40 units and a destination node (sink) t . Suppose that a classical push-relabel algorithm (Cormen et al., 2001; Goldberg and Tarjan, 1988; Kleinberg and Tardos, 2006) is used to maximise the throughput flow from the source s to the sink t . The first number on the labels of the edges denotes the capacity of the edge and the second number denotes the actual edge flow.

The preflow-push algorithm begins with assigning the highest label to the source s and zero to the rest of the nodes. The edge (1,2) coming out of the source s is fully saturated with flow (Fig.2b) and node 2 becomes an excess node. Edges (2,3) and (2,4) are augmentable; nodes 3 and 4 have the same label '0', and a relabeling operation is initiated for node 2. The label of node 2 is increased from 0 to 1, after which flow of 10 units is pushed towards node 3 and flow of 30 units towards node 4 (Fig.3c). Node 4 is the next selected excess node. Its label is '0', the same as the labels of the two eligible neighbours - nodes 3 and 7.

A relabelling operation is initiated, which increases the label of node 4 to 1. A flow of 10 units is then pushed towards node 3 and flow of 20 units is pushed towards node 7 (Figure 2d). Next, a relabelling operation is initiated for node 3 and its label is increased from 0 to 1, after which the excess flow of 20 units is sent towards node 5. Node 5 becomes an excess node with excess flow of 20 units. One of its eligible neighbours, node 4, has a label '1' and the other eligible neighbour, node 6, has a label '0'. A relabelling operation is initiated for node 5, whose label is increased to '1' and 10 units of flow are pushed towards node 6 (Fig.3e). Now, node 4 is an eligible neighbour towards which excess flow can be sent but it has the same label '1' as node 5. Following another relabelling operation, the label of node 5 becomes 2 and 10 units of excess flow are sent towards node 4 (Fig.2e). Now node 4 has 10 units excess flow which is pushed towards node 7 (Fig.2f).

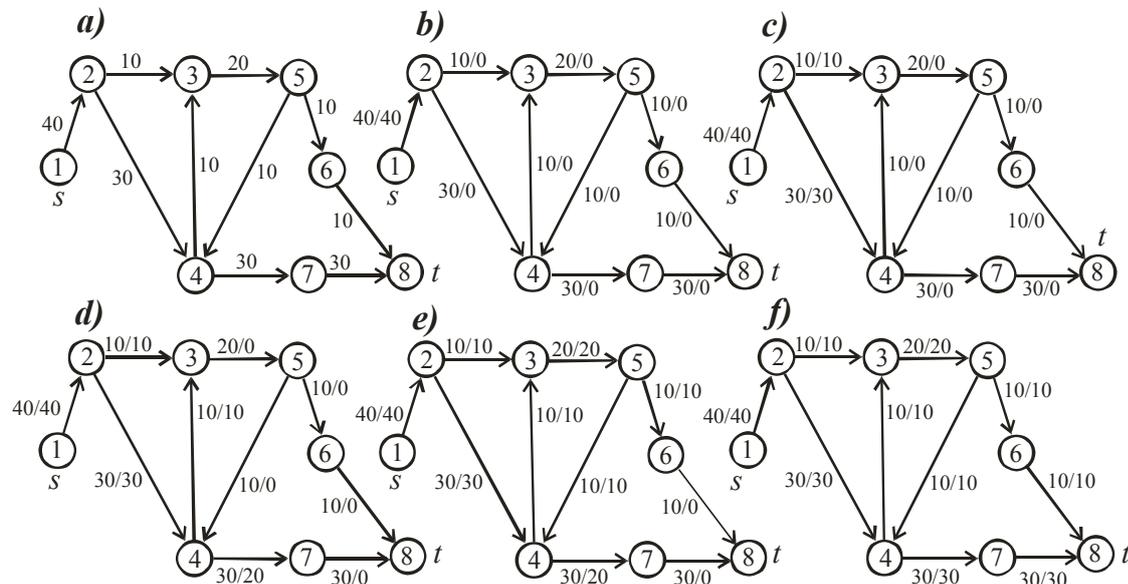


Figure 2. Maximising the throughput flow by using the preflow-push algorithm leads to a directed flow loop (3,5,4,3) carrying 10 units of flow.

After pushing 30 units of excess flow from node 7 towards node 8 and 10 units of excess flow from node 6 towards node 8, there are no more excess nodes in the network. Figure 2f gives the final state of the edge flows. The maximum throughput flow of 40 units has been attained but a parasitic flow loop (3,5,4,3) carrying 10 units flow now appears in the network. The circular flow of 10 units effectively never leaves the network.

A fundamentally different idea for maximising the throughput flow characterizes the draining algorithms (Dong et al., 2009, Todinov, 2013b). The central concept in these

algorithms is to start from a fully saturated with flow network and balance excess and deficit nodes until the excess and deficit flow at the nodes disappears. The necessary stages to achieve the correct value of the maximum throughput flow have been presented in (Todinov 2013b). If the sum of capacities of all edges going into a node (different from the source and the sink) is greater than the sum of capacities of all outgoing edges, the node is referred to as ‘excess node’. Conversely, if the sum of capacities of all edges going into a node (different from the source and the sink) is smaller than the sum of capacities of all outgoing edges, the node is referred to as ‘deficit node’. If the sum of capacities of all edges going into a particular node is equal to the sum of capacities of all outgoing edges, the node is referred to as ‘balanced node’. The amount of excess/deficit flow at a balanced node is zero. The algorithm essentially drains flow from paths connecting deficit nodes and excess nodes, until all nodes become balanced. It can be shown that at this point a maximum throughput flow is attained. The node balancing can be conducted along the shortest augmentable paths between excess and deficit nodes.

The problem with the draining algorithms can be illustrated with the simple network in Fig.3a, where the labels stand for edge capacities.

In the network from Fig.3a, there are no excess and deficit nodes (all nodes are balanced). According to the draining method, no flow is removed from any edge in the fully saturated with flow network and the existing edge flows essentially maximise the throughput flow (Fig.3b). The first number of the labels on the edges stands for the edge capacity and the second number stands for the actual flow along the edge. The correct maximum throughput flow of 30 units (Fig.3b) has indeed been attained, but a flow loop (3,2,4,3) with parasitic flow of 10 units remains in the network and never leaves it. The desirable solution has been shown in Fig.3c., where the maximum throughput flow is also 30 units but no parasitic flow loops exist.

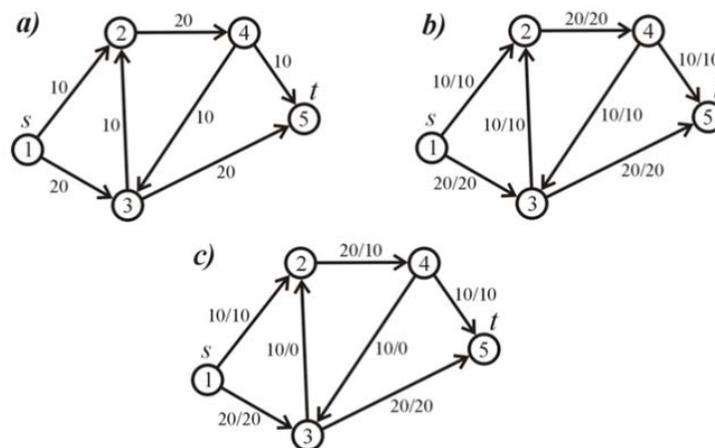


Figure 3 a) Initial network; b) A network with a parasitic loop (3,2,4,3) of 10 units flow, after a throughput flow maximisation by the draining algorithm proposed in Dong et al. (2009); c) The network from b) optimised by removing the closed parasitic flow loop.

The common feature of all discussed approaches to maximising the throughput flow is that despite that the maximum throughput flow is obtained correctly, the edge flows yielding the maximum throughput lead to a highly undesirable state where part of the flow which enters the network is effectively engaged in endless circulation along a closed path and never leaves the network.

Closed parasitic flow loops, in fact appear in the solutions of *published algorithms for maximising the throughput flow* since the creation of the theory of flow networks in 1956. None of these algorithms should be used without an additional stage aimed at removing

possible closed parasitic flow loops. Ironically, despite the decades of intensive research on static flow networks, the existence of closed parasitic flow loops in solutions maximising the throughput flow, remained unnoticed for 60 years.

One of the reasons for this critical omission is the counterintuitive result that a closed parasitic flow loop can be present despite that no commodity is physically moved around a closed loop.

Another reason for this critical omission can be found in the last example (Fig.3). The closed parasitic flow loops are easy to observe in networks with directed edges, directed cycles and balanced internal nodes, whose throughput flow has been maximized by using a draining algorithm. In this case, a directed parasitic flow loop appears immediately after the saturation of all edges with flow. In contrast, the appearance of closed parasitic flow loops is not at all straightforward in networks whose maximum throughput flow has been produced by an augmentation algorithm or a push-preflow algorithm. This is why the discovery of closed parasitic flow loops in networks had to wait the publication of the first 'draining algorithm' in 2009.

3. Probability of existence of closed parasitic flow loops in networks

The probability of existence of closed parasitic flow loops in networks is surprisingly high. Consider a system of random intersecting linear flow paths on a plane (Fig.4a). Each of the linear flow paths carries flow in a random direction. It is also assumed that there are at least three linear flow paths; there are no parallel paths and no three paths intersect into a single point. A unit vector \mathbf{u}_i can be assigned to each flow path, with a direction following the direction of the flow along the flow path (Fig.4b). The unit vector \mathbf{u}_i will be referred to as *direction vector*. The angle the direction vector subtends with the horizontal x -axis (Fig.4b) gives both the orientation of the flow path and the direction of its flow. A random orientation of a flow path i means that the angle which its direction vector \mathbf{u}_i subtends with the fixed horizontal x -axis is uniformly distributed in the interval $(0, 2\pi)$. In other words, any particular orientation of the flow path is characterised by the same probability. All direction vectors \mathbf{u}_i corresponding to the separate flow paths can be translated at the common origin O , as shown in Fig.4b. The following theorem then holds:

Theorem 1. *The necessary and sufficient condition for the non-existence of a closed parasitic flow loop between the intersection points of randomly oriented straight-line flow paths on a plane is the existence of a half-plane where all direction vectors reside.*

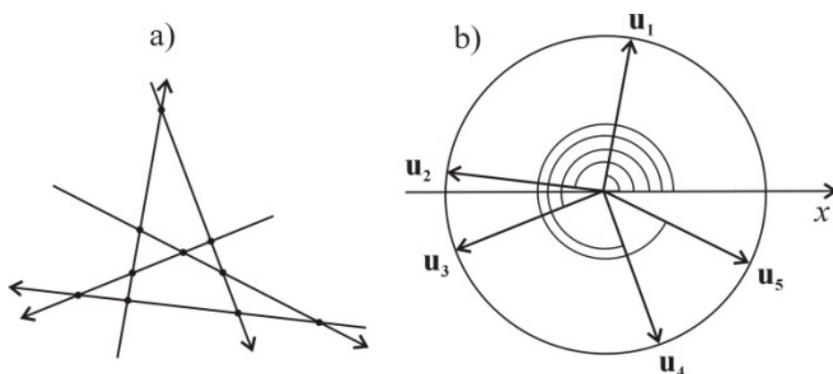


Figure 4. a) Randomly oriented intersecting linear flow paths on a plane; b) direction vectors of the flow paths.

