Identification of optimal left-turn restriction locations using heuristic methods

By

Murat Bayrak
Department of Civil and Environmental Engineering
The Pennsylvania State University
201 Transportation Research Building
University Park, PA 16802
Phone: 814-863-1897
mub321@psu.edu

Vikash V. Gayah*
Department of Civil and Environmental Engineering
The Pennsylvania State University
231L Sackett Building
University Park, PA 16802
Phone: 814-865-4014
gayah@engr.psu.edu

***Corresponding Author**

July 2020

Word count: 6,600

ABSTRACT

Restricting left turns throughout a network improves overall flow capacity by reducing conflicts between left-turning and through-moving vehicles. However, doing so comes with the drawback of requiring vehicles to travel longer distances on average. Implementing these restrictions at only a subset of locations can help by balancing this tradeoff between increased capacity and longer trips. Unfortunately, identifying exactly where these restrictions should be implemented is a complex problem due to the very large number of configurations that must be tested and interdependencies between left-turn restriction decisions at adjacent intersections. This paper implements three heuristic solution algorithms—population-based incremental learning, Bayesian optimization and a hybrid of the two—to identify optimal locations of left-turn restrictions at individual intersections in a grid network. Scenarios are tested in which restriction decisions are the same for all intersection approaches and in which this decision is only the same for approaches in the same direction. The latter case is particularly complex as it increases the number of potential configurations exponentially. The results suggest all methods can be effectively used to solve this problem, though the population-based incremental learning method appears to perform the best in the more complex scenario. The proposed framework and procedures can be applied to realistic city networks to identify where left-turn restrictions should be implemented to improve overall network operations.

INTRODUCTION

Conflicting left turns represent a significant safety issue at signalized intersections. Left-turning vehicles have to cross the path opposing through vehicles to traverse the intersection and managing these left-turning and opposing through vehicle conflicts is a primary driver of Signal Phasing and Timing (SPaT) plans (*1*, *2*). Providing protected phases for left turns is the safest option as it eliminates these conflicts. However, this takes time away from through movements and introduces additional lost times during which the intersection is not serving any vehicles, both of which can reduce overall intersection capacity (*3*, *4*). Serving left turns in a permitted manner requires drivers to select appropriate gaps in which to move, which is less safe since the conflicts still exist. This might be more operationally efficient if sufficient gaps exist for left-turning vehicles to move; however, the left-turning vehicles might block other vehicles from discharging if they have to wait a long time for an appropriate gap (*5*). Dedicated left-turn pockets can be installed to mitigate this, but queued left-turn vehicles can still spillover and block vehicles in adjacent lanes from discharging through the intersection (*6*).

Several strategies have been proposed to mitigate these issues. Alternative intersection designs have been developed that allow left-turning vehicles in non-traditional ways. These strategies manage left-turn conflicts by using additional features (e.g., downstream U-turns or additional signals) and/or changing the intersection geometry (*7–13*). However, these designs are generally not well-suited for urban areas with limited road space since they require large spatial footprints or long blocks.

Instead, conflicting left turns can be simply restricted at signalized intersections. This simplifies the SPaT plan and allows the intersection to achieve higher overall flow capacities due to fewer change intervals and use of only through/right-turn phases. However, such restrictions will require vehicles that would have otherwise made a left turn to reroute, which may induce longer average travel distances. Several recent studies have examined the competing impacts of such left-turn restrictions enacted across entire grid networks (*14–18*). The studies generally agree that eliminating left turns can improve overall network operation (specifically, the rate that trips can be completed in the network), particularly when the network is operating near its capacity. However, these prior studies fail to consider the optimal spatial location of such restrictions at individual intersections within a network.

This type of problem is most closely related to general facility location problems in the transportation research literature, which are classified as NP-hard optimization problems due to the large solution space and lack of analytical solution (*19*). Within urban networks, many studies have proposed methods to determine optimal locations of treatments along individual links—e.g., optimal bus lanes locations (*20–23*)–or at individual intersections—e.g., optimal transit signal priority locations (*24*, *25*). Very few studies have examined the application of left-turn restrictions or alternative left-turn treatments at individual intersections on a network. The problem is complex as changes in left-turn treatments at one intersection might influence traffic patterns and operations at adjacent intersections. Additionally, vehicles will reroute themselves due to the left-turn restrictions enacted. These issues do not facilitate the expression of the optimal left-turn restriction problem as a traditional mathematical program. Instead, one study proposed a generic methodology to identify intersections that should restrict left turns that relied on fairly simplistic traffic models that did not account for traffic dynamics, such as queue spillbacks and changes in vehicle routing (*26*). Others utilized simulation to test the performance of a set of candidate left-turn restriction configurations but did not necessarily perform any optimization (*27*, *28*).

The purpose of this paper is to apply and compare several methods to determine the optimal spatial configuration of left-turn restrictions within a dense, grid-like urban network. Several heuristic algorithms are considered to identify candidate configurations, the performance of which are tested in a micro-simulation environment. The first is a population-based incremental learning (PBIL) algorithm that combines the evolutionary nature of genetic algorithms with competitive learning features (*29*). The second is a Bayesian optimization algorithm that seeks to directly learn and leverage dependencies inherent within the solution space using Bayesian networks (*30*). The final method is a hybrid of the two that uses the PBIL method to generate a set of candidate solutions and then implements a Bayesian network to generate similar solutions that might also perform well. The results suggest that these methods are generally suitable for the optimal spatial left-turn optimization problem and provide insights into the conditions under which each might perform best.

The remainder of this paper is organized as follows. First, the proposed optimization methods and experimental setup are described. Then, the results of the proposed algorithms are provided for two optimization scenarios that are considered. Finally, some concluding remarks are provided.

METHODOLOGY

This section describes the methods that were used in this paper to determine where left turns should be restricted spatially across an urban network. The remainder of this section describes the methods used to optimize the left-turn restrictions and the network-setup.

Optimization of left-turn restriction configurations

The methodology used in this study can be classified as a bi-level optimization algorithm. The upper level involves the selection of left-turn restriction locations, the lower level simulates the traffic for the selected left-turn restriction locations and estimates the total travel time of the network. The next two subsections describe the methods used for the network evaluation and the selection of left-turn restriction locations.

The lower level network assessment

The lower of the bi-level optimization algorithm aims to calculate total travel time for a given left-turn restriction configuration. For this purpose, an accurate method is needed to capture routing behavior of drivers, the effects of network elements (e.g., traffic signals), and the related traffic phenomena (e.g., queue spillbacks, heterogeneous driver behavior). Due to the accuracy needed for travel time calculation, the Aimsun micro-simulation platform is selected for network assessment (*31*).

The stochastic c-logit route choice model is used for vehicle routing within the micro-simulation environment. This model emulates a user-equilibrium routing pattern in which drivers select the route to minimize their own expected travel time. The drivers make the routing decision when they initiate the trip and the decisions are based on the average travel time on links over the past 3 minutes. However, a portion of the drivers (50% in the tests performed in this paper) are allowed to change their route during their trip based on the current traffic situation. The rerouting occurs at regular intervals of 3 minutes. Previous research has shown that such adaptive traffic routing algorithms provide better network-wide operational performance (*32*).

The upper level optimization

The upper level optimization procedure seeks to identify the optimal configuration of left-turn restrictions—i.e., the locations at which left-turn restrictions should be enacted to optimal traffic performance. Here, traffic performance is measured using total travel time that is experienced by all vehicles during a simulated peak period. Three different optimization algorithms are considered for this upper level optimization. These methods are selected due to their capability of accounting for spatial dependencies in the left-turn restriction decisions at individual locations.

