International Journal of Applied Metaheuristic Computing
Volume 14 • Issue 1

Extreme Min – Cut Max – Flow Algorithm

Trust Tawanda, National University of Science and Technology, Zimbabwe*
https://orcid.org/0000-0003-2665-9711

Philimon Nyamugure, National University of Science and Technology, Zimbabwe

Elias Munapo, North West University, South Africa
https://orcid.org/0000-0001-6279-3729

Santosh Kumar, RMIT University, Australia
https://orcid.org/0000-0001-9321-5622

ABSTRACT

In this article, the authors propose a maximum flow algorithm based on flow matrix. The algorithm only requires the effort to reduce the capacity of the underutilized arcs to that of the respective flow. The optimality of the algorithm is proved by the max-flow min-cut theorem. The algorithm is table-based, thus avoiding augmenting path and residual network concepts. The authors used numerical examples and computational comparisons to demonstrate the efficiency of the algorithm. These examples and comparisons revealed that the proposed algorithm is capable of computing exact solutions while using few iterations as compared to some existing algorithms.

KEYWORDS

Maximum Flow Problem, Minimum Cut, Algorithm, Flow Matrix, Max-Flow Min-Cut Theorem

INTRODUCION

The maximum flow problem is a network-based problem, whose objective is to determine the maximum amount of units that can pass through a network. All links in the network have capacity restrictions. Given a network with given capacity restrictions on each link, the maximum flow problem usually deals with the determination of flow patterns thorough various links in the network, so that a maximum flow can arrive at a specified node known as the "destination node" from another specified node, known as the "origin," where the supply in unlimited. Ahmed et al. (2013), Kaml (2017), Munapo et al. (2021), Sarukhi et al. (2017), and Yuan et al. (2010), among others, tackled this problem. The maximum flow network can be directed or non-directed; water networks and road networks are examples of nondirected and directed networks, respectively.

The maximum flow problem has many applications, which include maximizing movement of water from dams, rivers, and boreholes to treatment sites, and then from treatment sites to residents. The maximum flow problem can be used to determine congestion points in urban road networks that require road expansion, so as to minimize congestion and delays. Also, the maximum flow problem is one of the network-based combinatorial optimization problems that requires the development of algorithms and heuristics to determine the maximum flow value in any given network. Researchers

DOI: 10.4018/IJAMC.322436

*Corresponding Author

have been developing methods to solve maximum flow problems using concepts based on augmenting path, maximum flow – minimum cut theorem, and residual networks, among others.

Motivation

The various applications and the complexity of the maximum flow problem have motivated this research. The minimum cut theorem is very powerful and helps to determine the maximum flow value in any given network. The drawback is that it consumes time to determine all the cuts in the network and evaluate them to determine the minimum cut in the set of all possible cuts. Making the position of the minimum cut deterministic in any given network has motivated this research. It becomes very easy to determine the maximum flow value in any given network when the minimum cut position is deterministic. Most algorithms use augmenting paths to determine the maximum flow value in a network; however, in this research, the authors exploited other directions to avoid the use of augmenting paths and residual network concepts.

This research has the following contributions:

1. The authors developed a new algorithm that is based on the concepts of maximum flow – minimum cut theorem. The design of the algorithm is simple; thus, it can be used for teaching purposes and without the aid of computer software.
2. The authors proposed table-based maximum flow algorithm concepts, thus avoiding the augmenting paths and residual networks. The literature revealed that most of the existing algorithms are based on augmenting paths and residual network concepts.
3. The authors proposed two theorems based on the max-flow min-cut theorem. These theorems are very useful in making the position of the minimum cut in any given network deterministic. The authors also proved their proposed theorems mathematically using max-flow min-cut theorem concepts.

This paper is organized in seven sections. The second section presents the literature related to the maximum flow problem; the third section presents the proposed algorithm and theorems, definitions of related terms, and notations; the fourth section offers numerical results and discussions; The fifth and sixth sections report the worst case time complexity of the algorithm and computational results, respectively; lastly, the seventh section provides the conclusion and further research suggestions.

