Operations Research Forum (2022) 3: 45
https://doi.org/10.1007/s43069-022-00152-w

ORIGINAL RESEARCH

Improving Patient Flow in a Primary Care Clinic

Nathan Preuss[1,2] · Lin Guo[3] · Janet K. Allen[4] · Farrokh Mistree[4]

Received: 17 December 2021 / Accepted: 25 June 2022 / Published online: 1 September 2022

Abstract

When patients visit primary care clinics, they can be subject to long wait times due to operational inefficiencies and bottlenecks, decreasing patient satisfaction and sometimes leading to worse health outcomes. The existing literature models primary care clinics primarily as agent-based models, which are excellent at tracking individual patients and their movements in a model of a clinic. While agent-based models can detect bottlenecks, a network flow model better detects bottlenecks in the model by correlating changes in patient flow and wait times in the healthcare network. In this paper, a network flow model is constructed, where patients flow along the capacitated edges of a network while receiving treatment at the nodes. This configuration easily identifies bottlenecks by analyzing the flow in and flow out of nodes through metrics such as efficiency and patient wait times. The capacities of the edges for this model are taken from an agent-based model of a case study of a primary care clinic and sampled as random variables. Ensemble runs of the network flow model are created to account for uncertainty in the synthetic data. By changing the topology of the network flow model, bottlenecks are removed, increasing the model efficiency and decreasing patient wait times. Finally, the model is subjected to a sensitivity analysis. The focus in this work is on the method rather than the results per se.

Keywords Primary care clinic · Network flow model · Efficiency · Patient wait times · Synthetic data

Highlights
- A network flow model can be optimized to minimize each individual patient's wait time, or to minimize all of the patient wait times in the network.
- By changing how patients move through a primary care clinic, patient wait times can be reduced.
- Synthetic data can be used to model a primary care clinic under a variety of circumstances.

✉ Janet K. Allen
janet.allen@ou.edu

Extended author information available on the last page of the article

1 Frame of Reference

When patients visit primary care clinics, they may be subject to long wait times due to bottlenecks or other operational inefficiencies. These long wait times decrease patient satisfaction and patient happiness, and in some cases can lead to worse patient health outcomes. There are many methods to model healthcare networks: the most common are agent-based models and network flow models. Potential solutions to long wait times in healthcare networks are identified in these models, such as redesigning a process or changing patient flow to eliminate bottlenecks within the healthcare network. The surrounding literature encompasses a wide variety of modeling techniques and measurements not just within a healthcare context, as shown in Table 1.

There are three broad clusters of papers: in the first, the focus is on agent-based models of healthcare networks at smaller timescales [1, 2]; in the second cluster, network flow models at longer timescales are used [3, 4]; and in the third, various modeling techniques are used which are adapted in this paper [5, 6], including stochastic models [7, 8].

The authors of the first cluster of papers use agent-based models to explore low-level patient processes, such as patient check-in or patient scheduling. Bobbie [1] uses an agent-based model to look at how various scheduling practices affect patient wait times at a primary care clinic where walk-in patients are plentiful. They note that modifying the scheduling of patients has a "significant impact on the wait time of scheduled patients when walk-in patients are present," decreasing average patient wait times by about 22.6% [1]. Su et al. [2] use business process redesign coupled with an agent-based model to decrease patient wait time while checking in to hospitals. They find that "simulation modeling can provide essential assistance in the healthcare service process evaluation and reengineering" while drastically reducing patient wait time from "50 min to 8 min." Furthermore, the authors use two different types of wait times in their analysis, maximum total wait time, and average total

Table 1 Papers of significance [1–8]

Group	Cluster 1 Agent-based models		Cluster 2 Network flow models		Cluster 3 Various modeling techniques			
Reference	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
Agent-based model	X	X						
Network flow model			X	X	X			X
Stochastic model							X	X
Integer model							X	X
Ensemble models	X		X			X		
Wait time measurement	X	X		X	X		X	X
Efficiency measurement				X			X	
Model modification	X	X	X					
Healthcare context	X	X	X	X			X	X

wait time. Agent-based models can identify bottlenecks in healthcare networks, and modifications to the model are effective at reducing patient wait times and increasing patient satisfaction.

The authors of the second cluster of paper use network flow models to explore higher-level patient processes at longer timescales. Akcali et al. [3] use a network flow model to "simultaneously determine the timing and magnitude of changes in bed capacity that minimizes capacity cost… while maintaining a desired level of facility performance… over a finite planning horizon." This time horizon is measured in quarters, unlike the time horizon in minutes that characterizes the agent-based models. Furthermore, the authors design this model with parameters that could be adjusted under slightly different scenarios, creating ensemble models to determine bed capacity. Bean et al. [4] considered a hospital as a collection of wards and used a network flow model to analyze the efficiency of moving patients among the different wards. The authors found that the ward flow has a "core" sub-network that "constitutes 83–90% of all flow" through the network, but the remaining flow went through a larger number of edges. The authors furthermore found that changes to the patient flow through a hospital "separate the best and worst-performing days in each hospital site," although the authors caution that this change in flow may not cause longer wait times [4].

The authors of the third cluster of papers introduce techniques of modeling from outside a healthcare context that may be of use in designing a network flow model of a primary care clinic. Zawack and Thompson [5] introduce two concepts relating to network flow models of traffic networks, user optimal wait time, and system optimal wait time. In the context of a healthcare network, a network is user optimal if each patient minimizes their wait time, and a network is system optimal if the total wait time is minimized. Skurichina and Duin [6] introduce three methods of building ensemble models from a limited data source: bagging, boosting, and the random subspace method. The random subspace method allows for the construction of ensemble models with parameters taken from a constrained solution space. Fu and Banerjee [7] use a stochastic integer model to manage the fluctuations of service time and increasing urgent requests brought by COVID-19, taking into account uncertainties such as no shows, cancelations, punctuality of patients, or overtime treatment. Yang and Rajgopal [8] formulate a multi-period integer stochastic model to design clinic outreach networks for vaccination and determine the worst-case solutions to address uncertainties.

These papers are not the only methods of generating stochastic parameters. In 1976, Box and MacGregor [9] proposed estimating parameters using closed-loop operating data. Bhatnagar and Patel [10] propose a method to use stochastic approximation to tune parameters for active queue management. In addition, Lee et al. [11] schedule physicians under different scheduling algorithms with various levels of pre-emption. Unlike this paper, the authors give additional tasks to physicians, including paperwork. For the parameters of the model, Lee et al. use a Poisson distribution to measure inter-arrival times, which is also used in this paper. Finally, the authors use the coefficient of variation to measure the sensitivity of their model to the underlying parameters, an approach that is replicated in Sect. 4.3. In this paper, we focus on the feasibility of incorporating stochasticity

and how it affects the wait times of the patients and the locations of bottlenecks in the network, instead of stochastic technology. Therefore, in this paper, we only incorporate one method to illustrate the stochasticity in a test problem.