Population-based incremental learning

Population-based incremental learning combines the generational evolution of genetic algorithms with competitive learning (*29*). The flowchart of the PBIL algorithm is shown in FIGURE 1. The algorithm has four basic steps that will be described in this section: 1) initiation, 2) generation and evaluation, 3) mutation and update of the probability vector, and 4) termination.

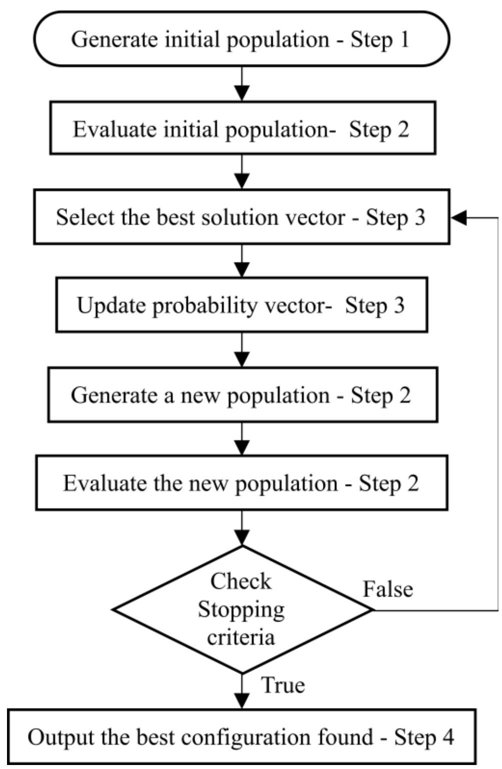


FIGURE 1. Flow chart of population-based incremental learning

The initiation step involves generating an initial probability vector, P^1, that serves as a starting point for the algorithm. Each element i of this vector is associated with a specific intersection (or intersection approach) and the corresponding value in the initial probability vector, P_i^1, represents the probability that left-turn restriction is applied at that intersection (or approach). Thus, the size of the probability vector is equal to the number of candidate locations that left-turn restrictions are considered for implementation. In this paper, the probability vector is initiated by setting the value of each element to 0.5. This represents a truly random guess with no prior knowledge on if left-turn restrictions should be applied at each candidate location.

The generation and evaluation step generates a population of candidate left-turn restriction configurations (60 individual configurations in this paper) by randomly sampling the solution space using the probability vector associated with the current generation, P^t. Each candidate configuration is represented by a vector of the same size as P^t and consists of binary values, where the value 1 represents a left-turn restriction and the value 0 represents no left-turn restriction. After the set of candidate configurations are generated, each is evaluated by the lower level to obtain the total travel time value associated with each configuration.

After the evaluation, the probability vector is updated by learning from the best solution vector (i.e. the configuration with the lowest total travel time) in the current generation, B^t, and the worst solution in the current generation. The former adjustment (1) represents positive learning, while the latter (2) represents negative learning**.**

$$P_i^{t+1} = \left(P_i^t \times (1 - LR^+)\right) + (B_i^t \times LR^+) \tag{1}$$

$$P_i^{t+1} = P_i^t \times (1 + LR^-) - w_i^t \times LR \tag{2}$$

where, P_i^t is the probability of LTs at a location i in generation t, LR^+ is the positive learning rate, LR^- is the negative learning rate, and B_i^t is the value (0 or 1) of the ith position of the best solution found in generation t, and w_i^t is the value of the ith position of the worst solution found in generation t. Equation (1) essentially adjusts the probability vector so that future solutions that are evaluated are more likely to adopt features of the best solution found in the current generation, while Equation (2) essentially adjusts the probability vector so that future less likely to adopt features of the worst solution found in the current generation.

Similar to other evolutionary algorithms, PBIL converges around a solution as the search progresses. However, PBIL allows explicit control of the speed of convergence with the learning rate parameters, LR^+ and LR^-. The learning rate parameters enables the PBIL to explore a larger portion of the solution space, which is essential for problems with dependencies, before starting to converge to a solution. The LR values seek to create a balance between the portion of the solution space explored and the convergence speed. In this study, we apply $LR^+ = 0.1$ and $LR^- = 0$. The latter is selected as negative learning might restrict the search for optimal solutions in some cases.

In addition to the learning process, the mutation operator is also responsible for expanding the explored solution space by randomly changing the probabilities in P^t by some magnitude of $\Delta m = 0.05$. Mutation is applied with a predefined random probability $m = 0.02$, known as the mutation rate. Each element in the solution vector is randomly updated according to (3) with probability m.

$$P_i^{t+1} = P_i^t \times (1 - \Delta m) + \Delta m \tag{3}$$

After the probability vector is updated using (1-3), steps 2 and 3 are repeated until the termination criteria is met. The termination criteria can be based on convergence of probability vector or maximum number of generations. In this study, the PBIL algorithm is terminated after 60 generations.

Bayesian optimization algorithm (BOA)

PBIL relies on random sampling of potential solutions and learning about the performance of these sampled solutions to generate better solutions. Unlike PBIL, the BOA does not assume an independent relationship between decision variables (e.g., locations of left-turn restrictions). BOA aims to learn the dependency structure between decision variables using Bayesian networks and uses this information to generate better solutions. The Bayesian networks represent the dependencies between left-turn restriction decisions at individual intersection. Each node of a Bayesian network corresponds to a possible left-turn restriction location, and each directed edge of a Bayesian network represents a dependent relationship between locations of left-turn restrictions. The flowchart of the BOA is shown in FIGURE 2.

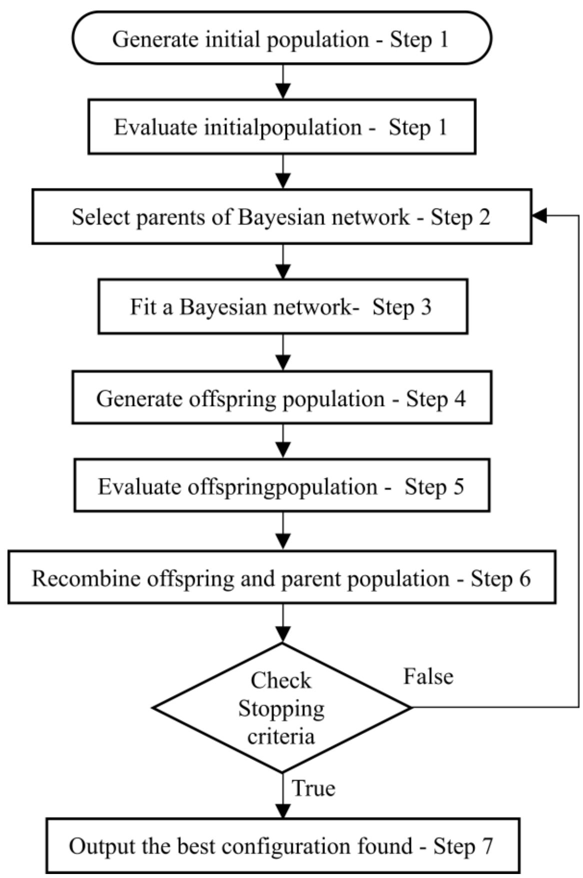


FIGURE 2. Flow chart of Bayesian optimization algorithm

The BOA has 7 steps: 1) initiation, 2) selection, 3) Bayesian network construction, 4) offspring creation, 5) evaluation, 6) recombination, and 7) termination. The algorithm starts by randomly generating an initial population (in this case, 60) individual left-turn restriction configurations. Similar to PBIL, each solution vector is a binary vector that represents the left-turn restriction at each location. After the initial population is generated, each individual solution is evaluated by the lower level to compute the total travel time associated with each left-turn restriction configuration.