LITERATURE REVIEW

Ma et al. (2019) developed maximum flow algorithms using the concepts of dynamic networks information to maximize the transmission rate in communication networks. Computational experiments have proved 40% increase in throughput and 30% decrease in completion time. Wang (2019) applied the maximum flow algorithms to determine the feasibility of foreign trade transformation and upgrading for China. More in detail, Wang implemented the maximum flow algorithm to design and to improve the China distribution system. This application proved how maximum flow algorithms can be used to solve real world challenges. Zhang et al. (2015) presented a maximum-flow-based data transmission algorithm for distributed computing systems. Rajalakshmi and Vaidyanathan (2019) applied Edmonds-Karp maximum flow algorithm to traffic management system, so as to minimize congestion by finding alternative routes and traffic flow regulating. Ahmed et al. (2013) introduced lexicographic search technique to obtain exact solutions to the problem of maximum flow with minimum attainable cost in a flow network. Their computational experiments revealed that the proposed algorithm computes the maximum flow value in any given network in the least time. Their research demonstrated that powerful maximum flow algorithm can be developed using borrowed or new techniques, thus departing from finding maximum flow using augmenting path and residual path concepts. Gharehbolagh et al. (2016) proposed a maximum flow optimization model that incorporated reliability analysis. Application of reliability in maximum flow networks helps the decision makers to make better decisions. Khanal et al. (2021) presented a new variant of the maximum flow problem named transshipment multicommodity maximum flow problem. In this problem, the maximum flow value is dependent on the storage capacity of intermediate nodes in the flow network. Tawanda (2015) presented a node merging approach for solving the transshipment flow network problem. Tawanda solved the transshipment commodity flow problem as a transportation problem. Munapo et al. (2021) developed a new method to determine the maximum flow by route merging concept, thus reducing the complexity of the network as iteration increases. Kaml (2017) presented a new approach to solve maximum flow problem with fuzzy weights. Indeed, Kaml used numerical examples to demonstrate their proposed method. Surakhi et al. (2017) developed a bio-inspired algorithm for determining the maximum flow value. The results of their computational experiments revealed that the algorithm running time is better when compared to other genetic algorithms. Sivasubramani and Swarup (2011) developed an algorithm that deals with the power flow problem and uses multi-objective harmony techniques. They compared the simulation results with genetic results and revealed that the algorithm is able to compute optimal solutions to the power flow problem. Mohammadi and Tayyebi (2019) presented an algorithm to deal with maximum capacity path interdiction problem with fixed costs. They proved the accuracy and efficiency of the proposed algorithm using computational experiments based on real world data sets. Boykov and Kolmogorov (2004) presented a maximum-flow minimum-cut algorithm and used computational experiments to compare their algorithm with existing algorithms. Their experiments proved that their algorithm is two to five times faster than Dinic maximum flow algorithm and push-relabel maximum flow algorithm. Yuan et al. (2010) developed multiplier-based maximum flow algorithms, proved the convergence of their algorithms using optimization theories, and presented computational experiments to validate the effectiveness of the proposed algorithms. Kobayashi and Otsuki (2014) considered a geographic maximum flow algorithm in a circular disk failure model. Their algorithm is polynomial, as a result it is considered a fast algorithm, when compared with others. Neumayer et al. (2015) proposed polynomial method, exact algorithm, heuristic algorithm, and integer linear programming formulations to solve geographic maximum-flow and minimum-cut problem under a circular disk failure model. Several numerical examples demonstrated the applicability of their proposed algorithms. Zhang et al. (2023) proposed the available flow neural network method to solve the damaged-network time-varying maximal flow problem. A road network in New York was used to prove the validity and efficiency of the available flow neural network method. Many scientists, mathematicians, engineers, and computer scholars have proposed different methods to solve the maximal flow problem and related variants, such as maximum flow problem instance space analysis (Alipour et al., 2023), network reconstruction method for maximum flow problem (Munapo et al., 2022), poly-logarithmic maximum flows (Cen et al., 2023), maximum flow routing strategy (Yang et al., 2023), maximum flow in fuzzy environments (Bavandi & Bigdeli, 2023), and multicommodity flow problem (Gupta et al., 2023).

Most of the existing algorithms in the literature use the concepts of augmenting path and residual networks. This makes these algorithms complex as well require many iterations to compute the maximum flow value. Furthermore, the min-cut max-flow theorem is a very powerful theorem that has not been utilized fully to help find the maximum flow value, because it takes time to determine the minimum cut in any given network. In this paper, the authors propose a new direction to determine the maximum flow value without making use of the augmenting path and the residual networks. The authors extended the concept of the min-cut max- flow theorem to make the position of the minimum cut in any given network deterministic.

Development of Extreme Min-Cut Max-Flow Algorithm

The concept of extreme min-cut max-flow is based on transforming the original flow network through reduction of underutilized arcs, thereby making the position of the minimum cut deterministic and extremely located in the transformed network, thus source and sink extreme. The authors proved the correctness of the concept through two proposed theorems, namely, sink extreme and source extreme theorems. These theorems facilitate the use of max-flow min-cut theorem, without taking much time trying to come up with a set of all possible cuts in any given network. The researchers used flow matrix transformations, thus avoiding the augmenting path and residual network concepts.

Terminology

Excess Flow

Excess flow presents stagnant flow in the network when the total pre-node arc capacity is greater than the total post-node arc capacity. Excess flow can be defined as the flow that is introduced in the network at the source node and fails to reach the sink node due to some arc capacity constraints in the network.

Underutilized Arcs

Arcs are said to be underutilized if, and only if, the actual value flowing through them is less than their actual flow capacity. This situation arises if, and only if, the total flow value of arcs into *v* is less than the total flow capacity of arcs out of *v*.

