

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: 6,951

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 considered 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 location of left-turn restrictions at individual intersections in a perfect and imperfect 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 hybrid method appears to perform the best under more complex scenarios with a larger solution space. 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. Application of these methods to square grid networks under uniform demand patterns also reveal a general pattern in which left-turns should be restricted at central intersections that carry larger vehicle flows, but allowed otherwise. Such findings can be used as a starting point for where to restrict left-turns in more realistic networks.

1 INTRODUCTION

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

15 One-way streets have traditionally been used to eliminate these left-turn conflicts and thus
16 provide higher vehicle flows (7-9). However, one-way streets have other negative externalities that
17 make them less desirable. This includes reduced economic activity (10,11), safety concerns
18 (particularly for non-motorized traffic) (12), circuitous routes for drivers (13), and confusion for
19 transit users (14), among others. Perhaps for this reason, there is a growing number of cities that
20 have performed one-way to two-way street network conversions (15). While such conversions
21 have been found to mitigate some the negative features of one-way streets, engineers have long
22 thought they came at the expense of traffic flow. And, these conversions lead to left-turning
23 conflicts at intersections that need to be mitigated.

24 Several strategies have been proposed to accommodate left-turns while considering the
25 issues they create. Alternative intersection designs have been developed that allow left-turning
26 vehicles in non-traditional ways. These strategies manage left-turn conflicts by using additional
27 features (e.g., downstream U-turns or additional signals) and/or changing the intersection
28 geometry (16–22). However, these designs are generally not well-suited for urban areas with
29 limited road space since they require large spatial footprints or long blocks.

30 Instead, conflicting left turns can be simply restricted at signalized intersections. This
31 simplifies the SPaT plan and allows the intersection to achieve higher overall flow capacities due
32 to fewer change intervals and use of only through/right-turn phases. However, such restrictions
33 will require vehicles that would have otherwise made a left-turn to reroute, which may induce
34 longer average travel distances. Several recent studies have examined the competing impacts of
35 such left turn restrictions enacted across entire grid networks (23–27). The studies apply
36 macroscopic traffic flow models (28) to quantify overall network performance under scenarios
37 with and without left turns. The findings generally agree that eliminating left-turns can improve
38 overall network operation (specifically, the rate that trips can be completed in the network),
39 particularly when the network is operating near its capacity. However, these prior studies fail to
40 consider the optimal spatial location of such restrictions at individual intersections within a
41 network.

42 This type of problem is most closely related to general facility location problems in the
43 transportation research literature, which are classified as NP-hard optimization problems due to
44 the large solution space and lack of analytical solution (29). Within urban networks, many studies

1 have proposed methods to determine optimal location of treatments along individual links—e.g.,
2 optimal bus lanes locations (30–33)—or at individual intersections—e.g., optimal transit signal
3 priority locations (34, 35). Very few studies have examined the application of left turn restrictions
4 or alternative left-turn treatments at individual intersections on a network. The problem is complex
5 as changes in left-turn treatments at one intersection might influence traffic patterns and operations
6 at adjacent intersections. Additionally, vehicles will reroute themselves due to the left-turn
7 restrictions enacted. These issues do not facilitate the expression of the optimal left-turn restriction
8 problem as a traditional mathematical program. Instead, one study proposed a generic
9 methodology to identify intersections that should restrict left turns that relied on fairly simplistic
10 traffic models that did not account for traffic dynamics, such as queue spillbacks and changes in
11 vehicle routing (36). Others utilized simulation to test the performance of a set of candidate left-
12 turn restriction configurations but did not necessarily perform any optimization (37, 38).

13 Another factor that complicates the problem of selecting left turn restriction locations is
14 the spatial dependency between the possible restriction locations. The spatial dependency can be
15 defined as the influence of restricting left turns at one intersection on the effect of restricting left
16 turns at adjacent intersections. This happens for multiple reasons. For one, restriction decisions at
17 one intersection will influence how vehicles traverse the intersection and arrive to the downstream
18 intersection (both in terms of amount and pattern of arrivals). Additionally, these decisions can
19 change routing of vehicles. The existing studies on determining optimal location of treatments
20 along individual links primarily use genetic algorithms to find the optimum locations (30, 39–42).
21 However, genetic algorithms assume independent relationship between decision variables, and
22 thus the unaccounted dependencies can significantly limit the performance of genetic algorithms
23 (43,44). There are two groups of evolutionary algorithms that can account the dependencies
24 between decision variables. Messy genetic algorithms conducts a partial enumeration of the
25 solution space to select promising building blocks to identify dependencies, and then uses these
26 building blocks to continue with a solution method similar to the basic genetic algorithms. The
27 need for partial enumeration makes these types of algorithms infeasible to use for simulation-based
28 transportation optimization problems due to the required computation time. On the other hand,
29 estimation of distribution algorithms identify the existing dependencies within the decision
30 variables by leveraging probability theory. These methods construct and sample probabilistic
31 models of dependencies, and do not require a computationally expensive partial enumeration.
32 Hence, this paper will explore the use of estimation of distribution algorithms to optimize left turn
33 restriction locations.

34 The purpose of this paper is to propose and compare several methods to determine the
35 optimal spatial configuration of left-turn restrictions. Several heuristic algorithms are considered
36 to identify candidate configurations, the performance of which are tested in a micro-simulation
37 environment. The first is a population-based incremental learning (PBIL) algorithm that combines
38 the evolutionary nature of genetic algorithms with competitive learning features (45). The second
39 is a Bayesian optimization algorithm that seeks to directly learn and leverage dependencies
40 inherent within the solution space using Bayesian networks (46). The final method is a hybrid of
41 the two that uses the PBIL method to generate a set of candidate solutions and then implements a
42 Bayesian network to generate similar solutions that might also perform well. These are applied
43 within a dense, grid-like urban network under uniform traffic patterns, though it can be applied in
44 more realistic cases as well. The results suggest that these methods are generally suitable for the
45 optimal spatial left-turn optimization problem and provides insights into the conditions under
46 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 relationship between a transport agency that decides the left turn restriction locations and the people in the network can be modeled as a leader-follower problem. The leader (transport agency) enacts a set of restrictions to reduce the total delay of the network. The followers (people in the network) react to that decision by adapting their route choice with the aim of minimizing their travel times. Since the reaction of the followers affects the resulting delay change in the network, a bi-level optimization approach is needed to find the optimum locations of left turn restrictions. The upper level uses a metaheuristic method to select promising left turn restriction configurations. Then the total travel time associated with the selected left turn restriction configuration is calculated by the lower level. The lower level uses a microsimulation model to accurately simulate the traffic and account for the changes in the routing behavior of the network users. The resulting total travel time is used by the upper level to select the next set of promising left turn restriction configurations. The next two subsections describe the methods used for the network evaluation and the selection of LT restriction locations.

The lower level network assessment

The lower of the bi-level optimization algorithm aims to calculate total travel time for a given LT 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 (47), since it accurately captures vehicle dynamics, queue spillover and other realistic features.

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 re-routing occurs at regular intervals of 3 minutes. Previous research has shown that such adaptive traffic routing algorithms provide better network-wide operational performance (48).

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 (45). 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.

