Todinov, M

The throughput flow constraint theorem and its applications.

Todinov, M (2014) The throughput flow constraint theorem and its applications. *International Journal of Advanced Computer Science and Applications*, 5 (3).

doi: 10.14569/IJACSA.2014.050302

The Throughput Flow Constraint Theorem and its Applications

Michael T. Todinov
Department of Mechanical Engineering and Mathematical Sciences
Oxford Brookes University, Oxford, Wheatley,
OX33 1HX, UK;
mtodinov@brookes.ac.uk

***Abstract*—The paper states and proves an important result related to the theory of flow networks with disturbed flows: "the throughput flow constraint in any network is always equal to the throughput flow constraint in its dual network".**

After the failure or congestion of several edges in the network, the throughput flow constraint theorem provides the basis of a very efficient algorithm for determining the edge flows which correspond to the optimal throughput flow from sources to destinations which is the throughput flow achieved with the smallest amount of generation shedding from the sources.

In the case where a failure of an edge causes a loss of the entire flow through the edge, the throughput flow constraint theorem permits the calculation of the new maximum throughput flow to be done in $O(m)$ time, where *m* is the number of edges in the network. In this case, the new maximum throughput flow is calculated by inspecting the network only locally, in the vicinity of the failed edge, without inspecting the rest of the network.

The superior average running time of the presented algorithm, makes it particularly suitable for decongesting overloaded transmission links of telecommunication networks, in real time. In the paper, it is also shown that the deliberate choking of flows along overloaded edges, leading to a generation of momentary excess and deficit flow, provides a very efficient mechanism for decongesting overloaded branches.

Keywords*— networks with disturbed flows; congestion; decongestion; maximum throughput flow; telecommunication networks.

Todinov M.T., The Throughput Flow Constraint Theorem and its Applications, *International Journal of Advanced Computer Science and Applications,* Vol. 5, No. 3, 2014, pp. 11-24.

I. THE NEED FOR A HIGH-SPEED CONTROL OF FLOW NETWORKS

Although almost all real networks are networks with disturbed flows, the focus of existing research on flow networks has been exclusively on static flow networks. Research and algorithms related to static flow networks has been presented in [1-3]. The first major category of algorithms for maximising the throughput flow in networks includes the augmentation algorithms which preserve the feasibility of the network flow at all steps, until the maximum throughput flow is attained [4-5]. The second major category of algorithms are based on the preflow concept used in [6] and subsequently in [7] and [8]. The central idea behind these algorithms is converting the preflow into a feasible flow.

The best of these methods however, have a polynomial running time and do not provide the necessary computational speed for re-optimising the throughput flow in a large and complex network in real time, after an edge failure or congestion. The main reason is that classical algorithms for maximising the throughput flow start from a network with empty edges and do not make use of special properties of the network providing a short cut to determining the maximum throughput flow.

The central question for networks with disturbed flows is how to re-optimise the network flows after an edge flow disturbance (caused by edge failure or congestion), so that a *new optimal throughput flow* is attained quickly. The concept 'new optimal throughput flow' means a throughput flow achieved with a minimum reduction of flow production from the flow generators (with a minimum generation shedding).

After an edge failure or edge congestion, often there exists a possibility for redirecting the flow through alternative paths with non-zero residual capacity, so that a new throughput flow is reached quickly, with a minimum loss of flow. Even for relatively simple networks, it is not obvious how the edge flows should be reset in order to attain the required throughput flow, with a minimum generation shedding. Without an appropriate algorithm, the task of resetting correctly the edge flows in order to attain the new optimum throughput flow is almost impossible, for large and complex flow networks. In addition, the computational time of the algorithm must be within milliseconds, if the algorithm is to be capable of re-optimising the network flows in real time, after a contingency event. For very large networks (>10000 transmission links) an algorithm with approximately linear average running time in the size of the network is needed.

The lack of optimisation of the network flow after a contingency event leads to a severe disruption of the flow, suboptimal performance and loss of throughput flow. The importance of dynamically rerouting the traffic in telecommunication networks has been stressed in [9].

Despite the critical importance of the problem related to re-optimising the flows upon disappearance of an edge due to failure, it is difficult to find a relevant theoretical discussion. Such a discussion would provide the necessary short cut speeding the performance of algorithms calculating the re-optimised throughput flow. This problem has been mentioned in question 6.35b from [1], where the authors propose to the reader to show that after an overestimation of the capacity of an edge by *k* units, the labelling algorithm can re-optimise the maximum flow in O(*km*) time, where m is the number of edges in the network.

Disregarding the fact that the running time O(*km*) of the labelling algorithm is too slow for large *k*, the direct application of the labelling algorithm to solve this problem leads to sub-optimal solutions. This can be demonstrated immediately with the network in Fig.1, where the first number on the edges denotes the throughput capacity of the edge and the second number denotes the actual flow through the edge. Initially, the throughput flow has been maximised by using the labelling algorithm. The path (1,2,7) has been augmented with 40 units and path (1,3,6,7) with 60 units. There are no more augmentable *s-t* paths and according to the Ford-Fulkerson theorem [4] the throughput flow in the network is the maximum throughput flow.

Now suppose that the capacity of edge (3,6) has actually been overestimated by 30 units and the actual capacity of the edge is only 30 units. In this case, the labelling algorithm re-optimises the flow by diminishing the flow along the s-t path (1,3,6,7) by 30 units. However, the total throughput flow in the network from the source *s* to the sink *t*, obtained by the labelling algorithm, is 70 units, which is a sub-optimal value (Fig.1b).

If before diminishing the flow along the path (1,3,6,7) an additional operation was performed, the total throughput flow would be 80 units, not 70 units. Due to constraining the flow along edge (3,6) by 30 units a “momentary excess flow” appears at node 3 and a “momentary deficit flow“ appears at node 6. The network should be augmented first with the momentary excess flow of 30 units at node 3 aimed to cancel first the momentary deficit flow of 30 units at node 6. As can be seen, the maximum of 10 units momentary excess flow can be sent from node 3 to node 6 through path (3,4,2,5,6). After cancelling 10 units of flow, the remaining momentary excess flow at node 3 is 20 units and the remaining momentary deficit flow at node 6 is also 20 units. Now, by using the labelling algorithm, the momentary excess flow and deficit flow can be reduced to zero by diminishing the flow along the s-t path (1,3,6,7) by 20 units. The result is the network in Fig.1c where the total throughput flow is 80 units, which is the maximum possible throughput flow.

A similar deficiency is present in the algorithm reported in [10], treating the problem of maximising the flow in a static network by starting from a network where all edges are fully saturated with flow. This causes unbalanced *excess* and *deficit* nodes to appear in the network. The sum of the flows going into an excess node is greater than the sum of the outgoing flows while for a deficit node, the sum of the ingoing flows is smaller than the sum of the outgoing flows. The essence of the draining algorithm presented in [10] is to cancel excess flow with a deficit flow by augmenting paths starting from excess nodes and ending at deficit node. The process of cancellation of excess and deficit flow in [10] was done only in a network with a back circulation edge, connecting the sink with the source. In [11], it was shown that this approach leads to suboptimal solutions where the obtained throughput flow is feasible but not maximal. It has also been demonstrated in [11] that to achieve an optimal solution, there is a need of two distinctive stages. In the first stage, cancelling of excess and deficit flow is done in a network without a circulation edge. In the second stage, draining of excess and deficit flow is done in a network with a circulation edge.

In short, applying the labelling algorithm without an intermediate stage consisting of cancelling as much as possible excess and deficit flow results in sub-optimal solutions.

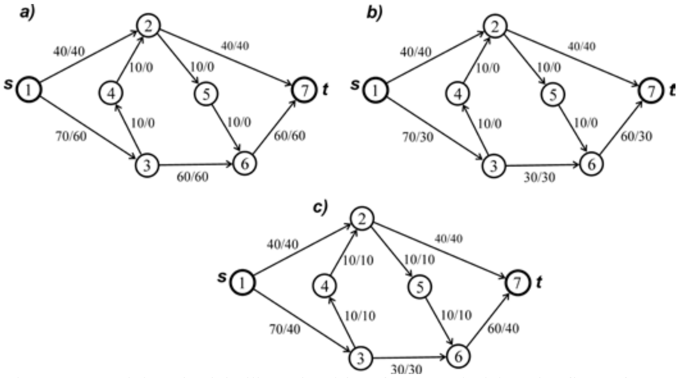

Figure 1. Applying the labelling algorithm for re-optimising the flow after an overestimation of the capacity of an edge results in a sub-optimal solution.

Component failures in flow networks and congestion are inevitable. These events lead to disappearance of flow capacity and the expected magnitude of the throughput flow from sources to destinations may not be guaranteed. As a result, the quality of service received from the network (which is a key performance characteristic) can be affected seriously.

