Ensemble Visualization of Bottlenecks in Planar Flow Networks

Tobias Post[1,a], Bernd Hamann[2,b], Hans Hagen[1,c] and Jan C. Aurich[3,d]

[1]Computer Graphics and HCI Group, University of Kaiserslautern, Germany

[2]Department of Computer Science, University of California, Davis, U.S.A.

[3]Institute for Manufacturing Technology and Production Systems,
University of Kaiserslautern, Germany

[a]tpost@rhrk.uni-kl.de, [b]hamann@cs.ucdavis.edu, [c]hagen@cs.uni-kl.de, [d]jan.aurich@mv.uni-kl.de

Keywords: Visual Analysis, Bottleneck Visualization, Ensemble Visualization, Flow Networks, Planar Graphs, Maximum Flows, Minimum Cuts

Abstract. The problem of finding bottlenecks in flow networks often appears in real world applications like production planning, factory layout, flow related physical approaches and even cyber security. This work introduces intuitive visual mechanisms to enable domain experts and users to visually analyze stable regions of a network and identify critical transitions. Those transitions form a varying bottleneck front for different configurations of network restraints. To tackle this problem, this work enhances the comparability of different network configurations by utilizing ensemble visualization techniques. The effectiveness of this approach is demonstrated by showing how this enables users to evaluate the progress of different bottlenecks and individual regions in a flow network.

Introduction

The analysis of flows is an important topic in various applications such as cyber security [14], biological pathways [20] and cyber physical manufacturing systems [15]. In particular, one aim of designing a manufacturing system is to identify weaknesses in the manufacturing system's layout in an early planning stage to minimize costs, raise the product quality and shorten production times. An important factor to optimize production system is the identification and elimination of bottlenecks [18].

The analysis of bottlenecks in cyber physical manufacturing systems can be described by flow networks with machines as nodes/vertices and product flows as edges of the network. Depending on the factory setting, each of these edges has a specific capacity, describing the maximum amount of products that can flow between the two connected machines. To identify the bottleneck of a manufacturing system, the correlating flow network is subject of analysis. Contrary to intuition, the bottleneck of a flow network is not a single edge between two nodes. Instead, a bottleneck is a whole set of edges. The minimum cut of a flow network can help in describing these bottlenecks. This cut separates the nodes of the flow network into two groups: one that can be reached by the network's source, and the other being the remaining nodes. In this mathematical setup, the question arises how to identify the true bottleneck edge in the group of minimum cut edges, how to visually encode this bottleneck, and how to compare various network configurations and their resulting bottlenecks. A sufficient solution targeting all mentioned problems was not provided so far, as shown in section "Related Work".

This work extends the definition of a minimum cut in a flow network by separating the nodes of a flow network into three groups: nodes that can be reached from the source, nodes that can reach the sink of the network, and the remaining nodes. This definition allows an enhanced classification of edges crossing these regions to identify those specific edges that are the bottlenecks of the network. To define an intuitive visualization for bottlenecks in a network, this work presents an intuitive visualization based on Voronoi diagrams [10] derived from the underlying graph's node layout. This work uses color-coded regions to indicate bottleneck transition of a flow network. Based on these regions, an ensemble visualization technique for multiple configurations of a flow network is presented in this work. The resulting ensemble visualization indicates common bottlenecks and differences in the underlying configurations by an intuitive color-coding of Voronoi cells (section "Methods"). Section "Conclusion" will summarize this work and point out future challenges.

Therefore, this work contributes:

- An extended definition of the minimum cut in flow networks
- An intuitive visualization of a minimum cut in a flow network
- An intuitive ensemble visualization for multiple configurations of a flow network

Related Work

This section will present the state of the art in minimum cut visualization as well as ensemble visualizations for flow networks.

Visualization of Minimum Cuts. Vehlow et al. [19] presented a state of the art report summarizing available network drawing methods with the goal of grouping the nodes of graphs. Although they presented a large variety of graph-drawing algorithms, an intuitive visual mapping of the minimum cut itself was not presented. In contrast to that, the presented approach introduces a visual encoding for the minimum cut based on Voronoi cells.