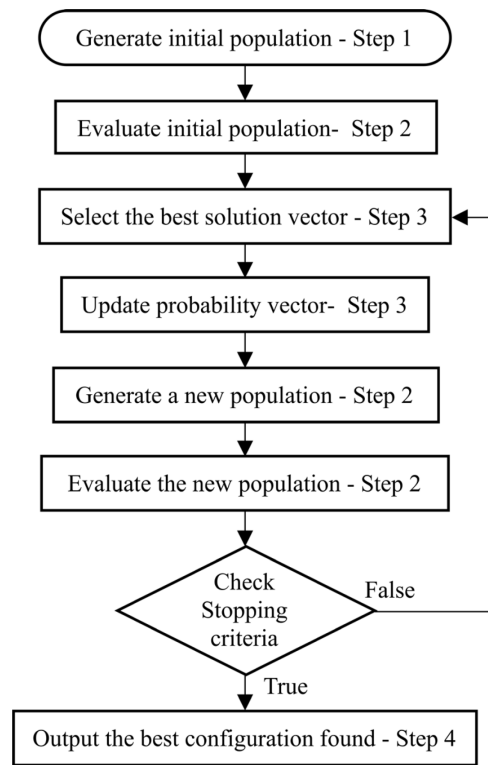


FIGURE 1. Flow chart of population-based incremental learning

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.

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

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

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 i th position of the best solution found in generation t , and w_i^t is the value of the i th 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 seeks 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 $LR^- = 0.05$. 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 .

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

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 LT restrictions). BOA aims to learn the dependency structure between decision variables using Bayesian networks and uses this information to generate better solutions. The structure of a Bayesian network is a directed acyclic graph which the nodes are the decision variables (i.e. left turn restriction locations). The links between nodes of the Bayesian network represent the dependencies between left-turn restriction decisions at individual intersections. The flowchart of the BOA is shown in FIGURE 2.

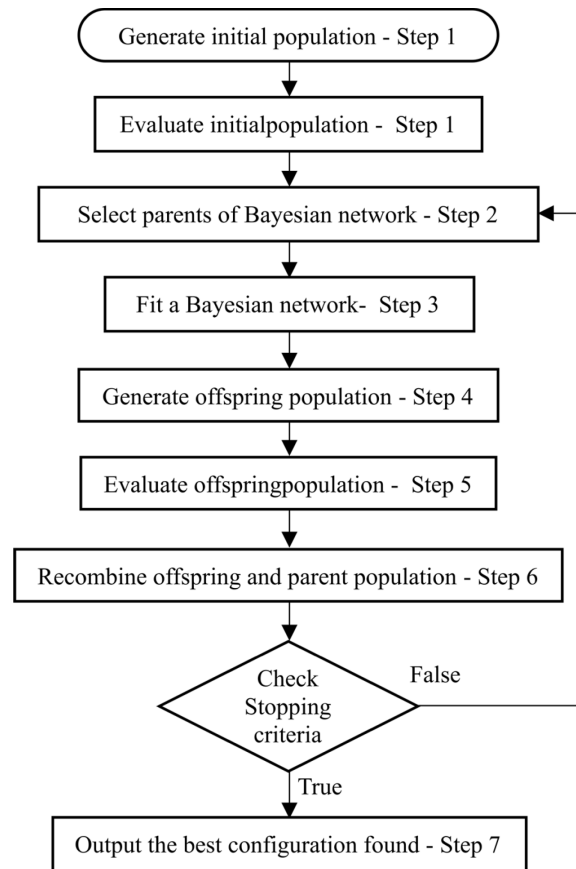


FIGURE 2. Flow chart of Bayesian optimization algorithm

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 LT restriction configurations. Similar to PBIL, each solution vector is a binary vector that represents the LT

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 LT restriction configuration.

The selection step selects the parent solutions that will be used for Bayesian network construction. A tournament selection method is used (49) in which two LT 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 operations 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) shown in (4) and (5) is used as a scoring metric (50).

$$BIC = \sum_{i=1}^n \left(-H(X_i|D_i)N - 2^{|D_i|} \frac{\log_2(N)}{2} \right) \quad (4)$$

$$H(X_i|D_i) = -\sum_{x_i, d_i} p(x_i, d_i) \log_2(x_i|d_i) \quad (5)$$

Where, n is the number of decision variables; D_i is the variables that decision variable X_i depend on; $H(X_i|D_i)$ is the conditional entropy of X_i given D_i ; N is the population size; $p(x_i, d_i)$ and $(x_i|d_i)$ are the observed probabilities from the population where $X_i = x_i$ and $D_i = d_i$. 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. The term with $\log_2(N)$ favors simpler network (i.e. network with fewer number of dependencies) and helps to eliminate inclusion of unnecessary dependencies to 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. For the sampling process the joint probability distribution (6) encoded in the Bayesian network is used. This enables algorithm to keep exploring the solution space with independent variables while keeping the values of dependent decision variables conditioned to the variables they depend on. Since low performing solutions are eliminated at the end of each generation (Step 6), the algorithm fits the Bayesian network to better solutions at each generation and finds successful value combinations of dependent variables and the variables they depend on.

$$p(X) = \prod_{i=1}^n p(X_i|D_i) \quad (6)$$

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 (51). 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 high 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 the 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 LT 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.

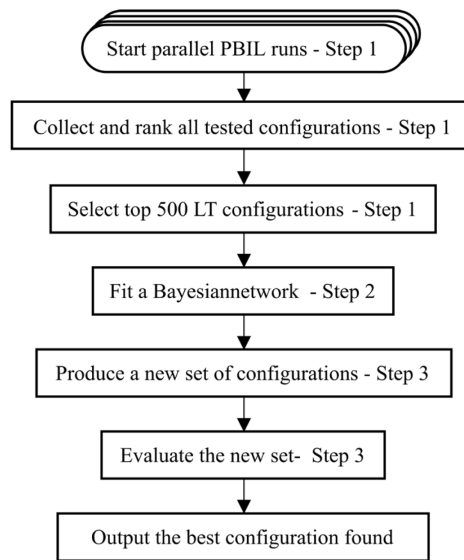


FIGURE 3. Flow chart of PBIL-BOA hybrid optimization algorithm

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 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 an 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 likely to be selected is used; however, the proposed methods can be applied to any generic demand pattern as desired. 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. Each LT configuration is simulated for 45 minutes followed by a recovery period of 15 minutes. The resulting cumulative count curves of vehicle entries and exits are used to determine the total travel time of the network users.

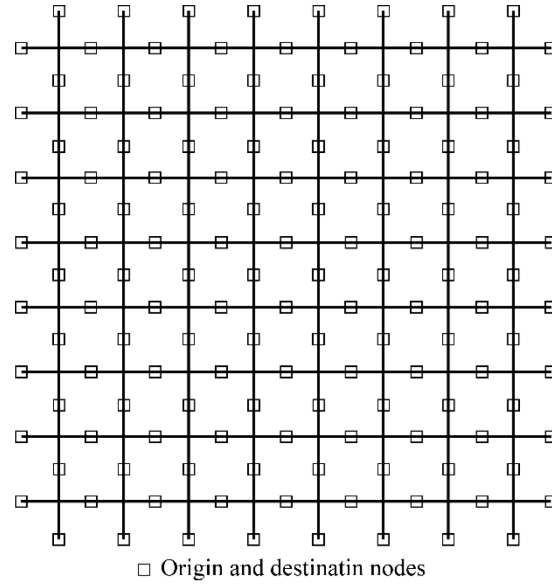


FIGURE 4. Test network and OD locations

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 LT 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 LT restriction configurations increases exponentially with the number of possible restriction locations, the number of possible LT restriction configurations for both scenarios is extremely large (2^{64} in the first scenario, $(2 \times 2)^{64}$ in the second). Several constraints/assumptions are implemented to reduce the solution space to examine in detail the performance of the proposed heuristics. 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 LT restriction configurations that are rotationally symmetrical around the center of the network. The second constraint is LT 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 LT 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 LT restriction configurations is low enough to use brute-force enumeration to determine the global optimal configuration. These constraints/assumptions are later relaxed to demonstrate the performance of the proposed algorithms in a more realistic setting.