Network Cut

A cut is any line that separates network nodes into two sets, such as set A and B, where $s \in A$ and $t \in B$, where s and t are the source and sink nodes, respectively. The total capacity of the arcs where the cut passes through is exactly equal to the network cut value.

Minimum Network Cut

This is the cut with the least cut value amongst the set of all possible cuts in any given network. According to the max-flow min-cut theorem, the value of this cut is exactly equal to the maximum flow value.

Flow Matrix

It is a square matrix whose elements are the flow values or weights between all possible flow network nodes. For a flow network with *V* nodes, $V \times V$ is the flow matrix dimension.

Notation

Table 1 gives all the symbols the authors used in their proposed algorithm and the symbol explanations.

Theorems and Proofs

Theorem One: Sink Extreme Min-Cut Theorem

The minimum cut in any network where $f^{+}\left(v\right) \geq f^{-}\left(v\right) \forall\, v \in V - \left\{s,t\right\}$ is the sink extreme cut.

The maximum flow in any network without underutilized arcs ($f^{+}\left(v\right) \geq f^{-}\left(v\right) \forall\, v \in V - \left\{s,t\right\}$) is exactly equal to the sum of arcs into the sink ($f^{+}\left(t\right)$).

Case One: All nodes are at equilibrium, thus $f^{+}\left(v\right) = f^{-}\left(v\right) \forall\, v \in V - \left\{s,t\right\}$.

Proof: Since all nodes are at equilibrium, the flow introduced at the source node $f = f^{-}\left(s\right)$ is exactly the same amount received at the sink node $f = f^{+}\left(t\right)$, thus $f = f^{-}\left(s\right) = f^{+}\left(t\right)$. Now, let $C = \left\{C_k\right\}$ where k = 0, 1, 2, . . , *n* and where C_0 and C_n are the source and sink extreme cuts. Using the max-flow mini-cut theorem, let $C = \left\{C_o, C_1, C_2, \ldots C_{n-1}, C_n\right\}$ be a set of all possible

Table 1. Symbols used in the extreme min-cut max-flow algorithm

Symbols	Explanation
f^*	Maximum flow value.
$f^+\left(v\right)$	Total flow value into vertex v.
$f^-\left(v\right)$	Total flow value out of vertex v .
$f^-\left(s\right)$	Total flow value out of source s.
$f^+\left(t\right)$	Total flow value into sink t.
RFD	Row flow difference.
CFD	Column flow difference.
V	Set of network nodes.

cuts in a network in their order from the source to the sink node, where $C_0 = f^-\left(s\right)$ and $C_n = f^+\left(t\right)$ are the source and sink extreme cuts, respectively. Since all the nodes are at equilibrium, Equation 1 holds $C_o = C_1 = C_2 = \ldots C_{n-1} = C_n$ as follows:

$$\text{Minimum Cut} = \text{Min}\left\{C_0, C_1, C_2, \ldots, C_{n-1}, C_n\right\} = C_n \tag{1}$$

Case Two: The network has either excess flow at all intermediate nodes or a mixture of excess flow and equilibrium nodes, thus $f^+\left(v\right) \geq f^-\left(v\right) \forall v \in V - \left\{s, t\right\}$.

Proof: Using the max-flow mini-cut theorem, let $C = \left\{C_o, C_1, C_2, \ \ldots \ C_{n-1}, C_n\right\}$ be a set of all possible cuts in a network in their order from the source to the sink node, where $C_0 = f^-\left(s\right)$ and $C_n = f^+\left(t\right)$ are the source and sink extreme cuts, respectively. Case Two is possible if, and only if, $f^+\left(v\right) > f^-\left(v\right)$ or a combination of $f^+(v) > f^-(v)$ and $f^+(v) = f^-(v) \forall v \in V - \left\{s, t\right\}$. Then, the following inequality holds $C_o \geq C_1 \geq C_2 \geq \ \ldots \ C_{n-1} \geq C_n$. Hence, Equation 2, which completes the proof, is as follows:

$$\text{Minimum Cut} = \text{Min}\left\{C_0, C_1, C_2, \ldots, C_{n-1}, C_n\right\} = C_n \tag{2}$$

Theorem Two: Source Extreme Min-Cut Theorem

The minimum cut in any network where $f^+\left(v\right) \leq f^-\left(v\right) \forall v \in V - \left\{s, t\right\}$ is the source extreme cut.

Proof: In this kind of network, all the nodes have post underutilized arcs. The authors made all arcs in the network to be utilized fully and created stagnant flow at all the nodes through reversing the direction of all the nodes, thus making the source the sink and the sink the source.

Then, the following inequality holds: $C_o \le C_1 \le C_2 \le \ldots. C_{n-1} \le C_n$ Hence, Equation 3, which completes the proof, is as follows:

$$\text{Minimum Cut} = \text{Min}\left\{C_0, C_1, C_2, \ldots, C_{n-1}, C_n\right\} = C_0 \tag{3}$$

Steps of the Proposed Algorithm

Algorithm 1 indicates the steps of the proposed algorithm.