Before proving this theorem a lemma will be stated, whose proof has been omitted.

Lemma 1. *If no half-plane can be selected where all direction vectors reside, it is always possible to select three direction vectors which do not reside in a single half-plane.*

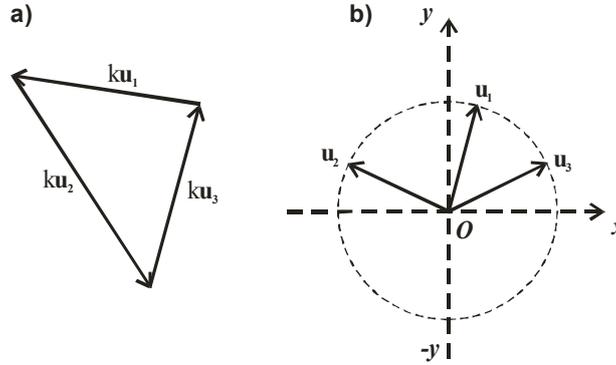


Figure 5. a) A directed triangular flow loop; b) the direction vectors of the flow paths cannot possibly reside in a single half-plane.

Now, Theorem 1 can be proved.

Proof. First note, that the existence of a closed flow loop implies the existence of at least one triangular closed flow loop (Fig.5a) because no two flow paths are parallel. Suppose that the flow paths with direction unit vectors \mathbf{u}_1 , \mathbf{u}_2 and \mathbf{u}_3 , define a triangular closed flow loop (Fig.5a). The vectors connecting the intersection points of the paths forming the triangular flow loop are $k_1\mathbf{u}_1$, $k_2\mathbf{u}_2$ and $k_3\mathbf{u}_3$, where $k_1 > 0$, $k_2 > 0$ and $k_3 > 0$ are some non-negative constants. Because the vectors $k_1\mathbf{u}_1$, $k_2\mathbf{u}_2$ and $k_3\mathbf{u}_3$ form a triangle, their sum is zero:

$$k_1\mathbf{u}_1 + k_2\mathbf{u}_2 + k_3\mathbf{u}_3 = 0 \quad (1)$$

A similar relationship holds regarding the projections of the vectors $k_1\mathbf{u}_1$, $k_2\mathbf{u}_2$ and $k_3\mathbf{u}_3$ on any specified axis. Thus, for an axis defined by a direction vector \mathbf{s} , the relationship

$$m_{1s} + m_{2s} + m_{3s} = 0 \quad (2)$$

holds, for the projections $m_{1s}=k_1\mathbf{u}_1\mathbf{s}$, $m_{2s}=k_2\mathbf{u}_2\mathbf{s}$ and $m_{3s}=k_3\mathbf{u}_3\mathbf{s}$ of vectors $k_1\mathbf{u}_1$, $k_2\mathbf{u}_2$ and $k_3\mathbf{u}_3$ on the axis with a direction given by vector \mathbf{s} . Relationship (2) is obtained by multiplying equation (1) by the vector \mathbf{s} .

Suppose that there is a single half plane where the direction vectors of all flow paths reside (Fig.5b). In this case, the direction vectors \mathbf{u}_1 , \mathbf{u}_2 and \mathbf{u}_3 of the paths forming the directed triangular loop from Fig.5a will also reside in the same half-plane. Without loss of generality, suppose that the axis x and the half-axis y define this half-plane (Fig.5b). The projections of the vectors $k_1\mathbf{u}_1$, $k_2\mathbf{u}_2$ and $k_3\mathbf{u}_3$ along the y -axis will then violate equation (2), because there will be a non-zero sum of projections along the half-axis (O,y) and zero sum of projections along the half-axis $(O,-y)$. As a result, $m_{1y} + m_{2y} + m_{3y} > 0$ and we arrive at a contradiction. Therefore, if a closed flow loop exists, there can be no half-plane where all direction vectors reside.

Now, suppose that there is no half-plane where all of the direction vectors \mathbf{u}_1 , \mathbf{u}_2 , ..., \mathbf{u}_n reside. According to Lemma 1, in this case, it is always possible to select three vectors \mathbf{u}_1 , \mathbf{u}_2 and \mathbf{u}_3 , which do not reside in a half-plane. For these three vectors, it is always possible to select constants $k_1 > 0$, $k_2 > 0$ and $k_3 > 0$, such that condition (1) is fulfilled. Because by assumption, no three source-destination pairs intersect in a single point, the flow paths which correspond to the selected direction vectors \mathbf{u}_1 , \mathbf{u}_2 and \mathbf{u}_3 will form a closed triangular parasitic flow loop.□

By using Theorem 1, the next theorem can be proved:

Theorem 2: The probability p of existence of a closed parasitic flow loop between the intersection points of n randomly oriented linear flow paths on a plane is $p = 1 - n/2^{n-1}$.

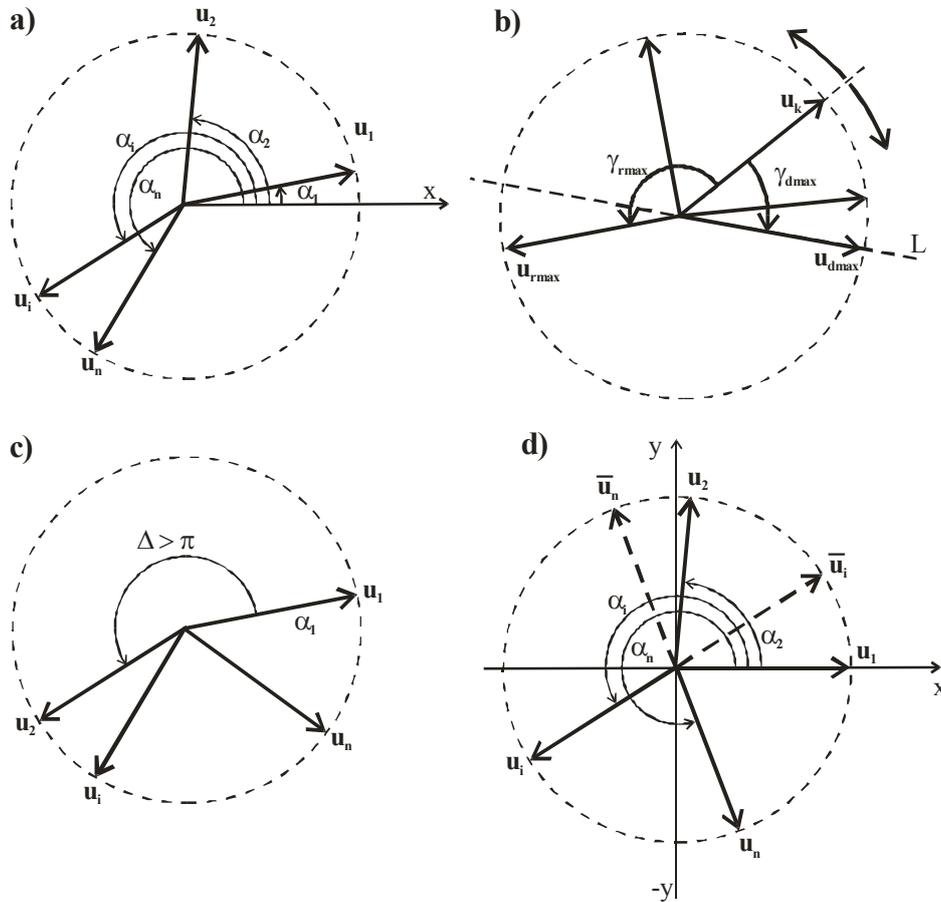


Figure 6. a) Ordering the direction vectors, according to the angle they subtend with the horizontal axis; b) If no half-plane can be selected where all direction vectors reside, it is always possible to select three direction vectors which do not reside in a single half-plane; c) a gap of size at least π between two random direction vectors; d) it is always possible to reverse the direction of the flows in the intersecting paths in such a way, that all direction vectors reside in the same half-plane.

Proof. The probability of a closed parasitic flow loop can be determined by determining first the probability of the complementary event that no parasitic flow loop exists. To determine this probability, the random direction vectors of the flow paths are ordered in ascending order, according to the magnitude of the angle they subtend with the horizontal x -axis (Fig.6a).

The probability that there will be no parasitic flow loop is equal to the probability that all random direction vectors will lie in a single half-plane. All random direction vectors will lie in a single half-plane if and only if a gap (angle) of minimum size $\Delta = \pi$ exists *between two adjacent direction vectors* (Fig.6c). There can be no more than a single gap of size $\Delta = \pi$, between two adjacent direction vectors, therefore, the random events corresponding to a gap on the left of the first direction vector, on the left of the second direction vector,...,on the left of the last (n -th) direction vector are mutually exclusive events. As a result, a single gap (angle) of minimum size $\Delta = \pi$ may be located in n distinct, mutually exclusive ways between the direction vectors.

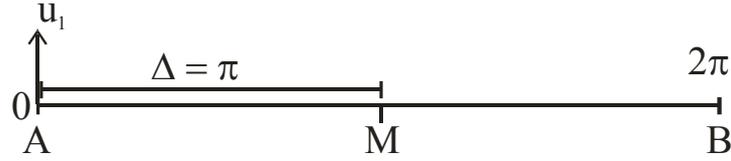


Figure 7. A gap of minimum length at least $\Delta = \pi$ between the first direction vector and the next direction vector can occur only if all unit direction vectors are located on the segment MB.

Let the circumference of the unit circle connecting the end-points of the direction vectors be represented by the segment with length 2π (Fig.7). A gap (angle) of length at least $\Delta = \pi$ on the left of direction vector u_1 , can only occur if the rest of the $n-1$ random direction angles are located in the segment MB and none of them is located in the segment AM (Fig.7). Because the probability that a random direction angle will be located on the segment MB is $\pi/(2\pi) = 1/2$, the probability that $n-1$ random direction angles will be located in the segment MB is $1/2^{n-1}$. Similarly, the probability of a gap on the left of the second direction vector is also $1/2^{n-1}$... and so on. As a result, the probability p of a gap angle of minimum size $\Delta = \pi$, between two direction vectors is a sum of the probabilities of these mutually exclusive events and becomes

$$p = \sum_{k=1}^n 1/2^{n-1} = n/2^{n-1} \quad (3)$$

which is also the probability that no closed parasitic flow loop will exist. Because there can be either a parasitic flow loop or no parasitic flow loop, the probability q of a closed parasitic flow loop becomes:

$$q = 1 - n/2^{n-1} \quad (4)$$

□

The probability of existence of a closed parasitic flow loop estimated from equation (4) has been plotted in Fig.8.

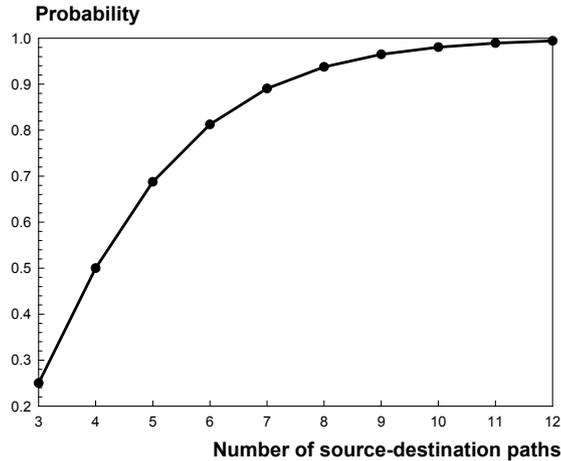


Figure 8. Probability of a closed parasitic flow loop as a function of the number of intersecting source-destination paths.