In addition to the scenarios with the symmetry constraint, two other optimization scenarios created to demonstrate the capabilities of the bi-level methodology under more realistic assumptions. For the first scenario, the symmetry constraint is relaxed, and the performance of the used upper-level algorithms are compared. The second scenario considers optimization of LT restriction locations for a non-perfect grid network. For both of the scenarios, the demand pattern, network parameters, and the algorithm specific optimization parameters are same as for the scenarios with the symmetry constraint.

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. Finally, the results of the tests under more realistic assumptions are presented.

Scenario 1: One left-turn restriction decision per intersection

This section provides the results when only one restriction decision per intersection is made. First, the results are provided when restriction configurations are assumed to be perfectly symmetric, to reduce the size of the solution space and examine if the heuristics can find the optimal solution under these assumptions. Then, the results are presented when this symmetry constraint is relaxed.

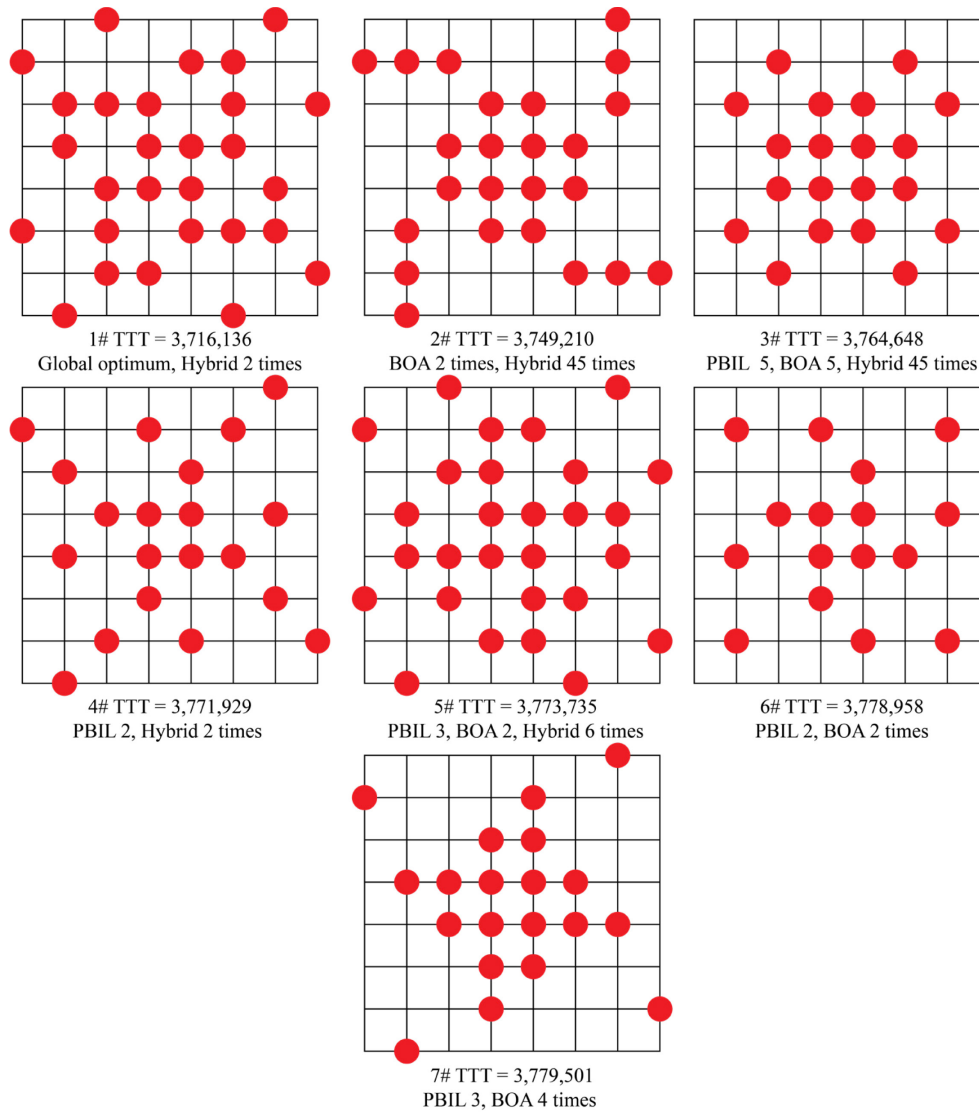
Tests under symmetric configurations

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 LT restriction configurations found during a subset of independent PBIL runs. At each instance of this algorithm, a Bayesian network that represents the dependencies between LT 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.

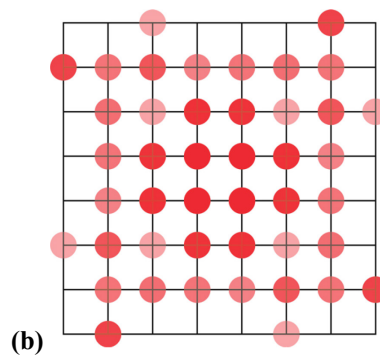
In addition to these heuristic methods, a brute-force enumerate approach was applied to identify the global optimal solution. In this method, all 32,768 potential configurations were tested to determine the configuration with the lowest travel time. The results suggest that the implementation of left-turn restrictions at a subset of intersections can reduce 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

1 congestion in the network. Under these conditions, it is not necessary to prohibit left turns
2 everywhere since some intersections are under-saturated.

3 All configurations obtained from the proposed optimization methods and the global
4 optimal configuration are presented in FIGURE 5a. In this figure, red dots represent intersections
5 at which left-turn restrictions are applied. Note that only seven unique configurations were
6 identified: five using PBIL, five using the BOA, and five using the hybrid approach. The results
7 reveal that all methods are able to identify a solution that provides travel times within 1.7% of the
8 global optimal solution; i.e., the maximum optimality gap of any solution identified was 1.71%.
9 Comparing the three methods, the average optimality gap of the PBIL, BOA and hybrid methods
10 are 1.51%, 1.44%, and 1.11%, respectively. Overall, this suggest strong performance across the
11 three optimization methods. However, the hybrid method appears to generally provide better
12 results. For one, it has the smallest average optimality gap. In addition, it is the only method to
13 identify the global optimal solution (though this was just found 2% of the times and can be
14 attributed to randomness). Still, was able to outperform or equal the PBIL method in 92% of runs
15 and outperform or equal the BOA method 47% of runs. A comparison of these methods by TTT
16 produced is illustrated in FIGURE 6.



(a)



(b)