The selection step selects the parent solutions that will be used for Bayesian network construction. A tournament selection method is used (*33*) in which two left-turn restriction configurations are randomly selected from the initial population and the better performing configuration of the two options is identified as a parent. The process repeats itself until a pre-specified number of parents are determined (30 in this paper), where repetition is allowed (i.e., a particular solution can be used as a parent multiple times if randomly selected).

A Bayesian network is then constructed using a greedy algorithm. The procedure used to construct the Bayesian network is essentially a separate optimization algorithm inside the BOA. The procedure starts with a Bayesian network with no edges (i.e., a network with no dependencies). Then, basic graph operations (edge addition, removal, and reversal) are applied to a random node pair. The operation that increases the quality of the Bayesian network most is kept and the others are discarded. These two steps (testing and selecting operations) are repeated until the network can no longer be improved. Notice that only one graph operation is performed at each cycle of the Bayesian network construction process. For the selection of the graph operation that is applied, the quality of the Bayesian network is assessed with a scoring metric. In this study, the Bayesian information criterion (BIC) is used as a scoring metric (*34*). The BIC assumes that the number of dependencies in the network is proportional to the amount of compression of the data allowed by the Bayesian network. Therefore, a Bayesian network structure that maximizes the BIC metric can describe the dependencies.

Next, offspring solutions are generated by sampling the fitted Bayesian network. In the Bayesian network, each variable (i.e., the of left-turn restrictions at a given intersection) can be categorized into one of three groups: 1) completely independent (i.e., no links are formed in the Bayesian network); 2) dependent on others; and, 3) others are dependent on this variable (i.e., the value of the variables in group 2 depend on the value of the variables in group 3). In order to sample from these sets of variables, a forward simulation process is used (*35*). This sampling is done based on the conditional probabilities encoded in the Bayesian network, by assigning first the value (0 – no left-turn restrictions, or 1 – left-turn restriction) of independent variables (those in group 1), next the value of variables in group 3 (since the values of these do not depend on other variables) and finally by assigning the values of variables in group 2 (since their values depend on the values of the variables in group 3). The sampling process is repeated until 30 offspring solutions are generated.

After the offspring solutions are generated, they are evaluated by the lower level. After the evaluation step, the offspring solutions and the previous population are recombined by replacing the worse half of the population with the offspring solutions. Similar to PBIL, BOA is run for 60 generations. The steps 2-6 are repeated until enough number of generations are evaluated.

PBIL-BOA hybrid algorithm

Both BOA and PBIL are capable algorithms to solve complicated problems but have several limitations. PBIL samples configurations from different parts of the solution space due to its exploration focus, but this makes the PBIL algorithm slow to converge to a point in solution space (*29*). BOA is faster to converge to a solution due to its focus on learning dependencies between decision variables, but requires a large population to accurately learn the dependencies between decision variables for problems with a large number of dependent decision variables. When the solution evaluation is computationally intensive (e.g., like in the micro-simulation approach that is applied here) large number of evaluations may not be feasible. Thus, in this study, a hybrid method that combines the exploration capability of PBIL and dependency learning capability of BOA is proposed.

The hybrid algorithm has three main steps. First, generate a large set of promising solutions (i.e., a set of left-turn restriction configurations that has good total travel time values). Second, use this large set of solutions to create a Bayesian network. Finally, use the constructed Bayesian network to generate new solutions. The flowchart of the Hybrid algorithm is shown in FIGURE 3.

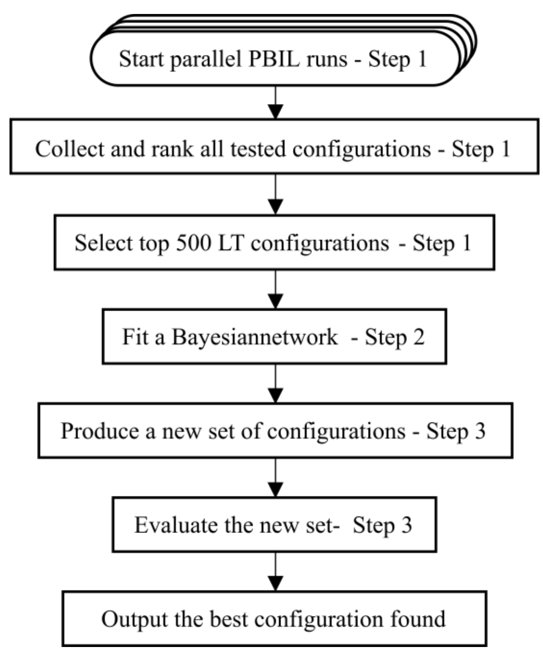


FIGURE 3. Flow chart of PBIL-BOA hybrid optimization algorithm

The first step involves shorter independent PBIL runs with small populations. The aim of this step is not to necessarily find the global optimum solution. Instead, the aim is to identify promising solutions from different parts of the solutions space. Thus, several PBIL runs with different initial populations are used to identify promising solutions. To select the sample that will be used for Bayesian network construction, all configurations evaluated during the independent PBIL runs are ranked, and the top 500 unique configurations are selected for Bayesian network construction.

The selected sample of promising configurations are then used to construct a Bayesian network. The algorithm used for Bayesian network construction is similar to the one used for BOA algorithm. Since a much larger population (500 unique configurations) is used in the hybrid algorithm to create the Bayesian network compared to BOA's procedure, the hybrid algorithm is more likely to capture dependencies between decision variables. After the Bayesian network creation, a new set of solutions is generated by forward simulation, similar to BOA. Finally, the new set of solutions are evaluated by the lower level to estimate total travel time values.

Experimental setup

The methodology described above is tested using an illustrative network shown in FIGURE 4. The test network is a 8x8 grid. The block length is 200 meters. Each link is coded as an arterial in Aimsun and has two lanes per direction and the speed limit and capacity of links are 48 km/h and 1600 veh/h. Every intersection in the network is a signalized intersection with 90-sec cycle length without any offset between adjacent intersections. During each cycle, each direction receives 42 seconds of green time with three seconds of change interval is applied between consecutive phases. Left turns are either set to permitted operation during each phase or restricted altogether. When permitted, left-turning vehicles waiting for a gap at the intersection tend to block one lane and reduce overall intersection capacity. Origin and destination nodes are located in the middle of each block and 32 entry/exit points on the periphery of the network. A uniform demand pattern in which each origin generates the same number of trips (on average) and each destination is equally like to be selected is used. The simulation run using an average of 367 total trips per minute. The selected demand level is enough to saturate the network when LTs are permitted at all intersections. However, due to the randomness of Aimsun's simulation process, there are small variations in the uniform demand pattern in each simulation.