These problems are particularly acute for telecommunication networks, oriented towards media applications, for transportation networks and power distribution networks, because they all require a high throughput flow rate. Selecting the shortest path for a data transfer, as it is commonly done [12] is often far from optimal. It is a common-sense strategy which often leads to overloading and congestion of network sections, and ultimately, to a low throughput flow.

Consequently, the objectives of this paper are: (i) to present a theoretical analysis of the important problem related to the flow constraint arising in the case of edge failures or congestion; (ii) to use the analysis for improving the efficiency of calculation of the new optimal throughput flow after failures or congestion of edges in the network and (iii) to achieve the new throughput flow in the network with a minimum generation shedding.

II. An new throughput flow with minimum generation shedding after failure or congestion of several edges

A flow network can always be modelled by a directed graph *G* = (*V*,*E*) consisting of a set of nodes *V* and a set of edges. The network flow is said to be *feasible*, if the next two conditions are fulfilled. At each node v_i, different from a source or a sink, the flow conservation law holds (equation 1).

$$\sum_{k\in\delta+} f(k,v_i) = \sum_{m\in\delta-} f(v_i,m) \tag{1}$$

The flow conservation law simply states that the sum of the edge flows $\sum_{k\in\delta+} f(k,v_i)$ entering node v_i is equal to the sum of the edge flows $\sum_{m\in\delta-} f(v_i,m)$ leaving the node. The second condition involves the capacity constraints imposed by the edges:

$$f(i,j) \le c(i,j) \tag{2}$$

The capacity constraint condition simply states that the flow $f(i,j)$ through edge (*i,j*) cannot exceed the capacity $c(i,j)$ of the edge – the maximum flow that the edge permits.

The performance of an algorithm for re-optimising the network flows after failure or congestion of edges can be increased significantly in comparison with classical algorithms, which always start the maximisation of the throughput flow from a network with empty edges.

This can be done by using the important circumstance that after a disturbance of the flow along a particular edge, *the rest of the edges are not empty but are already fully or partially saturated with flow*. An algorithm which starts the re-optimisation from a network with edges which are fully or partially saturated with flow avoids the augmentation of all feasible paths and has a clear advantage to an algorithm which starts the optimisation from a network with empty edges.

In the case of a single edge failure, a method for re-optimising the flows has already been outlined in [13]. However, the critical question related to eliminating the overloading and congestion along branches of a flow network, with minimum generation shedding, was not discussed in Ref.[13]. Finally, Ref.[13] treats only the special case where the throughput flow in the network is the maximum possible throughput flow. In communication networks, electrical networks and transportation networks however, the throughput flow is rarely the maximum possible throughput flow. For real networks the central issue is to re-optimise the network flow in such a way that the contingency event causes a minimum flow generation shedding. In this sense, the notion 'optimum throughput flow' used here stands for the restored new feasible throughput flow in the network *attained with the smallest decrease of flow generation (generation shedding)*.

As we shall see later, the deliberate choking of flows along overloaded edges, leads to a generation of momentary excess and deficit flow and provides a very efficient mechanism of relieving overloaded branches of the network. In this respect, it is important to state and prove a result related to the magnitude of the optimal throughput flow after the flows along several edges have been constrained (choked) to a particular level.

After the choking of the flow along an edge (e.g. edge (e_i, d_i)), from the initial level $f(e_i,d_i)$ to the level $f'(e_i,d_i)$ ($0 \le f'(e_i,d_i) < f(e_i,d_i)$), the network flow is disturbed at nodes e_i and d_i to which the edge (e_i,d_i) has been connected. The flow along edge (e_i,d_i) may be fully choked because of edge failure ($f'(e_i,d_i)=0$) or partially choked ($0 < f'(e_i,d_i) < f(e_i,d_i)$.

If edge (e_i,d_i) is not empty, after the choking of its flow, a momentary excess flow appears at one of the nodes (node e_i) equal to the amount of choked flow $f(e_i,d_i) - f'(e_i,d_i)$ along the edge. In other words, the sum of the edge flows going into node e_i is greater than the sum of the edge flows leaving the node. This difference will be referred to as '*momentary excess flow*' mef_i :

$$mef_i = \sum_{k\in\delta+} f(k,e_i) - \sum_{m\in\delta-} f(e_i,m) > 0 \tag{3}$$

and node e_i will be referred to as *momentary excess node*.

Alternatively, after choking the flow along edge (e_i , d_i), momentary deficit flow will be created at node d_i , equal to the amount of choked flow $f(e_i,d_i) - f'(e_i,d_i)$ along the edge (e_i,d_i). The sum of the edge flows going into node d_i is smaller than the sum of the edge flows leaving node d_i . The difference between the sum of the ingoing flows and the sum of the outgoing flows is negative, and will be referred to as *momentary deficit flow* mdf_i :

$$mdf_i = \sum_{k\in\delta+} f(k,d_i) - \sum_{m\in\delta-} f(d_i,m) < 0 \tag{4}$$

Accordingly, node d_i will be referred to as *momentary deficit node*.

After choking the flow along *n* edges, M_1 momentary excess nodes e_i , with momentary excess flows mef_i (*i*=1,..., M_1) and M_2 momentary deficit nodes d_j with momentary deficit flows mdf_j (*j*=1,..., M_2), will be created. In general, $M_1 \ne M_2$ because a momentary excess flow at a particular node from choking the flow along a particular edge may have been compensated or turned into a deficit flow by the momentary deficit flow from choking the flow along another edge incident to the node. The quantity $\sum_{i=1}^{M_1} mef_i > 0$ is the total amount of momentary excess flow at the excess nodes, after choking the flows along the M_1 edges.

The state of any node (neutral, momentary excess node or momentary deficit node) is determined by the algebraic sum of the momentary excess and deficit flows created at that node. However, the sum of the momentary excess flows at all excess nodes is always equal to the sum of the momentary deficit flows at all deficit nodes: $\sum_{i=1}^{M_1} mef_i = \sum_{j=1}^{M_2} mdf_i$.

These concepts will be illustrated by the network in Fig.2a, where the first number on each edge is the edge capacity and the second number is the actual magnitude of the flow through the edge. In the network from Fig.2a, edge (10,14) has been overloaded by 30 units of flow and the flow magnitude needs to be reduced from 60 units to 30 units. Temporary overloading of edges for example is common for electrical power networks. The flow along a power line could exceed for a short time its nominal capacity but if the flow through the line is not returned within the throughput capacity of the line, the result is a failure of the overloaded line.

After choking the flow along edge (10,14) in Fig.2a, from the initial level $f(10,14) = 60$ to the new level $f'(10,14) = 30$, the network flow is disturbed locally at nodes 10 and 14 to which the edge (10,14) is connected (Fig.2a). Edge (10,14) is not empty after the choking of its flow. After choking the edge, it can be thought as if a momentary excess flow appears at one of the nodes (node 10), equal to the amount of choked flow $f(10,14) - f'(10,14) = 30$ through the edge. In other words, the sum of the edge flows going into node 10 becomes greater than the sum of the edge flows leaving the node. This difference is the excess flow mef_{10} and node 10 becomes a momentary excess node.

Alternatively, after choking the flow along edge (10,14), a momentary deficit flow will be created at node 14, equal to the amount of choked flow $f(10,14) - f'(10,14) = 30$ of the edge. The sum of the edge flows going into node 14 is smaller than the sum of the edge flows leaving node 14. The difference between the sum of the ingoing flows and the sum of the outgoing flows is negative, and will be referred to as a *momentary deficit flow,* mdf_{14} . Accordingly, node d_{14} becomes *a momentary deficit node*.

Now, a new start node s_d with unlimited capacity can be introduced, connected with the momentary excess node 10 (Figure 2a) through a fully saturated edge (10, s_d) with capacity equal to the momentary excess flow at the momentary excess node 10. A new end node t_d is also introduced, connected with the momentary deficit node 14. The connecting edge (t_d ,14) is a fully saturated edge with capacity equal to the momentary deficit flow at the momentary deficit node. The result is the network in Fig.2a, which will be referred to as *dual network*.

As a result of introducing the fully saturated auxiliary edges, the momentary excess and deficit nodes disappear and feasible flow is established in all parts of the dual network. Suppose that a feasible throughput flow *Q* has been established in the original network before choking the flow along edge (10,14).