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

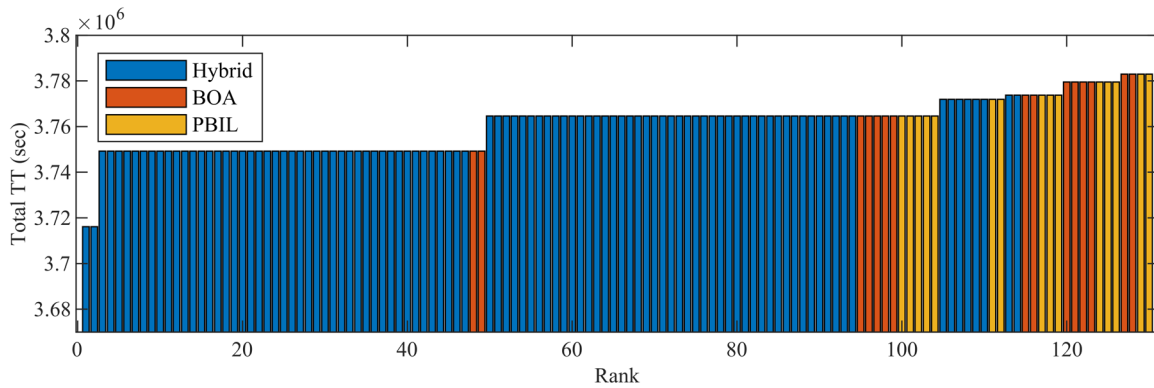
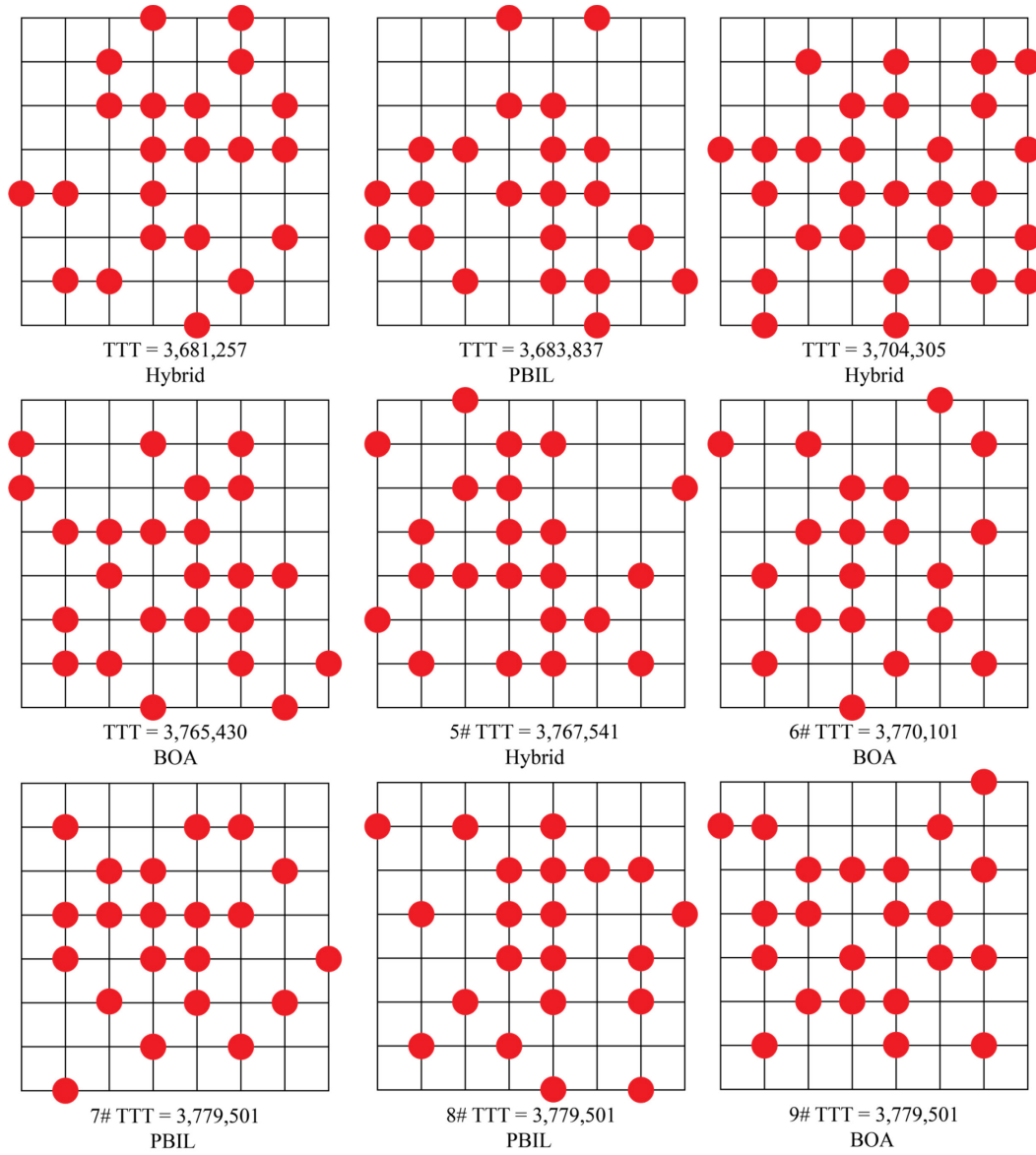


FIGURE 6. Ranking of best-performing left-turn restriction configurations identified (Scenario 1)

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 re-route 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 re-routing 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 flow 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 re-route. Finally, left-turn restrictions at these periphery locations are more likely to induce additional travel distance due to fewer routing options available at the edge of the network.

Tests under asymmetric configurations

The previous section finds that the heuristic algorithms work well to identify a (nearly) global optimal solution when the restriction configuration is assumed to be symmetric. Now, we relax this symmetry assumption to see how the performance and spatial restriction changes under more realistic settings. 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, in this case, is unknown. FIGURE 7 provides the set of best-performance solutions using each of the three algorithms. In all cases, travel time can be further reduced – as expected – when the symmetry assumption is relaxed. Note that unlike the previous section, the different algorithms do not identify the same solutions, likely due to the extremely large solution space. Nevertheless, the overall pattern is the same: left turns are more likely to be restricted at the central intersections and less likely to be restricted at the periphery intersections to improve travel time. Furthermore, the hybrid algorithm finds the best-performing configuration and three of the top five configurations.



(a)

(b)

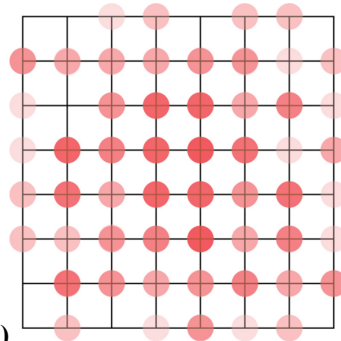


FIGURE 7. Results of Scenario 1 relaxing symmetry constraints. (a) Optimal left-turn restriction configurations obtained by proposed algorithms; and, (b) common locations of left-turn restrictions identified in top-performing algorithms.

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. As with Scenario 1, the PBIL, BOA and hybrid methods were repeated multiple times. However, the number of repeats was 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.

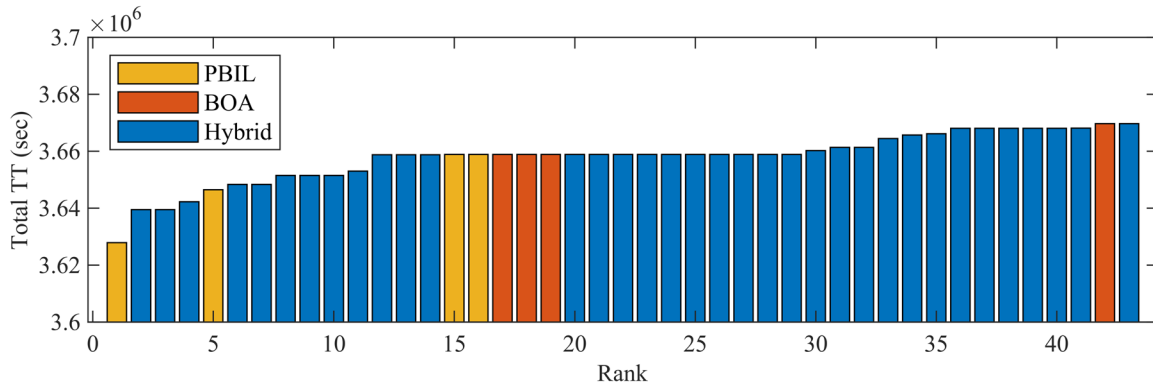
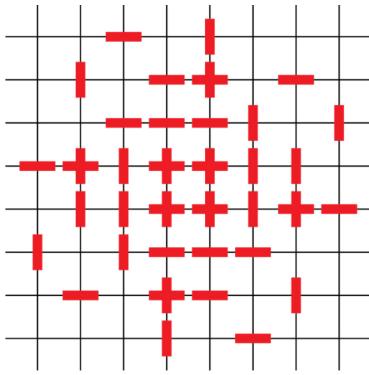