Brandes et al. [5] presented a planar visualization for the minimum cut in flow networks by arranging a network in an rectangular manner and adding a poly-line to indicate the cut. This method is widely used in open source solutions as [16, 17, 12]. Although this method gives a suitable first indication of the minimum cut, it can not indicate edges in a flow network that represent a bottleneck for the network. The presented approach utilizes the method of Brandes et al. as a starting point and refines the definition of a minimum cut to enhance transitions that form the bottleneck of the considered system.

Ensemble Visualization of Flow Networks. The ensemble visualization of graphs is an important feature to allow a comparative investigation of their behavior. A state of the art report was given by Borgo et al. [11]. They presented a large variety of visualization techniques for graph ensemble. The relevant techniques will be discussed below.

Using the third dimension to stack different network configurations on top of each other was presented by Brandes et al. [4] and Itoh et al. [13]. Although this provides a good overview over the capacity values in the considered network configurations, the approach is not able to indicate common bottlenecks. In contrast to that, the presented approach uses a visualization based on regions that indicate the specific location of nodes with respect to the bottleneck of a network.

Temporal variance [1, 2, 3], variety-based edge visualization and selective accumulated visualization [2] form techniques that are visualizing the variety of occurring capacity constraints in a network ensemble. Although they provide an interactive visualization for an overview of the occurring capacity values, they lack the ability to indicate common bottlenecks. Therefore, the presented approach allows to identify common bottlenecks of a given ensemble of a network's capacity configurations.

Cesario et al. [6] presented a technique to visualize graph ensembles by visually encoding multiple capacities per edge. This enables users to get an overview of different network configurations, but lacks the ability to show the resulting bottlenecks. In contrast to that, this work focuses on the visualization and accumulation of bottlenecks and their propagation.

Boyandin et al. [3] presented a visualization of graph capacities that are aligned in a plane and can therefore be reviewed in total. Although, the line representation of the minimum cut could be added to each of these representations easily, it would be hard to identify their similarities. Therefore, this work presents a single visualization for the entire ensemble that is able to indicate stable and divergent bottleneck regions.

Methods

The analysis of bottlenecks in flow networks is an essential task for many real world applications in planning and engineering. This section will introduce a method to visually inspect those bottlenecks. Also, a method to analyze the propagation of bottlenecks for a network with an ensemble of different configurations will be demonstrated.

Flow Networks. This work will rely on the general definition of flow networks with a single source and sink, which will be presented below. Fig. 1 shows an example for such a flow network. A *network* $N = (G, c, s, t)$ consists of a directed graph $G = (V, E)$ with a finite set of vertices V and a set of directed edges $E \subseteq V \times V$. Here, the edges should not include self loops or multiple edges in the same direction between any two nodes. The capacity function $c : E \to \mathbb{R}_+$ assigns a non-negative capacity value to every edge in the network. The vertices $s, t \in V$ with $s \neq t$ should be the only source and sink in the network, respectively.