As Ajorlou et al. [12] point out, the supply of healthcare is deterministic and can be calculated easily, "based on headcounts and available service hours," whereas the demand for healthcare is not so easily calculated and depends upon factors such as the age and gender of a patient, as well as possible comorbidities. These variances in patient demand should be captured within the model, as doctors may take longer on some days to treat the same number of patients. Moreover, Fletcher and Worthington [13] distinguish between generic hospital models, which can be adapted with different data to different hospitals, and specific hospital models, which are designed to work only at one hospital.

Agent-based models and network flow models provide different analysis of the same system, and both types of models are useful. While agent-based models can identify bottlenecks in the healthcare network, network flow models can easily correlate changes in patient flow and waiting times, indicating that bottlenecks are present in a system. Because it is assumed that all patients have the same priority and differences in treatment are not discussed, a network flow model may give an appropriate level of detail and a better understanding of network efficiency. Many agent-based models use patient wait times as a metric, but few use a measure of technical efficiency. In this paper, a network flow model is constructed to evaluate and, as appropriate, to predict network performance and efficiency is presented, allowing healthcare management to implement changes to reduce patient wait time and increase efficiency. In more complex networks with multiple bottlenecks, management can use this technique to explore multiple solutions that increase efficiency.

In Sect. 2, the attributes and equations of the network flow model of the primary care clinic are introduced. Then, various metrics that measure the efficiency of the network and the wait time of patients are introduced. Because data is unavailable, synthetic data is generated for a primary care clinic using the random subspace method with suggestions for values adapted from Bobbie's model to build an ensemble model that captures varying levels of average patient treatment times at different nodes. In Sect. 3, the creation of a network flow model is presented. In Sect. 4, a clinic is evaluated with metrics measuring user optimal wait times, system optimal wait times, and efficiency, while also exploring how changes to the constraints of the subspace affect the metrics. Closing remarks are presented in Sect. 5.

2 Designing a Network Flow Model

In this section, the various attributes of the nodes and edges are presented for network flow models, as well as equations describing patient behavior in the model. Various metrics are introduced, and synthetic data generation and sampling are also discussed.

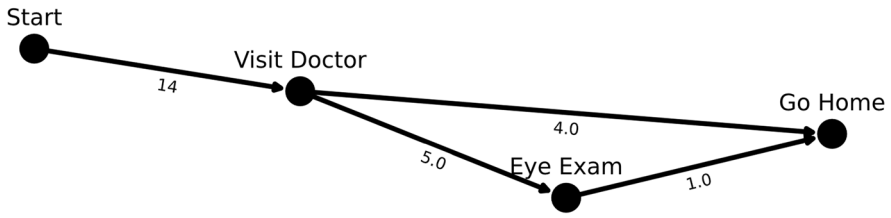

Fig. 1 Example network of a simple primary care practice. Node names are given above the nodes represented as circles, the capacities of the edges are given below the edges, and the arrows on the edges represent the direction of patient flow

2.1 Attributes of a Network Flow Model

Instead of an agent-based model that tracks individual patients as they move through a primary care clinic, a network flow model tracks how patients move through a network with capacitated edges that connect various nodes representing various procedures. This network flow model is assumed to be a directed acyclic graph, requiring patients to only flow in one direction, and requiring that no loops are present in the network. A sample primary care clinic is shown in Fig. 1, where patients visit an eye doctor and are either sent back home or referred for an eye exam. In this example, the "Visit Doctor" node is a bottleneck. As the network becomes more complex, this method should be able to be applied to other primary care clinics so long as the directed acyclic property is maintained.

In the paper, it is assumed that patients arrive at a given node equally interspersed throughout a given time period. Therefore, the time between patient arrivals at a given node is constant. Similarly, patient treatment times at a node are assumed to take a constant amount of time, but this constant time can change between models. For example, in one of the ensemble models, a certain node may treat 10 patients per hour, for a constant time between patients of 6 min; but in a different model, that same node may treat 12 patients per hour, for a constant time of 5 min per patient. Attempts to assign an exact arrival time require keeping track of the location of patients within the network, essentially converting this network flow model into an agent-based model. Furthermore, it is assumed that patients visiting the primary care clinic do not require urgent care, and it is predicted that this model will not work well in settings beyond a primary care clinic, where patients with varying urgency for treatment appear.

Nodes have the following six attributes:

- Node ID: A unique number identifying the node, for example, node x.
- Node Name: A name that gives a brief description of the procedure occurring at node x.
- Number of patients treated at a given node x (P_x): The number of patients the node can treat within a given time interval. Each patient's treatment time is constant within a given time period, but in a different model, the number of patients treated within a given time period changes, causing different treatment times between models.

- *Flow In*$_x$: The number of patients flowing into a given node x within a given time interval.
- *Flow Out*$_x$: The number of patients flowing out of a given node x within a given time interval.
- *Total Wait Time* $\left(W_x\right)$: The total wait time experienced by all patients at node x.

Edges have the following five attributes:

- Node from: The Node ID of the node the edge is coming from.
- Node to: The Node ID of the node the edge is connected to.
- Capacity ($C_{d,e}$): The maximum number of patients flowing that can flow through an edge between nodes d and e within a given time interval.
- Weight: The *Longest Single Patient Wait Time* for the node the edge flows into.
- Branching coefficient ($b_{d,e}$): The probability of a patient flowing down the edge from node d to node e.

The *Flow Out* of a node x is the sum of all capacities of i edges leaving the node.

$$Flow\ Out_x = \sum_{k=1}^{i} C_{x,k} \tag{1}$$

The *Flow In* to a node x is the sum of all capacities of j edges entering the node.

$$Flow\ In_x = \sum_{k=1}^{j} C_{x,k} \tag{2}$$

When the *Flow In* to node x is greater than the *Flow Out* of node x, a bottleneck appears:

$$Flow\ In_x > Flow\ Out_x \tag{3}$$

To ensure that the number of patients flowing out of a given node along multiple edges is equivalent to the number of patients being treated at a given node, a branching coefficient b is introduced. The branching coefficient can be modeled as a multinomial random variable, representing the probability that a given patient will go down each edge. Each edge i leaving a given node is assigned a coefficient b such that the sum of all branching coefficients of edges leaving node x is 1.

$$\sum_{k=1}^{i} b_{x,k} = 1 \tag{4}$$

$$0 < b_{x,i} \leq 1 \tag{5}$$

The capacity for an edge leaving node x to node i is defined as the minimum of the *FlowIn* for a given node and the number of patients being treated at a given node, and that quantity times the branching coefficient for that edge:

$$C_{x,i} = MIN\left(Flow\ In_x, P_x\right) * b_{x,i} \tag{6}$$

The effect of Eq. (6) is that the flow of patients through an edge will always equal the capacity of that edge. It is not necessary to know the theoretical maximum number of patients flowing through an edge. Setting capacity equal to the number of patients flowing through the edges makes it easier to use a graph algorithm[1] to calculate the flow through the network. Using Eqs. (1), (2), and (6), the capacity for each edge of the network can be calculated once the initial *FlowIn* to the network and the various values of P_x are known.