As can be verified, with increasing the number of intersecting randomly oriented flow paths, the probability of a parasitic flow loop increases significantly. For five intersecting random flow paths, the probability of a parasitic flow loop is already 69%. The results from equation (4) have been confirmed by Monte Carlo simulations.

Note from equation (4) that no matter how large the number n of intersecting flow paths is, the probability of a closed parasitic flow loop is always smaller than 1. Furthermore, for any possible number and for any possible orientation of straight-line flow paths on a plane, it is always possible to orient the flows along the paths in such a way, that no closed parasitic flow loops exist between the points of intersection. To do this, the direction of the flow in some of the paths is reversed in the opposite direction so that all direction vectors are located in a single half-plane (Fig.6d).

The probability of a closed parasitic flow loop has also been estimated on a lattice consisting of n vertical flow paths intersected with n horizontal flow paths where, for each flow path, the direction of the flow along the path is random (Fig.9). Thus, for a horizontal flow path, the flow is directed with probability 0.5 rightwards or leftwards. Similarly, for a vertical flow path, the flow is directed with probability 0.5 upwards or downwards.

Consider now a fixed lattice consisting of n vertical flow paths $x_{a1}, x_{b1}, \dots, x_{an}, x_{bn}$, intersected with n horizontal flow paths $y_{a1}, y_{b1}, \dots, y_{an}, y_{bn}$ (Fig.9). For each flow path, the direction of the flow along the path is random.

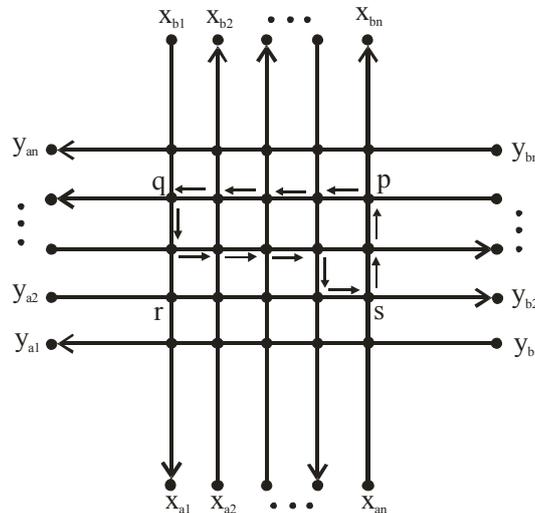


Figure 9. A lattice with intersecting source-destination paths with random directions of the flows creates the possibility of directed loops of flow.

The likelihood that the flow will follow the same direction along a closed contour (Fig.9) will be termed ‘probability of a closed parasitic flow loop on a lattice-type network’.

In a lattice-type network, a parasitic flow loop between intersections of the flow paths always implies a parasitic flow loop around the rectangular envelop of the intersections. Thus, the parasitic flow loop denoted by arrows in Fig.9, implies a parasitic flow loop along the rectangular envelope p,q,r,s,p of the intersections. Conversely, the existence of a rectangular flow loop implies the existence of a parasitic flow loop. Consequently, the probability that a parasitic flow loop will exist, can be assessed simply by assessing the probability that a rectangular parasitic flow loop will exist.

To derive an analytical expression for the probability of a rectangular parasitic flow loop, the state of the flows along k vertical flow paths, for example, can be represented by a chain of k bits which will be referred to as V -chain. Each bit of the V -chain represents the direction of the flow in the corresponding flow path. Thus, a bit ‘0’ codes downward direction of the flow while a bit ‘1’ codes upward flow. Similarly, a chain of k bits can also represent the horizontal flow paths. This chain of bits will be referred to as ‘ H -chain’. A bit ‘0’ in the H -chain codes a leftward orientation of the flow while a bit ‘1’ codes a rightward orientation. For example, the vertical flows in Fig.9, starting from left to right are coded by the V -chain

‘01101’ and the horizontal flows, starting from top to bottom are coded by the H-chain ‘00110’.

The number of all possible combinations of flow directions for k vertical paths is therefore 2^k and the number of all possible combinations of flow directions for k horizontal paths is also 2^k . All distinct combinations of flow directions along k horizontal and k vertical paths is therefore equal to $2^k \times 2^k = 2^{2k}$. A directed flow loop in clockwise direction requires the existence of at least a single bit combination 1-0 somewhere in the V -chain and the same bit combination 1-0 somewhere in the H -chain. This is because for a clockwise flow loop to be present, there must be flow in upward direction (bit ‘1’), flow in downward direction (bit ‘0’), flow in rightward direction (bit ‘1’) and flow in leftward direction (bit ‘0’).

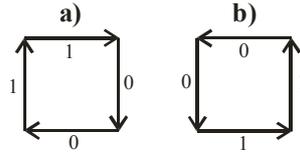


Figure 10. The two basic types of loops (clockwise and counter-clockwise loop) and the bits indicating the directions of the flows for the vertical and the horizontal flow paths.

The total number of V -chains which have at least a single bit combination ‘1-0’ somewhere, can be obtained by subtracting from the total number 2^k of possible V -chains, the number of V -chains which do not have a bit combination ‘1-0’. For a 4-bit chain for example, the only bit chains which *do not* have a bit combination ‘1-0’ are: 0000, 0001, 0011, 0111 and 1111. This result is easily generalized and for a k -bit chain, the number of chains which do not have a bit combination ‘1-0’ is equal to $k+1$. As a result, the total number of V -chains which have the bit combination ‘1-0’ is $2^k - (k + 1) = 2^k - k - 1$. In order to have a clockwise directed loop, at least one bit combination ‘1-0’ must also be present in the H -chain. Because H -chains and V -chains are combined independently, the total number of flow path configurations in which a clockwise directed flow loop exists is

$$N_1 = (2^k - k - 1)^2 \quad (5)$$

From Fig.10b, it can be seen that for a counter-clockwise directed loop to be present, at least one bit combination ‘0-1’ is required in the V -chain and at least one bit combination ‘0-1’ is required in the H -chain. With a reasoning very similar to the one used for the clockwise directed loops, the total number of configurations in which a counter-clockwise flow loop exists is also

$$N_2 = (2^k - k - 1)^2 \quad (6)$$

Note that in estimating the numbers of clockwise and counter-clockwise flow loops, some of the configurations have been counted twice because they contain both clockwise and counter-clockwise loops. Both types of flow loops (clockwise and counter-clockwise) are present if both bits combinations ‘1-0’ and ‘0-1’ are present somewhere in the V -chain and both bit combinations ‘1-0’ and ‘0-1’ are also present somewhere in the H -chain. The number of V -chains containing both bit combinations ‘1-0’ and ‘0-1’ simultaneously can be obtained by subtracting from the total possible number 2^k of distinct V -chains, the number of chains which do not contain both bit combinations simultaneously. For a 4-bit V -chain for example, the bit chains which do not contain both bit combinations (‘1-0’ and ‘0-1’) simultaneously are: ‘0000’, ‘0001’, ‘0011’, ‘0111’, ‘1111’, ‘1000’, ‘1100’, ‘1110’. This result is easily generalized for a k -bit chain for which, the number of chains not containing a bit combination ‘1-0’ and a bit combination ‘0-1’ simultaneously is equal to $2k$. As a result, the total number of V -chains which contain both bit combinations (‘1-0’ and ‘0-1’) simultaneously is $2^k - 2k$.

In order to have both: clockwise oriented directed loop and a counter-clockwise oriented loop, both bit combinations ‘1-0’ and ‘0-1’ must also be present in the H-chain. Because H-chains and V-chains are combined independently, the total number of flow path configurations in which both a clockwise and a counter-clockwise directed flow loop exists is

$$N_{12} = (2^k - 2k)^2 \quad (7)$$

The total number flow path configurations with a flow loop in clockwise direction or with a flow loop in counter-clockwise direction therefore are equal to $N_1 + N_2 - N_{12}$, where N_1 is given by equation (5), N_2 is given by equation (6) and N_{12} by equation (7). The probability of a directed flow loop (in clockwise or counter-clockwise direction) on a lattice of k vertical and k horizontal paths is therefore given by:

$$p = \frac{2 \times (2^k - k - 1)^2 - (2^k - 2k)^2}{2^{2k}} \quad (8)$$

The result given by equation (8) has been confirmed by a Monte Carlo simulation.

Equation (8) can be used for determining the probability of a parasitic flow loop for a different number of intersecting flow paths. The results have been presented in Fig.11. From the graph in Fig.11, it can be seen that the probability of a directed flow loop increases quickly with increasing the number of intersecting flow paths and even for 4 horizontal and 4 vertical paths (8 intersecting paths), the probability of a closed parasitic flow loop is nearly 70%. For 6 horizontal and 6 vertical intersecting pairs (12 intersecting paths), this probability is already nearly 93%. These results mean that in lattice networks (which include many intersecting flow paths, the existence of closed parasitic loops of flow is practically guaranteed.

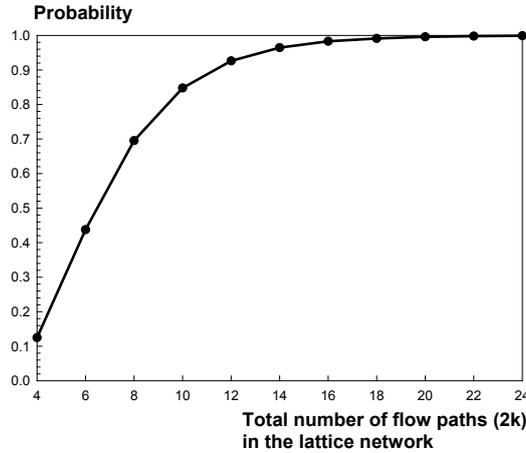


Figure 11. Results from equation 8, showing the variation of the probability of a closed parasitic flow loop, on a lattice with random directions of the flows.

As a result, the state where closed parasitic flow loops are present can be considered to be the ‘natural state’ of flow networks.

4. Dominated flow loops

A cyclic path in which more than half of the path length is occupied by flow along one of the directions of traversal will be referred to as ‘dominated flow loop’.

The dominated flow loop (Fig.12a) is a sequence of n sections/edges ($n \geq 3$) in which the flow along $n - k$ sections/edges (not necessarily sequential) points against the direction of traversing of the contour and the rest of the k sections/edges (the closing edges) are either empty or partially saturated with forward flow.

Figure 12a features the dominated flow loop (2,7,6,5,4,3,2) on a simple transportation network including three origins s1, s2 and s3 and three destinations d1, d2 and d3. Each origin is connected to a single destination through a path carrying 1 unit of flow.

The backward edges are the edges (2,7),(7,6),(6,5) and (5,4). The closing edges are the empty edges (4,3) and (3,2). Each of the non-empty edges carries exactly 1 unit of flow.