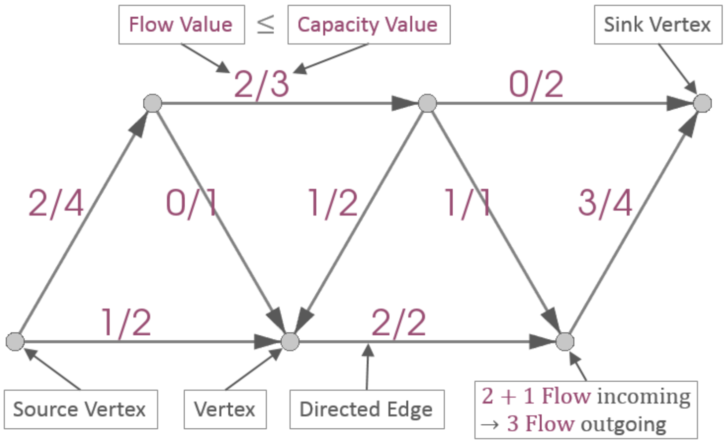
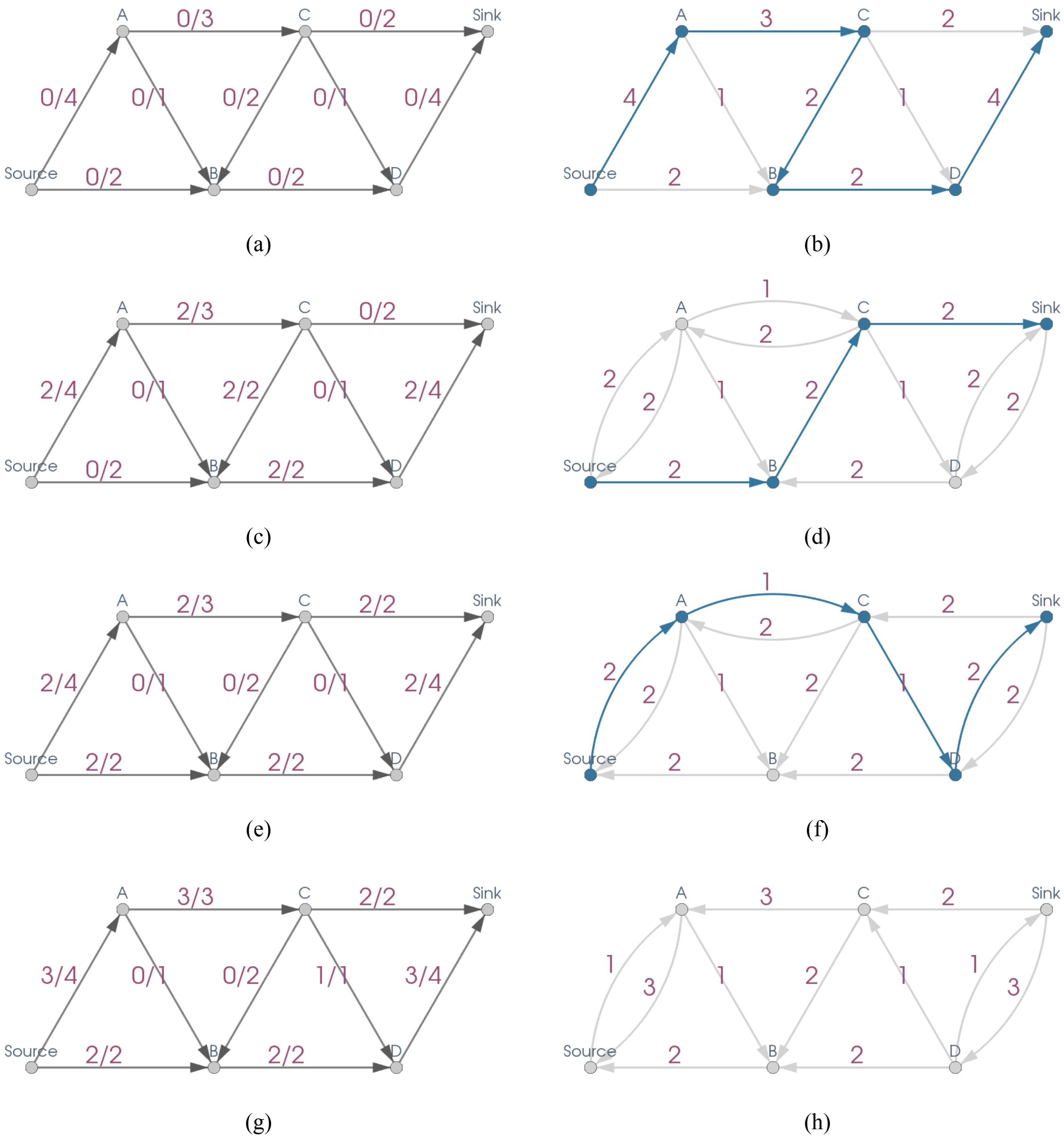


Fig. 1: Flow network consisting of vertices, directed edges, a source and sink vertex, and a flow and capacity value per edge. The capacity limits the flow. Except the source and sink, all vertices preserve the flow.

A *flow* $f : E \to \mathbb{R}_+$ is a function assigning a non-negative flow value to each edge in the network. Hence, a *flow network* is a network together with a specific flow on it. There are several constraints that have to apply for such a flow. The flow should be limited by the capacity, so $\forall e \in E : f(e) \leq c(e)$, meaning, that the flow along an edge is never larger than the edge's capacity. Also, all vertices except the source and sink should preserve the flow, so $\forall v \in V \setminus \{s, t\} : \sum_{(w,v) \in E} f(w, v) = \sum_{(v,w) \in E} f(v, w)$, meaning, that the total incoming flow is equal to the total outgoing flow for a vertex. For the source, the total outgoing flow is larger than the total incoming flow, and for the sink this is reversed.