2.2 Metrics for a Network Flow Model

There are three metrics to measure the severity of the bottlenecks of the network:

- Efficiency (E): The percentage of patients that flow out of the network compared to the original flow in of the network.
- *Longest Single Patient Wait Time*: A user optimal metric that measures the longest possible wait time a single patient could experience on a path through the network.
- *Total Wait Time*: A system optimal metric that measures the total wait time of all patients within the network.

To measure the efficiency of the network, the *FlowIn* for the first node a is compared to the flow out for the last node z and converted to a percentage. Efficiency can be applied to any sub-network in the larger overall network. Kawaguchi et al. [14] differentiate between revenue efficiency and technical efficiency; because revenue is not considered in the data, the efficiency described here is a type of technical efficiency. As Cinaroglu [15] and Berry et al. [16] point out, specific hospital characteristics such as quality management, technology level, structure of the hospital department, and degree of professionalism of healthcare workers potentially affect the efficiency and productivity of a hospital. These potential sources of inefficiency would be baked into the underlying model parameters discussed in Sect. 2.3.

$$E = \frac{Flow\ Out_z}{Flow\ In_a} * 100 \tag{7}$$

The wait time for a given patient n at node x depends upon the difference in time between the patient's arrival at a node and when they are treated.

$$Wait\ Time_{x,n} = Treatment\ Time_{x,n} - Arrival\ Time_{x,n} \tag{8}$$

[1] https://networkx.org/documentation/stable//reference/algorithms/generated/networkx.algorithms.flow.maximum_flow.html#networkx.algorithms.flow.maximum_flow. This graph algorithm is used to find the maximum single-commodity flow between a start node and an end node, that is, the path from the start node to the end node that has the greatest flow using the capacities of the edges. The algorithm cannot return a path a patient cannot take because of the directed acyclic nature of the graph. Accessed 7/26/21.

Table 2 Wait times for patients at the Visit Doctor node from Fig. 1

Patient number	Arrival time (minutes)	Treatment start time (minutes)	Wait time (minutes)
1	0 min	0 min	0 min
2	4.29 min	6.67 min	2.38 min
3	8.57 min	13.33 min	4.76 min
4	12.86 min	20.00 min	7.14 min
5	17.14 min	26.67 min	9.52 min
6	21.43 min	33.33 min	11.90 min
7	25.71 min	40.00 min	14.29 min
8	30.00 min	46.67 min	16.67 min
9	34.29 min	53.33 min	19.05 min
10	38.57 min	60.00 min	21.43 min
11	42.86 min	66.67 min	23.81 min
12	47.14 min	73.33 min	26.19 min
13	51.43 min	80.00 min	28.57 min
14	55.71 min	86.67 min	30.95 min

The treatment time for each patient is dependent upon the time period T and the number of patients P_x that can be seen within the time period at a given node x. Because the first patient arriving during the time interval is immediately treated (has a treatment time of 0), the treatment time for the nth patient can be given by Eq. (9).

$$Treatment\ Time_{x,n} = \frac{T}{P_x} * (n-1) \tag{9}$$

The arrival time for each patient is dependent upon the time period T and the number of patients flowing into a given node x. Again, the first patient arrives immediately at the beginning of the time interval, so the arrival time for the nth patient is given by Eq. (10).

$$Arrival\ Time_{x,n} = \frac{T}{\text{Flow In}} * (n-1) \tag{10}$$

In Fig. 1, if 14 patients flow into the network at the Start Node with a time period of 1 h, and assuming that patient arrivals and patient treatment times are interspersed throughout the hours, the wait times for each patient can be calculated at the Visit Doctor node using Eqs. (8), (9), and (10) and are displayed in Table 2.

There is a bottleneck at the Visit Doctor node because the *FlowIn* to the Visit Doctor node (14) is greater than the *FlowOut* of that node (9). As shown in Table 2, the later a patient arrives to a node with a bottleneck, the longer that patient waits. The wait time increases at a constant rate for each additional patient entering a node.

Combining Eqs. (8), (9), and (10), the wait time for the nth patient to arrive at a given node x for a given time interval T and $1 \leq n \leq Flow\ In_x$ is given in Eq. (11).

$$Wait\ Time_{x,n} = \left(\frac{T}{P_x} - \frac{T}{Flow\ In_x}\right) * (n-1) \tag{11}$$

The last patient to visit node x will have the longest wait time. For Table 2, the *Longest Single Patient Wait Time* is 30.95 min, and this value is assigned to each edge flowing into the Visit Doctor node as the *Weight* attribute. A graph algorithm[2] can compute the path through the network with the highest total *Weight*, determining the *Longest Single Patient Wait Time*.

The total wait time of all n patients being treated at a given node x can also be calculated. If there is no bottleneck, Eq. (12) will sum to zero.[3]

$$Wait\ Time_x = \sum_{n=1}^{Flow\ In_x} MAX(0, \left(\frac{T}{P_x} - \frac{T}{Flow\ In_x}\right) * (n-1)) \tag{12}$$

Summing this attribute over all the nodes of the network produces the final metric for measuring the severity of a bottleneck, *Total Wait Time*.

When a bottleneck exists in a network, some patients are stopped at the bottleneck and do not flow through to the rest of the model. While they still contribute to the wait time at the node with the bottleneck, they do not contribute to the wait times of later nodes in the network. This property enables the measurement of the efficiency of the network but results in wait time measurements that are a lower bound for the true wait time experienced by patients in the network. If a bottleneck is resolved close to the beginning of the network, the newly freed patients may encounter another bottleneck later in the network. Therefore, the removal of this bottleneck may not decrease patient wait times by as much as the removal of a similarly sized bottleneck towards the end of the network. Bringing the patients back inside the network could be achieved by adding a new attribute to the edges or relaxing the assumptions about patient arrival times.

2.3 Sampling from a Random Subspace

Unlike Fig. 1, the reality of a primary care clinic is that there are a variable number of patients seeking treatment each day, and each patient takes a variable amount of time to be treated. Jiang et al. [17] find that unscheduled patient arrivals at hospitals can be modeled as a Poisson random variable. In this network flow model, a Poisson random variable determines the *FlowIn* to the start of the model as well as the number of patients being treated at each node. These parameters

[2] https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.dag.dag_longest_path.html. This algorithm returns the longest path in a directed acyclic graph based on the weight attribute of each edge. Accessed 7/26/21.

[3] For example, using Eq. (12) to calculate the total wait time at the Visit Doctor node in Fig. 1 would result in a total wait time of 216.66 min.