Algorithm 1. Extreme min-cut max-flow algorithm

Step 1: Convert the maximal flow network into an $n \times n$ flow matrix.
Step 2: Compute row and column flow totals $f^-(v)$ and $f^+(v)$, respectively.
Step 3: Calculate row and column flow differences using the following conditions:
a. If $f^+(v) \ge f^-(v)$, then CFD and RFD are $\left(f^+(v) - f^-(v)\right)$ and 0, respectively.
b. If $f^+(v) < f^-(v)$, then CFD and RFD are 0 and $\left(f^+(v) - f^-(v)\right)$, respectively.
Step 4: If all $CFD = RFD = 0$, then $f^* = f^-(s) = f^+(t)$.
If all $RFD = 0$, then $f^* = f^+(t)$.
If all $RFD < 0$ and all $CFD = 0$, then $f^* = f^-(s)$.
Otherwise, if some $RFD = 0$ and $CFD < 0$, then update the flow matrix using step 5.
Step 5: Subtraction of negative RFD from respective row entries:
a. When all $CFD = 0$: Consider first the row elements in the sink column, if there is any. Otherwise, subtract RFD from the elements in the corresponding row in such a way that (Row elements $\ge$ 0) after the subtraction operation.
b. When $\sum CFD < RFD$: Consider first the row elements that correspond to $CFD > 0$ and subtract those corresponding to CFD. Next, consider the row element that lies in the sink column. Otherwise, consider the row elements with $CFD = 0$. Subtraction of RFD should be in such a way that (Row elements $\ge$ 0) after the subtraction operation.
c. When $\sum CFD \ge RFD$: Consider only row elements whose $CFD > 0$; the subtracted value should not exceed the respective CFD.
d. When at least one CFD $\ge RFD$: Subtract one row element that has $CFD \ge RFD$.
Step 6: Repeat steps 2—5 until all $RFD = 0$.
Step 7: Optimality Conditions: The flow matrix is optimal when all $RFD = 0$.
Step 8: Compute $f^* = f^+(t)$.

NUMERICAL RESULTS AND DISCUSSION

Example One

The authors considered a 6-node network. Nodes 1 and 6 are the source and sink nodes, respectively, and nodes 2, 3, 4, and 5 are the intermediate nodes. The objective is to demonstrate how the proposed method determines the maximum flow of units that can pass through the network at any given time.

Figure 1. Flow chart of the extreme min-cut max-flow algorithm

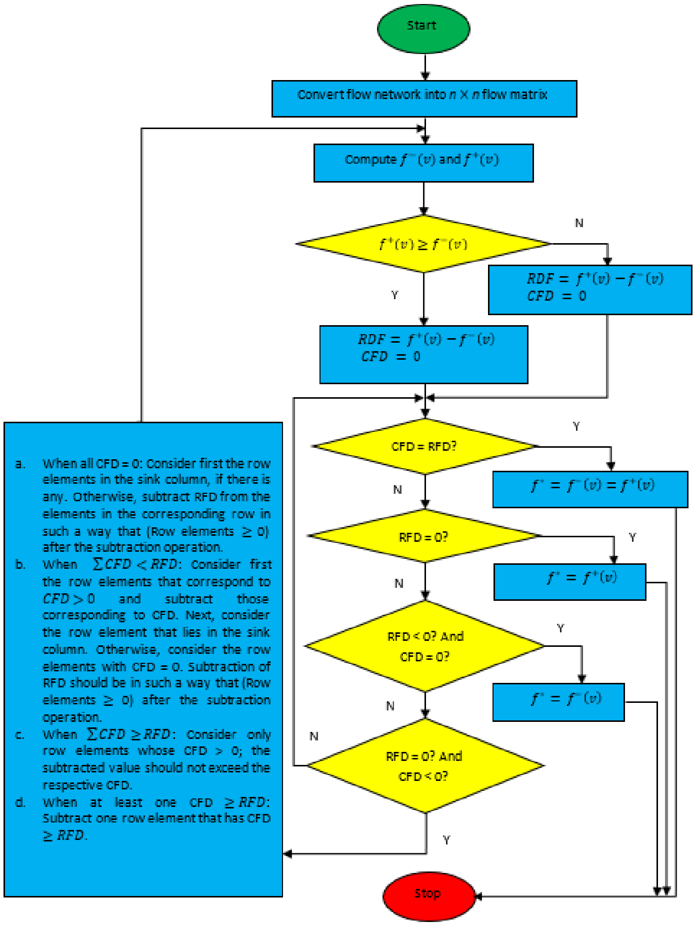
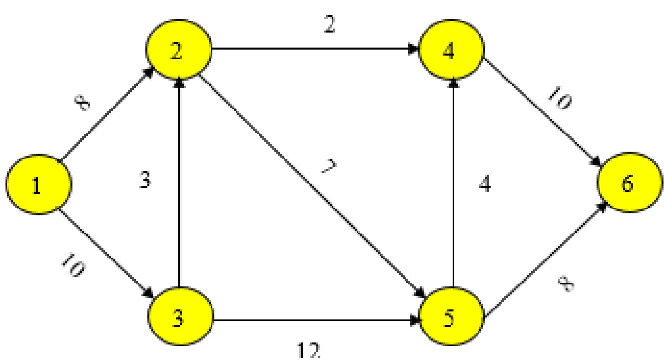


Figure 2. Maximum flow problem one



Iteration One: This iteration includes the following steps:

1. Convert the network problem to flow matrix (Table 2).
2. Compute flow matrix totals and respective differences (Table 3). Optimality is not reached, yet, since some RFD are negative.
3. Go to the next iteration.