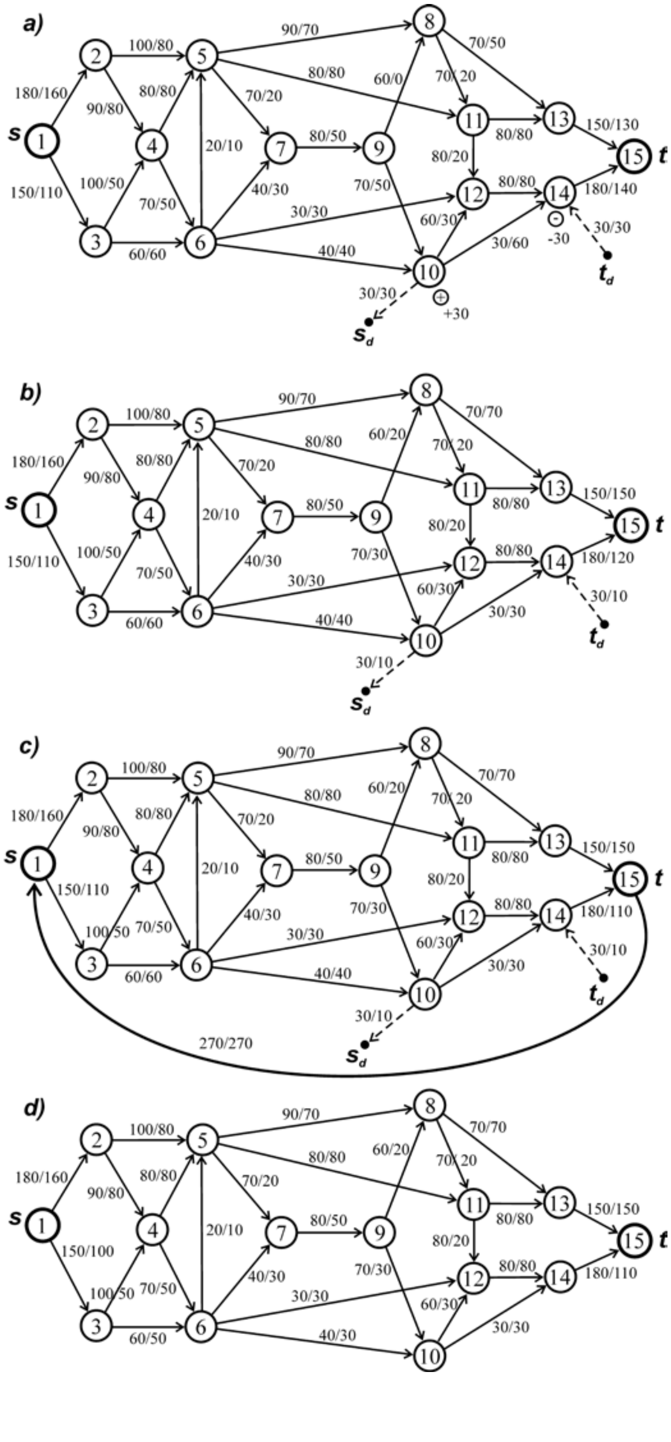

Figure 2. The two basic stages of the fast decongestion algorithm.

The algorithm for determining the edge flows corresponding to the new optimal throughput flow in the network, characterised by a minimum generation shedding, is based on the next theorem:

Theorem 1.
a) *The edge flows corresponding to the optimal throughput flow after choking the flow along several edges can be obtained by a two-stage procedure which consists of: (i) augmenting with flow the dual network until no more augmentable paths can be found, followed by (ii) augmenting with flow the dual circulation network until no more augmentable paths can be found.*
b) The optimal throughput flow in any flow network after choking the flow along several edges is equal to the throughput flow in the network before choking the edge flows, minus the total amount of momentary excess flow at the excess nodes, plus the maximum throughput flow in the dual network.

The path augmentation is a Ford-Fulkerson type augmentation [4]. In this sense, a path is augmentable if all edges along the path are characterised by a nonzero residual capacity, i.e. if no forward edge from the path is fully saturated and no backward edge is empty. For the network in Fig.2a, the path (s_d ,10,9,8,13,15,14, t_d) is augmentable and can be augmented with 20 units of flow. The result is the network in Fig.2b. During the augmentation of paths in the dual network the originalsource and sink are treated as ordinary nodes. Because there are no more augmentable paths in the dual network, and there is still 10 units momentary excess flow at node 10 (Fig.2b), a circulation edge (*t*, *s*) is introduced with capacity equal to the flow (270) towards the sink (Fig.2c). The network obtained from the dual network, where the original sink *t* has been connected with the original source *s* through a circulation edge (*t*,*s*), will be referred to as *dual circulation network* (Fig.2c).

Augmenting path (s_d ,10,6,3,1,15,14, t_d) with 10 units of flow, removes the momentary excess and deficit flow of 10 units at nodes 10 and 14 and the flow in the network becomes feasible (Fig.2d). The flow is also optimal because the new feasible flow has been achieved with a minimum decrease of the flow production from the source *s* (10 units only). Note that a feasible flow could have been achieved simply by draining 30 units of flow from the network by augmenting the path (s_d ,10,6,3,1,15,14, t_d) in Fig.2a with 30 flow units in the dual circulation network (with the circulation edge (*t*,*s*) included). This would mean however that the generation from the source *s* would have been decreased by 30 units (not by 10 units) hence, the throughput flow will not be optimal. Note that *augmenting the dual circulation network, before the maximum possible augmentation in the dual network has been done, leads to suboptimal solutions*. Executing the two stages of the algorithm in the correct sequence *is absolutely essential to optimizing the network flow*. The described algorithm works equally well if choking of the flow has been done along several edges.

III. Proof of Theorem 1 and an algorithm

Theorem 1 can be proved by using the following two lemmas, whose detailed proofs are given in the Appendix.

Lemma 1. *If there are no augmentable* $s_d - t_d$ *paths in the dual network, the momentary excess flow at the excess nodes and the momentary deficit flow at the deficit nodes can always be reduced by augmenting a* $s_d - t_d$ *path in the dual circulation network, where the circulation edge (t,s) belongs to the augmented path.*

Lemma 2. *If no augmentable* $s_d - t_d$ *path exists in the dual network, augmentation of a* $s_d - t_d$ *path in the dual circulation network results in the absence of augmentable s-t paths in the original network.*

Proof of Theorem 1. Suppose that by augmenting $s_d - t_d$ paths in the dual network, the entire momentary excess flow $q_d^{\max}$ has been purged from all backward edges (s_d , e_i), (*i*=1,..., M_1) connecting the excess nodes e_i with the new start node s_d . Because the total momentary excess flow at the excess nodes e_i (*i*=1,..., M_1) is always equal to the total momentary deficit flow at the deficit nodes, d_i (*i*=1,..., M_2), eliminating the momentary excess flow will also eliminate the momentary deficit flows. In this case, the resultant throughput flow in the original network will be feasible and, at the same time, it will be equal to the initial throughput flow Q . This flow will also be optimal because the path augmentations have been done in the dual network only and no reduction of flow generation from the original source has been made.
Now, consider the second possibility: (ii) the maximum purged momentary excess flow $q_d^{\max}$ from edges (s_d , e_i), *i*=1,..., M_1 , in the dual network is smaller than the sum of the momentary excess flow from the excess nodes ($q_d^{\max} < \sum_{i=1}^{M_1} mef_i$).

The augmentation of $s_d - t_d$ paths in the dual network terminates when no more augmentable $s_d - t_d$ paths can be found. Let $q_d^{\max}$ be the maximum throughput flow, with which the dual network has been augmented at the end of the first stage. The remaining excess flow q_{rem} to be purged from the edges connecting the excess nodes with the new start node

s_d is given by $q_{rem} = \sum_{i=1}^{M_1} mef_i - q_d^{\max}$. According to Lemma 1, there is always an augmentable $s_d - t_d$ path in the dual circulation network, which reduces the remaining excess flow q_{rem}. Because an augmentable $s_d - t_d$ path in the dual circulation network always includes the circulation edge (*t*,*s*), each augmentation of an $s_d - t_d$ path subtracts equal amount of flow from edges going out of the source *s* and from edges going into the sink *t*. According to Lemma 2, after each augmentation of a $s_d - t_d$ path in the dual circulation network, there will be no augmentable *s*-*t* path in the original network. The augmentation of $s_d - t_d$ paths in the dual circulation network is repeated, until the entire remaining excess flow q_{rem} from the first stage disappears. Consequently, the new throughput flow Q' after choking the flows along several edges, will be

$$Q' = Q - \sum_{i=1}^{M_1} mef_i + q_d^{\max}, \tag{5}$$

where Q is the throughput flow in the original network before choking the edge flows. This new throughput flow is achieved with a minimum generation shedding. Simultaneously, according to Lemma 2, there will be no augmentable *s*-*t* path in the network. According to the Ford-Fulkerson theorem [4], the absence of augmentable *s*-*t* paths in the original network means that the throughput flow in the original network is indeed the maximum possible. Consequently, the flow Q' given by equation (5) is the new optimal flow in the network, achieved with a minimal generation shedding. This proves the theorem.□

A. An algorithm for re-optimising the throughput flow after choking the flows along multiple edges

The algorithm for determining the edge flows corresponding to the maximum throughput flow in the network has essentially been formulated by Theorem 1. It consists of the following two basic steps:

1. ***Augment $s_d - t_d$ paths in the dual network until no more augmentable paths can be found***.