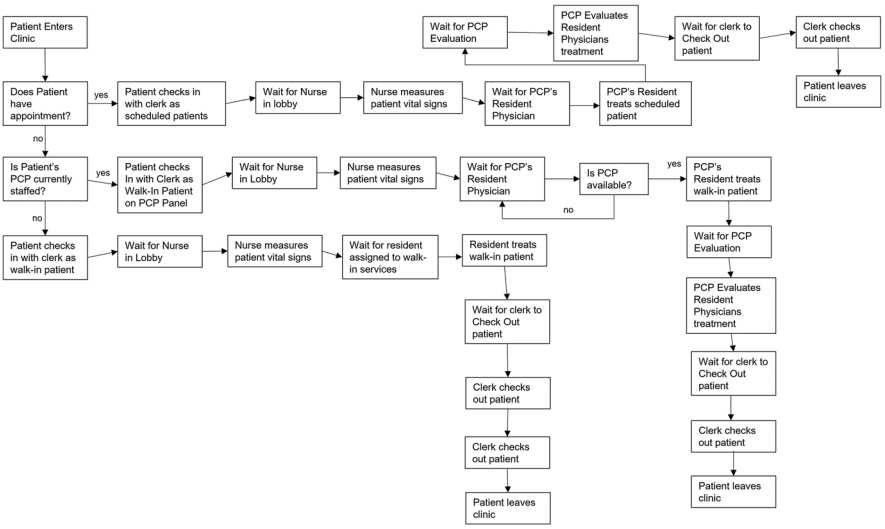

Fig. 2 Recreation of Original Orlando VAMC model [1]. Patients move through the flowchart following the direction of arrows and doing tasks at each box. If a question is asked in a box, the answer to the question determines which subsequent box the patient visits

can be modified to account for variation in the care environment (the number of physicians on staff, technological capability, professionalism of staff, etc.) that are specific to either certain days (in case hospital demand varies greatly day to day) or are specific to certain hospitals. In turn, sampling from random variables affects the capacity of various edges for the network flow model in this paper. Alternatively, robust optimization could be used to estimate the model parameters without knowledge of the probability distribution of the model parameters as demonstrated by Aslani et al. [18].

Therefore, the solution space for this model is constrained by the standard deviation and mean of the values sampled from a Poisson random variable for P_x and the *Flow* In_x to the start of the network and a multinomial random variable for$b$.[4] By sampling from these distributions, a random subspace is generated and used as the basis for a model in the ensemble. By repeating this process, additional models are generated and added to the ensemble. The models in the ensemble are averaged to determine where bottlenecks are likely to appear and generate summary statistics.

[4] Other random variables or combinations thereof can be used to as constraints to the subspace if they better represent patient arrivals and treatment at primary care clinics.

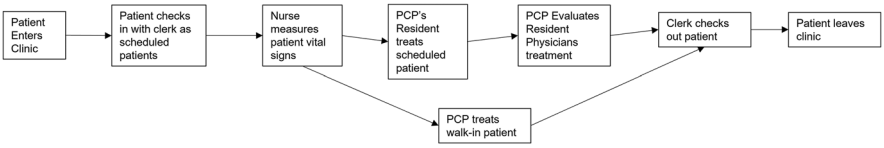

Fig. 3 Orlando VAMC model with removed nodes. Patients move through the flowchart following the direction of arrows and doing tasks at each box

3 Creating a Network Flow Model of a Primary Care Clinic

3.1 Constructing a Network Flow Model

While the framework for the model could be applied to many different settings, in this paper, the network flow model is largely adapted from the agent-based model of the Orlando Veteran Affairs Medical Center (VAMC). Figure 2 is a modified flowchart of patient flow through the Orlando VAMC from Bobbie's paper [1]. Nodes dedicated to patients waiting and duplicate nodes are eliminated, simplifying the flowchart to Fig. 3, and in Fig. 4 the names of the nodes are shortened for ease of reference.

In the model of the Orlando VAMC as shown in Fig. 4, patients start at the Start Node. From there, the patients check in with the receptionists, before traveling to the Nurse and getting their vitals checked. In the upper branch, scheduled patients are treated by a Resident Physician, whose treatment is then evaluated by a Primary Care Physician (PCP). The patients then Check Out at the receptionist desk. In the lower branch, walk-in patients are treated by a Primary Care Physician and are sent to Check Out.

Next, patient treatment times (P_x) are determined for each node, as well as the initial number of patients being treated each day. On average, 50 patients visit the Orlando VAMC each day, 38 of which are scheduled, and 12 of which are walk-in [1]. Furthermore, it appears that "there are 4 first year residents who see 1 new

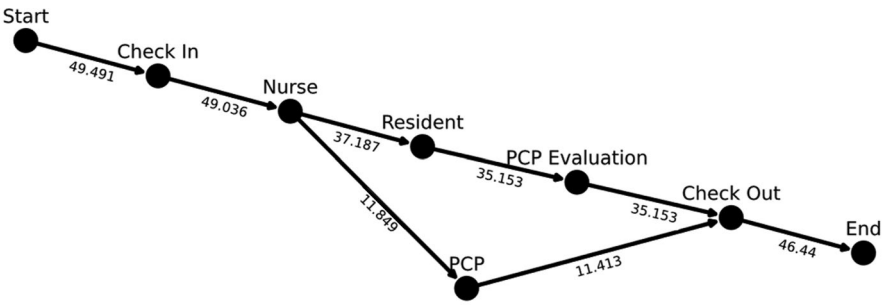

Fig. 4 Topology of the base ensemble model. Node names are given above the nodes represented as circles, the capacities of the edges are given below the edges, and the arrows on the edges represent the direction of patient flow

Table 3 Theoretical number of patients that are treated per day on average at each given node at the Orlando VAMC

Node	Mean number of patients treated per day (8 h)
Start	50
Check In	64
Nurse	192
Resident	40
PCP Evaluation	160
PCP	12
Check Out	64

patient and 3 return patients, and 4 s year residents who see 1 new patient and 5 return patients" [1]. Therefore, the Resident node is able to treat 40 scheduled patients each day. Furthermore, the six PCPs treat about 2 walk-in patients each, resulting in that node treating 12 patients per day [1]. Further patients per day numbers are estimated from patient treatment times [1].

The subspace for each model run is sampled from Poisson random variables with lambda equal to the number of patients treated per day in Table 3. A Poisson random variable has a standard deviation equal to the square root of λ. The probability of the Poisson random variable returning a value of x for a given lambda λ is given by Eq. (13). For example, at the Resident node, there would be a 6.3% chance for $P_{Resident} = 40$ and a 4.85% chance for $P_{Resident} = 35$.

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!} \tag{13}$$

For the only branch in the network, after leaving the nurse node, patients have a 24% change of going to the PCP node, and a 76% chance to go to the resident node as determined by a multinomial random variable.