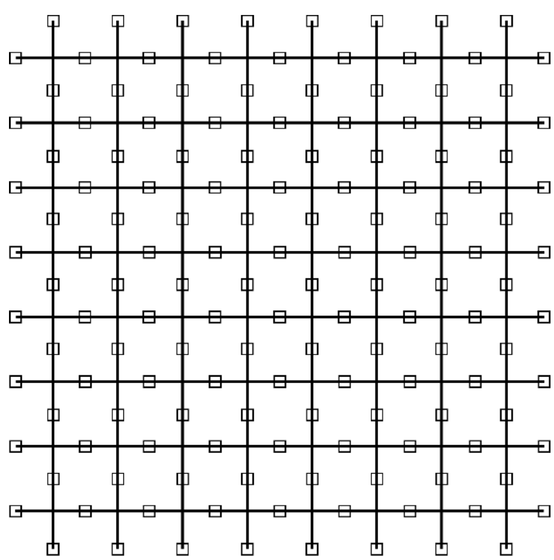


FIGURE 4. Test network and OD locations

The BOA, PBIL, and hybrid algorithms described in the methodology section are tested on two different optimization scenarios related to how left-turn restrictions are implemented at individual intersections. The first scenario considers the case where a single left-turn restriction is made for each intersection. In this case, the same decision (restrict or allow left turns) is made at all intersection approaches at a given intersection. In this scenario, there are 64 possible left-turn restriction locations for this scenario. The second scenario considers the case where multiple left-turn restriction decisions are made at each intersection. Specifically, left-turn restrictions decisions are made independently at each intersection for each of the two competing directions (e.g., north/south and east/west) at each intersection. In this scenario, there are two left-turn restriction decisions that can be made for each of the 64 intersections for a total of 128 binary decisions.

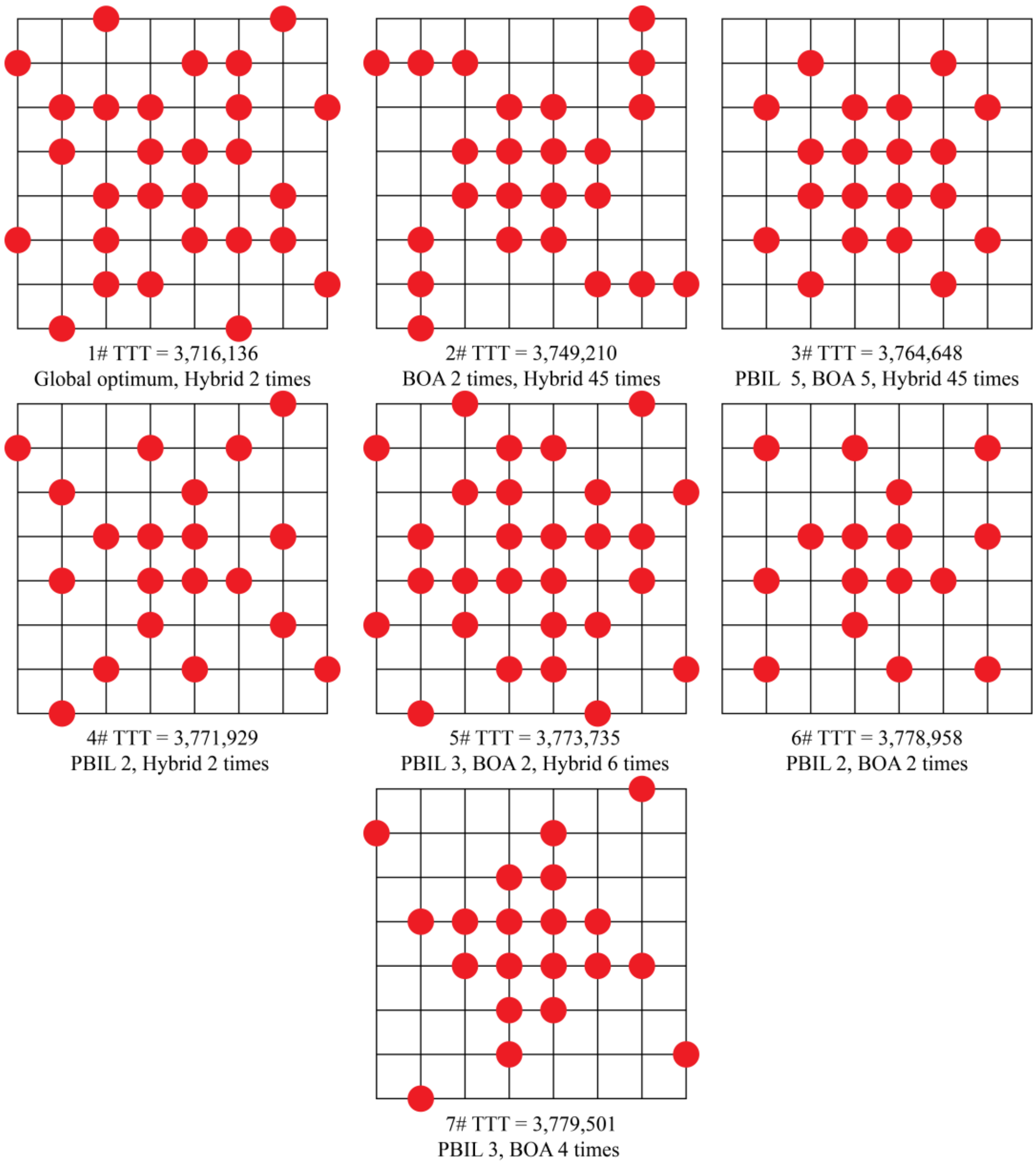
Given that the number of possible left-turn restriction configurations increases exponentially with the number of possible restriction locations, the number of possible left-turn restriction configurations for both scenarios is extremely large (2^{64} in the first scenario, $(2 \times 2)^{64}$ in the second). To reduce the size of the solution space, several constraints are implemented. First, a symmetry constraint is implemented. Since the network is a perfect grid with a uniform demand pattern, the global optimum solution is likely to be rotationally symmetrical. For both scenarios, the upper-level algorithms only search for left-turn restriction configurations that are rotationally symmetrical around the center of the network. The second constraint is left-turn restrictions are never applied at the intersections at the corners of the network to always provide a feasible path between all OD pairs. These two constraints reduced the number of possible left-turn restriction configurations to 32,768 and 1.07×10^9 for first and second scenarios, respectively. For the first scenario, the reduced number of possible left-turn restriction configurations is low enough to use brute-force enumeration to determine the global optimal configuration.

RESULTS

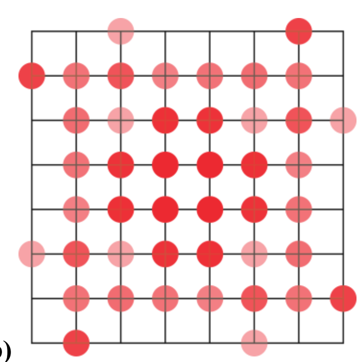
This section describes the results of the optimal left-turn restriction configuration problem using the three proposed methods. First, the results of the first scenario with only one restriction decision per intersection are provided. Then, the results of the second scenario with two decisions per intersection are provided.