2. ***If there is still remaining flow in the backward edges connecting the new start node s_d, augment $s_d - t_d$ paths in the dual circulation network until the remaining flow is removed.***

Theorem 1 establishes a very important link between the optimal throughput flow Q in a flow network after constraining the flow capacity of an edge and the maximum throughput flow $q_d^{\max}$ in its dual network. Theorem 1 replaces the task of determining the optimal throughput flow in the original network, with the task of determining the maximum throughput flow in the dual network. In the case of choking the flow of a single edge (for example after an edge failure), there are only two unbalanced nodes. In this case, determining the maximum throughput flow in the dual network is significantly easier than determining the maximum throughput flow in a network with empty edges. The reason for this important trade-off is that the dual network *is already saturated with flow*.

Because augmenting a single path or few paths is a procedure of worst-case complexity *O*(*m*) (where *m* is the number of edges in the network), in the cases where the momentary excess flow is eliminated after augmenting a single $s_d - t_d$ path or few $s_d - t_d$ paths, the running time of the proposed re-optimisation algorithm will be proportional to the number of edges *m* in the network. Because the momentary excess and deficit nodes, resulting from choking the flow along an edge, are adjacent nodes, in many cases, the momentary excess and deficit flow are eliminated after a single augmentation along a single path or after augmenting few paths. The proposed algorithm has a significantly smaller average running time compared to classical algorithms starting from a network with empty edges. As it will be demonstrated later, in some cases, the algorithm re-optimises the flow in time, independent of the size of the network.

Numerous experiments with different network topologies indicated that apparently, only in extreme, deliberately designed cases, the running time of the proposed re-optimisation algorithm approaches the running time of the classical Edmonds and Karp algorithm [5]. Experiments with networks of different size and topology indicated that the average running time of the re-optimisation algorithm appears to be increasing approximately linearly with increasing the number of edges of the network.

The algorithm can also be used for a real-time control of flow networks, upon contingency events. Its high computational speed makes it appropriate for decongesting overloaded edges of networks in real time. This is particularly important for telecommunication networks, which need to be controlled within the range of milliseconds, upon congestion or failure of a transmission link. The ever increasing demand on the existing computer networks and communication networks requires a faster congestion management if the number of dropped calls, the delays caused by congestion are to be minimised. The algorithm has also important applications for real-time control of active power networks, production networks (e.g. oil and gas production networks), manufacturing networks, and supply logistics networks.

Another significant advantage of the proposed algorithm is that upon choking the flows along edges, the edge flows can be simultaneously re-optimised by several independent agents attached to the edges of the network. In this case, achieving a global maximum throughput flow from sources to destinations is guaranteed, *as long as no imbalanced nodes remain in the network after the re-optimisation from the independent agents*.

This point will be illustrated with the network in Fig 3a in which edge (6,5) and edge (10,12) have failed. The result is a momentary excess flow of 10 units at node 6 and 30 units at node 10, and the same quantities of momentary deficit flows at nodes 5 and 12, respectively (Fig.3a). Suppose that two independent agents one attached to edge (6,5) and one attached to edge (10,12) are independently re-optimising the throughput flow. The agent attached to edge (6,5) is trying to cancel the momentary excess and deficit flow of 10 units at nodes 6 and 5 by sending 10 units of momentary excess flow from node 6 towards node 5. This can be done by augmenting the path (6,7,5) by 10 units of flow. The agent attached to edge (10,12) is re-optimising the flow by trying to cancel the 30 units of momentary excess flow at node 10 with the 30 units of momentary deficit flow at node 12. This can be done by augmenting the path (10,9,8,11,12) with 30 units of flow. The result is the network in Fig.3b, where the momentary excess and deficit flows no longer exist. The network flow has been re-optimised by the independent actions of the agents.

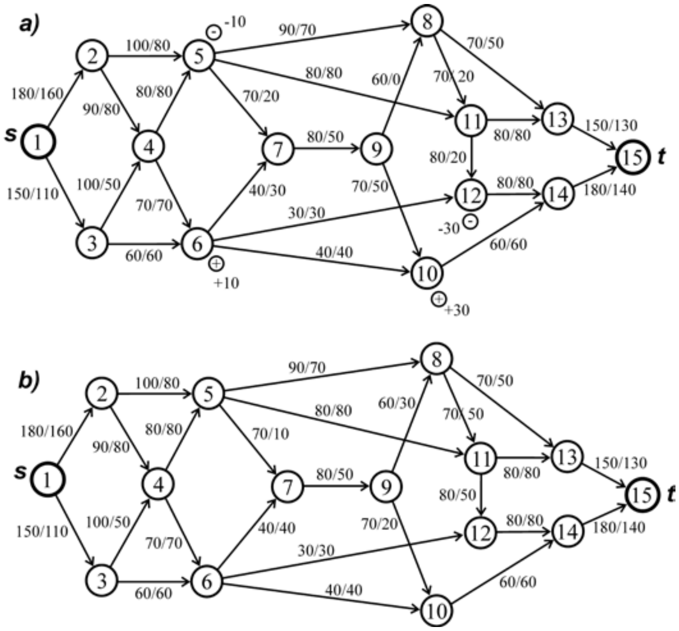

Figure 3. The network flow can be simultaneously re-optimised by the independent actions of agents attached to the separate edges of the network.

IV. THE THROUGHPUT FLOW CONSTRAINT

Suppose that in a particular flow network, a number of sources supply flow to a number of consumers. The total maximum amount of flow which all sources can supply is Q_g. Suppose, for the sake of simplicity, that the total amount of throughput flow transmitted to the consumers, in the absence of edge failures, is also Q_g. Because of capacity degradation, the new maximum possible throughput flow Q' transmitted through the network after edge failures is usually smaller than the maximum possible generated quantity Q_g. The difference $Q_g - Q'$ will be referred to as *throughput flow constraint in the original network*.

Now, let us introduce a similar concept related to the dual network. The amount of momentary excess flow in the dual network is $\sum_{i=1}^{M_1} mef_i$ and the maximum possible throughput flow from the new start node s_d to the new end node t_d is $q_d^{\max}$. The difference $\sum_{i=1}^{M_1} mef_i - q_d^{\max}$ will be referred to as *throughput flow constraint in the dual network*.

Note that a network with multiple distributed sources can always be transformed to a network with a single source by linking the separate sources with a super-source, through edges whose capacities are equal to the amount of generated flow from the separate sources.

Thus, the network in Fig.4a features three sources of flow *s*1, *s*2 and *s*3, whose flow generation powers are: *s*1 = 15 GB/h, *s*2 = 20 GB/h and *s*3 = 30 GB/h. The possible transmission paths and their throughput capacities have been specified as labels on the edges.

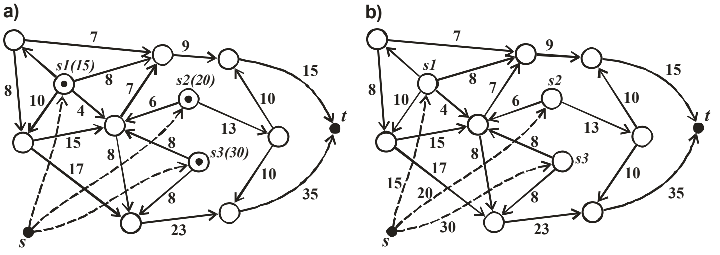

Figure 4. Transforming a network with multiple sources of generation into a single-source network.

The network in Fig.4a, with multiple sources of generation can be transformed into an *s*-*t* network (with a single source) by introducing a super-source *s* as it is shown in Fig.4b. The flow capacities of the edges (*s*,*s*1), (*s*,*s*2) and (*s*,*s*3) connecting the super-source *s* with the sources of generation (nodes *s*1, *s*2 and *s*3) are equal to the flow generation power of the separate sources. As a result, the multiple sources disappear and throughput edges appear instead (Fig.4b).
It can now be shown that the following theorem holds:

Theorem 2. (Throughput flow constraint theorem).
The throughput flow constraint in any network after restricting the flows along some of the edges is always equal to the throughput flow constraint in its dual network.

Proof.
Suppose that the throughput flow Q is equal to the generated by the sources flow ($Q = Q_g$).