3.2 Coding the Network Flow Model

With all the necessary data gathered, the network flow model of the Orlando VAMC is implemented in Python, with graphs and statistics generated in R.[5] Object-oriented programing and inheritance are used to minimize code duplication and mistakes in the code base. NetworkX is the main package used to build and visualize the network[6]; ggplot2[7] is used to create the figures. The base model is the parent class, and any topological changes to the base model are implemented as child classes, using the parent methods whenever possible. The base model has a constructor which declares the variables, and methods to initialize, build, analyze, and visualize the model. Additional functions are present to help gather data and calculate wait times.

[5] GitHub Repository: https://github.com/nbpreussOU/HERE1
[6] https://networkx.org/ Accessed 6/24/21.
[7] https://ggplot2.tidyverse.org/ Accessed 6/24/21.

Table 4 Base ensemble model results for 500 runs

Metric	Mean	Extreme
Flow in	49.49 patients per day	69 patients per day
Flow out	46.44 patients per day	62 patients per day
Efficiency	94.39%	70.91%
Longest single patient wait time	47.82 min	316 min
Total wait time	879 min	5213 min
Average wait time per patient	17.8 min	75.6

In the GitHub repository, there are four different versions of the model (as of 6/05/2021): VAMCv0 is the base class, VAMCv1 is the modified model where some patients are sent home, VAMCv2 is a discarded model looking at modifying the underlying distribution, and VAMCv3 modifies the constraints of the subspace. The driver program creates an arbitrary number of ensemble models (500 in this case) to be analyzed, gathers the data in dataframes, performs analysis, and saves the data in a.csv to make graphs in R. If a model had a *FlowIn*, *FlowOut*, *E*, *Total Wait Time*, or *Longest Single Patient Wait Time* with a value that was more than three standard deviations from the mean value of all models, it was discarded as an outlier.

4 Results and Discussion

4.1 Base Model Results

To create the results, 500 runs of the base model are simulated and combined into an ensemble model to ensure adequate coverage of the subspace distributions. To gain an understanding of what an average day at the Orlando VAMC is, a number of patients flowing along each edge are averaged together from all models in the ensemble and are displayed below the edges in Fig. 4. From comparing the C of edges entering and leaving each node, it appears that there is a small bottleneck at the Check In node, a larger bottleneck appears at the PCP node, and a large bottleneck exists at the Resident node. These bottlenecks are the causes of the inefficiencies and long wait times present in Table 4.

As noted in Table 4, bottlenecks reduce the efficiency of the network, with the worst bottleneck resulting in a model run with an efficiency of 70.91%. On average, the bottlenecks in the network combine to cause a decrease in efficiency, with the average model run having an efficiency of 94.39%. The histogram in Fig. 5 further breaks down the distribution of efficiency: 221 of the 470 runs are 100% efficient, with a long tail distribution of inefficient model runs. In most cases, the ensemble models run smoothly at 100% efficiency and patients wait for 0 min, but in the worst case, wait times increase and efficiency decreases. Figure 6 shows the correlation between the *Longest Single Patient Wait Time* and the *Total Wait Time*. In general, the two metrics are highly correlated, but there appear to be two separate bottlenecks

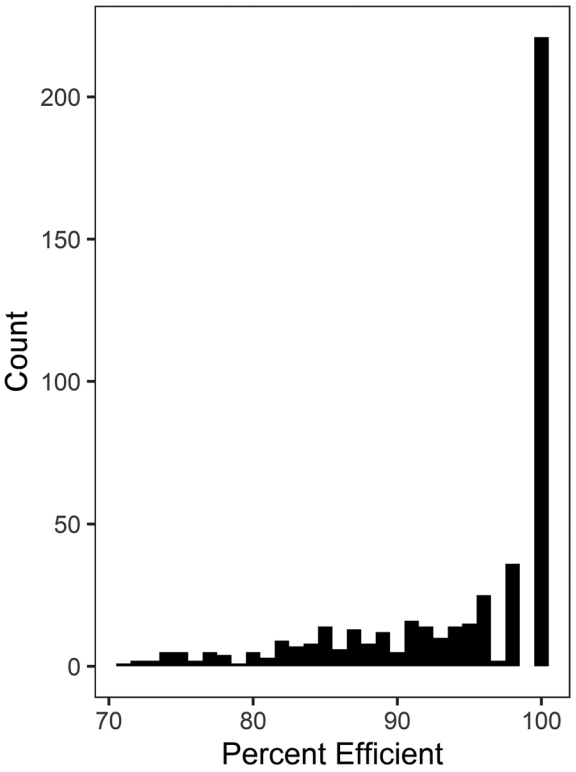

Fig. 5 Efficiency histogram of the base ensemble model — 221 of 470 models are 100% efficient

causing wait times. There is one bottleneck that causes higher single patient wait times relative to the total wait time, and it occurs relatively infrequently compared to the other, more common bottleneck. Therefore, to make users' lives better and improve the model, this bottleneck should be removed. Likewise, improvements to the primary care clinic should prevent the worst-case scenarios from happening rather than modifying the day-to-day operations of the clinic.

These model results somewhat corroborate the results found by Bobbie in her agent-based model of the Orlando VAMC. Their model has an average wait time per patient of 21.4035 min [1]. In contrast, this model has an average wait time per patient of 17.8 min, as shown in Table 4. Bobbie does not use an efficiency metric, nor a metric equivalent to *Longest Single Patient Wait Time*. It is possible to come close to replicating the results of an agent-based model with a network flow model.

4.2 Modifying the Topology of the Base Model

There are multiple ways to potentially remove the bottlenecks in the model, such as by hiring more staff or changing patient treatment techniques, it is also possible to remove bottlenecks by modifying the topology of the network. Since it is more difficult to turn away scheduled patients than unscheduled patients at the primary care

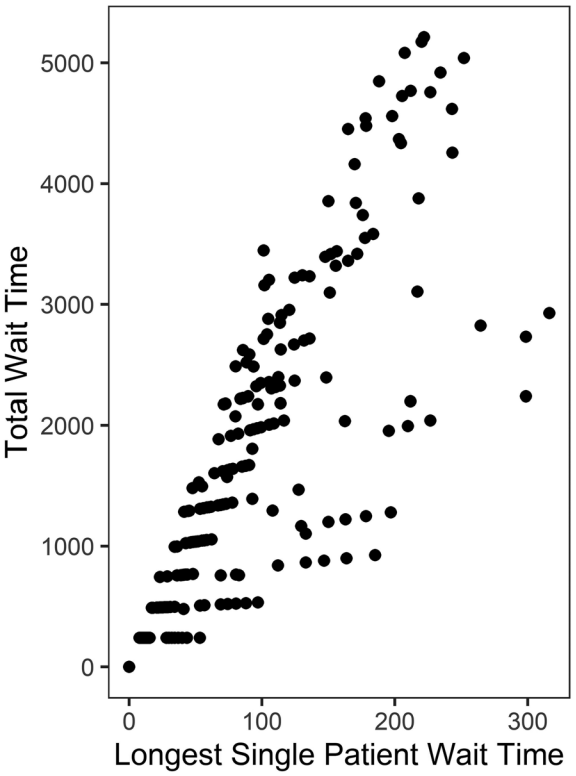

Fig. 6 The *Longest Single Patient Wait Time* is correlated with the total wait time for the base ensemble model ($R^2 = .8021$)

clinic, the topological change in this model will turn away unscheduled patients if the wait time at the PCP node is too long. To do this, an edge between the PCP node and the End node (as shown in Fig. 7) is added. While this method may not be feasible in reality, it is used here to illustrate the ability of topological changes to remove bottlenecks in the network. The results from an ensemble model of 500 runs appear in Table 5.