Iteration Two: This iteration includes the following steps:

1. Update the flow matrix by subtracting all the negative RDF (Table 4).
2. Compute flow matrix totals and differences (Table 5).

Since all the RFD are equal to zero, all the underutilized arcs have been reduced and, hence, the optimality is reached. Since v =6 is the sink, the maximum flow value can be given by:

Table 2. Flow matrix

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$
$v=1$	∞	8	10	-	-	-
$v=2$	-	∞	-	2	7	-
$v=3$	-			-	12	
$v=4$	-	-	-	∞	-	10
$v=5$	-	-	-	4	∞	8
$v=6$	-	-	-	-	-	∞

Table 3. Computed Totals and Differences

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$	$f^{-}(v)$	RFD
$v=1$	∞	8	10	-	-	-	18	-
$v=2$	-	∞	-	2	7	-	9	0
$v=3$	-	3*	∞	-	12*		15	-5
$v=4$	-	-	-	∞	-	10*	10	-4
$v=5$	-	-	-	4	∞	8	12	0
$v=6$	-	-	-	-	-	∞	-	
$f^{+}(v)$	-	11	10	6	19	18		
CFD		+2	0	0	+7	-		

Rows (v = 3 and v = 4) show RFD of -5 and -4, respectively. These negative RFD are supposed to be subtracted from their respective row pivotal elements. Elements marked with an asterisk (*) are the possible respective pivotal elements.

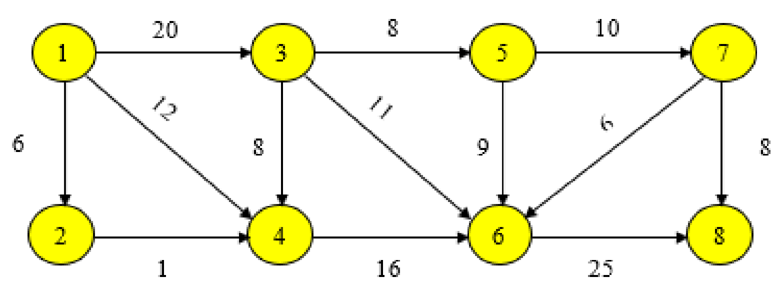
Table 4. Updated flow matrix

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$
$v=1$	∞	8	10	-	-	-
$v=2$	-	∞	-	2	7	-
$v=3$	-	3	∞	-	(12 - 5)	-
$v=4$	-	-	-	∞	-	(10-4)
$v=5$	-	-	-	4	∞	8
$v=6$	-	-	-	-	-	∞

Table 5. Computed totals and differences

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$	$f^{-}(v)$	RFD
$v=1$	∞	8	10	-	-	-	18	-
$v=2$	-	∞	-	2	7	-	9	0
$v=3$	-	3	∞	-	7	-	10	0
$v=4$	-	-	-	∞	-	6	6	0
$v=5$	-	-	-	4	∞	8	12	0
$v=6$	-	-	-	-	-	∞	-	-
$f^{+}(v)$	-	11	10	6	14	14		
CFD	-	+2	0	0	+2	-		

Figure 3. Maximum flow problem two



$$f^{+}\left(v=6\right)=6+8=14\ units$$

Example Two

The author considered an 8-node network. Nodes 1 and 8 are the source and sink nodes, respectively, and nodes 2, 3, 4, 5, 6 and 7 are the intermediate nodes. The objective is to demonstrate how the proposed method determines the maximum flow of units that can pass through the network at any given time.

Iteration One: This iteration includes the following steps:

1. Convert the network problem to flow matrix (Table 6).
2. Compute flow matrix totals and respective differences (Table 7). Optimality is not reached, yet, since some RDF are negative.
3. Go to the next iteration.

Iteration Two: This iteration includes the following steps:

1. Update the flow matrix by subtracting all the negative RFD (Table 8).
2. Compute flow matrix totals and differences (Table 9). Optimality is not reached, yet, since some RFD are negative.
3. Go to the next iteration.

Iteration Three: This iteration includes the following steps:

1. Update the flow matrix by subtracting all the negative RFD (Table 10).
2. Compute flow matrix totals and differences (Table 11).