FIGURE 8. Ranking of best-performing left-turn restriction configurations identified (Scenario 2)

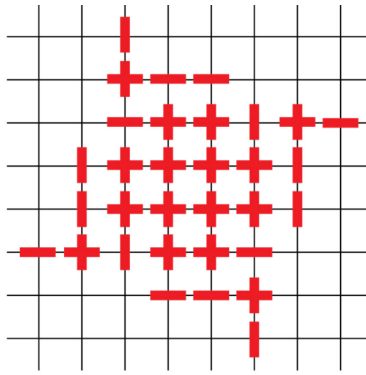
The unique configurations that were identified are shown in FIGURE 8a. In this figure, red vertical or horizontal lines represent left-turn restrictions in the north-south or east-west direction, respectively, implemented at a specific intersection. As shown, only 17 unique configurations were identified. 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

1 portion of the network and generally do not occur in the periphery of the network. These reasons
2 are the same as those in the previous scenario. However, there are some key differences. Left-turns
3 are generally permitted at approaches that lead from the central portion of the network toward the
4 periphery of the network, whereas they are restricted at those approaches leading toward the central
5 portion of the network.

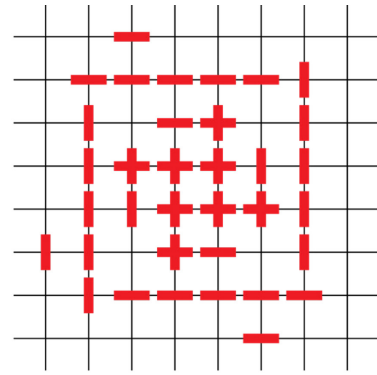
6



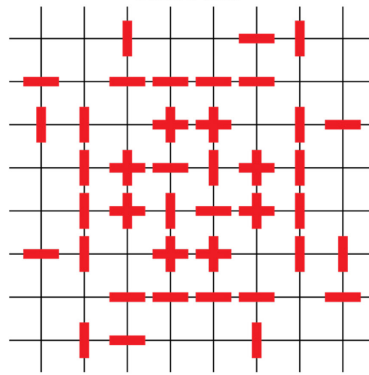
1# TTT = 3,627,872
PBIL 1 time



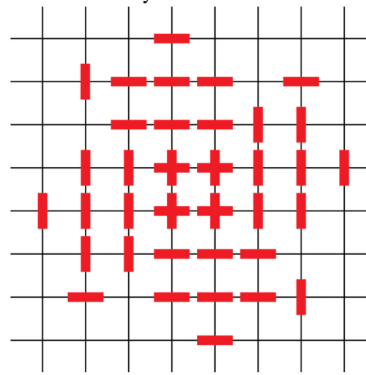
2# TTT = 3,639,460
Hybrid 2 times



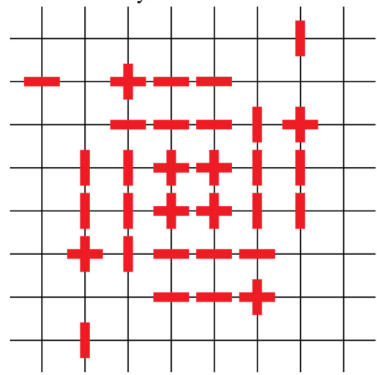
3# TTT = 3,642,241
Hybrid 1 time



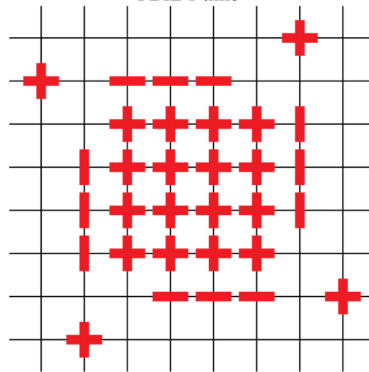
4# TTT = 3,646,469
PBIL 1 time



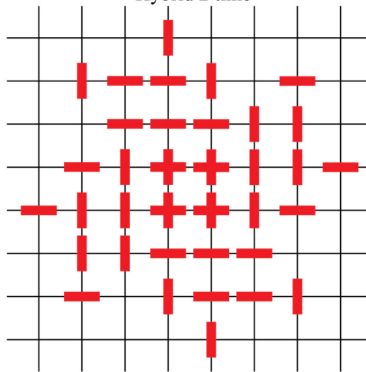
5# TTT = 3,648,314
Hybrid 2 time



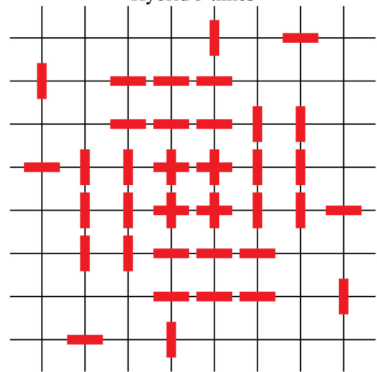
6# TTT = 3,651,474
Hybrid 3 times



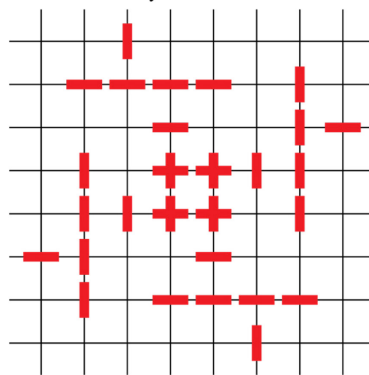
7# TTT = 3,652,978
Hybrid 1 time



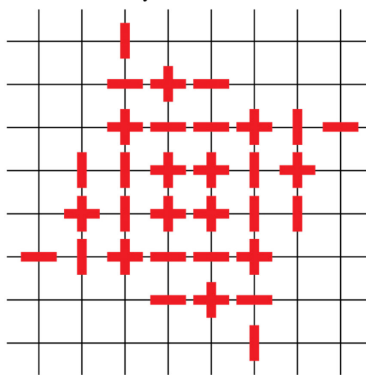
8# TTT = 3,658,715
Hybrid 3 times



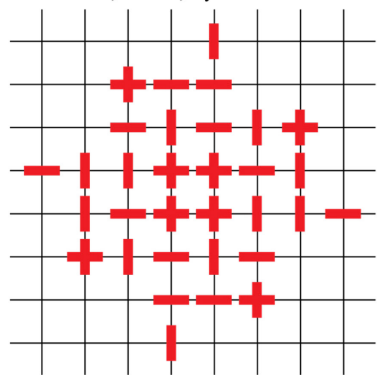
9# TTT = 3,658,822
PBIL 2, BOA 3, Hybrid 10 times