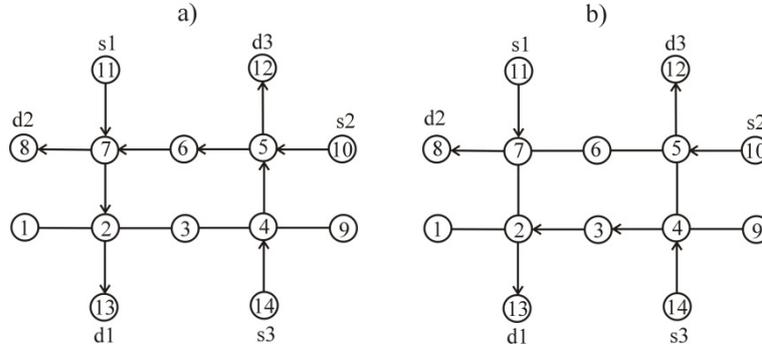


Figure 12. a) A dominated flow loop in a transportation network; b) Removal of the dominated flow loop.

The $n-k$ edges with backward flow will be referred to as ‘backward edges/sections’ while the remaining ‘ k ’ edges will be referred to as ‘closing edges/sections’.

For a dominated flow loop to exist, the cyclic path must be augmentable and the sum of the lengths of the $n-k$ edges with backward flow must be greater than the sum of the lengths of the closing k edges:

$$\sum_{i=1}^{n-k} l_i^{(b)} > \sum_{j=1}^k l_j^{(c)} \quad (9)$$

where $l_i^{(b)}$ is the length of the i th backward edge and $l_j^{(c)}$ is the length of the j th closing edge.

Essentially, the sum of the lengths of the edges with backward flows dominates the sum of the lengths of the closing edges. The dominated flow loop can be thought as an augmentable broken flow loop for which some of the flow travels in backward direction and the sum of the lengths of the backward sections dominates the sum of the lengths of the closing sections. To be augmentable, the closing edges *must either be empty or partially saturated with forward flow*.

Let Δ be the flow magnitude by which a dominated flow loop can be augmented without violating the capacity constraints on the edges. The dominated flow loop can be augmented (essentially drained) by decreasing the flow along the $n-k$ edges with dominating (backward) flow by Δ and increasing with the same amount Δ the flow along the closing k edges. The draining operation does not violate the flow conservation at each node and the capacity constraints along the edges and leads to a new feasible flow in the network.

Note that a closed parasitic loop of flow is always a dominated flow loop but the converse is not true in general. An essential distinction between a closed parasitic flow loop and dominated flow loop consists of the circumstance that the flow of the closed parasitic loop essentially never leaves the network while the flow of the dominated flow loop always leaves the network.

Dominated flow loops are also associated with losses and their removal/draining is highly beneficial. After the draining of a dominated flow loop, either one or more edges with backward flow become empty or one or more of the k closing edges loop become fully saturated with flow. As a result, of the flow augmenting operation, the dominated flow loop is removed.

The dominated loop in Fig.12a can be drained by augmenting it with one unit of flow in the direction (2,7,6,5,4,3,2). The result is the transportation network in Fig.12b, where no dominated flow loops are present.

Dominated flow loops and augmentable broken loops with dominating flow along a particular direction of traversing are equivalent concepts.

If the cost of transportation per unit distance does not vary along the different edges, draining an augmentable broken flow loop with dominating flow always results in a reduction of the transportation cost.

4.1 Transportation cost associated with dominated flow loops in networks

Consider a network with undirected edges. A *path* in a flow network is a unique sequence of edges between two nodes. Suppose that the flow along a particular path can be increased by Δ without violating the capacity constraints on the edges and the flow conservation law at the nodes along the path. Such a path will be referred to as *augmentable path*.

Consider an edge i whose flow direction is opposite to the direction of traversing of the augmentable path. According to an earlier convention, such edges are referred to as ‘backward edges’. Similarly, edges which are empty or whose flow direction is along the direction of traversing the augmentable path are referred to as ‘closing edges’.

Suppose that the cost of transportation per unit length is c units. Augmenting the path with Δ means that the flow along each backward edge is decreased and the flow along each forward edge is increased.

Because, during the path augmentation, the flow along the closing edges is increased, the transportation cost along the closing edges will increase. For the i th closing edge, the increase of the cost of transportation ΔC_i is equal to

$$\Delta C_i = \Delta \times c \times l_i^{(c)} \quad (10)$$

where $l_i^{(c)}$ is the length of the i th closing edge.

The increase ΔC_c of the cost of transportation for a path composed of M_c closing edges only, is given by

$$\Delta C_c \approx \Delta \times c \times \sum_{i=1}^{M_c} l_i^{(c)} \quad (11)$$

Suppose that a path containing both closing edges and backward edges is augmented with flow with magnitude Δ .

After the path augmentation, along any backward edge, the flow Δ is no longer transported but is prevented from being transported, because the flow along backward edges has been *decreased* by Δ . Therefore, the increase in the transportation cost ΔC_j , along a backward edge with index ‘ j ’, has a negative sign:

$$\Delta C_j = -\Delta \times c \times l_j^{(b)} \quad (12)$$

For M_b backward edges, the total increase ΔC_b of the cost of transportation can be approximated by

$$\Delta C_b \approx -\Delta \times c \times \sum_{j=1}^{M_b} l_j^{(b)} \quad (13)$$

where $l_j^{(b)}$ is the length of the j th backward edge.

Consequently, the total increase of the cost of transportation ΔC along a path including both backward and closing edges can be approximated by

$$\Delta C = \Delta C_f + \Delta C_b \approx \Delta \times c \left(\sum_{i=1}^{M_c} l_i^{(c)} - \sum_{j=1}^{M_b} l_j^{(b)} \right) \quad (14)$$

The quantity

$$\gamma = \sum_{i=1}^{M_c} l_i^{(c)} - \sum_{j=1}^{M_b} l_j^{(b)} \quad (15)$$

will be referred to as a ‘*cost factor of a path*’. A positive sign of the cost factor γ means that if the path is augmented with flow, the transportation cost will increase. A negative sign of the cost factor γ means that after the path augmentation, the transportation cost will decrease. The larger the magnitude of the cost factor γ , the larger is the amount of the transportation cost increase/decrease following the path augmentation.

A dominated parasitic flow loop is always characterised by a negative cost factor. The augmentation of a dominated parasitic flow loop results in a decrease of the transportation cost. At the same time, the amount of flow transported from sources to destinations is not affected. Effectively, augmenting a parasitic flow loop in a direction opposite to the direction of its dominant flow, is a process of draining the flow loop. An indication of the presence of a dominated flow loop is the presence of an augmentable cyclic path with a negative cost factor, anywhere in the network. The following theorem can then be stated:

Theorem 3. *Dominated flow loops are associated with increased transportation cost which can be reduced by augmenting the loops with flow, in a direction opposite to the direction of the dominant flow.*

4.2 Estimating the probability of a dominated flow loop

The unexpectedly high probability of existence of dominated parasitic flow loops will be demonstrated by considering randomly oriented intersecting straight-line flow paths on a plane (Figure 13a).

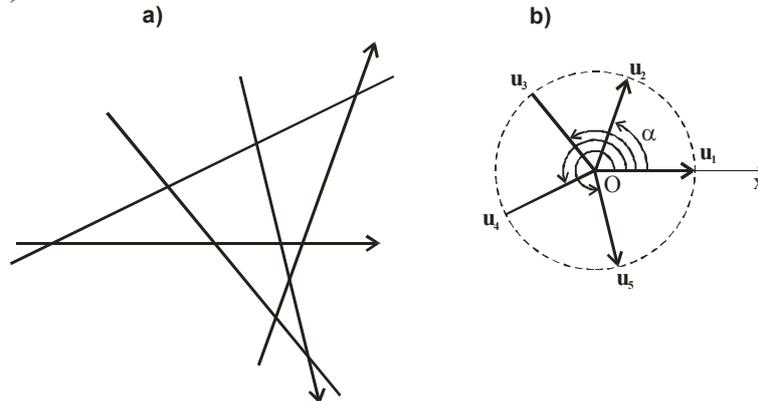


Figure 13. a) Randomly oriented intersecting flow paths; b) All direction vectors of the source-destination paths can be translated to start from a common point O.

All flow paths transport the same type of interchangeable commodity (e.g. petrol) and along each flow path, there is either flow with a particular direction (Fig.13a, the lines with an arrow) or no flow (Fig.13a, the lines without arrows).

It is assumed that there are at least three intersecting flow paths; there are no parallel paths and no three paths intersect into a single point. These conditions are natural and do not restrict generality. For randomly oriented straight-line flow paths on a plane, it is very unlikely to find two parallel paths or three paths intersecting into a single point. The likelihood that a

dominated flow loop will be present between the intersection points of the network, given that the orientation of the intersecting flow paths is random, will be termed ‘probability of a dominated flow loop for random flow paths on a plane’.

The existence of a dominated flow loop anywhere between the points of intersection of the flow paths implies the existence of a triangular dominated flow loop between some of the intersection points. This is because for any dominated flow loop, it is always possible to select one closing edge and two non-zero flow paths with flows following the direction of the dominant flow. These three paths will form a triangular dominated flow loop. As a result, the existence of a triangular dominated flow loop is a necessary condition for the existence of a dominated flow loop. The existence of a triangular dominated flow loop is also a sufficient condition for the existence of a dominated flow loop.

Consequently, the probability of a dominated flow loop for randomly oriented flow paths on a plane can be estimated by estimating the probability of a triangular dominated flow loop. Similar to the case of closed parasitic flow loops, a unit vector can be assigned to each flow path with non-zero flow (Fig.13b), pointing in the direction of the flow along the path.

The angle α which the unit vector subtends with the horizontal axis (Fig.13b) gives the orientation of the flow path and the direction of the flow along the path, if the path carries non-zero flow. Each flow path is randomly oriented, which means that the angle which its direction vector subtends with the fixed horizontal x -axis is uniformly distributed in the interval $(0, 2\pi)$. In other words, all possible flow path orientations are characterized by the same probability.

The direction vectors of the separate paths can all be translated at the common origin O , as is shown in Fig.13b. Two types of directions will be distinguished: *non-zero direction vectors (nz-vectors)*, corresponding to paths with non-zero flow and *zero direction vectors* - corresponding to paths whose flow is zero. It is assumed that the flow along any non-zero flow path fills the entire available throughput capacity of the path.

Note that if all nz-vectors lie in the same half plane, no closed parasitic flow loop exists. This follows from Theorem 1. Conversely, if it is possible to select three nz-vectors which do not lie in the same half plane, then a closed parasitic flow loop will certainly be present. (See the proof of Theorem 1).

Here, it needs to be pointed out immediately, that two non-zero flow paths intersected by a zero flow path may form a dominated flow loop (Fig.14a).

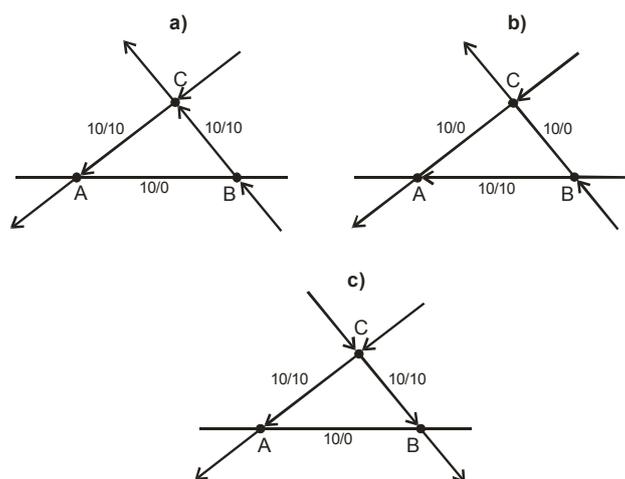


Figure 14. a) Intersecting two non-zero flow paths with a zero flow path yields a dominated parasitic flow loop b) The drained dominated parasitic flow loop from a); c) Intersection of two non-zero flow paths with a zero path does not always result in a dominated parasitic flow loop.