Since all the RFD are equal to zero, all the underutilized arcs have been reduced and, hence, the optimality is reached. Since v =8 is the sink, the maximum flow value can be given by:

$$f^{+}\left(v=8\right)=25+8=33\ units$$

Table 6. Flow matrix

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$	$v=7$	$v=8$
$v=1$	∞	6	20	12	-	-	-	-
$v=2$	-	∞	-	1	-	-	-	-
$v=3$	-	-	∞	8	8	11	-	-
$v=4$	-	-	-	∞	-	16	-	-
$v=5$	-	-	-	-	∞	9	10	-
$v=6$	-	-	-	-	-	∞	-	25
$v=7$	-	-	-	-	-	6	∞	8
$v=8$	-	-	-	-	-	-	-	∞

Table 7. Computed totals and differences

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$	$v=7$	$v=8$	$f^{-}(v)$	RFD
$v=1$	∞	6	20	12	-	-	-	-	38	-
$v=2$	-	∞	-	1	-	-	-	-	1	0
$v=3$	-	-	∞	8*	8*	11*	-	-	27	-7
$v=4$	-	-	-	∞	-	16	-	-	16	0
$v=5$	-	-	-	-	∞	9*	10*	-	19	-11
$v=6$	-	-	-	-	-	∞	-	25	25	0
$v=7$	-	-	-	-	-	6*	∞	8*	14	-4
$v=8$	-	-	-	-	-	-	-	∞	-	
$f^{+}(v)$	-	6	20	21	8	42	10	33		
CFD		+5	0	+5	0	+17	0	-		

Table 8. Updated flow matrix

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$	$v=7$	$v=8$
$v=1$	∞	6	20	12	-	-	-	-
$v=2$	-	∞	-	1	-	-	-	-
$v=3$	-	-	∞	(8 - 5)	8	(11 - 2)	-	-
$v=4$	-	-	-	∞	-	16	-	-
$v=5$	-	-	-	-	∞	(9 - 9)	(10 - 2)	-
$v=6$	-	-	-	-	-	∞	-	25
$v=7$	-	-	-	-	-	(6 - 4)	∞	8
$v=8$	-	-	-	-	-	-	-	∞

ALGORITHM COMPUTATIONAL COMPLEXITY

Computing row and column totals require an effort of $2E$, computing column flow differences and row flow differences require an effort of $2V$. Updating flow matrix requires an effort of E and combining all these efforts we have the following expression $2E + 2V + E$. This expression reduces to $E + V$. The algorithm requires at most V iterations to terminate. Time complexity of the algorithm is then given by $O\left(VE + V^2\right)$ which reduces to O(VE) when $E > V$.

Table 9. Computed totals and differences

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$	$v=7$	$v=8$	$f^{-}(v)$	RFD
$v=1$	∞	6	20	12	-	-	-	-	-	
$v=2$	-	∞	-	1	-	-	-	-	1	0
$v=3$	-	-	∞	3	8	9	-	-	20	0
$v=4$	-	-	-	∞	-	16	-	-	16	0
$v=5$	-	-	-	-	∞	0	8	-	8	0
$v=6$	-	-	-	-	-	∞	-	25	25	0
$v=7$	-	-	-	-	-	2*	∞	8*	10	-2
$v=8$	-	-	-	-	-	-	-	∞		
$f^{+}(v)$	-	6	20	16	8	27	8	33		
CFD		+5	0	0	0	+2	0			

Table 10. Updated flow matrix

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$	$v=7$	$v=8$
$v=1$	∞	6	20	12	-	-	-	-
$v=2$	-	∞	-	1	-	-	-	-
$v=3$	-	-	∞	3	8	9	-	-
$v=4$	-	-	-	∞	-	16	-	-
$v=5$	-	-	-	-	∞	0	8	-
$v=6$	-	-	-	-	-	∞	-	25
$v=7$	-	-	-	-	-	(2 - 2)	∞	8
$v=8$	-	-	-	-	-	-	-	∞

COMPUTATIONAL COMPARISONS

Three instances (EX-1, EX-2, and EX-3) are considered for computational comparisons. Two (EX-1 and EX-2) of these instances see (Mallick et al., 2016) and the third (EX-3) instance see (Dash and Rahman, 2019). The proposed algorithm is compared to seven state of the art existing algorithms. The algorithms are compared in terms of the solution found, number of iterations and augments. Table 12 summarizes the computational comparisons.