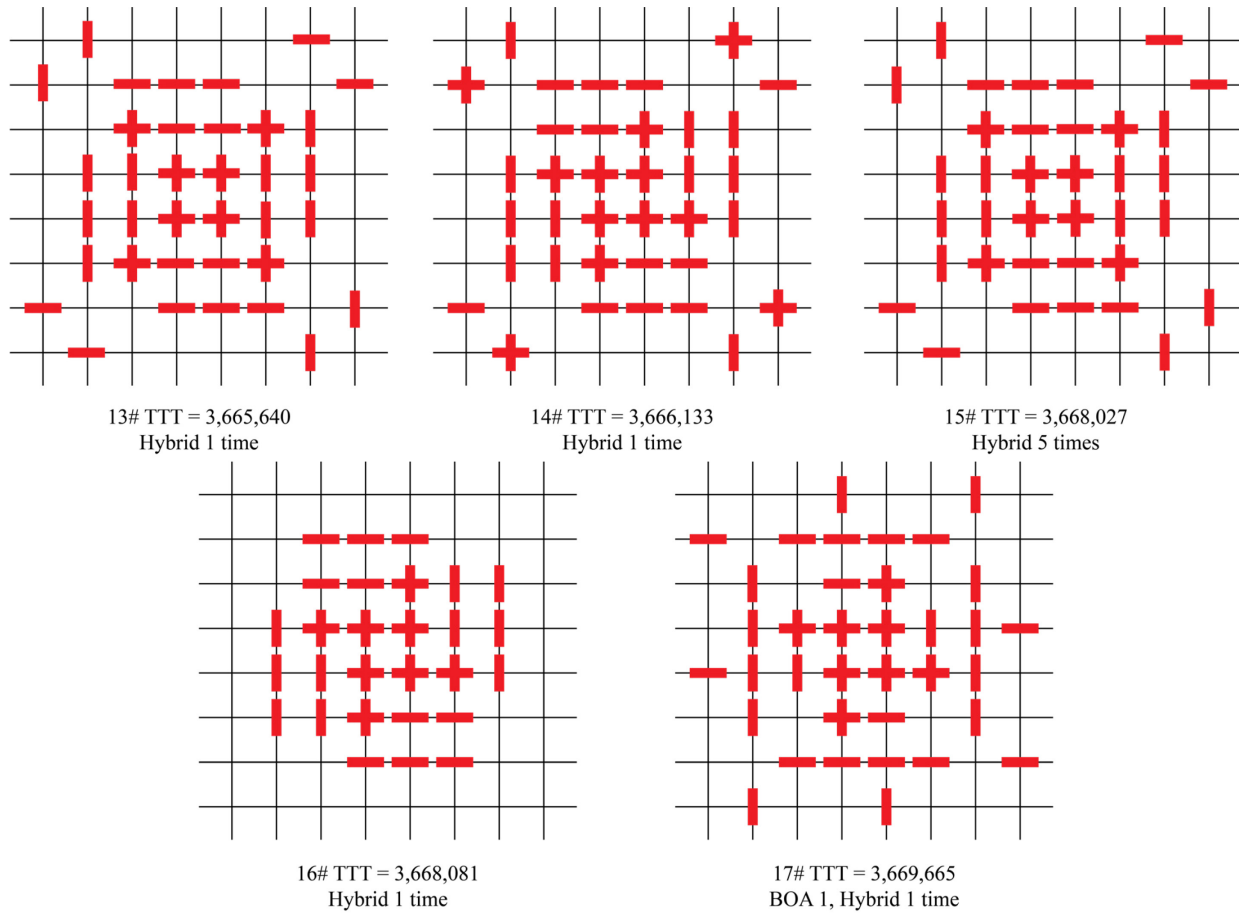
10# TTT = 3,660,201
Hybrid 1 time



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



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



(a)

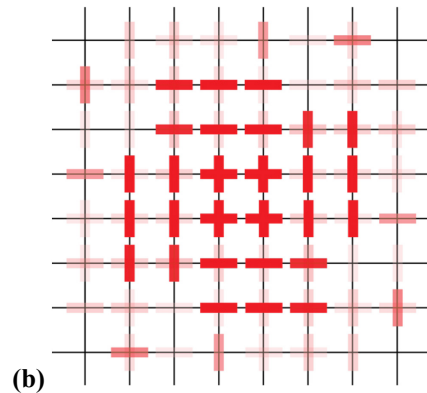
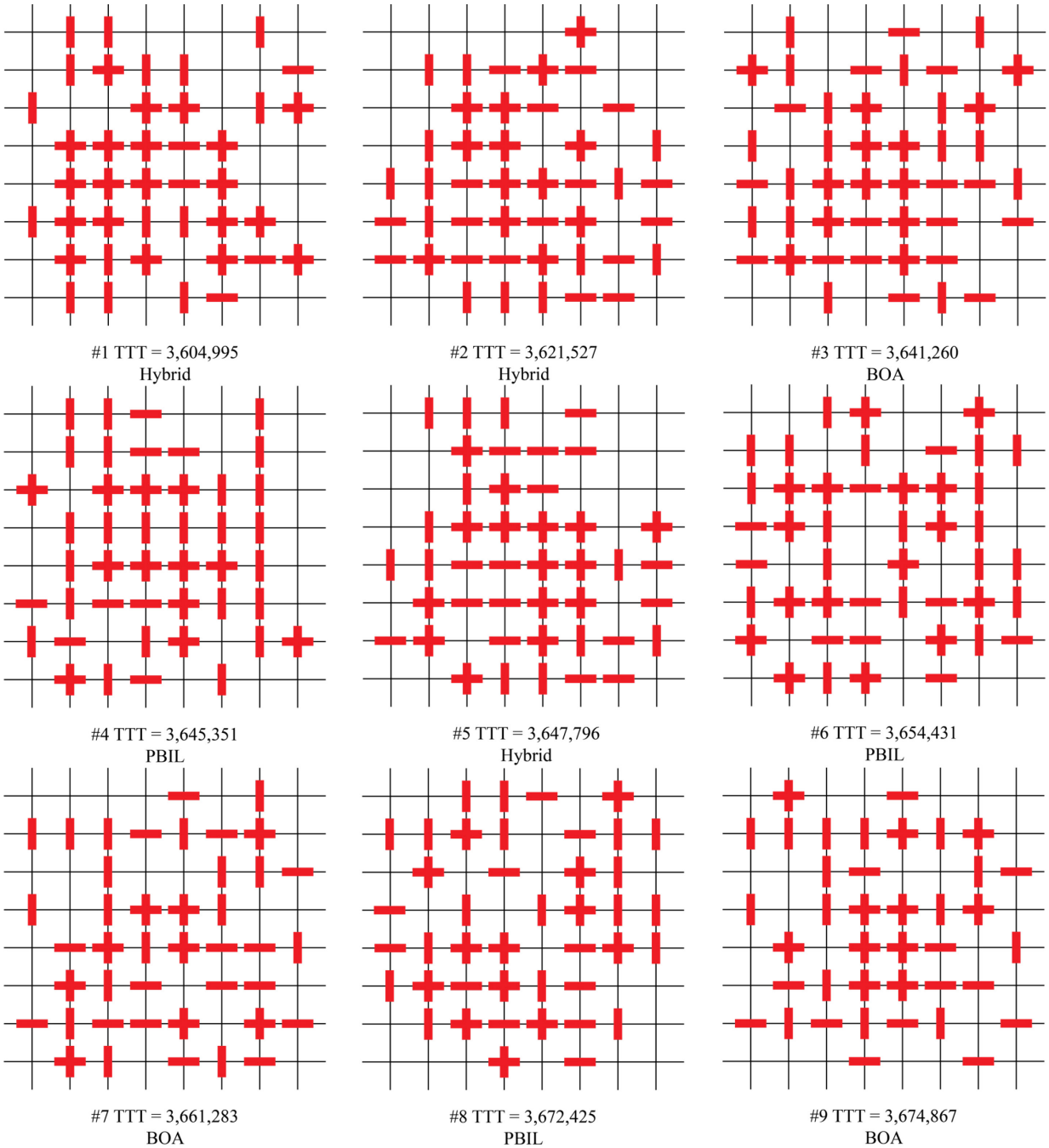


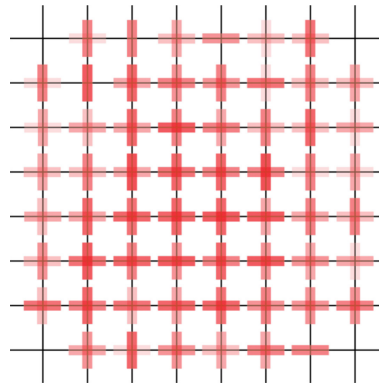
FIGURE 9. Results of Scenario 2 under symmetry constraints. (a) Optimal left-turn restriction configurations obtained by proposed algorithms; and, (b) common locations of left-turn restrictions identified in top-performing algorithms.