According to the earlier discussion, a network with multiple sources which supplies Q_g total amount of throughput flow can always be reduced to a single source network which supplies the maximum of Q_g throughput flow. Let the new maximum throughput flow after choking the flows along some of the edges be Q'. According to Theorem 1, equation (5) holds:

$$Q' = Q_g - \sum_{i=1}^{M_1} mef_i + q_d^{\max} \qquad (6)$$

After rearranging the terms, equation (6) becomes

$$Q_g - Q' = \sum_{i=1}^{M_1} mef_i \; - q_d^{\max}\,, \qquad (7)$$

which proves Theorem 2.□

For a single edge flow constraint in a saturated with flow network, it is easier to determine the throughput flow constraint in the dual network rather than the throughput flow constraint in the original network. This explains the efficiency of the optimisation algorithm working with the dual network.

Using these ideas, a similar invariant can also be formulated for static flow networks. Consider a static flow network with edges fully saturated with flow. Similar to networks with disturbed flows, excess and deficit nodes can also be defined. If the sum of capacities of all edges going into a node *e* (different from the source and the sink) is greater than the sum of capacities of all outgoing edges, the node is said to be an *excess node*. The amount of momentary excess flow ef at an excess node *e* is given by:

$$ef = \sum_{i \in \delta+} c(i,e) - \sum_{j \in \delta-} c(e,j) > 0\,. \qquad (8)$$

Conversely, if the sum of capacities of all edges going into a node *d* (different from the source and the sink) is smaller than the sum of capacities of all outgoing edges, the node is said to be a *deficit node*. The amount of momentary deficit flow *df* in the deficit node is:

$$df = \sum_{i \in \delta+} c(i,d) - \sum_{j \in \delta-} c(d,j) < 0\,. \qquad (9)$$

Finally, if the sum of capacities of all edges going into a particular node is equal to the sum of capacities of all edges going out of the node, the node is referred to as *balanced node*. The amount of excess/deficit flow at a balanced node is zero. Unlike the momentary excess and deficit nodes in networks with disturbed flows, the excess and deficit flows in static networks with fully saturated edges are real.

Consider now the static network in Fig.5a, whose edges are fully saturated with flow. As a result, excess and deficit nodes appear in the network: the excess node '4' with 30 units excess flow and the deficit node '3' with 20 units deficit flow. These are imbalanced nodes and the network flow is not feasible. The purpose is to make it feasible and maximise the throughput flow, by appropriate flow redistribution between excess and deficit nodes and by draining flow from the network.

Now let us connect all excess nodes (for the network in Fig.5a the only excess node is node 4) with the sink *t*, by fully saturated *ghost edges* directed to the sink, with flow capacities equal to the amount of excess at the excess nodes. Simultaneously, let us also connect the source *s* with all deficit nodes (for the network in Fig.5a the only deficit node is node 3) by fully saturated ghost edges directed towards the deficit nodes, with flow capacities equal to the deficit flows at the deficit nodes. This operation transforms the original network into a network where all internal nodes are balanced. The ghost edges have been drawn by dashed lines (Fig.5b).

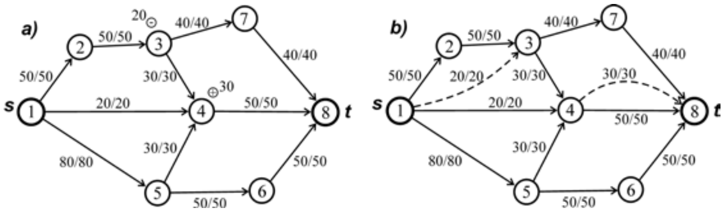

Figure 5. By using ghost edges (the dashed lines), the excess and deficit nodes in any static flow network with fully saturated edges can be transformed into a network where all internal nodes are balanced.

After the introduction of ghost edges, the network flow is feasible everywhere (Fig.5b). In other words, the flow conservation at the nodes and the capacity constraints of the edges are honored in the network.

Now, suppose that the ghost edges in the network from Fig.5b 'fail' simultaneously. Because, by saturating all edges with flow, the throughput flow in the network has essentially been maximised, the problem is now reduced to the problem treated earlier - in a network with feasible flow several edges (the ghost edges) are choked. Because the conditions of Theorem 1 are fulfilled, it can be applied for determining the new maximum throughput flow in the network, after the 'failure' of all ghost edges. Theorem 2 will also be valid. If a dual network is now constructed, the following theorem will hold:

Theorem 3. (Throughput flow constraint theorem for static flow networks)
The throughput flow constraint in any static flow network is always equal to the throughput flow constraint in its dual network.

$$\sum_i c(s,i) + \sum_k c_g(s,k) - Q_{\max} = \sum_i ef_i - q_d^{\max} \qquad (10)$$

In equation (10), $\sum_i c(s,i)$ is the sum of capacities of all real edges coming out of the source *s*; $\sum_k c_g(s,k)$ is the sum of capacities of all ghost edges coming out of the source *s*; $\sum_i ef_i$ is the sum of the excess flow formed by the 'failure' of all ghost edges in the network; $Q_{\max}$ is the maximum throughput flow in the original network (without any ghost edges) and $q_d^{\max}$ is the maximum throughput flow in the dual network (with failed ghost edges). Equation (10) can also be rearranged as

$$Q_{\max} = \sum_i c(s,i) + \sum_k c_g(s,k) - \sum_i ef_i + q_d^{\max} \ , \quad (11)$$

which permits computing the maximum throughput flow $Q_{\max}$ in static networks immediately, after obtaining the maximum throughput flow $q_d^{\max}$ in the corresponding dual network. If there are no augmentable $s_d - t_d$ paths in the dual network, $q_d^{\max} = 0$ and the maximum throughput flow in the original network can be established instantly:

$$Q_{\max} = \sum_i c(s,i) + \sum_k c_g(s,k) - \sum_i ef_i$$

Because the absence of augmentable paths is established in $O(m)$ time, where *m* is the number of edges in the network, the maximum throughput flow in this case will be established in $O(m)$ time.

As a result, *in some cases, the invariant throughput flow constraint theorem provides the opportunity to determine the maximum throughput flow in a static network by inspecting the network only locally, without considering the rest of the network.*

For the network in Fig.5b, the maximum throughput flow in the dual network, after the failure of the ghost edges, is $q_d^{\max} = 20$ units. Because $\sum_i c(s,i) = 150$, $\sum_k c_g(s,k) = 20$ and $\sum_i ef_i = 50$, from equation (11) the maximum throughput flow becomes

$$Q_{\max} = 150 + 20 - 50 + 20 = 140$$

V. Application of the Dual Network Theorem for Decongesting Overloaded Transmission Paths in Telecommunication Networks

Consider the telecommunication network in Fig.6a, where data is transmitted from node 1 to node 7 and where sections (1,2) and (2,5) have been congested and causing delays. To relieve congestion, the data flow along these edges should, for example, be reduced by 5 GB/h. The first step is to choke the flows along the congested transmission links by limiting the capacities of the corresponding edges to the desired amount of flow – to 20 GB/h for edge (1,2) and to 15 GB/h for edge (2,5). As a result, momentary excess and deficit flow of 5 GB/h appears at the beginning and at the end of the edges whose flow has been choked. Node 2 however, remains a balanced node, because the momentary excess data flow of 5 GB/h from choking the flow along edge (2,5) has been cancelled canceled by the momentary deficit data flow of 5 GB/h from choking the flow along edge (1,2).

Additional start node s_d is then added, connected to the excess node '1' by the backward, fully saturated edge (s_d,1), with capacity 5 GB/h. Similarly, additional end node t_d is also added, connected to the deficit node 5, by the backward, fully saturated edge (t_d,5), with capacity 5 GB/h (Fig.6b). The algorithm for redistributing the flow in the resultant network, then proceeds as follows. The shortest augmentable path (s_d,1,3,6,5,t_d) is augmented with 5 GB/h, which results in the network flow from Fig.6c. The momentary excess and deficit flows disappear, the network flow is feasible; the throughput flow (35 GB/h) is equal to the throughput flow before the redistribution. The transmission links (1,2) and (2,5) however, are no longer congested.

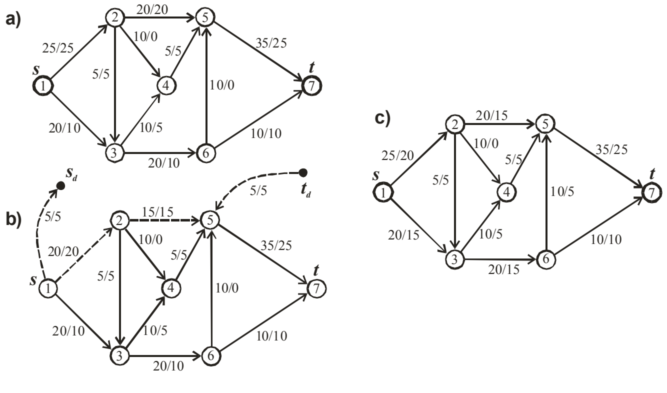

Figure 6. Decongestion of edges (1,2) and (2,5).