Scenario 1: One left-turn restriction decision per intersection

The PBIL, BOA, and hybrid algorithms were run multiple times to obtain best-performing solutions. Multiple runs of each algorithm were performed since they are heuristic methods that rely on random processes, particularly the first set of randomly generated solutions. The PBIL and BOA algorithm were each run 15 times since they were the most well-known of the three methods tested but also the most computationally intensive. The hybrid optimization procedure was repeated 100 times to learn more about how well it is able to improve upon the PBIL and BOA methods. The sample used for Bayesian network construction contains the best 500 left-turn restriction configurations found during a subset of independent PBIL runs. At each instance of this algorithm, a Bayesian network that represents the dependencies between left-turn restriction locations was constructed by using these 500 configurations. Using the constructed Bayesian network, 500 new configurations are generated. Since Bayesian network construction and new configuration generation are heuristic processes, each run of the hybrid algorithm created a different set of new solutions. The best solutions found in each instance of the Hybrid algorithm is presented below. The results of the optimal configurations obtained from these methods are provided in FIGURE 5a. From the 130 total optimization runs, only seven unique left-turn configurations were identified: five unique configurations were found using the PBIL approach, five unique configurations were found using the PBIL approach, and four of the configurations are common across the two methods. In addition, to these heuristic methods, a brute-force enumerate approach was applied to identify the global optimal solution. In this method, all 32,768 potential configurations were tested to determine the configuration with the lowest travel time.



(a)



(b)

FIGURE 5. Results of Scenario 1. (a) Optimal left-turn restriction configurations obtained by proposed algorithms; and, (b) common locations of left-turn restrictions identified in top-performing algorithms.

The implementation of left-turn restrictions at a subset of intersections using any of the methods reduces travel time significantly compared to implementing left-turn restrictions at no intersections (at least 3.8% reduction) or at all intersections (at least 12.3% reduction). Note that accommodating left turns at all intersections outperforms restricting left turns at all intersections due to the low level of congestion in the network. Under these conditions, it is not necessary to prohibit left turns everywhere since some intersections are under-saturated.

All configurations obtained from the proposed optimization methods and the global optimal configuration are presented in FIGURE 6. In this figure, red dots represent intersections at which left-turn restrictions are applied. Note that only seven unique configurations were identified: five using PBIL, five using the BOA, and five using the hybrid approach. The results reveal that all methods are able to identify a solution that provides travel times within 1.7% of the global optimal solution; i.e., the maximum optimality gap of any solution identified was 1.71%. Comparing the three methods, the average optimality gap of the PBIL, BOA and hybrid methods are 1.51%, 1.44%, and 1.11%, respectively. Overall, this suggest strong performance across the three optimization methods. However, the hybrid method appears to provide better results. For one, it has the smallest average optimality gap. In addition, it is the only method to identify the global optimal solution (though this was just found 2% of the times and can be attributed to randomness). Still, was able to outperform or equal the PBIL method in 92% of runs and outperform or equal the BOA method 47% of runs. A comparison of these methods by TTT produced is illustrated in FIGURE 6.

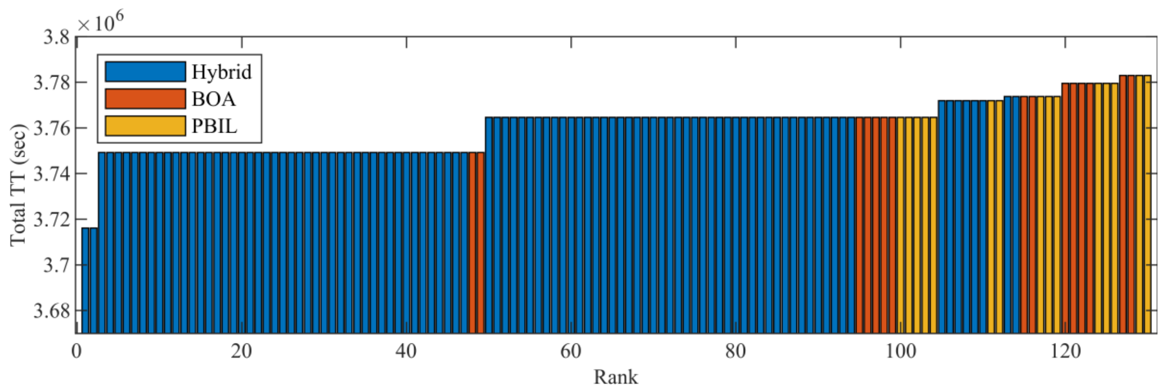


FIGURE 6. Ranking of best-performing left-turn restriction configurations identified (Scenario 1)

A review of these configurations suggests some common features about where left-turn restrictions should be enacted. FIGURE 5b overlays the configurations so that darker points represent intersections at which left-turn restrictions are applied more commonly in the configurations shown in FIGURE 5a. Notice that left-turn restrictions in the best-performing solutions are more likely in the central portion of the network. This is reasonable as these locations serve the highest flows (combined across all approaches) in the network. Restricting left turns at these locations increases the capacity at these locations, which helps to improve overall network operations. Although vehicles will have to reroute when left turns are restricted, the central portion of the network has many alternative routes that can be used for those vehicles wishing to make left turns. When left turns are restricted at these central locations, this rerouting can often occur without additional incurred travel distance. By contrast, left-turn restrictions are less likely on the periphery of the network. This is reasonable as these intersections generally carry lower total flows so the additional capacity gained by restricting left turns is less beneficial. Furthermore, these intersections experience a higher proportion of vehicles turning left so left-turn restrictions would cause more vehicles to reroute. Finally, left-turn restrictions at these periphery locations are more likely to induce additional travel distance due fewer routing options available at the edge of the network.

Scenario 2: Two left-turn restriction decisions per intersection

The optimization procedure was then repeated for the scenario in which two left-turn restrictions decisions are made per intersection. This is a much more challenging problem as there are 32,768 times more feasible configurations. Because of this, a brute-force enumeration is not possible and thus, the global optimal configuration is not known in this scenario. As with Scenario 1, the PBIL, BOA and hybrid methods were repeated multiple times. However, the number of times were reduced due to the computational burden required. Thus, the PBIL method was performed four times, the BOA method was performed four times, and the hybrid method was performed 35 times.

Comparing the results of this scenario with Scenario 1, we find that separating the left-turn restrictions between the north/south and east/west approaches provides more efficient network operations (i.e., reduced travel time). This is expected since having more options is more flexible and left-turn restrictions do not necessarily need to be enacted simultaneously at all intersection approaches. Overall, travel time is found to be reduced by an additional 1.25% (worst-performing configuration identified) to 2.38% (best-performing configuration identified) over the global optimal solution in Scenario 1. This represents a reduction in total travel time over not implementing any left-turn restrictions at any intersections of 7.7% and a reduction in total travel time over implementing left-turn restrictions at all approaches of all intersections of 15.8%. All methods were always able to outperform the best-performing configuration identified in Scenario 1, which suggests that these methods are still able to improve upon the problem in this more flexible case even though the solution space is much larger. FIGURE 7 provides a graphical comparison of the configurations identified using each method.