The *value of a s-t-flow* $|f| = \sum_{(s,w) \in E} f(s, w) - \sum_{(w,s) \in E} f(w, s)$ is the value of the outgoing flow of the source s minus its incoming flow. Since all vertices except the source s and sink t preserve the flow, this is the same as the value of the incoming flow of the sink s minus its outgoing flow. This work focuses on planar flow networks, so to restrict the general definition of (flow) networks to planar (flow) networks, the respective graph G should be planar. This means, that G can be plotted in a 2D plane without edges crossing each other. Fig. 1 shows an example for a planar flow network with a proper embedding into the image plane.

Maximum Flows. A *maximum flow* $\hat{f}$ on a network N has the largest value among all possible flows on N, so there exists no other flow f with $|f| > |\hat{f}|$. Maximum flows are interesting, since lower capacity constraints could be used to achieve flows with smaller values. This means, that given capacity constraints limit maximum flows only. So to evaluate the full potential of networks, the maximum flows have to be analyzed. This leads to the question of how to find a maximum flow for a given network. The method of Ford and Fulkerson [9] is a general approach to find such a maximum flow. Fig. 2 shows the individual steps of this approach. To understand this approach, the definition of a residual network needs to be understood.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 2: Iterations of the Ford and Fulkerson method to improve the flow of a network. To a given flow network (left images), the residual network is calculated and a path from source to sink is found (right images). This augmenting path is used to increase the flow along the path's edges in the original network. When no more path is found (see 2(h)), the flow is a maximum flow (see 2(g)).

For the flow networks in Fig. 2(a)-2(g), the respective residual networks are shown in Fig. 2(b)-2(h). For a given flow network $N = (G, c, s, t)$ with flow f the *residual network* is defined as $N_f = (G_f, c_f, s, t)$ with $G_f = (V, E_f)$. So the vertices V and the source s and sink t of the residual network are the same as the ones of the given network, though the edges E_f and their capacities c_f change. The edges and capacities of the residual network are defined as follows. For each edge $(v, w) \in E$ a forward edge (v, w) is added to E_f if $f(v, w) < c(v, w)$. The capacity of such a new forward edge (v, w) is set to $c_f(v, w) = c(v, w) - f(v, w)$. For each edge $(v, w) \in E$ a backward edge (w, v) is added to E_f if $f(v, w) > 0$. The capacity of a new backward edge (w, v) is set to $c_f(w, v) = f(v, w)$. Following this definition, a residual network describes the amount of flow that can be added to an edge before the capacity limit is reached (forward edge), and the amount of flow that can be subtracted from an edge before a negative flow would arise (backward edge).

The method of Ford and Fulkerson now operates on these residual networks. For a flow network in Fig. 2(a)-2(g) the residual network in Fig. 2(b)-2(h) is calculated. A directed path from the source to the sink is found in the residual network. This path is called an augmenting path, since the flow of the edges in the original network on this path can be improved, thereby increasing the value of the overall flow in the network. So for a forward edge in the residual network the flow of the original edge is increased, and for a backward edge in the residual network the flow of the original edge is decreased. This procedure is iterated as long as no more augmenting paths can be found in the residual network. It can be shown, that the value of the resulting flow is maximal, so the resulting flow is a maximum flow. Fig. 2(g) shows the maximum flow with a value of 5.

The algorithm of Edmonds and Karp [7] uses a breadth-first-search from the source to always find a shortest augmenting path in the residual network. This ensures the termination of the algorithm as well as a polynomial bound of the algorithm's run-time, leading to an efficient algorithm to find maximum flows. It can be shown, that the run-time complexity of this algorithm is in $O(|V| \cdot |E|^2)$, meaning that the run-time is bounded asymptotically by $k \cdot |V| \cdot |E|^2$ for a fixed constant k, $|V|$ vertices and $|E|$ edges, and therefore is not dependent on the capacities. Although even faster algorithms with a complexity of nearly up to $O(|V| \cdot |E|)$ are known, the method of Ford and Fulkerson was demonstrated above, as the shown definitions like augmenting paths will be used in the following.