Note, that for a long directed flow path, where the choked flow along each edge is the same, only two imbalanced nodes will appear after the choking. The start node of the directed path will appear as an excess node and the end node will appear as a deficit node. At any other node *i* along the directed path, the momentary deficit flow from choking edge (*i-1*,*i*) will be cancelled by the momentary excess flow from choking edge (*i*,*i+1*). The result will be a neutral node *i*.

In this example, the optimisation has been achieved without decreasing the generation of flow from the source. In some cases however, correcting the generated flow is necessary to avoid exceeding the permitted throughput capacities of the transmission links as in the example from Fig.2. Exceeding the permitted transmission capacities of transmission links occurs for example in overloaded power lines.

VI. Advantage of the Proposed Method to Classical Optimisation Methods

To demonstrate the advantage of the proposed method, consider an illustration example, featuring the telecommunication network in Fig.7, where data flow of magnitude 280 GB/h is transmitted from a source *s* to a destination *t*. The capacities of the transmission lines ‘c’ and the actual data flows ‘f’ along them are shown as edge labels ‘c/f’.

Suppose that the transmission link (6,5), carrying 30 GB/h, has actually been overloaded and its flow needs to be reduced from 30 GB/h to 20GB/h. According to the earlier discussion, the congestion can be eliminated by choking the flow of edge (6,5) from 30 GB/h to 20 GB/h, by reducing the capacity of the edge from 30GB/h to 20 GB/h. The throughput flow in the network after reducing the capacity of edge (6,5) (Fig.7), was re-optimised by using the classical *Edmonds and Karp* algorithm [5], which starts from a network with empty edges. For the network in Fig.7, one million runs of the Edmonds and Karp algorithm, on a computer with processor *Intel(R) Core(TM) 2 Duo CPU T9900 @ 3.06 GHz*, took 11.3 seconds.

The proposed re-optimisation algorithm from section 2 was also run on the network in Fig.7, after introducing a start node s_d and end node t_d . Because the flow through edge (6,5) before its choking was 30 GB/h, after the choking, 10 GB/h momentary excess flow will appear at node 6 and 10 GB/h momentary deficit flow will appear at node 5. Following the re-optimisation algorithm, the throughput flow in the dual network was maximised by augmenting the shortest paths starting at the new source s_d and ending at the new sink t_d . One million runs of the re-optimisation algorithm were executed for only 0.98 seconds, more than an order of magnitude faster than the running time of the classical Edmonds and Karp algorithm working on a network with empty edges.

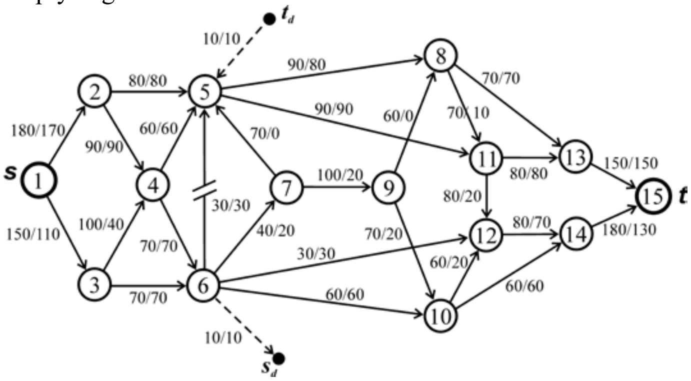

Figure 7. A network, demonstrating the performance of the fast re-optimisation algorithm.

The re-optimisation algorithm augments essentially the shortest path ($s_d,6,7,5,t_d$) with 10 GB/h. After the augmentation and the removal of the connecting edges, the momentary deficit and excess flow at nodes 6 and 5 disappear and feasible edge flows are set up everywhere in the network. The maximum throughput flow in the network after the re-optimisation is still 280 GB/h. Even if the network is increased significantly in size, by adding many nodes between node 7 and the sink *t* for example, the running time of the re-optimisation algorithm will not increase and will remain the same!

A fast re-optimisation of the network flow after an edge overloading or congestion is critically important for large flow networks including thousands of edges and nodes (e.g. telecommunication networks and computer networks). Restoring quickly the throughput flow minimises the flow disruption and optimises the network performance in real time. Another study conducted on a computer with a processor Intel (R) Core (TM) 2 Duo CPU T9900 @ 3.06 GHz, has indicated that after a component failure in a network with *m*=10000 edges, an augmentation algorithm with average running time proportional to m^2, needs many seconds to maximise the throughput flow.

Indeed, for a network with 10000 edges, the average running time of such algorithm is proportional to (10000 x 10000) x Δt, where Δt =2.5x10^{-6} s is the average time expended on a single edge. This equates to an average running time of 250 seconds, which is unacceptable for re-optimising the flows in real-time. If the proposed re-optimisation algorithm with approximately linear average running time in the size *m* of the network is used, in a network with 10000 nodes, the average running time would be proportional to 10000 x Δt s, which means a running time of about 25 milliseconds!

The proposed re-optimisation algorithm is also very useful in cases where only the maximum throughput flow is needed but not the values of the edge flows. This application is relevant to designing fast discrete-event simulators for determining the throughput availability of flow networks by calculating the throughput flow hundreds of thousands times, upon failures of various edges.

There are many cases where a failure of an edge causes the entire flow through the edge to be lost. For example, the entire class of networks with tree topology possesses this feature. With respect to these cases, the following theorem can be formulated:

Theorem 4. *If an edge failure causes a loss of the entire flow through the edge, the new maximum throughput flow can be computed in* $O(m)$ *time.*

Proof. The excess flow associated with the failure of the edge (*i,j*) is equal to the flow $f(i,j)$ through the edge. According to equation (5),

$$Q' = Q - f(i,j) + q_d^{\max} \qquad (12)$$

holds for the new maximum throughput flow Q' . Simultaneously, if the failure of an edge causes a loss of the entire flow through the edge,

$$Q' = Q - f(i,j) \qquad (13)$$

From (12) and (13), it follows that $q_d^{\max} = 0$. In other words, there is no augmentable path in the dual network. In a connected network, discovering that there is no augmentable path is an operation with worst-case running time $O(m)$, where *m* is the number of edges in the network. Computing the expression $Q' = Q - f(i,j)$ after discovering that $q_d^{\max} = 0$ has running time $O(1)$. Consequently, the magnitude of the new maximum flow can be computed in $O(m)$ time. □

As an illustrating example, consider the communication network in Fig.8. After the failure of edge (3,5), the new maximum throughput flow in the network has been obtained by the proposed re-optimisation algorithm. Because, there are no augmentable $s_d - t_d$ paths in the dual network, $q_d^{\max} = 0$ and, according to equation (5), the new optimal throughput flow after the failure of edge (3,5) is simply $Q' = Q - f(3,5) = 180 - 30 = 150$ GB/h (Fig.8). The re-optimisation algorithm discovers that no augmentable $s_d - t_d$ path exist in the dual network in O(m) time. One million runs of the Edmonds and Karp shortest-path algorithm for determining the maximum throughput flow (starting from a network with empty edges), took 9.3 seconds. In contrast, one million runs of the proposed re-optimisation algorithm took only 0.53 seconds, 17.5 times faster!

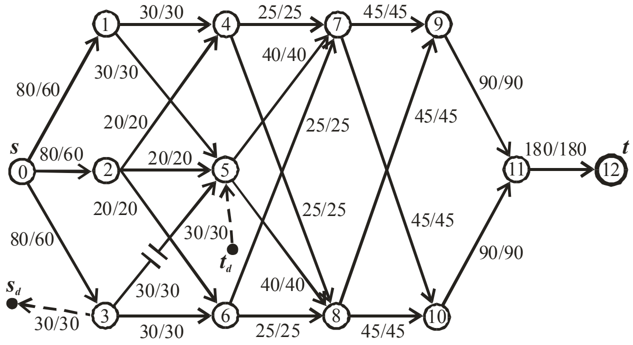

Figure 8. The maximum flow of 150 after the failure of edge (3,5) is obtained in $O(m)$ time, after discovering that no augmentable flow paths exist.

While the running time of classical re-optimisation methods based on maximising the throughput flow always increases with increasing the size of the network, as can be seen from the example in Fig.8, *the running time of the proposed re-optimisation algorithm does not necessarily increase with increasing the size of the network.* In the example from Fig.8, the algorithm recalculated the new maximum network flow by inspecting the network locally, in the vicinity of the failed edge, without actually considering the rest of the network. The throughput flow constraint theorem permits the recalculation of the new optimum throughput flow to be done by a local inspection of the network. This feature of the proposed method is an important contributing factor determining its high computational speed.