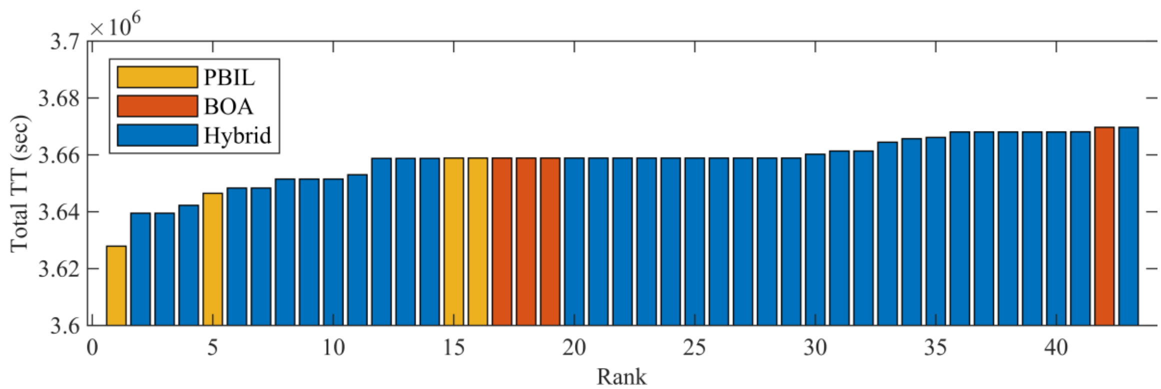
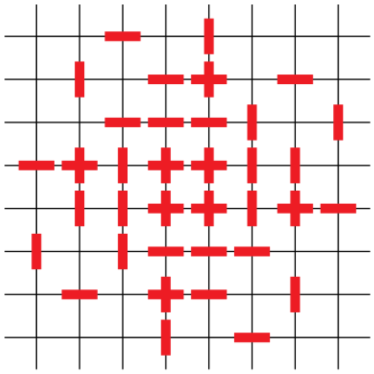
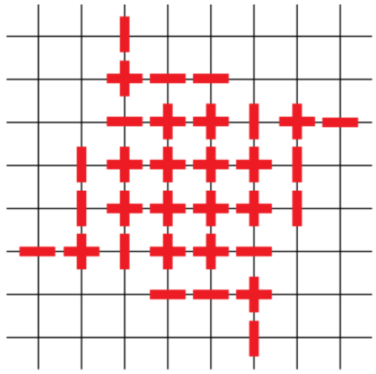


FIGURE 7. Ranking of best-performing left-turn restriction configurations identified (Scenario 2)

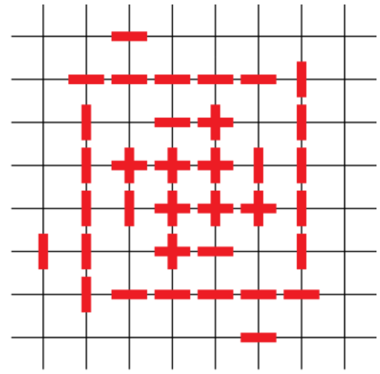
The unique configurations that were identified are shown in FIGURE 8a. In this figure, red vertical or horizontal lines represent left-turn restrictions in the north-south or east-west direction, respectively, implemented at a specific intersection. As shown, only 17 unique configurations were identified. The PBIL was able to provide the best-performing configuration, while the configurations identified using the BOA and hybrid methods performed a bit worse. A review of these configurations again reveals common features about where left-turn restrictions should be enacted. FIGURE 8b overlays the configurations so that darker lines represent approaches at which left-turn restrictions are applied more commonly in the configurations shown in FIGURE 8a. A very clear pattern of common left-turn restriction locations emerges from this overlapping figure. The general pattern is similar to Scenario 1: left-turn restrictions occur frequently in the central portion of the network and generally do not occur in the periphery of the network. These reasons are the same as those in the previous scenario. However, there are some key differences. Left turns are generally permitted at approaches that lead from the central portion of the network toward the periphery of the network, whereas they are restricted at those approaches leading toward the central portion of the network.



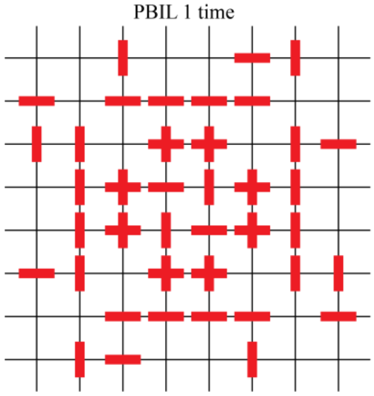
1# TTT = 3,627,872
PBIL 1 time



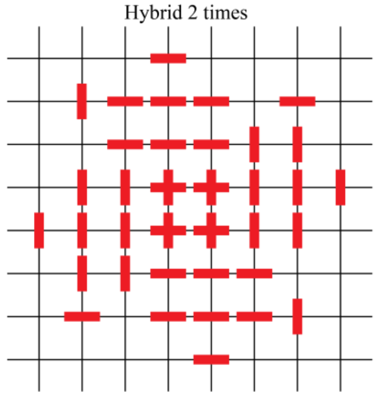
2# TTT = 3,639,460
Hybrid 2 times



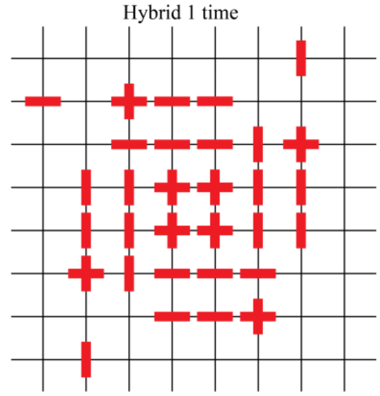
3# TTT = 3,642,241
Hybrid 1 time



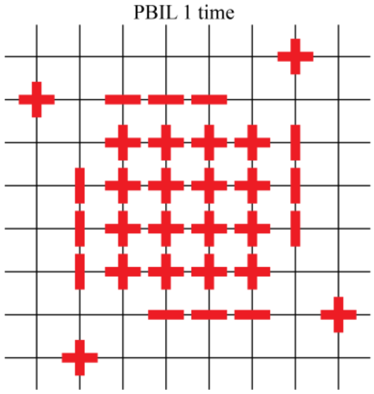
4# TTT = 3,646,469
PBIL 1 time



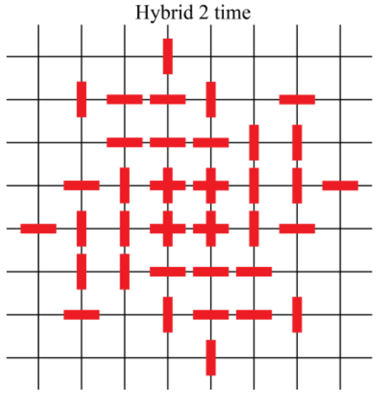
5# TTT = 3,648,314
Hybrid 2 time



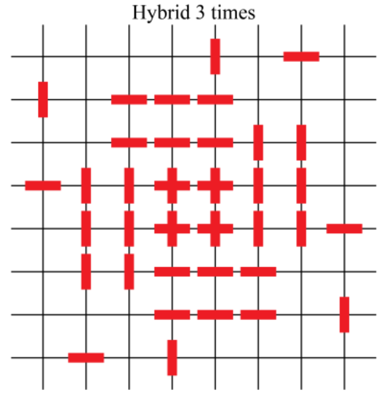
6# TTT = 3,651,474
Hybrid 3 times



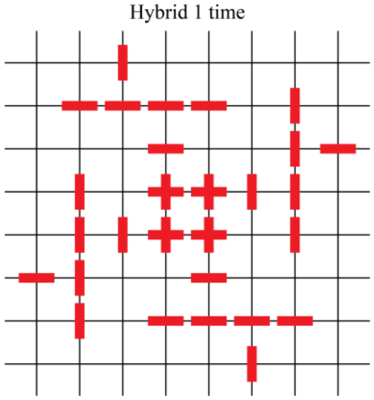
7# TTT = 3,652,978
Hybrid 1 time



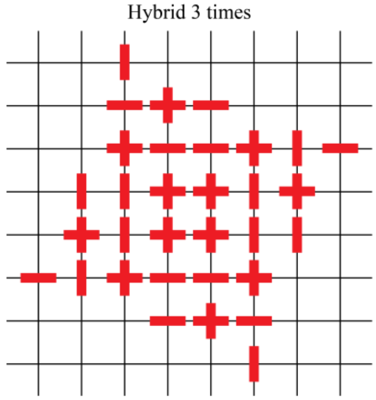
8# TTT = 3,658,715
Hybrid 3 times



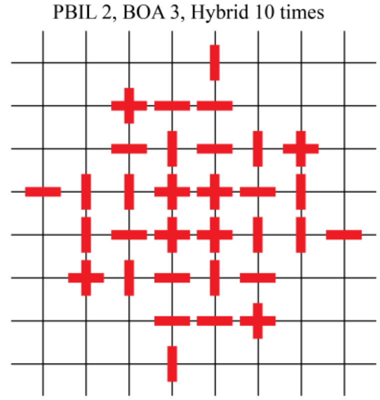
9# TTT = 3,658,822
PBIL 2, BOA 3, Hybrid 10 times