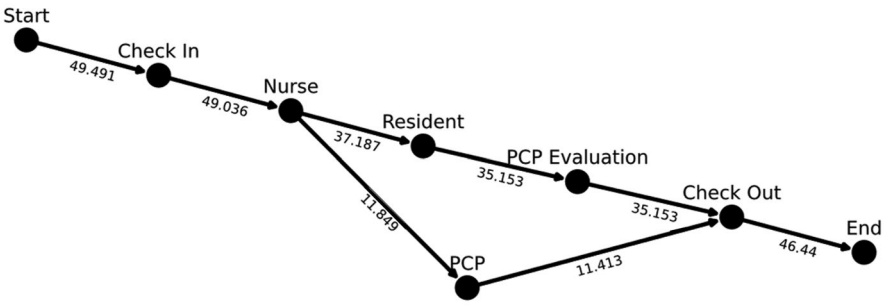

Fig. 7 Topology of the send home ensemble model. Node names are given above the nodes represented as circles, the capacities of the edges are given below the edges, and the arrows on the edges represent the direction of patient flow

Table 5 Send home ensemble model results for 500 runs

Metric	Mean	Extreme
Flow in	49.64 patients per day	67 patients per day
Flow out	47.67 patients per day	62 patients per day
Efficiency	96.5%	78.8%
Longest single patient wait	25.24 min	190 min
Total wait	542 min	3634 min
Average wait time per patient	10.9 min	54.2

Using the data provided in Table 5, the average efficiency of the send home ensemble model was 96.5%, compared to 94.39% for the base ensemble model. The minimum efficiency increased for the send home ensemble model, up to 78.8%. The *Total Wait Time* decreased from 879 to 542 min on average, and the *Longest Single Patient Wait Time* decreased to 25.24 min on average from 47.82 min. By removing the bottleneck at the PCP node, the metrics for patient wait time decreased and the overall efficiency of the network increased.

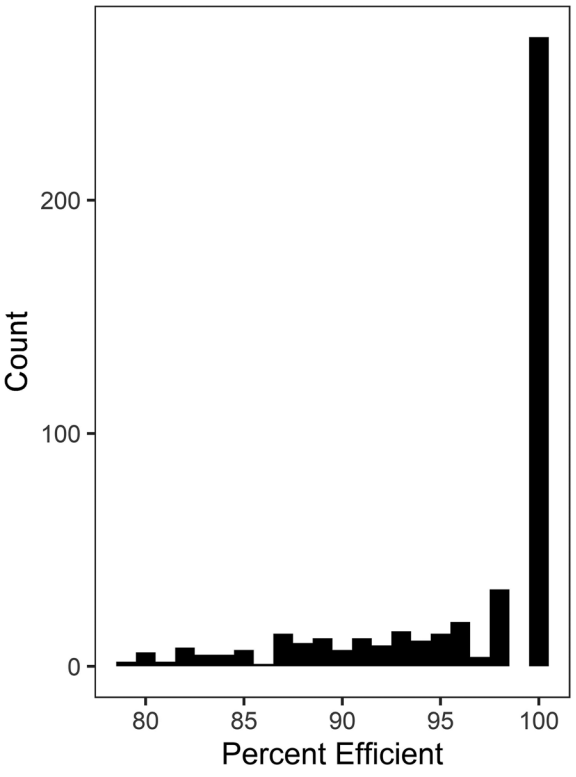

Fig. 8 Efficiency in the send home ensemble model — 270 of 466 models 100% efficient

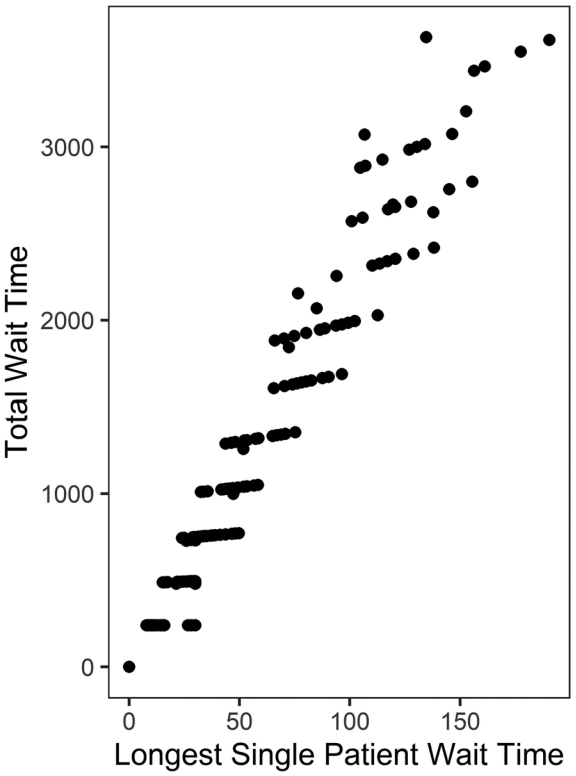

Fig. 9 The *Longest Single Patient Wait Time* is correlated with the *Total Wait Time* for the send home ensemble model ($R^2 = .9761$)

In Fig. 7, the topological change reduces wait times for many patients, at the expense of sending 1.779 patients home on average across the ensemble. While this is good for the patients receiving treatment, the patients who delay their care may end up with worse health outcomes. By making the topology change, the efficiency, *Longest Single Patient Wait Time*, and the *Total Wait Time* all significantly decreased. Looking at the histograms of the efficiency metric in Fig. 8, it appears that the data has become more skewed, increasing the chances for a model run with 100% efficiency to 270 of 466 models, but not mitigating the worst effects of the bottlenecks. Moreover, the system optimal wait time measurement decreases as the network becomes more user optimal. Meanwhile, in Fig. 9, the *Longest Single Patient Wait Time* and the *Total Wait Time* become more highly correlated compared to Fig. 6, indicating that most patients in the send home ensemble model experience the same bottleneck, and that this bottleneck appears relatively frequently. Looking at the capacities of the edges for Fig. 7, the Resident node shows the greatest decrease in capacities across the node. This suggests that the bottleneck at the PCP node is contributing to the high values for *Longest Single Patient Wait Time* for a select few patients. Modifying the topology of a network flow model results in improvements to the model metrics, decreasing patients wait times and increasing patient satisfaction.