The re-optimisation after a failure of unreliable node can easily be reduced to the already considered case related to unreliable edges and perfectly reliable nodes. Each flow network with unreliable nodes and unreliable edges can always be reduced to a network with perfectly reliable nodes and unreliable edges. In order to do this, each node '*i*' of the network (Fig.9a), characterised by a failure rate (expected number of failures per unit time) $\lambda_i > 0$, can be presented by a pair of perfectly reliable nodes *i1*, *i2* ($\lambda_{i1} = 0, \lambda_{i2} = 0$) connected with an unreliable edge (*i1*,*i2*), characterised by a throughput flow capacity equal to the sum of the flow capacities of the edges entering node *i* and failure rate λ_i, equal to the failure rate of the unreliable node *i*. For each unreliable node *i*, the first perfect node *i1* collects flow from all edges entering node *i*. The second perfect node *i2* is incident to all edges leaving the unreliable node *i*.

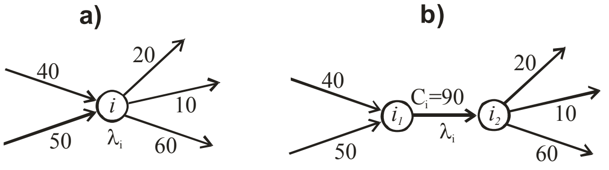

Figure 9. Representing an unreliable node '*i*' by splitting it in two perfectly reliable nodes i_1 and i_2, connected with unreliable edge (i_1, i_2).

Now, the failure of the unreliable node *i* is treated as a failure of the unreliable replacing edge (i_1, i_2). The advantage of this approach is that after the failure of the unreliable node *i*, only one excess node (node i_1) and one deficit node (node i_2) appear in the network.

The re-optimisation method proposed in this paper has the potential to deliver a significant improvement in the real-time control of real networks with disturbed flows, for example for telecommunication networks.

Telecommunication networks are an important example of networks with disturbed flows. The increased need for exchanging data and information is increasing the need for efficient telecommunication networks. The increased network efficiency leads to increased access, and hence an increase in network traffic. Maintaining a high quality of service during the transfer of large media files requires optimal management of the available bandwidth capacity. Finally, the telecommunication network is the backbone of the smart grid with active control of the power flows – the power network of the future [14]. In order to fulfil its function, the smart grid requires a supporting telecommunication network needed to accommodate and control the large volumes of data generated from distributed sensors, meters, generators and loads and the data flows channelled to the hardware control devices.

To improve the automation of network flows, recently, the autonomous agent-based type of control has been gaining popularity [15-18]. However, despite the intensive recent research on multi-agent systems control, currently there is a lack of algorithms for optimal flow management, which guarantees that the independent interventions of the autonomous agents upon overloading and congestion will eventually lead to a minimum generation shedding from the sources and to an optimum utilization of the residual capacity of the network. In the case of component failures, the mitigating actions from the autonomous agents are reduced to sending signals to shed load from the sources of flow. This approach requires special control systems in place, each monitoring for a different scenario and requiring a different control [15]. This approach not only leads to very complex control actions that are not at all straightforward and transparent. As the example from Fig.1 demonstrates, this approach may result in unnecessary reduction of the generated flow. As a result, this approach provides no guarantee that the optimal flows will be set up, which minimise the generation shedding and maximise the throughput flow delivered from sources to destinations.

The approach presented in this paper can be used with success for re-optimising the flow after congestion and failure by the actions of independent agents.

CONCLUSIONS

1. A new result has been stated and proved: "the throughput flow constraint in any network is always equal to the throughput flow constraint in its dual network".

2. After choking the flow along several edges of a network, the new throughput flow is equal to the throughput flow in the network before choking the edge flows, minus the total amount of momentary excess flow at all excess nodes, plus the maximum throughput flow in the dual network.

3. In the case where a failure of an edge causes a loss of the entire flow through the edge, the throughput flow constraint theorem permits the calculation of the new maximum throughput flow to be done in $O(m)$ time, where *m* is the number of edges in the network.

In this case, the maximum throughput flow is determined by inspecting the network only locally, in the vicinity of the failed edge, without inspecting the rest of the network.

4. The throughput flow constraint theorem provides the basis for an efficient algorithm for determining the edge flows which correspond to the optimal throughput flow from sources to destinations – the throughput flow achieved with the smallest amount of generation shedding from the sources.

5. The deliberate choking of flows along overloaded edges, leading to a generation of momentary excess and deficit flow, provides a very efficient mechanism for decongesting overloaded branches of the network.

6. The very high average running time of the presented algorithm, makes it particularly suitable for decongesting overloaded transmission links of telecommunication networks, in real time.

7. The proposed algorithm can also be used for re-optimising the flow upon failure or congestion of edges, by independent agents.

APPENDIX

Lemma 1 *If there are no augmentable $s_d - t_d$ paths in the dual network, the momentary excess flow at the excess nodes e_i and the momentary deficit flow at the deficit nodes d_i can always be reduced by augmenting an $s_d - t_d$ path in the dual circulation network, where the circulation edge (t,s) belongs to the augmented path.*

Proof. The network flow in the dual circulation network is feasible, because the flow conservation law is honored at each node and the edge capacity constraints are not violated. There are no excess and deficit nodes in the network, except at the new source s_d and at the new sink t_d (Fig.A1).

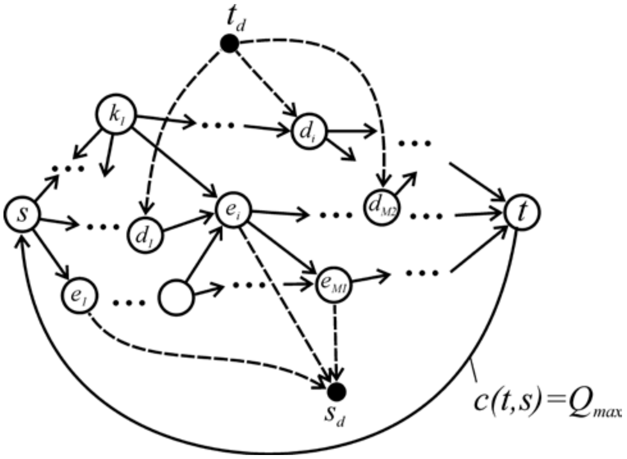

Figure A1. Determining directed $s_d - t_d$ paths, consisting of backward edges only.

Let us start from the start node s_d, by selecting a non-empty backward edge (s_d, e_i), and consider the edges going into node e_i. Because node e_i is now a balanced node, if edge (e_i , s_d) carries some flow out of node e_i, there must be another backward edge (k_1, e_i) going into node e_i and carrying flow greater than zero. Let us consider the start node k_1 of this edge (Fig.A1). The reasoning, which has been applied to node e_i can now be applied to node k_1 and so on, until either a visited node is reached or the end node t_d is reached. Suppose that a visited node *v* has been reached, before the new sink t_d. This means that a cyclic path has been encountered, consisting of backward edges only. Next, the edge carrying the smallest amount of flow belonging to the encountered directed cyclic path is identified, and its flow is subtracted from all edges of the cyclic path. During this operation, the edge carrying the smallest amount of flow becomes empty and the flow conservation law at the repeated node *v*, will not be violated.

Because the edge through which the repeated node *v* has been first reached is not part of the cyclic path, we continue from node *v* and the same process is repeated until no more directed cyclic paths are encountered. After each flow subtraction from the edges of the encountered directed cyclic path, at least one edge from the network becomes empty and is *never filled with flow again*, because only backward edges are selected for the augmented s_d - t_d path. Reaching the end node t_d is guaranteed after at most *m* repetitions of this process, where *m* is the number of edges in the network. Reaching the end node t_d is always guaranteed because all nodes in the network are balanced, except nodes s_d and t_d. Note that during this process, the $s_d - t_d$ path must necessarily include the circulation edge (*t*,*s*). Otherwise, it will follow that there is an augmentable $s_d - t_d$ path in the dual network, which contradicts the condition of the lemma. □

Lemma 2. *If no augmentable $s_d - t_d$ path exists in the dual network, an augmentation of $s_d - t_d$ path in the dual circulation network, results in the absence of augmentable s-t paths in the original network.*

Proof. Consider the dual network in which there are no augmentable $s_d - t_d$ paths, for example the dual network, immediately after the first stage (Fig.A1).