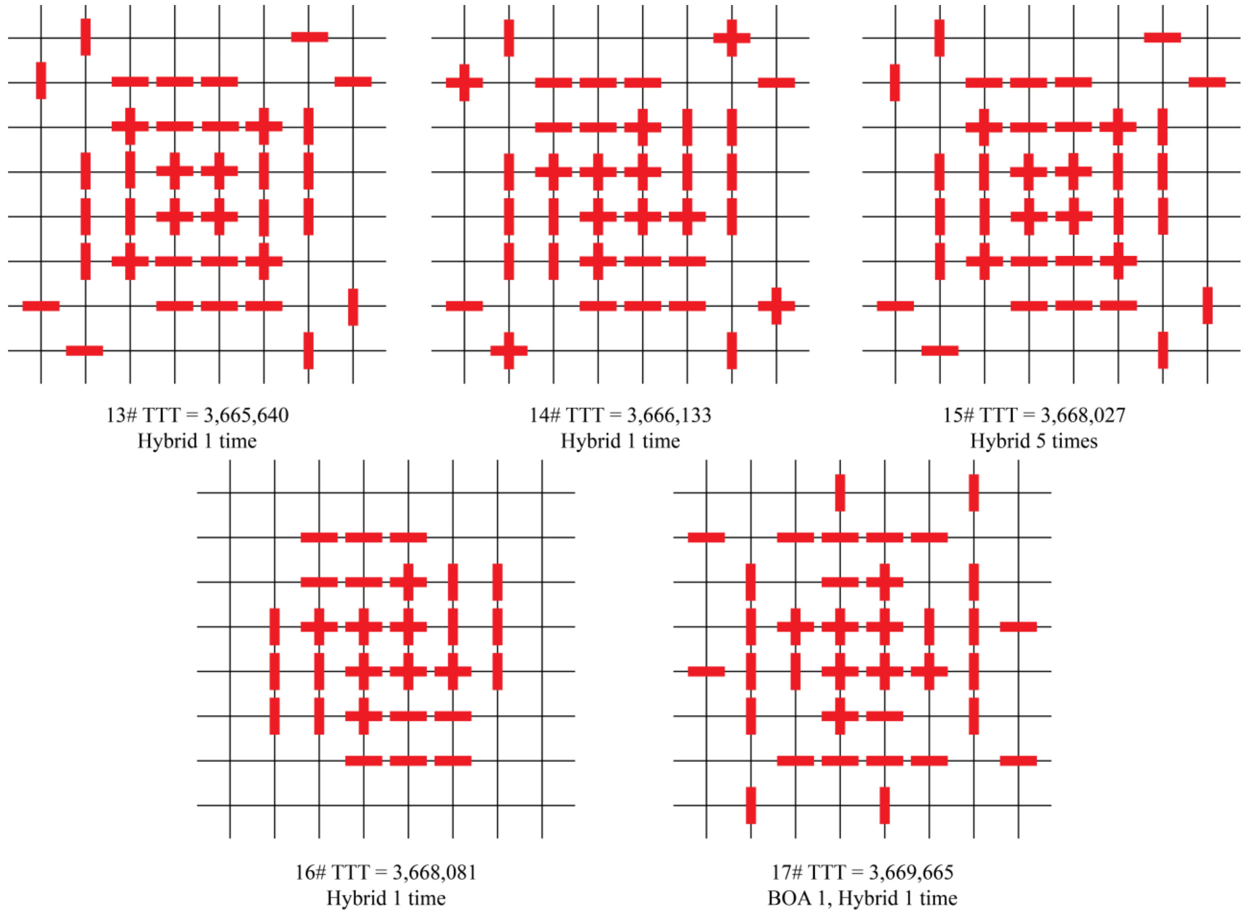
10# TTT = 3,660,201
Hybrid 1 time



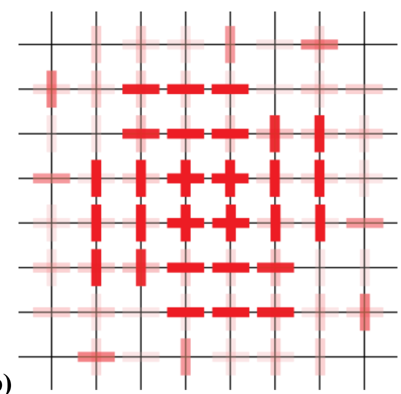
11# TTT = 3,661,335
Hybrid 2 times



12# TTT = 3,664,448
Hybrid 1 time



(a)



(b)

FIGURE 8. Results of Scenario 2. (a) Optimal left-turn restriction configurations obtained by proposed algorithms; and, (b) common locations of left-turn restrictions identified in top-performing algorithms.

DISCUSSION AND CONCLUDING REMARKS

This paper has extended recent work on the impact of restricting left turns at signalized intersections. It specifically considers the identification of which intersections left-turn restrictions should be implemented at to maximize overall network efficiency, measured by the total travel time for all vehicles on the network. Two scenarios are considered: one where the same left-turn restriction decision is made for all approaches at an intersection and another where two decisions are made (one for the north-south approach and another for the east-west approach). Both are complex combinatorial problems with an incredibly large solution space. To determine optimal left-turn restriction configurations, three heuristic methods are compared: a population-based incremental learning algorithm, a Bayesian optimization algorithm and a hybrid of the two.

The results reveal that all three methods are fairly reasonable for solving this problem and identifying a left-turn configuration that reduces total travel time within the network. In general, the population-based incremental learning algorithm performs slightly better than the other two methods, particularly in the second scenario where multiple left-turn restriction decisions are made at each intersection. However, the three methods provide configurations with consistent and similar features. Namely, that left turns should be restricted at intersections in the inner portion of the network that carry the largest vehicle flows, and left turns should be allowed at the intersections in the periphery where flows are low and the proportion of vehicles making a left turn is higher. In general, the two-decision-per-intersection case always provides lower travel times than the one-decision-per-intersection case. While this is expected because it is more flexible, the fact that better-performing configurations can be obtained even though the solution space is exponentially greater is promising.

The results provide a general framework for how to implement such decisions at more complicated and realistic network structures. While the actual configuration on any network would be subject to its network-specific features and demand pattern, the overall pattern that were observed here should be fairly general and serve as a good starting point that could be refined due to network-specific features. Further work should also consider how these central left-turn restrictions might influence overall network resilience to disruptions that might occur along links, such as traffic crashes or bottlenecks caused by freight vehicles.