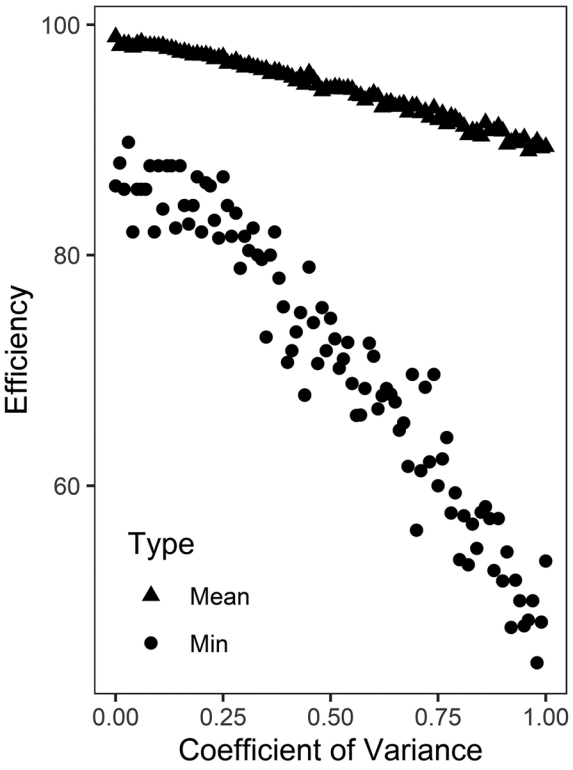

Fig. 10 The relationship between the minimum and mean efficiency and the coefficient of variance

4.3 Sensitivity Analysis

While modifying the topology of the network made the network more user optimal, inefficiencies and long wait times remain. Looking at Fig. 7, the Resident node appears to be the culprit due to the difference between the *FlowIn* and *FlowOut* remaining constant through both models. As mentioned previously, the doctors at the Resident node can treat 40 patients per day, but because of the sampling from the subspace, the number of patients treated at the Resident node is not constant. It is likely that the large standard deviation of the Poisson distribution gave unrealistic values for the number of patients treated at the Resident node in some cases. To test the changes on the constraints for sampling from the subspace, a third model is created where the underlying distribution is changed from a Poisson distribution to a normal distribution. A normal distribution enables fine-grained control over the standard deviation and mean, making it easy to manipulate. The standard deviation of each node's number of patients treated is derived from the Coefficient of Variance, which is varied between 0 and 1. When this coefficient of variance equals approximately 0.707, the standard deviations of the variables in this model are equal to the standard deviations of the variables in the base model. The mean for all sampled variables is constant between the

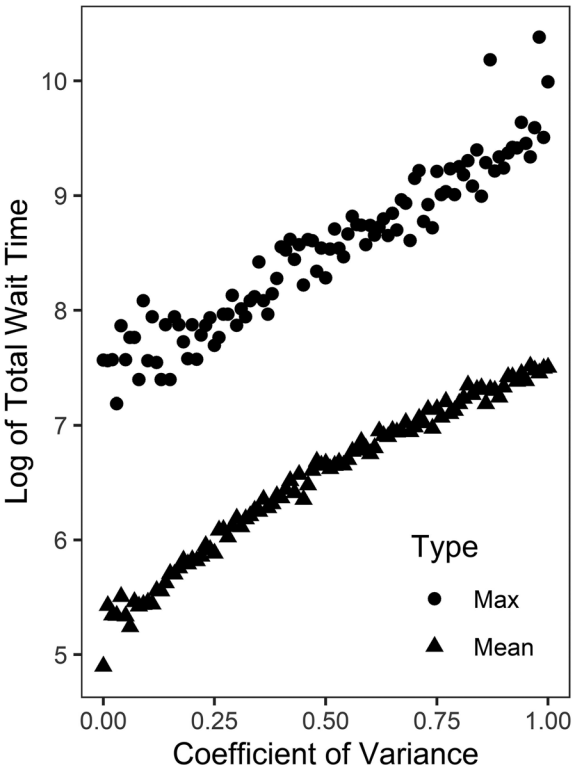

Fig. 11 The relationship between the log of the maximum and mean *Total Wait Time* and the coefficient of variance

models, and no runs are removed as outliers. The changes in metrics with respect to the coefficient of variance are shown in Figs. 10, 11, and 12.

The mean efficiency decreases almost constantly as the coefficient of variance increases, and the minimum efficiency steadily decreases towards 40% as the coefficient of variance approaches 1 in Fig. 10. The decrease in efficiency is constant, unlike the increases in Figs. 11 and 12, which, due to the log scale, increase at an exponential rate. The *Total Wait Time* increases exponentially in Fig. 11 as the Coefficient of Variance increases, and a similar pattern is seen in Fig. 12 for the *Longest Single Patient Wait Time*.

From Figs. 10, 11, and 12, it is highly likely that the model is sensitive to the parameters. To measure the sensitivity, Eq. (14) is used to model the sensitivity of the metrics to the Coefficient of Variance, Z_E, as used by Lee et al. [11]. The sensitivity is calculated by subtracting $MAX\left(E_{CV}\right)$, the maximum efficiency for a given coefficient of variance (CV), by $MIN\left(E_{CV}\right)$, the minimum efficiency for a given CV. This is divided by the expected value, or mean (μ), of efficiency.

$$Z_E = \frac{MAX}{CV} \frac{\left(MAX\left(E_{CV}\right) - MIN\left(E_{CV}\right)\right)}{\mu} * 100\% \tag{14}$$

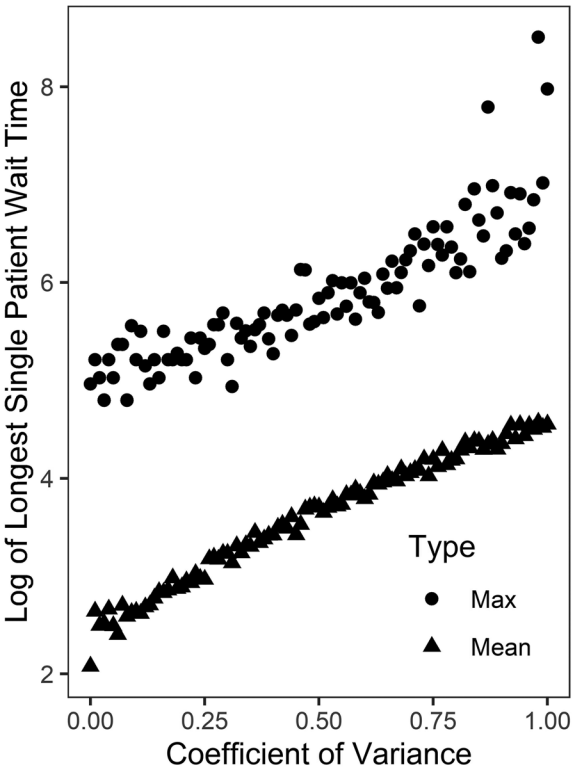

Fig. 12 The relationship between the log of the maximum and mean *Longest Single Patient Wait Time* and the coefficient of variance

Evaluating Eq. (14) results in a sensitivity of 61.6%, suggesting that the model is highly dependent upon its parameters and the distribution. This is much higher than Lee et al. [11] maximum sensitivity of 1.22%. The metrics used to evaluate the are sensitive to the coefficient of variance and mean of the variables, and caution is needed when building network flow models to ensure that the underlying standard deviations are accurate. If the subspace being randomly sampled is not modeled correctly, the ensemble model will be unable to provide useful results.