Indeed, consider the example in Fig.14a where the non-zero flow paths BC and CA, each with capacity 10 units flow per unit time, carrying 10 units of flow per unit time, have been

intersected by the zero flow path AB with capacity 10 units of flow per unit time and carrying no flow. The cyclic path ACBA is augmentable dominated flow loop because the sum of the lengths of two sides in a triangle is always greater than the third side. From the dominated flow loop ACBA 10 units of flow can be drained. The result from the augmentation is the network in Fig.14b. In Fig.14c, the non-zero flow paths have been intersected with a zero flow path but in this case, no dominated flow loop is present.

Lemma 2. *For a single linear path with zero flow and several linear non-zero flow paths, a dominated flow loop is absent only if all nz-vectors corresponding to the non-zero flow paths reside in a single half-plane defined by the direction of the zero-flow path.*

Lemma 3. *For a single linear path with zero flow and several linear non-zero flow paths, a dominated flow loop is present only if the non-zero flow paths do not all reside in any of the two half-planes defined by the direction of the zero flow path.*

Lemmas 2 and 3 can be proved with reasoning similar to the reasoning used for proving Theorem 1. According to Lemma 3, in the flow path configurations presented in Fig.15a and Fig.15b, a dominated flow loop exists because the nz-vectors do not reside in one of the two half-planes defined by the direction of the zero-flow path. The flow loop is dominated because the sum of the lengths of the two edges with non-zero flow (pointing against the direction of traversal) is always greater than the length of the edge with zero flow.

In Fig.15c and 15d, both nz-vectors reside solely in one of the two half-planes defined by the direction of the zero-flow path and, according to Lemma 2, no dominated flow loop exists. Dominated flow loops do not exist if two zero-flow paths are intersected with a single non-zero flow path. In this case, condition (9) for the existence of a dominated flow loop is violated — there is absence of dominating flow in a particular direction of traversal.

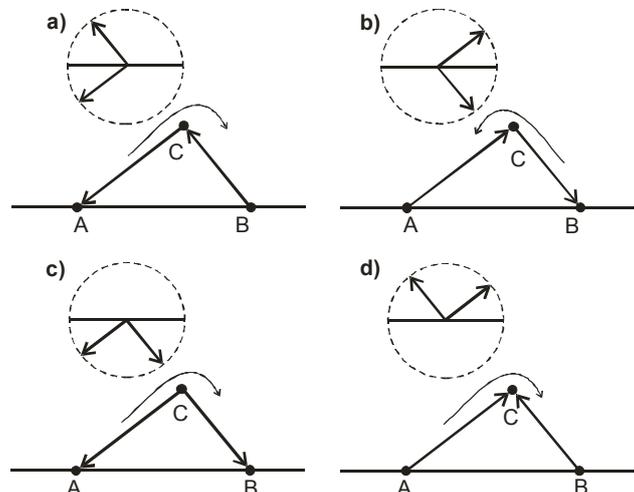


Figure 15. a,b) Flow configurations which: a,b) result in a dominated flow loop; c,d) do not result in a dominated flow loop.

The probability that all random nz-vectors reside in one of the two half-planes defined by the direction of the zero flow path is $1/2^{n-1}$ (Fig.16a). Consequently, for n non-zero randomly oriented flow paths, intersected with a single randomly oriented zero-flow path, the probability of existence of a dominated flow loop is $p = 1 - 1/2^{n-1}$.

Theorem 4. *For more than one linear zero-flow paths and more than one non-zero linear flow paths, dominated flow loops between the intersection points of the flow paths are absent if and only if all nz-vectors reside in a single sector (defined between two zero-flow paths) within which no other zero-flow path is present.*

To clarify this theorem, in Fig.16b, the four sectors β_1 , β_2 , β_3 and β_4 are defined by the two zero-flow paths z_1 and z_2 . No dominated flow loop is present for this orientation of the flow paths because all nz-vectors u_1, u_2 and u_3 , defining the directions of the flows in the nz-paths, reside in a single sector (sector β_3). In Fig.16c and 16d however, not all nz-vectors reside in a single sector defined by the directions of the zero-flow paths. As a result, a dominated flow loop will be present.

Proof. Suppose that no dominated flow loop is present. Consider the first zero-flow path. If a dominated flow loop is absent, according to Lemma 2, all nz-vectors must lie in a single half-plane defined by the first zero-flow path. Consider now the second zero-flow path. All nz-vectors must also lie in a single half-plane defined by the second zero-flow path. This is possible only if all nz-vectors lie in the intersection of the two half-planes (defined by the first and the second zero-flow paths), which is a sector. Within this sector of intersection, the first and the second zero-flow paths are not present. Consider now the third zero-flow path. All nz-vectors must also lie in a single half-plane defined by the third zero-flow path. This is possible only if all nz-vectors lie in the non-empty intersection of the existing sector (defined by the intersection of the first and the second half-plane) and the half-plane defined by the third zero-flow path. The non-empty intersection of a half-plane and a sector, is a sector. Within this sector of intersection, the first, the second and the third zero-flow paths are not present. Continuing this inductive reasoning with the rest of the nz-paths, it is derived that all nz-vectors must lie in the non-empty intersection of all half-planes defined by the zero-flow paths. This intersection is a sector within which no zero-flow paths are present.

Suppose now that all nz-vectors reside in a single sector. For a dominated flow loop to exist, a triangular dominated flow loop must necessarily exist with two intersecting nz-paths and one zero-flow path. Because all nz-vectors reside in a single sector, all nz-vectors reside in a single half-plane defined by the first zero-flow path. According to Lemma 3, the first zero flow path and the nz-paths cannot form a dominated flow loop. This reasoning is valid for the second zero-flow path, and so on. Consequently, no dominated parasitic flow loop exists if all nz-vectors reside in a single sector. The theorem has been proved \square

Similar to closed parasitic flow loops, the probability of existence of dominated flow loops in networks is also surprisingly high. Consider the system of intersecting randomly oriented linear flow paths on a plane depicted in Fig.17a: two perpendicular linear paths which do not carry flow and n randomly oriented linear flow paths, each of which carries flow to its full capacity.

For this system of flow paths, the probability of existence of a dominated flow loop between the intersection points formed by the linear paths is

$$p = 1 - 1/4^{n-1} \quad (16)$$

which is a value very close to unity even for small values of n . Indeed, the probability that the first nz-vector will lie in the first quadrant is $1/4$. Consequently, the probability that all n nz-vectors will lie in the first quadrant is $(1/4)^n$. All nz-vectors can lie in a single quadrant in four different, mutually exclusive ways: if they all lie in the first quadrant, in the second quadrant, in the third quadrant or in the fourth quadrant. The probability q of the union of these mutually exclusive events is equal to the sum of the probabilities of the separate events:

$$q = 1/4^n + 1/4^n + 1/4^n + 1/4^n = 1/4^{n-1} \quad (17)$$

The probability $p = 1 - q$ that not all nz-vectors will lie in a single quadrant is given by equation (16). This is the probability of a dominated flow loop.

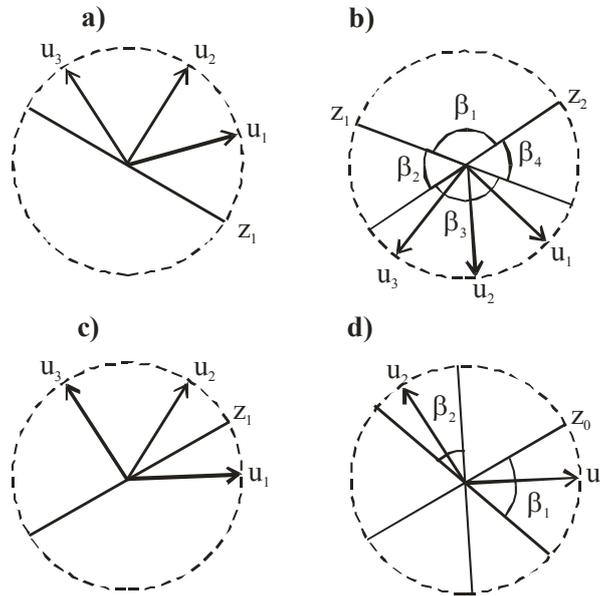


Figure 16. a) and b) A dominated parasitic flow loop does not exist; c) A dominated triangular parasitic flow loop exists; d) Two nz-vectors in two different sectors.

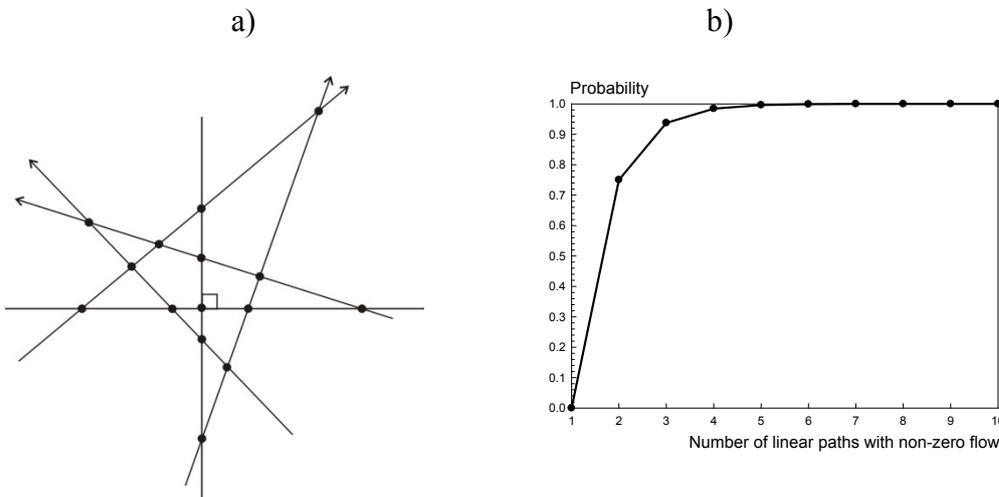


Figure 17. a) A network formed by the intersection points of randomly oriented linear flow paths on a plane. b) The probability of a dominated flow loop increases drastically, with increasing the number of paths with non-zero flow.

A simulation algorithm for estimating the probability of a dominated flow loop

For different numbers of randomly oriented zero-flow paths and non-zero flow paths, the probability that no dominated flow loop will be present can be estimated by a simulation algorithm. The algorithm simply determines the probability that all nz-vectors will reside in a single sector formed by the directions of two zero-flow paths. This can be performed easily, if the random orientation angles of the zero-flow paths and the nz-vectors are sorted (separately) in ascending order.

The algorithm is given in Appendix A. For randomly oriented non-zero flow paths and zero flow paths, the probability of a dominated flow loop for a different total number of flow paths has been plotted in Fig.18. The number of zero-flow paths is equal to the number of non-zero flow paths.