Minimum Cuts. To find bottlenecks in networks, maximum flows could be considered. As the maximum flow in Fig. 2(g) shows, there are exhausted edges. For these edges the flow value equals the capacity value, so the flow can not be increased any further. An example for such an edge is the edge $(Source, B)$ with values "2/2". One could easily think, that increasing the capacity of this edge would result in a larger maximum flow, meaning, that this edge would be called a bottleneck edge in the following. It turns out, that this early intuition is wrong, and an increase of the capacity of this edge would not increase the value of the maximum flow. To countervail this effect, this work focuses on cuts instead of flows.

A *s-t-cut* $C = (S, S')$ is a partition of the vertices V into the disjunct sets $S \subset V$ with $s \in S$ and $S' \subset V$ with $t \in S'$ such that $S \cup S' = V$. The *capacity of a s-t-cut* $|C| = \sum_{(v,w) \in E \,:\, v \in S \,\wedge\, w \in S'} c(v, w)$ is the sum of the capacities of edges from a vertex in S to a vertex in S'. A *minimum cut* $\check{C}$ of a network N has the smallest capacity among all possible cuts of N, so there exists no other cut C with $|C| < |\check{C}|$.

The max-flow min-cut theorem [8] from graph and optimization theory states $|\hat{f}| = |\check{C}|$, so the value of a maximum flow is equal to the capacity of a minimum cut and vice versa. This means, that instead of considering maximum flows for the analysis of the performance and bottlenecks of networks, minimum cuts can be utilized.

The standard approach to find a minimum cut for a given network is to first calculate the maximum flow as described above, and then collect all vertices that are reachable from the source vertex in the resulting final residual network. Those vertices form the set $\check{S}$ of the cut, with $\check{S}' = V \setminus \check{S}$ being the set of remaining vertices. The desired minimum cut then is $\check{C} = (\check{S}, \check{S}')$.

Fig. 3(a) shows the residual network of the maximum flow (compare Fig. 2(h)), the collection of vertices starting from the source in blue, and the remaining vertices in white. To enhance the intuitiveness of the visualization and enable users to easily analyze minimum cuts, Fig. 3(b) colors the Voronoi cells [10] of each vertex by a partition specific color, blue for the vertices in S and white for all other vertices in S'. The Voronoi cell of a vertex is the area that is closer to this vertex then to all other

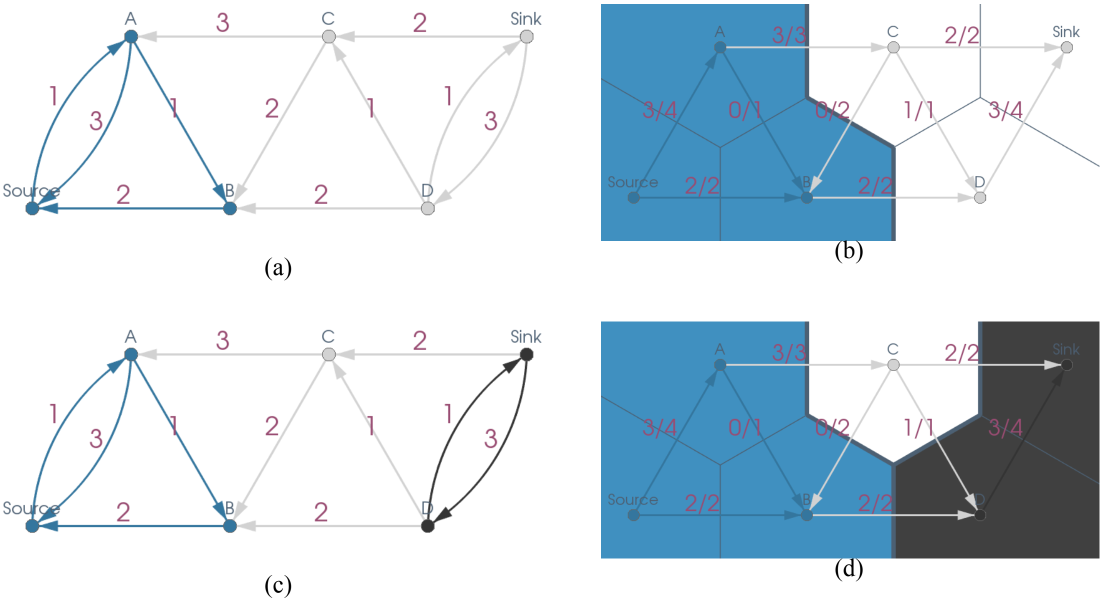


Fig. 3: Residual network of an exemplary network with maximum flow (compare Fig. 2(h)) with vertices/edges reachable forwards from source or backwards from sink (left images), and the original network with minimum cut and classified Voronoi cells (right images). The classical construction of a minimum cut (upper images) suggest wrong bottleneck edges, while the new extended construction (lower images) shows the true bottleneck transitions (blue to black).

vertices. By using Voronoi cells that share a common border to other cells of the same color, regions for both partitions of the minimum cut are formed.