Table 11. Computed totals and differences

	$v=1$	$v=2$	$v=3$	$v=4$	$v=5$	$v=6$	$v=7$	$v=8$	$f^{-}(v)$	RFD
$v=1$	∞	6	20	12	-	-	-	-	-	-
$v=2$	-	∞	-	1	-	-	-	-	1	0
$v=3$	-	-	∞	3	8	9	-	-	20	0
$v=4$	-	-	-	∞	-	16	-	-	16	0
$v=5$	-	-	-	-	∞	0	8	-	8	0
$v=6$	-	-	-	-	-	∞	-	25	25	0
$v=7$	-	-	-	-	-	0	∞	8	8	0
$v=8$	-	-	-	-	-	-	-	∞		
$f^{+}(v)$	-	6	20	16	8	25	8	33		
CFD		+5	0	0	0	0	0			

Table 12. Computational comparisons

Name of the algorithm	Solution found			Number of iterations			Number of augmentation		
	EX-1	EX-2	EX-3	EX-1	EX-2	EX-3	EX-1	EX-2	EX-3
Ford-Fulkerson algorithm Ford and Fulkerson (1956)	72	19	39	9	6	4	8	5	4
Edmonds-Karp algorithm Edmonds and Karp(1972)	72	19	39	7	5	3	7	5	3
An innovative approach Md. Al-Amin Khan et al. (2013)	72	19	39	7	5	4	7	5	3
Improved Edmond - Karp Chintan J. & Deepak G (2012)	72	19	39	6	4	4	6	3	3
An efficient algorithm Ahmed et al. (2014)	72	19	39	6	4	3	6	5	3
Modified Edmonds-Karp algorithm Mallick et al. (2016)	72	19	39	4	2	3	6	3	3
Network reconstruction method Munapo et al. (2022)	72	19	39	6	3	2	N/A	N/A	N/A
Extreme Min – cut Max – flow algorithm (proposed in this paper)	72	19	39	2	3	3	N/A	N/A	N/A
Optimal solution	72	19	39						

CONCLUSION

In this paper, the authors presented a new maximum flow algorithm, which is based on a flow matrix. The algorithm does not use of augmenting path and residual network concepts, thus making it unique from the existing maximum flow algorithm. Further, the authors proposed two theorems that support the idea behind the proposed algorithms. The proposed theorems help to make the position of the min-cut deterministic, thus reducing computational time in finding the exact min-cut among a set of all possible network cuts. In addition, the authors presented numerical examples to demonstrate the validity and efficiency of the algorithm. They compared the algorithm with the existing algorithms on three small-sized instances. The proposed algorithm computed exact maximum flow values on all the three instances. The proposed algorithm terminates after a few numbers of iterations, as compared to other algorithms, thus the algorithms performed better than all on instance EX-1, second best on instance EX-2, and second best performing algorithm on instance EX-3. Further research requires coding of the proposed algorithm, so that computational comparisons using large problem instances can be carried out.

DECLARATIONS

Ethics Approval and Consent to Participate

The authors certify that this research does not involve animal or human participants.

Consent for Publication

N\A.

Availability of Data

Data supporting this research are contained in this manuscript.

Competing Interests

N\A.

Funding

The authors received financial support for the research and publication of this article from the North West University, Mafikeng, South Africa.

Authors' Contributions

All authors contributed equally.

REFERENCES

Ahmed, F., Khan, M., Khan, A. R., Ahmed, S. S., & Uddin, M. (2014). An efficient algorithm for finding maximum flow in a network-flow. *The Journal of Physiological Sciences; JPS*, *19*, 41–50.

Ahmed, N., Das, S., & Purusotham, S. (2013). The problem of maximum flow with minimum attainable cost in a network. *OPSEARCH*, *50*(2), 197–214. doi:10.1007/s12597-012-0106-1

Alipour, H., Muñoz, M. A., & Smith-Miles, K. (2023). Enhanced instance space analysis for the maximum flow problem. *European Journal of Operational Research*, *30*(2), 411–428. doi:10.1016/j.ejor.2022.04.012

Bavandi, S., & Bigdeli, H. (2023). *A maximum flow network interdiction model in fuzzy stochastic hybrid uncertainty environments*. Yugoslav Journal of Operations Research., doi:10.2298/YJOR220415038B

Boykov, Y., & Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(9), 1124–1137. doi:10.1109/TPAMI.2004.60 PMID:15742889

Cen, R., He, W., Li, J., & Panigrahi, D. (2023). Steiner connectivity augmentation and splitting-off in poly-logarithmic maximum flows. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (pp. 2449—2488). Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611977554.ch95

Dash, P., & Rahman, Md. M. (2019). Developing algorithm to obtain the maximum flow in a network flow problem. *Journal of Advanced Research in Dynamic and Control Systems*, *11*(2), 455–459.

Gharehbolagh, H. H., Hafezalkotob, A., Makui, A., & Raissi, S. (2016). Cooperative strategies for maximum-flow problem in uncertain decentralized systems using reliability analysis. *Mathematical Problems in Engineering*, *2016*, 2349712. doi:10.1155/2016/2349712

Gupta, S. P., Pyakurel, U., & Dhamala, T. N. (2023). Multi-commodity flow problem on lossy network with partial lane reversals. *Annals of Operations Research*, *323*(1-2), 45–63. doi:10.1007/s10479-023-05210-y

Kaml, B. S. (2017). Fuzzy type-2 in shortest path and maximal flow problems. *Global Journal of Pure and Applied Mathematics*, *13*(9), 6595–6607.