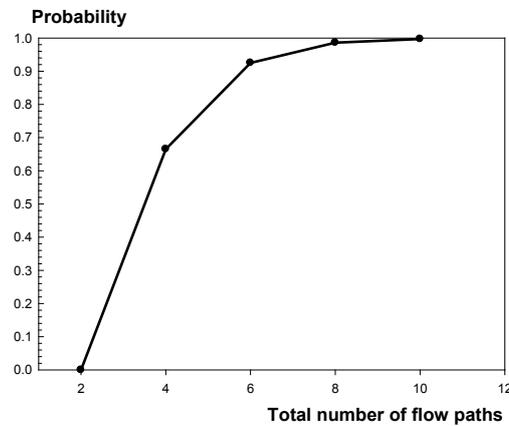


Figure 18. Probability of a dominated parasitic flow loop as a function of the total number of flow paths. The number of non-zero flow paths is equal to the number of zero flow paths.

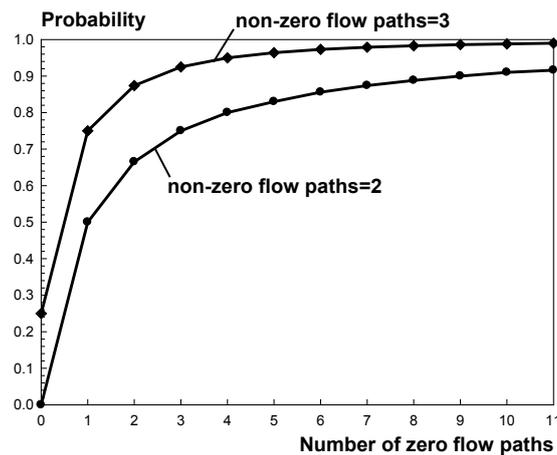


Figure 19. Probability of a dominated flow loop as a function of the number of randomly oriented zero-flow paths, for $n=2$ and $n=3$ nz-paths.

With increasing the total number of flow paths, the probability of a dominated flow loop approaches unity very quickly.

In Fig.19, the probability of a dominated flow loop has been plotted as a function of the number of zero-flow paths, for a mixture of zero-flow paths and non-zero flow paths, where the number of non-zero flow paths has been fixed to $n=2$ and $n=3$, respectively.

Again, with increasing the number of randomly oriented zero-flow paths, the probability of a dominated flow loop increases very rapidly. For three intersecting non-zero flow paths and two zero flow paths, the probability of a dominated flow loop is about 87%!

These results demonstrate that the existence of dominated flow loops on a set of intersecting zero-flow and non-zero flow paths is practically inevitable and the state where dominated flow loops are present can be considered to be the ‘natural state’ of flow networks.

5. Minimising transportation cost in a network with interchangeable sources

Suppose that a number of interchangeable sources and a number of destinations have been specified on a transportation network. A connection between a source and a destination will be thought as a unit flow sent from the source to the destination. It can be shown that for a set of interchangeable origins and a set of destinations, the next theorem holds.

Theorem 5. *The necessary and sufficient condition for a minimal total length of the transportation routes from a number of interchangeable origins to a number of destinations is the non-existence of closed parasitic flow loops and dominated flow loops.*

The proof that the condition is necessary is trivial and will be omitted. To prove that the condition is sufficient, Lemma 4, stated and proved in Ahuja et al. (1993) will be used.

Lemma 4. *In a flow network, if there are two different feasible edge flows $f_1(i, j)$ and $f_2(i, j)$ resulting in the same throughput flow, the flow $f_2(i, j)$ can always be presented as a sum of the flow $f_1(i, j)$ and the augmented flows along a set of augmentable cyclic paths.*

Proof of Theorem 5. Suppose that for edge flows $f_1(i, j)$, no closed parasitic loops or dominated loops exist and the throughput flow is equal to n units (from n interchangeable sources to n destinations). We will show that the total length D_1 of connections, associated with edge flows $f_1(i, j)$ is the smallest possible. Suppose that this is not the case and edge flows $f_2(i, j)$ exist, associated the same throughput flow of n units and a smaller total length $D_2 < D_1$ of the connections. According to Lemma 4, the edge flows $f_2(i, j)$ can be obtained from edge flows $f_1(i, j)$ after adding the augmented flows $\Delta\gamma_i$ along k cyclic paths.

$$f_2(i, j) = f_1(i, j) + \Delta\gamma_1 + \Delta\gamma_2 + \dots + \Delta\gamma_k \quad (18)$$

The total length of connections D_2 corresponding to edge flows $f_2(i, j)$ is linked with the total length D_1 of connections corresponding to edge flows $f_1(i, j)$ through the expression:

$$D_2 = D_1 + \Delta D_1 + \Delta D_2 + \dots + \Delta D_k \quad (19)$$

where ΔD_i is the change of the length of connections associated with augmenting the i th cyclic path. Because, according to the statement of the sufficient condition, there are no closed or dominated flow loops in the network, all k augmentations $\Delta\gamma_i$ ($i=1, \dots, k$) can only result in non-negative changes ΔD_i of the length of connections ($\Delta D_i \geq 0$). Consequently, $D_2 > D_1$, which contradicts the assumption that edge flows $f_2(i, j)$ are associated with a smaller total length of connections D_2 . We arrived at a contradiction. Therefore, edge flows $f_1(i, j)$ are associated with the smallest total length of connections. The sufficient condition has been proved. \square

It can also be shown that that the process of augmenting with flow dominated flow loops is finite and will end after a finite number of steps.

Theorem 6. *The process of augmenting cyclic paths with dominating flow until no such paths can be found, is finite and will end after a finite number of augmentations.*

Proof. The network connections on a transportation network can be thought as flow paths, each carrying a single flow unit. Because the lengths of the edges can always be expressed as

rational numbers, multiplying the lengths of the edges by their common denominator will express all lengths with integer numbers.

Suppose that there are m edges in the network and the maximum possible length of an edge is u_{\max} units. The total length of connections among interchangeable origins and destinations therefore, cannot exceed mu_{\max} length units. Because only cyclic paths with dominating flow are augmented and after each flow path augmentation a dominated flow loop ceases to exist, each augmentation of a dominated flow loop reduces the total length of connections by at least one length unit. Because there can be at most mu_{\max} length units, after a finite number of steps, there will be a step at which no dominated parasitic flow loop will be present.

During this operation, the invariant property (“each interchangeable origin is connected to exactly one destination”) is preserved. The obtained solution is a feasible solution and also an optimal solution.

6. Dominated parasitic flow loops resulting from building successive shortest-path routes from multiple origins to multiple destinations

Repeated draining of flows along dominated flow loops can be used to reduce the total length of transportation routes/connections between multiple interchangeable origins and multiple destinations.

Consider a network including multiple interchangeable origins and multiple destinations. The network in Fig.20a features three taxi cars belonging to the same company. The taxi cars are currently at locations $s1$, $s2$ and $s3$, and must be dispatched at the same time to three passengers at locations $d1$, $d2$ and $d3$. Suppose that the lengths of the edges represent the actual lengths of the road sections on the map. The successive shortest-path strategy in this case fails to find the optimal solution. The shortest path from taxi car $s1$ to passenger $d1$ is the road section (5,11); the shortest path from taxi car $s2$ to passenger $d2$ is the road section (12,8) and the shortest path from the remaining taxi car $s3$ to the remaining passenger $d3$ is the road section (3,6).

As can be seen from Fig.20, the cyclic path (5,11,12,8,3,6,5), is a cyclic path with dominating flow because more than half of the length of the cyclic path is occupied with flow in the specified direction of traversal. The dominated flow loop can be drained by pushing a unit flow in the opposite direction of the specified direction of traversal. As a result, the dominated parasitic flow loop disappears while the connections from the taxi cars to the passengers are preserved (Fig.20b).

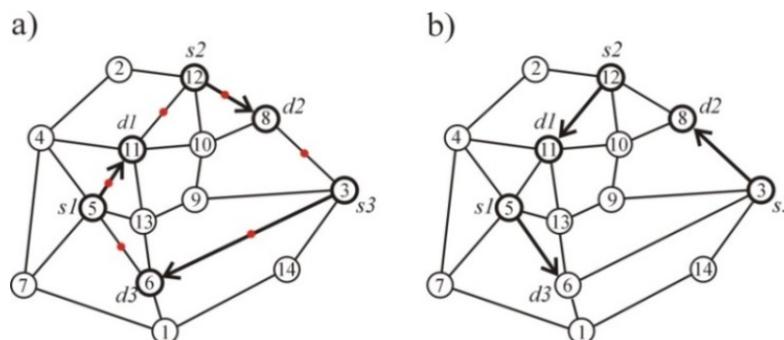


Figure 20. a) A counterexample network with edges with unlimited capacity. Contrary to the conventional wisdom, selecting the nearest available destination to each origin or the nearest available origin to each

destination does not guarantee a minimum total sum of the routes because of the dominated flow loop (5,11,12,8,3,6,5). b) The network with the optimal routes.

It can be shown that for any number of interchangeable origins connected to a number of destinations, where each origin is connected to exactly one destination, augmenting a dominated parasitic flow loop with flow in the opposite direction of the dominating flow always maintains the invariant: *each interchangeable origin services exactly one destination and each destination is being serviced by an origin.*

Theorem 7. *The augmentation of cyclic paths preserves the invariant property: each origin services exactly one destination and each destination is being serviced by exactly one origin.*

Proof: Because all n origins initially supply a unit flow to each of the n destinations, after an augmentation of a cyclic path, n origins will still originate n units of flow (one flow unit per origin) and n destinations will still accept exactly n units of flow (one unit per destination). In addition, at each node belonging to the cyclic path, the flow conservation will be preserved after the augmentation. It can be shown that after any number of augmentations of cyclic paths, each origin remains connected with exactly one destination.

Indeed, a process of path tracing can be started from one of the origins s . An edge coming out of the origin can always be selected, carrying a unit flow towards a particular node i . According to the flow conservation law, there must be another edge incident to node i , carrying a unit flow which goes out of node i . By selecting this edge, the node $i+1$ is reached and so on. By continuing this process, the traced path visits either (i) an already visited node or (ii) a destination t (Fig.21).

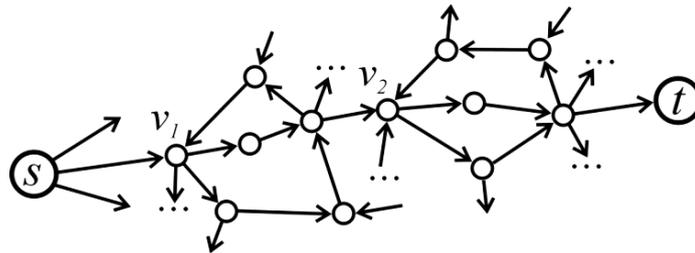


Figure 21. Tracing a connecting path from a source s to a destination t .