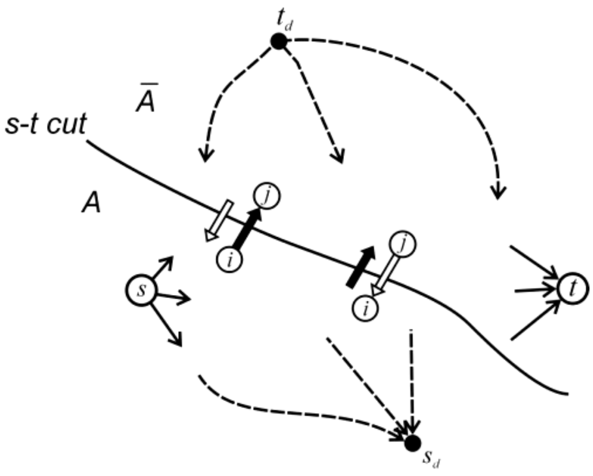

Figure A2. An *s-t* cut, defined by the possibility to reach nodes from the new sink s_d.

In the dual network, define a set *A* of nodes that can be reached through augmentable paths from node s_d (Fig.A2). An augmentable path is a path along which there are no fully saturated forward edges or empty backward edges.

The set $\overline{A}$ includes all nodes that cannot be reached from s_d through augmentable paths. The new sink t_d does not belong to set *A* because, according to the condition of the lemma, there is no augmentable $s_d - t_d$ path in the dual network. The original source *s* belongs to set A because, according to Lemma 1, there is always an augmentable path in the dual circulation network, therefore, there is always an augmentable path from the start node s_d to the original source *s*.

On the other hand, the original sink *t* does not belong to set A, because, according to Lemma 1, there is always an augmentable path in the dual circulation network from the start node s_d to the end node t_d, which includes the circulation edge (*t*,*s*). This means that there is always an augmentable path from the original sink *t* to the end node t_d in the dual network. If the original sink *t* was reachable from s_d through an augmentable path (if the sink *t* belonged to set *A*), the end node t_d would also be reachable from s_d and this will contradict the condition of Lemma 2, that there is no augmentable $s_d - t_d$ path in the dual network. Consequently, the original sink *t* belongs to the set $\overline{A}$. Because a node in the network either belongs to set *A* or to set $\overline{A}$ and no node can simultaneously belong to both sets, $A \cap \overline{A} = \varnothing$ and $A \cup \overline{A} = V$, where *V* is the set of all nodes in the network. In addition, $s \in A$ and $t \in \overline{A}$. As a result, the sets A and $\overline{A}$ define an *s-t* cut $A - \overline{A}$ in the original network.

Edges which cross the *s-t* cut from set *A* to set $\overline{A}$ are fully saturated with flow while edges which cross the cut in the opposite direction, from set $\overline{A}$ to set *A,* are empty (Fig.A2). Indeed, if a forward edge (*i*,*j*) crossing the *s-t* cut is not fully saturated with flow or a backward edge (*i*,*j*) is not empty, this will make edge (*i*,*j*) augmentable and node *j* will be reachable from s_d, because node *i* belongs to set *A* and is therefore reachable from s_d through an augmentable path. Node *j* however, is in set $\overline{A}$ and cannot be reached through an augmentable path from the new source s_d.

The augmented $s_d - t_d$ path in the dual circulation network cannot possibly cross the *s-t* cut through a directed edge, from a node in set *A* towards a node in set $\overline{A}$, because all forward edges crossing the *s-t* cut are fully saturated and all backward edges are empty – therefore none of these edges can be augmented.

The augmented $s_d - t_d$ path however may cross the *s-t* cut through an edge whose starting node *j* is in set $\overline{A}$ and whose end node *i* is in set A (Fig.A3). To do so, the augmented path must either enter the *A* set through a fully saturated backward edge, thereby decreasing its flow, or through an empty forward edge, thereby increasing its flow (Fig.A3). After entering the set *A*, in order to reach the new sink t_d, the augmented $s_d - t_d$ path must come back and cross the *s-t* cut again. Now, except the edge through which the $s_d - t_d$ path entered the *A* set, there are no other augmentable edges crossing the *s-t* cut from set A to set $\overline{A}$, along which the $s_d - t_d$ path can return to set $\overline{A}$.

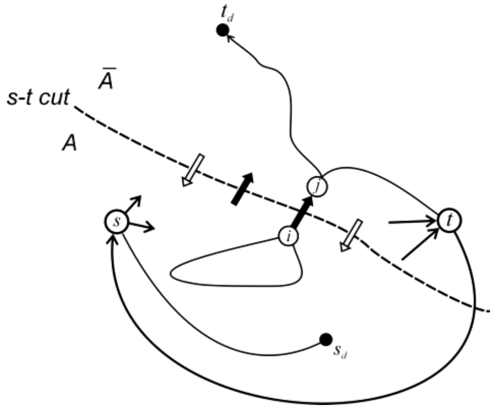

Figure A3. An augmented $s_d - t_d$ path in the dual circulation network.

The only possibility for the $s_d - t_d$ path *to return to set* $\overline{A}$ *is through the same edge through which it has entered the set A*. This means that the augmented $s_d - t_d$ path will restore the state of the edge through which it first entered set *A* and leave the edge in the way it was before entering the set *A* (Fig.A3). This is because the bottleneck flow with which the path is augmented is the same for all edges along the path. Because the augmented $s_d - t_d$ path must reach the new sink t_d, the path can only cross the *s-t* cut (A, $\overline{A}$) an *even number* of times. Therefore, the state of the edges crossing the *s-t* cut from set A to set $\overline{A}$ will be exactly the same as it was before the augmentation of the $s_d - t_d$ path – fully saturated forward edges and empty backward edges. Consequently, there will be no augmentable *s-t* path in the original network, after the augmentation of the $s_d - t_d$ path in the dual circulation network. This finally proves the lemma. □

REFERENCES

[1] Ahuja R.K., T.L.Magnanti, J.B.Orlin, *Network flows: Theory, Algorithms and Applications*, Prentice Hall, 1993.

[2] Asano T., Y.Asano, Recent developments in maximum flow algorithms, *Journal of the Operations Research Society of Japan*, vol.43 (1), pp.2-31, 2000.

[3] Cormen T.H., T.C.E.Leiserson, R.L.Rivest, and C.Stein, *Introduction to Algorithms*, 2nd ed., MIT Press and McGraw-Hill, 2001.

[4] Ford L.R. and D.R. Fulkerson, Maximal flow through a network, *Canadian Journal of Mathematics*, vol.8(5), pp.399-404, 1956.

[5] Edmonds J. and R.M.Karp, Theoretical improvements in algorithmic efficiency for network flow problems, *Journal of the ACM*, vol.19(2), pp.248-264, 1972.

[6] Karzanov A., Determining the maximal flow in a network by the method of preflows., Sov.Math.Dokl., vol.15, pp.434-437, 1974.

[7] Goldberg A.V. and R. E. Tarjan, A new approach to the maximum flow problem, Journal of ACM, vol.35, pp.921-940, 1988.

[8] Sleator D.D. and R.E.Tarjan. An O(nm log n) algorithm for maximum network flow. Technical report STAN-CS-80-831, Department of Computer Science, Stanford University, Stanford, CA, 1980.

[9] J. Ahmed, F. Solano, P. Monti1, L. Wosinska, Traffic Re-Optimization Strategies for Dynamically Provisioned WDM Networks; In the proceeding of the 15th International Conference on Optical Network Design and Modeling (ONDM), 2011.

[10] Dong J., W.Li, C.Cai, Z.Chen, Draining algorithm for the maximum flow problem, *2009 International conference on communications and mobile computing*, pp.197-200.

[11]. Todinov M.T., The dual network theorem for static flow networks and its application for maximising the throughput Flow, *Artificial Intelligence Research*, **2**(1), pp.81-106 (2013).

[12] Tanenbaum A.S., *Computer networks*, 4th ed., Pearson Education International, 2003.

[13] Todinov M.T., Fast Augmentation Algorithms for Maximising the Flow in Repairable Flow Networks After a Component Failure, *Proceedings of the 2011 11th IEEE International Conference on Computer and Information Technology*, Paphos, pp.505-512, 2011.

[14] Overbeeke F., Active networks: Distribution networks facilitating integration of distributed generation, In *Proc. of 2nd international symposium on distributed generation: power system and market aspects*, Stockholm, 2002.

[15] Catterson V., E.M.Davidson, S.D.J.McArthur, Embedded Intelligence for electrical network operation and control, *IEEE Intelligent systems*, vol.26 (2), pp.38-45, 2011.

[16] Jennings N. and S.Bussmann, “Agent-based control systems: Why are they suited to engineering complex systems?” *IEEE Control Systems Magazine*, vol.23 (3) pp.61-73, 2003.

[17] Nagata T., T. Tao, K.Kimura, H.Sasaki, H.Fujita. A multi-agent approach to distribution system restoration, *The 47th IEEE International Midwest Symposium on Circuits and Systems*, 2004.

[18] Saleem A. K.Heussen and M.Lind, Agent services for situation aware control of power systems with distributed generation, *IEEE PES General Meeting*, 2009.