5 Closing Remarks

Synthetic data is generated via the random subspace method discussed in Sect. 2.3 and the data is used to populate the various ensemble models in generated in Sect. 3. Bottlenecks readily appear in network flow models, and in Sect. 4.1 they are identified and assessed for potential removal. Moreover, modifying the topology of the model to send walk-in patients' home when wait times are long partially removed a bottleneck at a node, leading to shorter wait times and increased

efficiency. In this paper, the topology modification is only an example; primary care centers should consult with domain experts before making modifications to their patient flow. After borrowing some model parameters from an agent-based model, the average patient wait times remained similar between the two models, indicating that network flow models can represent the same system effectively. As mentioned in Sect. 4.3, changing the constraints on the subspace by manipulating the standard deviation of random variables has a large effect on the efficiency of the model. Domain experts should verify that the data is randomly sampled from a viable subspace. Poorly chosen means and standard deviations for the underlying variables can cause highly inefficient and inaccurate model runs that do not reflect reality, potentially invalidating the results of the model.

A limitation of a network flow model is that it will not give any sense of what the "structural" wait time is in any primary care clinic. If every patient waits a constant 20 min before being treated, the model will not pick up that the wait time exists, because the rate at which new patients are being added to the line is equivalent to the rate at which they are being treated.

Furthermore, this model will not give a sense of what improvements could be made in real time to reduce the wait time of the network. In a crisis like COVID-19, where many more patients than average visit a primary care clinic, this model is unable to assist in mitigating patients' wait times in real time. In a crisis, the solution to the bottlenecks is almost always to hire more staff or get more equipment. A network flow model excels at detecting bottlenecks and giving a sense of how the efficiency changes with regard to a variety of conditions but should not be used in real time to make network topology changes.

In the future, it may be possible to apply a network flow model to a scenario where only the topology and the number of patients flowing into the model are known. This model is used to help the clinic reach a certain level of efficiency, and the idea shares several principles with bagging and training neural networks. The efficiency of the model is a function that depends upon the underlying variables while being subject to certain constraints and taking the gradient of this function yields the direction in which to shift the underlying variables to reach a sufficient level of efficiency. This more general solution to the problem is derived by considering a larger solution space and can reach a satisficing solution to the given problem. An alternative method for prioritizing metrics to yield a good enough solution in a healthcare context is discussed by Proano and Agarwal [19].

Funding This work was supported by the L.A. Comp Chair, the John and Mary Moore Chair at the University of Oklahoma, and the Start-up Fund and the Pietz Fund at South Dakota School of Mines and Technology.

Data Availability The data used in this work is taken from the dissertation by Bobbie [1].

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. Bobbie A (2017) A simulation-based evaluation of efficiency strategies for a primary care clinic with unscheduled visits. Dissertation, University of Central Florida. https://stars.library.ucf.edu/cgi/viewcontent.cgi?article=6262&context=etd
2. Su Q, Yao X, Su P, Shi J, Zhu Y, Xue L (2010) Hospital registration process reengineering using simulation method. J Healthc Eng 1:67–82
3. Akcali E, Côté MJ, Lin C (2006) A network flow approach to optimizing hospital bed capacity decisions. Health Care Manag Sci 9:391–404
4. Bean DM, Stringer C, Beeknoo N, Teo J, Dobson RJB (2017) Network analysis of patient flow in two UK acute care hospitals identifies key sub-networks for A&E performance. PLoS ONE 12:e0185912
5. Zawack DJ, Thompson GL (1987) A dynamic space-time network flow model for city traffic congestion. Transp Sci 21:153–162
6. Skurichina M, Duin RPW (2002) Bagging, boosting and the random subspace method for linear classifiers. Pattern Anal Appl 5:121–135
7. Fu Y, Banerjee A (2021) A stochastic programming model for service scheduling with uncertain demand: an application in open-access clinic scheduling. Oper Res 2(3): Article 43
8. Yang Y, Rajgopal J (2021) Outreach strategies for vaccine distribution: a multi-period stochastic modeling approach. Oper Res 2(2):Article 24
9. Box GE, MacGregor JF (1976) Parameter estimation with closed-loop operating data. Technometrics 18(4):371–380
10. Bhatnagar S, Patel S (2018) A stochastic approximation approach to active queue management. Telecommun Syst 68(1):89–104
11. Lee S et al (2021) A Markov chain model for analysis of physician workflow in primary care clinics. Health Care Manag Sci 24(1):72–91
12. Ajorlou S, Shams I, Yang K (2015) An analytics approach to designing patient centered medical homes. Health Care Manag Sci 18:3–18
13. Fletcher A, Worthington D (2009) What is a 'generic' hospital model?—a comparison of 'generic' and 'specific' hospital models of emergency patient flows. Health Care Manag Sci 12:374–391
14. Kawaguchi H, Tone K, Tsutsui M (2014) Estimation of the efficiency of Japanese hospitals using a dynamic and network data envelopment analysis model. Health Care Manag Sci 17:101–112
15. Cinaroglu S (2020) Integrated k-means clustering with data envelopment analysis of public hospital efficiency. Health Care Manag Sci 23:325–338
16. Berry M, Berry-Stölzle T, Schleppers A (2008) Operating room management and operating room productivity: the case of Germany. Health Care Manag Sci 11:228–239
17. Jiang F-C, Shih C-M, Wang Y-M, Yang C-T, Chiang Y-J, Lee C-H (2019) Decision support for the optimization of provider staffing for hospital emergency departments with a queue-based approach. J Clin Med 8:2154
18. Aslani N, Kuzgunkaya O, Vidyarthi N, Terekhov D (2021) A robust optimization model for tactical capacity planning in an outpatient setting. Health Care Manag Sci 24:26–40
19. Proano RA, Agarwal A (2018) Scheduling internal medicine resident rotations to ensure fairness and facilitate continuity of care. Health Care Manag Sci 21:461–474

Authors and Affiliations

Nathan Preuss[1,2] · Lin Guo[3] · Janet K. Allen[4] · Farrokh Mistree[4]

1 The School of Computer Science, The University of Oklahoma, Norman, OK, USA

2 The Department of Economics, The University of Oklahoma, Norman, OK, USA

3 Department of Industrial Engineering, South Dakota School of Mines and Technology, Rapid City, SD, USA

4 The Systems Realization Laboratory, The University of Oklahoma, Norman, OK, USA