FIGURE 10 provides solutions obtained for this scenario when symmetry constraints are relaxed. Notice that travel times are able to be improved when the solution space is made larger, as expected. However, the general spatial pattern of left-turn restriction configurations remains the same. This suggests that this is a general pattern that should optimize operations in perfect grid networks. Notice also that the hybrid method is again the best-performing method. It found the top two configurations and three of the top five.



(a)

1
2



(b)

FIGURE 10. Results of Scenario 2 relaxing symmetry constraints. (a) Optimal left-turn restriction configurations obtained by proposed algorithms; and, (b) common locations of left-turn restrictions identified in top-performing algorithms.

Tests on imperfect networks

The previous results were all obtained in a perfect grid network. While useful for exploratory purposes, most networks do not have such an idealistic structure. To demonstrate the applicability under more realistic settings, the PBIL method was performed on an imperfect grid network structure. This imperfect grid was created by randomly removing links from the perfect grid in such a way that all OD pairs could still be served. This could represent the applicability of these methods to a real-world network configuration. The imperfect networks also lead to non-uniform traffic patterns even under a uniform traffic demand; thus, these tests also demonstrate the applicability of these methods under non-uniform traffic patterns.

The results of the optimization under Scenario 2 (two restriction decisions made per intersection) without any symmetry constraints are shown in FIGURE 11. The total travel times found represent a reduction of 15.4 percent compared to the case where left-turns are allowed at all intersections and 48.3 percent when left-turns are restricted at all intersections. This suggests that the algorithms work and can be applied to a more realistic setting.

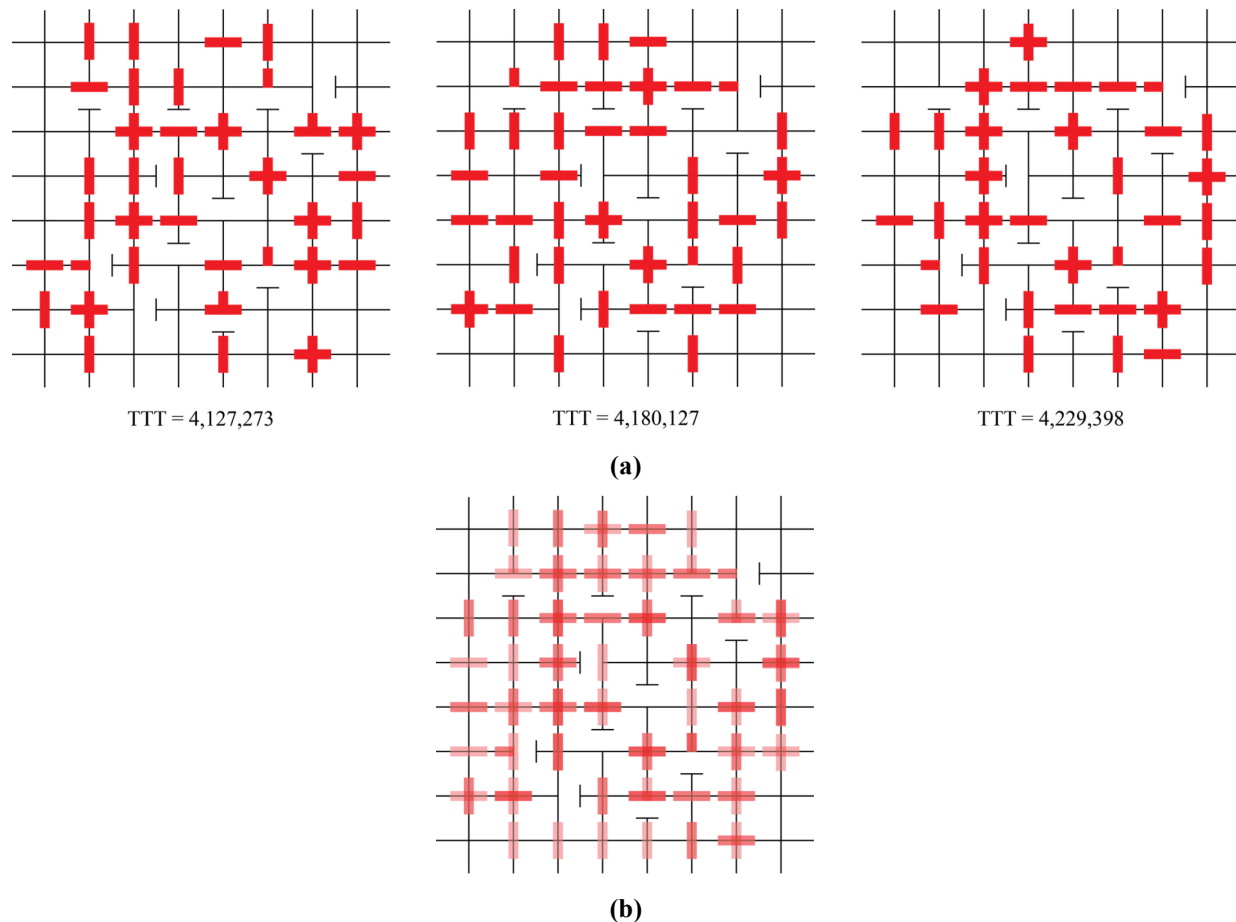


FIGURE 11. Results of PBIL method applied to an imperfect grid network. (a) Individual solutions; and, (b) common locations of left-turn restrictions identified.

DISCUSSION AND CONCLUDING REMARKS

This paper demonstrates that restricting left-turns at a subset of signalized intersections can help improve overall network traffic conditions, which contributes to the growing body of literature on left-turn restrictions in urban environments. 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

1 methods when the solution space is small, while the hybrid method performs better as the solution
2 space increases. In general, the two-decision-per-intersection case always provides lower travel
3 times than the one-decision-per-intersection case. While this is expected because it is more
4 flexible, the fact that better-performing configurations can be obtained even though the solution
5 space is exponentially greater is promising.

6 Regardless of the method and scenario, the results suggest that a fairly consistent and
7 generic spatial pattern of left-turn restrictions that minimizes vehicle travel and maximizes
8 network efficiency might exist in grid networks. In this pattern, left-turns are restricted at
9 intersections in the inner portion of the network that carry the largest vehicle flows, and left-turns
10 are allowed at the intersections in the periphery where flows are low and the proportion of vehicles
11 making a left-turn is higher. While this is obtain under a uniform demand pattern, it might provide
12 general insights for how to implement such decisions at more complicated and realistic network
13 structures. While the actual configuration on any network would be subject to its network-specific
14 features, signal settings and demand pattern, the overall pattern that was observed here should be
15 fairly general and serve as a good starting point that could be refined due to network-specific
16 features without the need for an exhaustive search. However, any search would benefit from the
17 methods provided here. The results also reveal that the livability and economic benefits of two-
18 way streets can be achieved while simultaneously reducing total vehicle travel time by simply
19 restricting left-turning vehicles at specific intersection locations. While not specifically studied
20 here, the reduction of delay could also have other inherent benefits – such as reducing harmful
21 vehicle emissions and fuel/energy consumption – since improving vehicle travel times would
22 reduce unnecessary delays and vehicle stopping maneuvers (52).