(i) Consider the first alternative. If a node v_1 has been visited again before the destination t has been reached (Fig.21), a directed flow cycle has been discovered in the network. A unit flow can then be removed from all edges of this cycle, which will result in a new feasible network flow where the flow conservation law is preserved at each internal node (different from an origin or a destination). After removing the unit flow from all edges belonging to the directed flow cycle, the process of path tracing continues from the repeated node v_1 . The continuation is guaranteed, because exactly one unit of flow entering node v_1 and exactly one unit of flow leaving node v_1 had been removed. Therefore, according to the flow conservation law, at node v_1 there must be a non-empty edge leaving the node. If another cycle is encountered at node v_2 , a unit of flow is removed from all edges and the process of path tracing continues from node v_2 . By continuing this process, it is guaranteed that a destination t will finally be reached. This is guaranteed, *because the flow conservation law holds at each encountered internal node and the network has been saturated with a finite number of flow units.*

(ii) Consider now the second alternative for which a destination t has been reached. The traversed edges, each carrying a unit flow, then constitute a directed s - t path. From each of the forward edges of the traced s - t path, exactly one unit of flow is removed. This procedure will subtract one unit of flow from the only non-empty outgoing edge from the origin s and exactly one unit of flow from the only non-empty edge entering the destination d . This procedure effectively disconnects the origin s and the destination d from the network and leaves a network with $n-1$ origins and $n-1$ destinations.

Another origin with a non-empty outgoing edge can then be selected and the entire process of origin-destination tracing is repeated until another origin-destination match is found. This is guaranteed because the flow conservation law at the nodes has been preserved by the removal of the previous traced s - t path. Continuing this procedure will, in the end, match each origin to each destination with a unique directed connecting path, carrying a single unit of flow. \square

7. Dominated parasitic flow loops from building successive shortest-path routes from multiple origins to a single destination and from a single origin to a single destination

Consider now Fig.22a, which illustrates the transportation of interchangeable commodity from three supply sources (origins) $s1$, $s2$ and $s3$ with unlimited production capacity to a common destination d which requires the delivery of 30 units flow per unit time. Each of the links $(5,d)$, $(4,d)$ and $(6,d)$, connecting the common destination d with the rest of the network, has the same length l_d and capacity limited to 10 units of flow per unit time. The shortest paths from $s1$ to d is $(1,4,d)$, along which 10 flow units per unit time can be dispatched; the shortest path from $s2$ to d is $(2,6,d)$, along which 10 more units per unit time can be dispatched and the shortest path from $s3$ to d is $(3,5,d)$, along which another 10 units of flow per unit time can be dispatched. Despite the selection of the shortest paths, a dominated flow loop $(1,4,2,6,3,5,1)$ appears in the ‘optimised’ network. As can be verified, in the network from Fig.22, the successive shortest path strategy *fails to determine the optimal solution if the links connecting the destination with the rest of the network have a limited throughput capacity*.

Augmenting the dominated flow loop $(1,4,2,6,3,5,1)$ yields the network in Fig.22b, which is characterised by the smallest total length of connections from the interchangeable origins to the common destination.

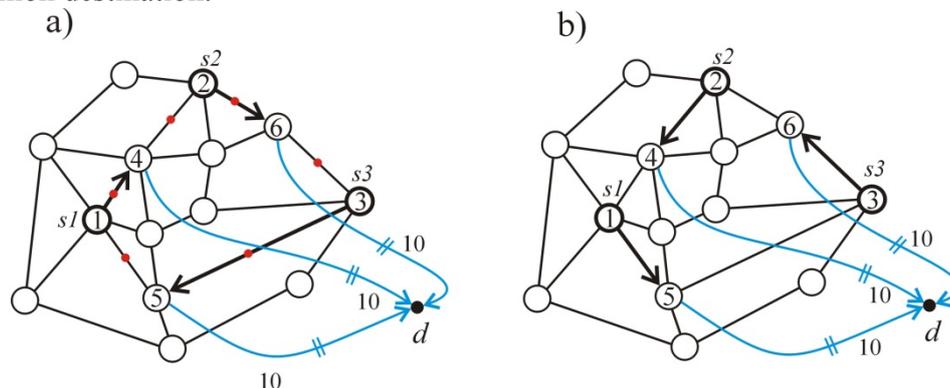


Figure 22. A counterexample network which demonstrates that the successive shortest-path strategy does not even guarantee a minimum total length of the routes in the case of multiple origins and a single destination

These examples emphasize the importance of an algorithm for removing dominated flow loops from networks. The approach based on repeated augmentations of dominated flow loops is applicable only to small networks.

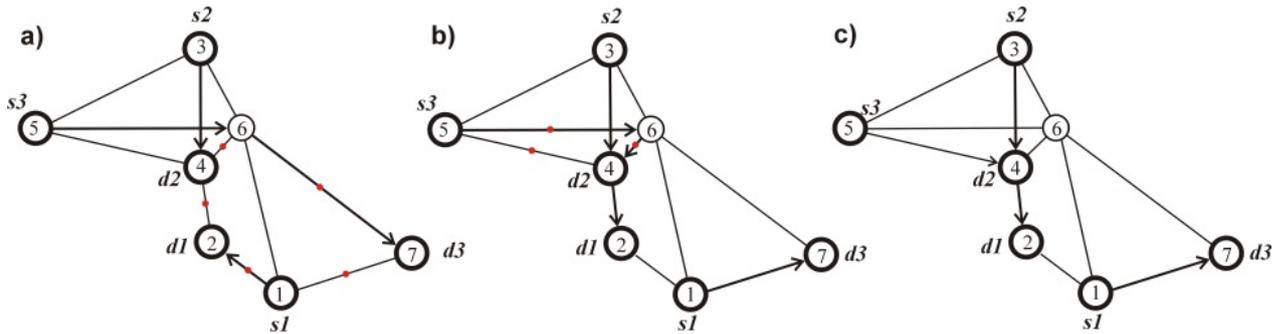


Figure 23. Repeated draining of dominated flow loops by reversing the dominating flow ensures a minimal total length of the connections and is a way of drawing value from existing supply networks.

Consider the network in Fig.23a which features three interchangeable origins s_1 , s_2 and s_3 , connected to three destinations d_1, d_2 and d_3 . Suppose that the lengths of the edges represent the distances between the nodes. The shortest path from s_1 to d_1 is the edge (1,2); the shortest path from origin s_2 to destination d_2 is the edge (3,4) and the shortest path from the remaining origin s_3 to the remaining destination d_3 is (5,6,7).

The cyclic path (1,2,4,6,7,1) however, is a cyclic path with dominating flow and can be drained by augmenting it with a unit flow in the opposite direction of the direction of the dominant flow. As a result, the dominating flow disappears from the cyclic path while the connections between origins and destinations are preserved (Fig.23b). In Fig.23b, the cyclic path (5,6,4,5) is now with dominating flow which can be drained by augmenting with unit flow in a direction (5,4,6,5). This operation results in the network from Fig.23c where no cyclic paths with dominating flow are present and the total length of connections for delivering the service from the interchangeable origins to the destinations is the smallest possible. The congestion associated with this transportation network is also minimised. In the example from Fig.23a, the successive shortest-path strategy fails to find the optimal solution.

Unlike the approach based on repeated augmentations of dominated flow loops, the next algorithm is suitable for a network of any size and complexity.

At the first step of the algorithm, the flow network with multiple origins s_1, s_2, \dots, s_n and destinations d_1, d_2, \dots, d_n (Fig.24a) is reduced to a flow network with a single super-source s (Fig.24b) and a single super-destination t . The multiple origins in Fig.24a are replaced by a single super-source s with infinite flow generation, feeding each of the initial origins s_i through lines whose capacity is exactly one flow unit. In a similar fashion, the multiple destinations in Fig.24a are replaced by a single super-destination t connected with each of the separate destinations d_i through edges whose capacity is exactly one flow unit. The sources of flow (the origins) and the destinations become ordinary throughput edges, with flow capacities equal to one. As a result, the interchangeable origins and the destinations ‘disappear’, and instead, ordinary throughput edges appear.

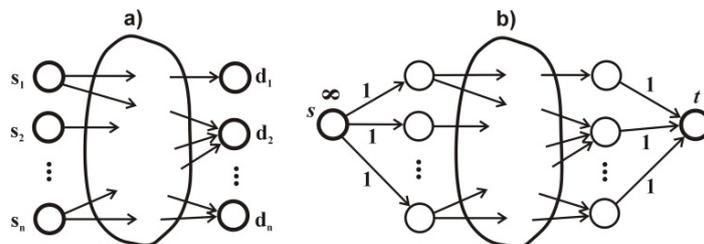


Figure 24 a) An example of a flow network with n interchangeable sources and n destinations. b) The network (a) has been reduced to a flow network with a single super-source s , and a single super-destination t .

The second step consists of assigning zero length to the edges connecting the super-source s and the super-destination t with the rest of the network. For the rest of the edges, the length of the edges corresponds to the real distances between the nodes.

The third step consists of maximising the flow in the transformed network (Fig.24b), from the super-source s to the super-destination t , at a minimum total transportation cost. The transportation cost associated with the separate edges is assumed to be proportional to their length. A number of algorithms have already been proposed for maximising the flow at a minimum total cost, some of which are characterised by a strictly polynomial running time (Tardos 1985; Orlin 1993). The method based on modified weights for example, can be used for maximising the throughput flow at a minimum cost (Todinov, 2013a). It has a worst-case running time $O(|f^*| m \log n)$ where $|f^*|$ is the magnitude of the throughput flow, m is the number of edges and n is the number of nodes in the network. Because the interchangeable origins are connected with the super-source s through edges with unit capacity, and the number of these edges does not exceed n , the worst-case running time of the algorithm for removing all dominated flow loops is $O(nm \log n)$.

At the last step, the super-source s , the super-destination t and their connecting edges are removed from the transformed network. The nodes connected to the super-source and the super-destination become origins and destinations again. The edge flows in the resultant network define the optimal solution.

After conducting the third step of the algorithm, it can be shown that there are no dominated parasitic flow loops left in the network. Indeed, suppose that the specified maximum throughput flow of n units has been guaranteed and the total sum of the length of connections between origins and destinations is the smallest possible. Suppose that there exists a dominated parasitic flow loop in the network. In this case, the dominated flow loop can be augmented in a direction opposite to the dominating flow. After the operation, each of the origins will be connected to a destination but the total length of connections will be reduced, therefore the total transportation cost will be reduced. However, this contradicts the assumption that the total transportation cost is the smallest possible. Consequently, there can be no dominated parasitic flow loops in a network where the total transportation cost is minimal.

Here is an illustrating example. Consider 5 volunteers belonging to the same organisation and living in different parts of a city allocated to 5 patients also living in different parts of the city (Fig.25a). The services of the volunteers are interchangeable and each volunteer must be assigned to exactly one patient. For the sake of simplicity, it has been assumed that all edges in the transportation network have the same length. One possible way of satisfying the demand from patients is shown in Fig.25a. Each origin has been connected with the corresponding destination through a direct, straight-line path carrying a single unit of flow. If the cost of transportation per edge is 100 units, the total cost of transportation to all destinations is 2500 units.

The solution produced by the proposed algorithm is shown in Fig.25b. The resultant network is characterised by a total transportation cost of 1500 units only, which constitutes a 40% reduction of the initial total transportation cost. As can be verified, no dominated parasitic flow loops exist, hence, the obtained transportation plan is associated with the smallest total transportation cost.