Khanal, D. P., Pyakurel, U., & Dhamala, T. N. (2021). Maximum multicommodity flow with intermediate storage. *Mathematical Problems in Engineering*, *2021*, 5063207. doi:10.1155/2021/5063207

Kobayashi, Y., & Otsuki, K. (2014). Max-flow min-cut theorem and faster algorithms in a circular disk failure model. In *Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications* (pp. 1635—1643). IEEE. doi:10.1109/INFOCOM.2014.6848100

Ma, P., Wang, J., & You, J. (2019). An improved maximum flow routing algorithm for multi-homing. In *Proceedings of the Information-Centric Networking, 2019 28th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1—8). IEEE. https://doi.org/ doi:10.1109/ICCCN.2019.8846939

Mallick, K. K., Khan, A. R., Ahmed, M. M., Arefin, M., & Uddin, M. S. (2016). Modified Edmonds-Karp algorithm to solve maximum flow problems. *Ozean Journal of Applied Sciences*, *6*(2), 131–140. doi:10.4236/ojapps.2016.62014

Mohammadi, A., & Tayyebi, J. (2019). Maximum capacity path interdiction problem with fixed costs. *Asia-Pacific Journal of Operational Research*, *36*(4), 1950018. doi:10.1142/S0217595919500180

Munapo, E., Kumar, S., & Mpeta, K. (2021). A new direction to the maximal flow problem: A route merging approach. *Applied Mathematics & Information Sciences*, *15*(4), 487–495. doi:10.18576/amis/150411

Munapo, E., Tawanda, T., Nyamugure, P., & Kumar, S. (2022). Maximum flow by network reconstruction method. In *Intelligent Computing & Optimization: Proceedings of the 5th International Conference on Intelligent Computing and Optimization 2022 (ICO2022)* (pp. 926—935). Springer International Publishing. https://doi.org/10.1007/978-3-031-19958-5_87

Neumayer, S., Efrat, A., & Modiano, E. (2015). Geographic max-flow and min-cut under a circular disk failure model. *Computer Networks*, *77*, 117–127. doi:10.1016/j.comnet.2014.10.026

Rajalakshmi, V., & Ganesh Vaidyanathan, S. (2019). Efficient traffic management on road network using Edmonds-Karp algorithm. In C. Panigrahi, A. Pujari, S. Misra, B. Pati, & K. C. Li (Eds.), *Progress in advanced computing and intelligent engineering: Advances in intelligent systems and computing* (Vol. 714). Springer., doi:10.1007/978-981-13-0224-4_52

Sivasubramani, S., & Swarup, K. S. (2011). Multi-objective harmony search algorithm for optimal power flow problem. *International Journal of Electrical Power & Energy Systems*, *33*(3), 745–752. doi:10.1016/j.ijepes.2010.12.031

Surakhi, O. M., Qatawneh, M., & Al Ofeishat, H. A. (2017). A parallel genetic algorithm for maximum flow problem. *International Journal of Advanced Computer Science and Applications*, *8*(6), 159–164.

Tawanda, T. (2015). A node merging approach to the transshipment problem. *International Journal of System Assurance Engineering and Management*, *8*(1), 370–378. https://doi.org/10.1007/s13198-015-0396-9

Wang, H. (2019). Feasibility study of China's foreign trade transformation and upgrading mode based on maximum flow algorithm. *Cluster Computing*, *22*(S6), 14445–14450. doi:10.1007/s10586-018-2312-y

Yang, H., Liu, W., Wang, X., & Li, J. (2023). Group sparse space information network with joint virtual network function deployment and maximum flow routing strategy. *IEEE Transactions on Wireless Communications*, 1. Advance online publication. doi:10.1109/TWC.2022.3233067

Yuan, J., Bae, E., & Tai, X. (2010). A study on continuous max-flow and min-cut approaches. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2217-2224). IEEE. doi:10.1109/CVPR.2010.5539903

Zhang, B., Huang, W., & Zhao, F. (2023). An available-flow neural network for solving the dynamic groundwater network maximum flow problem. *Soft Computing*, 1—12. https://doi.org/10.1007/s00500-023-07912-8

Zhang, X., Jiang, J., Zhang, X., & Wang, X. (2015). A data transmission algorithm for distributed computing system based on maximum flow. *Cluster Computing*, *18*(3), 1157–1169. doi:10.1007/s10586-015-0467-3

Trust Tawanda is faculty member at the National University of Science and Technology, Zimbabwe. Has published papers and book chapters, and is currently a PhD student.

Philimon Nyamugure is Associate Professor and Executive Dean, Faculty of Applied Sciences, National University of Science and Technology, Zimbabwe. Has published in international journals, author of a book and has published book chapters.

Elias Munapo is Professor of OR at North West University, South Africa. Has published in international journals, author of books and edited books and contributed book chapters.

Santosh Kumar is Honorary Adjunct Professor in Mathematics Discipline, RMIT University, Melbourne. He has published papers in international journals, author of books and edited books and contribute book chapters.