ACKNOWLEDGEMENTS

This research was supported by NSF Grant CMMI-1749200

AUTHOR CONTRIBUTION STATEMENT

The authors confirm contribution to the paper as follows: study conception and design: MB, VG; analysis and interpretation of results: MB, VG; draft manuscript preparation: MB, VG. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. Roess, R. P., E. S. Prassas, and W. R. McShane. *Traffic Engineering, 4th Edition*. 2010.

2. Cottrell, B. H. Guidelines for Protected/Permissive Left-Turn Signal Phasing. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1069, 1986, pp. 54–61.

3. Messer, C. J., and D. B. Fambro. Effects of Signal Phasing and Length of Left-Turn Bay on Capacity. *Transportation Research Record: Journal of the Transportation Research Board*, No. 644, 1977, pp. 95–101.

4. Newell, G. F. The Effect of Left Turns on the Capacity of a Traffic Intersection. *Quarterly of Applied Mathematics*, Vol. 17, No. 1, 1959, pp. 67–76.

5. Fambro, D. B., C. J. Messer, and D. A. Andersen. Estimation of Unprotected Left-Turn Capacity at Signalized Intersections. *Transportation Research Record: Journal of the Transportation Research Board*, No. 644, 1977, pp. 113–119.

6. Haddad, J., and N. Geroliminis. Effect of Left Turns for Arterials with Queue Spillbacks. *92nd Annual Meeting of the Transportation Research Board*, 2013.

7. Chowdhury, M. S. An Evaluation of New Jersey Jug-Handle Intersection (NJJI) with and without Pre-Signals. 2011.

8. Hummer, J. E. Unconventional Left-Turn Alternatives for Urban and Suburban Arterials--Part One. *Institute of Transportation Engineers. ITE Journal Sep*, Vol. 68, No. 9, 1998, pp. 26–29.

9. Hughes, W., R. Jagannathan, D. Sengupta, and J. E. Hummer. Alternative Intersections/Interchanges: Informational Report (AIIR). *FHWA Technical Report FHWA-HRT-09-060*.

10. Berkowitz, C., C. Bragdon, and F. Mier. Continuous Flow Intersection: A Public Private Partnership. *Vehicle Navigation and Information Systems Conference*, Vol. 7, 1996, pp. 277–287. https://doi.org/10.1109/VNIS.1996.1623758.

11. Reid, J., and J. E. Hummer. Travel Time Comparisons between Seven Unconventional Arterial Intersection Designs. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1751, 2007, pp. 56–66.

12. Xuan, Y., V. V. Gayah, M. J. Cassidy, and C. F. Daganzo. Presignal Used to Increase Bus- and Car-Carrying Capacity at Intersections. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2315, 2012, pp. 191–196. https://doi.org/10.3141/2315-20.

13. Xuan, Y., C. F. Daganzo, and M. J. Cassidy. Increasing the Capacity of Signalized Intersections with Separate Left Turn Phases. *Transportation Research Part B: Methodological*, Vol. 45, No. 5, 2011, pp. 769–781. https://doi.org/10.1016/j.trb.2011.02.009.

14. Gayah, V. V., and C. F. Daganzo. Analytical Capacity Comparison of One-Way and Two-Way Signalized Street Networks. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2301, 2012, pp. 76–85.

15. Ortigosa, J., V. V. Gayah, and M. Menendez. Analysis of Network Exit Functions for Various Urban Grid Network Configurations. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2491, 2015, pp. 12–21.

16. Ortigosa, J., V. V. Gayah, and M. Menendez. Analysis of One-Way and Two-Way Street Configurations on Urban Grids. *Transportmetrica B: Transport Dynamics*, Vol. 7, No. 1, 2019, pp. 61–81.

17. DePrator, A., O. Hitchcock, and V. V. Gayah. Improving Urban Street Network Efficiency by Prohibiting Left Turns at Signalized Intersections. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2622, No. 1, 2017, pp. 58–69.

18. Ortigosa, J., and M. Menendez. Traffic Performance on Quasi-Grid Urban Structures. *Cities*, Vol. 36, 2014, pp. 18–27.

19. Wu, P., F. Chu, and C. Ada. Mixed-Integer Programming for a New Bus-Lane Reservation Problem. No. 2015-October, 2015, pp. 2782–2787.

20. Mesbah, M., M. Sarvi, and G. Currie. Optimization of Transit Priority in the Transportation Network Using a Genetic Algorithm. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 3, 2011, pp. 908–919. https://doi.org/10.1109/TITS.2011.2144974.

21. Mesbah, M., M. Sarvi, and G. Currie. New Methodology for Optimizing Transit Priority at the Network Level. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2089, No. 1, 2008, pp. 93–100. https://doi.org/10.3141/2089-12.

22. Bagherian, M., M. Mesbah, and L. Ferreira. Using Delay Functions to Evaluate Transit Priority at Signals. *Public Transportation*, Vol. 7, 2015, pp. 61–75. https://doi.org/10.1007/s12469-014-0090-3.

23. Bayrak, M., and S. I. Guler. Optimizing Bus Lane Placement on Networks While Accounting for Queue Spillbacks. No. 2018-November, 2018, pp. 920–925.

24. Bayrak, M., and S. I. Guler. Determining Optimum Transit Signal Priority Implementation Locations on a Network. *Transportation Research Record: Journal of the Transportation Research Board*, 2020, p. 036119812093479. https://doi.org/10.1177/0361198120934792.

25. Ahn, K., and H. Rakha. System-Wide Impacts of Green Extension Transit Signal Priority. 2006.

26. Tang, Q., and B. Friedrich. Minimization of Travel Time in Signalized Networks by Prohibiting Left Turns. *Transportation Research Procedia*, Vol. 14, No. 14, 2016, pp. 3446–3455. https://doi.org/10.1016/j.trpro.2016.05.306.

27. Hajbabaie, A., J. C. Medina, and R. F. Benekohal. Effects of ITS-Based Left Turn Policies on Network Performance. *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 80–84. https://doi.org/10.1109/ITSC.2010.5625269.

28. Chowdhury, M., N. Derov, P. Tan, and A. Sadek. Prohibiting Left-Turn Movements at Mid-Block Unsignalized Driveways: Simulation Analysis. *ASCE Journal of Transportation Engineering*, Vol. 4, No. 279, 2005, pp. 279–295.

29. Baluja, S. Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. *Report No. CMS-CS-94-163*, 1994.

30. Pelikan, M., D. E. Goldberg, and E. Cantú-Paz. *BOA: The Bayesian Optimization Algorithm*.

31. Barceló, J., and J. Casas. Dynamic Network Simulation with AIMSUN. In *Simulation Approaches in Transportation Analysis*, Springer, pp. 57–98.

32. Daganzo, C. F., V. V. Gayah, and E. J. Gonzales. Macroscopic Relations of Urban Traffic Variables: Bifurcations, Multivaluedness and Instability. *Transportation Research Part B: Methodological*, Vol. 45, No. 1, 2011, pp. 278–288.

33. Miller, B. L., and D. E. Goldberg. *Genetic Algorithms , Tournament Selection, and the Effects of Noise*. 1995.

34. Schwarz, G. Estimating the Dimension of a Model. *Annals of Statistics*, Vol. 6, No. 2, 1978, pp. 461–464. https://doi.org/10.1214/AOS/1176344136.

35. Henrion, M. Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling. In *Machine Intelligence and Pattern Recognition*, North-Holland, pp. 149–163.