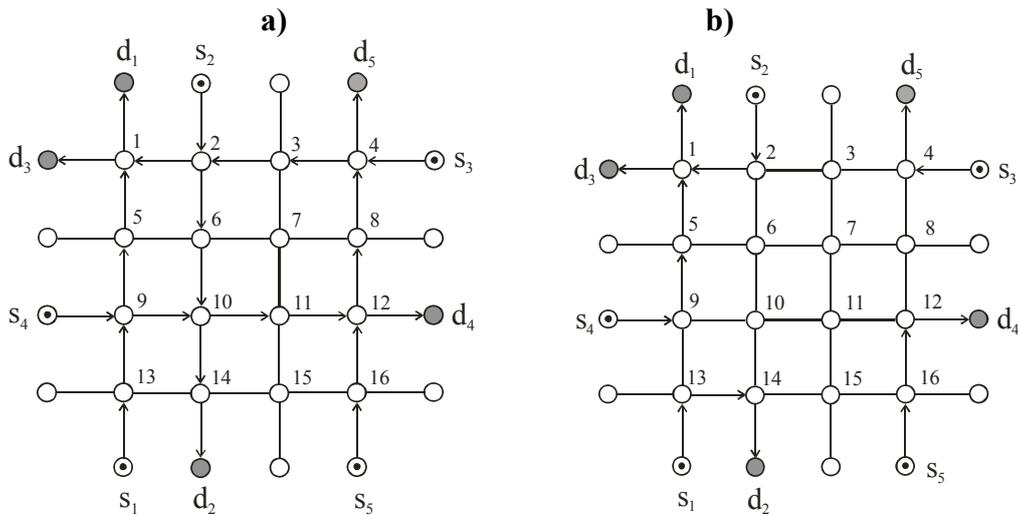


Figure 25. A transportation network a) before the optimisation and b) after the optimisation.

Conclusions

The paper demonstrates that despite the six decades of intensive research on static flow networks, existing algorithms for maximising the throughput flow published since 1956 yield solutions with closed parasitic flow loops. Effectively, part of the flow which enters the network is involved in endless circulation along a closed path and never leaves the network.

The paper demonstrates that a closed routing loop can effectively be present even if the routed commodity does not physically travel along a closed loop, as long the routed commodity is interchangeable commodity. In this case, despite that no commodity is physically moved around a closed loop, effectively, the same type of commodity is being circulated around a closed loop.

Consequently, the paper introduces a theoretical and computational framework related to the appearance of closed and dominated flow loops in flow networks. By using the developed framework, it has been demonstrated that the probability of existence of closed and dominated flow loops in networks is surprisingly high.

Closed-form expressions have been obtained for the probability of a parasitic flow loop among the intersection points of randomly oriented linear flow paths on a plane and for flow paths forming a lattice. A simulation programme has been developed for determining the probability of a dominated flow loops for randomly oriented linear flow paths on a plane.

Necessary and sufficient conditions have been derived for the existence of closed parasitic flow loops and dominated flow loops among the intersection points of randomly oriented linear flow paths on a plane.

The paper demonstrated that the successive shortest path strategy for minimising the total length of transportation routes from multiple interchangeable origins to multiple destinations fails to minimise the total length of the routes.

The paper demonstrates that that the successive shortest path strategy fails to minimise the total length of the transportation routes, even in the case of multiple interchangeable origins and a single destination, if the connections of the single destination with the rest of the network have limited throughput capacities.

By using the developed theoretical framework, it is demonstrated that a minimum total length of the transportation routes in a network with multiple interchangeable origins, is attained if and only if no closed parasitic flow loops or dominated flow loops exist in the network.

Accordingly, an algorithm for minimising the total length of transportation routes by eliminating all closed parasitic flow loops and dominated flow loops has been proposed.

The paper exposes the highly detrimental effect of parasitic flow loops in networks. Parasitic flow loops:

- increase the cost of transportation of the flow unnecessarily,
- consume residual capacity from the edges of the network;
- increase the likelihood of deterioration of perishable products;
- increase congestion and wearout of the transportation links;
- energy is unnecessarily wasted for maintaining these loops.

Optimizing supply networks by draining highly undesirable dominated flow loops derives significant value by reducing the transportation costs, the risk of congestion and accidents, and the environmental pollution. The end result is a huge amount of resources saved to the world economy.

APPENDIX A. Algorithm for determining the probability of a dominated flow loop.

This is the algorithm in pseudo-code for determining the probability of a closed parasitic flow loop or a dominated flow loop. The variable ‘count’ counts the number of times a dominated flow loop has not been found and the variable ‘Probability’ contains the probability of a dominated flow loop. The statements within braces ‘{...}’ are treated as a compound statement.

// The number n of non-zero flow paths is $n \geq 2$;
// The number m of zero flow paths is $m \geq 0$;

counter = 0;

if (m=0) **then** *// there are no zero flow paths*

{
 if (n=2) **then** Probability = 0;
 else if (n>=3) **then** Probability = $1 - n / 2^{n-1}$;
 }

else *// there are zero flow paths (m>0)*

{
 for i=1 **to** number_of_trials **do**

{
 Generate ‘n’ random nz-vectors corresponding to the directions of the flows in the non-zero flow paths;
 Sort their orientation angles in ascending order;

Generate ‘m’ random directions which correspond to the zero flow paths; Sort in ascending order the angles defining the sectors formed by the directions of the zero flow paths;

Check if all nz-random directions reside in a single sector; Set the variable Y to 1 if this is the case and to 0, otherwise;

```

    if (Y=1) then count = count + 1;
  }

  Probability = 1 - count/number_of_trials;
}
Print ('Probability of a closed or dominated flow loop = ', Probability);

```

References

- Ahuja R.K., Orlin J.B. (1991) 'Distance-directed augmenting path algorithms for maximum flow and parametric maximum flow problems', *Naval Research Logistics*, Vol.38, pp.413-430.
- Ahuja R.K., T.L.Magnanti, J.B.Orlin (1993) *Network flows: Theory, Algorithms and Applications*, Prentice Hall.
- Asano T, Asano Y. (2000) 'Recent developments in maximum flow algorithms', *Journal of the Operational Research Society in Japan*, Vol.43 No 1, pp.2-31,.
- Bohdanowicz F., Dickel H., Ch.Steigner Ch. (2009), 'Deterction of routing loops', *Proceeding of the International Conference on Information Networking*, ICON 2009, pp.1-5.
- Chandran B.G., Hochbaum D.S. (2009), 'A Computational Study of the Pseudoflow and Push-Relabel Algorithms for the Maximum Flow Problem', *Journal Operations Research*, Vol.57 No.2, pp.358-376.
- Cormen T.H., Leiserson T.C.E., Rivest R.L. and Stein C. (2001), *Introduction to Algorithms*, 2nd ed., MIT Press and McGraw-Hill.
- Dinic E.A. (1970) 'Algorithm for solution of a problem of maximum flow in a network with power estimation', *Soviet Math. Dokl.* Vol.11 No 8, pp.1277-1280.
- Edmonds J, Karp R.M. (1972), 'Theoretical improvements in algorithmic efficiency for network flow problems', *Journal of the ACM*. Vol.19, No 2, pp.248-264.
- Elias P, Feinstein A, Shannon C.E. (1956), 'Note on maximum flow through a network', *IRE Trans. Inf. Theory*. IT2, pp.117-119.
- Ford L.R. and D.R. Fulkerson (1956), 'Maximal flow through a network', *Canadian Journal of Mathematics*, Vol.8, No 5, pp.399-404.
- Fu J., Sjodin P., Karlsson G. (2008) 'Loop-free updates of forwarding tables', *Proceedings of IEEE Transactions on Network and Service Management*, Vol.5 No.1.
- Hengartner U., Moon S., Mortier R., Diot C. (2002) 'Detection and analysis of routing loops in packet traces', *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pp.107-112.
- Dong J., W.Li, Cai C., Chen Z. (2009) 'Draining algorithm for the maximum flow problem', *International conference on communications and mobile computing*, pp.197-200.
- Goldberg A.V., Tarjan R.E. (1988) 'A new approach to the maximum flow problem', *Journal of the ACM*, Vol.35, pp.921-940.

- Goldberg A., Hed S., Kaplan H., Werneck R.F., (2011) ‘Maximum Flows by Incremental Breadth-First Search’, *Proceedings of the 19th European conference on Algorithms - ESA 2011*, pp.457-468.
- Goldberg A.V., Rao S., (1998) ‘Beyond the flow decomposition barrier’, *Journal of the ACM*, Vol.45 No.5, pp.783-797.
- Goodrich M.T. and Tamassia R. (2001) *Algorithm design: Foundations, Analysis and Internet examples*, John Wiley & Sons.
- Hochbaum D.S. (2008) ‘The pseudoflow algorithm: a new algorithm for the maximum-flow problem’, *Operations Research*, Vol.56 No.4, pp.992-1009.
- Karzanov A. (1974) ‘Determining the maximal flow in a network by the method of preflows’, *Sov Math Dokl.*, Vol.15, pp.434-437.
- Kleinberg J. and Tardos E. (2006) *Algorithm design*, Addison Wesley.
- Li S., Duan H., Wang Zh., Li X., (2015) ‘Route leaks identification by detecting routing loops’, in the series *Lecture Notes of the Institute of Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol.164, pp.313-329.
- Madry A. (2016) Computing Maximum Flow with Augmenting Electrical Flows, Proceedings of the 2016 IEEE 57th Annual Symposium on Foundations of Computer Science, pp.593-602.
- Mallick K.K., Khan A.R., Ahmed M.M., Arefin M.S., Uddin M.S. (2016), Modified Edmonds-Karp algorithm to solve maximum flow problems, *Open Journal of Applied Sciences*, Vol.6, pp.131-140.
- Orlin J., (2013) ‘Max flows in $O(nm)$ time, or better’, Proceedings of the forty-fifth annual ACM symposium on theory of computing, STOC 13, pp.765-774.
- Orlin J.B. (1993) ‘A faster strongly polynomial minimum cost flow algorithm’, *Operations Research*, Vol.41 No.2, pp.338-350.
- Sharathkumar R., Agarval P.K. (2012) ‘Algorithms for the transportation problem in geometric settings’, *Proc. of the 23rd annual ACM-SIAM symposium on discrete algorithms*, pp.306-317.
- Shiloach Y., Vishkin U. (1982) ‘An $O(n^2 \log n)$ parallel max-flow algorithm’, *Journal of Algorithms*, Vol.3, pp.128-146.
- Sleator D.D., Tarjan R.E. (1980) ‘An $O(nm \log n)$ algorithm for maximum network flow’, Technical report STAN-CS-80-831, Department of computer science, Stanford, CA, 1980, Stanford University.
- Tardos E. (1985). A strongly polynomial minimum cost circulation algorithm, *Combinatorica* Vol.5 No.3, pp.247-255.
- Todinov M.T. (2013a) *Flow networks*, Elsevier.
- Todinov M.T. (2013b) ‘The dual network theorem for static flow networks and its application for maximising the throughput flow’, *Artificial Intelligence Research*, Vol.2 No.1, pp.81-106.
- Wang I.L., Johnson E.L., Sokol J.S., (2005) ‘A multiple pairs shortest path algorithm’, *Transportation science*, Vol.39 No.4, pp.465 - 476.

- Yousef M., Hunter D.K., Al-Kandan A. (2015) 'Analysis of network layer routing loops', *Proceedings of the 4th International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*.
- Zhao L., W.Wang W. (2014) 'A new path-searching algorithm for providing paths from multiple origins and one single destination', *International journal of computer science and applications*, Vol. 3 No.1, pp.29-33.