As can be seen, the previously considered edge $(Source, B)$ starts and ends in the blue region and can not be increased to increase the value of the maximum flow. Hence, this edge is not a bottleneck edge. In general, for all edges ending in S (blue region) by construction there exists a directed path in the residual network from the source to the endpoint of the edge. So instead of increasing the capacity of such an edge, the flow along this path could be improved. So an edge ending in S (blue region) can not be a bottleneck edge. In contrast to this, one could investigate the behavior of an edge starting in the blue region and leading to the white region. As and example, the edge (A, C) with values “3/3” is considered. But again the intuition fails and the considered edge is not a bottleneck edge.

This shows, that the general definition of a cut is not enough to find bottleneck edges. To compensate this shortcoming, this work extends the construction of a minimum cut by adding a third set $T \subset V$ to the partition. Fig. 3(c) and 3(d) show the same visualizations as before, but this time all vertices that have a directed path to the sink in the residual network are collected in the set T and colored in black. All vertices that are not reached from the source or do not reach the sink form the set $R \subset V$ with $R = V \setminus (S \cup T)$ and are left white. So the new partition is $P = (S, R, T)$ (blue / white / black regions) with disjoint sets $S, R, T \subset V$, and $S \cup R \cup T = V$, and $s \in S$ and $t \in T$.

Analog to before, all edges starting in T (black region) can not be bottleneck edges, since by construction there exists a directed path in the residual network from the starting point of the edge to the sink. All edges ending in R (white region) also can not be bottleneck edges, since by construction they do not have a directed path in the residual network from their endpoint to the sink. Increasing the capacity of such an edge could increase the value of a flow from the source to the edge’s endpoint, but not to the sink. So the overall flow would not increase, hence the edge is no bottleneck. Out of analog reasons, edges starting in R (white region) also can not be bottleneck edges.

The only edges left that can be bottlenecks are edges starting in S (blue region) and leading to T (black region). As can be seen in Fig. 3(d), the edge (B, D) with values “2/2” is of this kind and is a bottleneck edge, so increasing the capacity of this edge would lead to a larger maximum flow (compare Fig. 4(c)). In general, not only some, but all edges leading from S (blue region) to T (black region) are bottleneck edges.

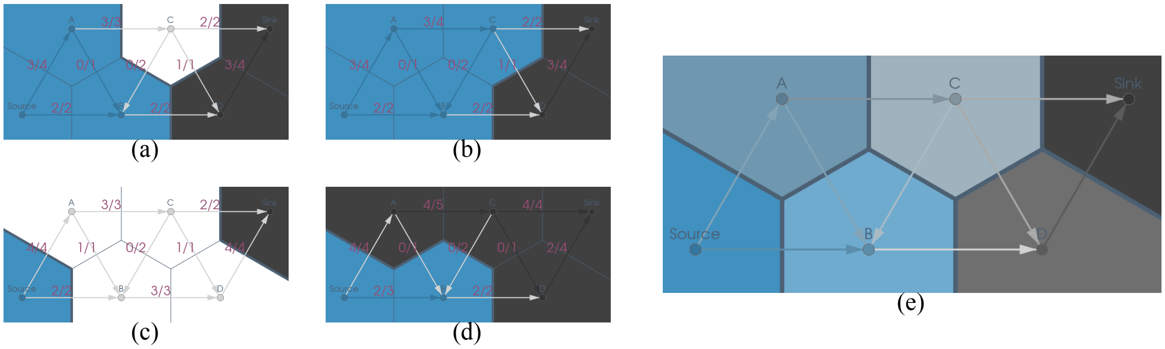
Proof: Let $(v, w) \in E$ with $v \in S$ and $w \in T$ be an edge leading from S (blue region) to T (black region). By construction, there exists a directed path $(v_1, v_2, ..., v_n)$ with $v_1, v_2, ..., v_n \in V$ and $v_1 = s$ and $v_n = v$ in the residual network from the source s to the starting point v of the edge. By construction there also exists a directed path $(w_1, w_2, ..., w_m)$ with $w_1, w_2, ..., w_m \in V$ and $w_1 = w$ and $w_m = t$ in the residual network from the endpoint w of the edge to the sink t. If both paths had a common vertex $v_i = w_j$, the path $(s = v_1, v_2, ..., v_{i-1}, v_i = w_j, w_{j+1}, ..., w_{m-1}, w_m = t)$ would be an augmenting path, and hence the given flow would not have been a maximum flow. So both paths are disjoint and do not increase the overall flow without modifying $c(v, w)$. Also, the flow $f(v, w)$ of the given edge equals its capacity $c(v, w)$, because otherwise the edge (v, w) would be included in the residual network and the path $(s = v_1, v_2, ..., v_n = v, w = w_1, w_2, ..., w_n = t)$ would be an augmenting path. But by modifying the capacity $c(v, w)$ to a greater value $c'(v, w) > c(v, w)$ it holds, that $f(v, w) < c'(v, w)$, so the edge (v, w) will be included in the modified residual network. This leads to an augmenting path $(s = v_1, v_2, ..., v_n = v, w = w_1, w_2, ..., w_n = t)$ that can be used to increase the value of the overall flow. Hence, increasing the capacity of an edge from S to T increases the value of the maximum flow, so all edges from S (blue region) to T (black region) are bottleneck edges.

So the overall approach works by first performing a max-flow calculation followed by two separate breadth-first-searches in the residual network starting forwards from the source and backwards from the sink, respectively. Since the residual network has the same number of vertices and at most twice the number of edges than the original network, the run-time complexity of the breadth-first-searches is in $O(|V| + |E|)$, meaning that the complexity and limitations of the overall approach are dependent only on which max-flow algorithm is chosen, as described above.

Ensemble Visualization. The visualization in Fig. 3(d) (compare Fig. 4(a)) intuitively shows the different regions of a network with maximum flow. By construction, there is enough capacity left to increase the flow from the source to the vertices in the blue region. On the other hand, there is enough capacity left to increase the flow to the network’s sink from one of the vertices in the black region. The white region consists of the remaining vertices that are not able to increase the flow in either direction. So to identify bottlenecks for a single network, users can intuitively consider transitions from a blue to a black region. Since multiple Voronoi cells form regions, in general there is not just a single bottleneck edge, meaning a single transition, but a closed front of bottleneck transitions. In addition to that, the different regions deliver insight into whether vertices and edges are in front of the overall bottleneck front or behind it.

In real world application there often is the need to compare results for different layouts or configurations. Here, the capability to get an overview of different capacity configurations for the same underlying graph is desired. To tackle this problem, Fig. 4(a)-4(d) show an ensemble of networks, their maximum flow and the different regions of the extended minimum cut. The different configurations were chosen arbitrarily to demonstrate a variety of possible scenarios. The individual Voronoi cells are accumulated and averaged as shown in Fig. 4(e).

Fully blue or black regions in the accumulated image indicate a completely stable location in front of or behind the bottleneck, respectively. Fully white regions never occur in any other (blue or black) region. The advantage of having chosen a blue / white / black color theme is, that these colors accumulate uniquely. A region that is bluer appears more often before the bottleneck front in the individual ensemble members, while blacker regions appear more often behind the bottleneck. The



(a) (b) (c) (d) (e)

Fig. 4: Ensemble of the same network with four different exemplary capacity configurations leading to four different maximum flows and minimum cuts (smaller images). Transitions from a blue to a black region are bottlenecks. The samples of the ensemble are accumulated and averaged, thereby giving an overview of stable regions (fully blue or black) and the progress of the bottlenecks (larger image).

same holds for whiter regions in the accumulated image that appear more often in neither of the two (blue or black) regions.

Fully blue, black or white regions indicate a stable behavior of the respective vertices in all ensemble members regardless of their different capacity configurations. In contrast to that, regions with a not so well defined color indicate unstable regions, meaning a divergent behavior for the different ensemble members. Also, a fully blue to fully black transition indicates a stable bottleneck regardless of the different ensemble configurations. Overall, the presented visualization methodologies enable an intuitive identification of network bottlenecks and the analysis of bottleneck propagation for an ensemble of a network with different capacity configurations.

Conclusion

This work has introduced a novel approach to visualize and compare bottlenecks in flow networks for real world applications like cyber physical manufacturing systems. Therefore, product flows and constraints of a manufacturing system were mapped to a network. This work extended the definition of a minimum cut of a network to identify bottleneck edges. This extended definition was used as a basis to visualize minimum cuts and bottlenecks in production systems based on Voronoi regions. This approach allowed a fast and intuitive identification of bottleneck transitions in a flow network.

As the planning of production systems involves the comparison of different factory settings, various configurations of a network need to be compared. This work has presented an ensemble visualization technique based on the visualization of extended minimum cuts. The presented examples demonstrated, that the ensemble visualization technique indicates stable and divergent bottleneck transitions and regions, thereby enabling users to visually identify and analyze similarities and differences of bottlenecks for an ensemble of flow network configurations.

Acknowledgments

This research was funded by the German Research Foundation (DFG) within the IRTG 2057 “Physical Modeling for Virtual Manufacturing Systems and Processes”.

References

[1] B. Bach, E. Pietriga, and J.-D. Fekete. Graphdiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization Computer Graphics*, 20(5):740–754, 2014.

[2] L. Borisjuk, M. Hajirezaei, C. Klukas, H. Rolletschek, and F. Schreiber. Integrating data from biological experiments into metabolic networks with the DBE information system. *In Silico Biology*, 5(2):93–102, 2004.

[3] I. Boyandin, E. Bertini, and D. Lalanne. A qualitative study on the exploration of temporal changes in flow maps with animation and small-multiples. *Computer Graphics Forum*, 31(3pt2):1005–1014, 2012.

[4] U. Brandes and S. R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization*, 2(1):40–50, 2003.

[5] U. Brandes, S. Cornelsen, and D. Wagner. *How to Draw the Minimum Cuts of a Planar Graph*, pages 89–119. Springer Berlin Heidelberg, 2001.

[6] N. Cesario, A. Pang, and L. Singh. Visualizing node attribute uncertainty in graphs. *Proc. SPIE*, 7868:78680H–78680H–13, 2011.

[7] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

[8] P. Elias, A. Feinstein, and C. Shannon. A note on the maximum flow through a network. *Information Theory, IEEE Transactions on*, 2(4):117–119, 1956.

[9] L. R. Ford and D. R. Fulkerson. Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[10] S. Fortune. Voronoi diagrams and delaunay triangulations. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 377–388. CRC Press, Inc., 1997.

[11] S. Hadlak, H. Schumann, and H.-J. Schulz. A Survey of Multi-faceted Graph Visualization. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015.

[12] S. Halim. https://visualgo.net/maxflow. *online*, 2017.

[13] M. Itoh, D. Yokoyama, M. Toyoda, Y. Tomita, S. Kawamura, and M. Kitsuregawa. Visual exploration of changes in passenger flows and tweets on mega-city metro network. *IEEE Transactions on Big Data*, 2(1):85–99, March 2016.

[14] J. Jaffe. Bottleneck flow control. *IEEE Transactions on Communications*, 29(7):954–962, 1981.

[15] M. Kikolski. Identification of production bottlenecks with the use of plant simulation software. *Ekonomia i Zarzadzanie*, 8(4):103–112, 2017.

[16] S. Klamt, J. Saez-Rodriguez, and E. D. Gilles. Structural and functional analysis of cellular networks with cellnetanalyzer. *BMC Systems Biology*, 1:open access, 2007.

[17] S. Klamt and A. von Kamp. An application programming interface for cellnetanalyzer. *BioSystems*, 105:162–168, 2011.

[18] C. G. Lee and S. C. Park. Survey on the virtual commissioning of manufacturing systems. *Journal of Computational Design and Engineering*, 1(3):213 – 222, 2014.

[19] C. Vehlow, F. Beck, and D. Weiskopf. Visualizing group structures in graphs: A survey. *Computer Graphics Forum*, pages n/a–n/a, 2016.

[20] H. Yu, P. M. Kim, E. Sprecher, V. Trifonov, and M. Gerstein. The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics. *PLoS Computational Biology*, 3(4), 2007.