23 Further work should also consider how these central left-turn restrictions might influence
24 overall network resilience to disruptions that might occur along links, such as traffic crashes or
25 bottlenecks caused by freight vehicles. Additionally, methods are needed to simultaneously
26 identify optimal restriction locations while changing signal timings at intersections with/without
27 left turn restrictions. This is a much more complex problem that requires both binary and
28 continuous decisions. The authors are currently exploring methods to address this problem.

30 ACKNOWLEDGEMENTS

31 This research was supported by NSF Grant CMMI-1749200

33 AUTHOR CONTRIBUTION STATEMENT

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

38 REFERENCES

- 39 1. Roess, R. P., E. S. Prassas, and W. R. McShane. *Traffic Engineering, 4th Edition*. 2010.
- 40 2. Cottrell, B. H. Guidelines for Protected/Permissive Left-Turn Signal Phasing.

- 1 *Transportation Research Record: Journal of the Transportation Research Board*, No. 1069,
- 2 1986, pp. 54–61.
- 3 3. Messer, C. J., and D. B. Fambro. Effects of Signal Phasing and Length of Left-Turn Bay on
- 4 Capacity. *Transportation Research Record: Journal of the Transportation Research Board*,
- 5 No. 644, 1977, pp. 95–101.
- 6 4. Newell, G. F. The Effect of Left Turns on the Capacity of a Traffic Intersection. *Quarterly*
- 7 *of Applied Mathematics*, Vol. 17, No. 1, 1959, pp. 67–76.
- 8 5. Fambro, D. B., C. J. Messer, and D. A. Andersen. Estimation of Unprotected Left-Turn
- 9 Capacity at Signalized Intersections. *Transportation Research Record: Journal of the*
- 10 *Transportation Research Board*, No. 644, 1977, pp. 113–119.
- 11 6. Haddad, J., and N. Geroliminis. Effect of Left Turns for Arterials with Queue Spillbacks.
- 12 *92nd Annual Meeting of the Transportation Research Board*, 2013.
- 13 7. O'Toole, R. Should Cities Convert One-Way Streets to Two-Way? *The Thoreau Institute*,
- 14 Camp Sherman, Ore., 2005. <http://ti.org/vaupdate30.html>. Accessed November 2020.
- 15 8. Stemley, J. One-Way Streets Provide Superior Safety and Convenience. *ITE Journal*, Vol.
- 16 68, No. 8, 1998, pp. 47–50.
- 17 9. Johnson, D. Economic Analysis of the One-Way Grid in Roseburg, Oregon. *ITE Journal*,
- 18 Vol. 26, 1955, pp. 67–70.
- 19 10. Forbes, G. Vital Signs: Circulations in the Heart of the City—An Overview of Downtown
- 20 Traffic. *ITE Journal*, Vol. 68, No. 8, 1998, pp. 26–29.
- 21 11. Riggs, W. and B. Appleyard. The economic impact of one to two-way street conversions:
- 22 advancing a context-sensitive framework. *Journal of Urbanism: International Research on*
- 23 *Placemaking and Urban Sustainability*, Vol. 11, No. 2, 2018, pp.129-148.
- 24 12. Tindale, S., and P. Hsu. Crash Data and Signal Coordination: A One-Way Pair Case Study.
- 25 *Journal of Safety Research*, Vol. 36, No. 5, 2005, pp. 481–482.
- 26 13. Gayah, V. V., & Daganzo, C. F. (2012). Analytical capacity comparison of one-way and
- 27 two-way signalized street networks. *Transportation Research Record*, 2301(1), 76-85.
- 28 14. Dunphy, R., D. Myerson, and M. Pawlukiewicz. Ten Principles for Successful Development
- 29 Around Transit. *ULI- the Urban Land Institute*, Washington, D.C, 2003.
- 30 15. R., C. Faulkner, and A. Syed. Conversion of Streets from One-Way to Two-Way Operation.
- 31 Michigan Department of Transportation, Lansing, 2000.
- 32 16. Chowdhury, M. S. An Evaluation of New Jersey Jug-Handle Intersection (NJJI) with and
- 33 without Pre-Signals. 2011.
- 34 17. Hummer, J. E. Unconventional Left-Turn Alternatives for Urban and Suburban Arterials--
- 35 Part One. *Institute of Transportation Engineers. ITE Journal Sep*, Vol. 68, No. 9, 1998, pp.
- 36 26–29.
- 37 18. Hughes, W., R. Jagannathan, D. Sengupta, and J. E. Hummer. Alternative
- 38 Intersections/Interchanges: Informational Report (AIIR). *FHWA Technical Report FHWA-*
- 39 *HRT-09-060*.

- 1 19. Berkowitz, C., C. Bragdon, and F. Mier. Continuous Flow Intersection: A Public Private
2 Partnership. *Vehicle Navigation and Information Systems Conference*, Vol. 7, 1996, pp.
3 277–287. <https://doi.org/10.1109/VNIS.1996.1623758>.
- 4 20. Reid, J., and J. E. Hummer. Travel Time Comparisons between Seven Unconventional
5 Arterial Intersection Designs. *Transportation Research Record: Journal of the*
6 *Transportation Research Board*, No. 1751, 2007, pp. 56–66.
- 7 21. Xuan, Y., V. V. Gayah, M. J. Cassidy, and C. F. Daganzo. Presignal Used to Increase Bus-
8 and Car-Carrying Capacity at Intersections. *Transportation Research Record: Journal of*
9 *the Transportation Research Board*, No. 2315, 2012, pp. 191–196.
10 <https://doi.org/10.3141/2315-20>.
- 11 22. Xuan, Y., C. F. Daganzo, and M. J. Cassidy. Increasing the Capacity of Signalized
12 Intersections with Separate Left Turn Phases. *Transportation Research Part B:*
13 *Methodological*, Vol. 45, No. 5, 2011, pp. 769–781.
14 <https://doi.org/10.1016/j.trb.2011.02.009>.
- 15 23. Gayah, V. V., and C. F. Daganzo. Analytical Capacity Comparison of One-Way and Two-
16 Way Signalized Street Networks. *Transportation Research Record: Journal of the*
17 *Transportation Research Board*, No. 2301, 2012, pp. 76–85.
- 18 24. Ortigosa, J., V. V. Gayah, and M. Menendez. Analysis of Network Exit Functions for
19 Various Urban Grid Network Configurations. *Transportation Research Record: Journal of*
20 *the Transportation Research Board*, No. 2491, 2015, pp. 12–21.
- 21 25. Ortigosa, J., V. V. Gayah, and M. Menendez. Analysis of One-Way and Two-Way Street
22 Configurations on Urban Grids. *Transportmetrica B: Transport Dynamics*, Vol. 7, No. 1,
23 2019, pp. 61–81.
- 24 26. DePrator, A., O. Hitchcock, and V. V. Gayah. Improving Urban Street Network Efficiency
25 by Prohibiting Left Turns at Signalized Intersections. *Transportation Research Record:*
26 *Journal of the Transportation Research Board*, Vol. 2622, No. 1, 2017, pp. 58–69.
- 27 27. Ortigosa, J., and M. Menendez. Traffic Performance on Quasi-Grid Urban Structures.
28 *Cities*, Vol. 36, 2014, pp. 18–27.
- 29 28. Geroliminis, N., & Daganzo, C. F. (2008). Existence of urban-scale macroscopic
30 fundamental diagrams: Some experimental findings. *Transportation Research Part B:*
31 *Methodological*, 42(9), 759–770.
- 32 29. Wu, P., F. Chu, and C. Ada. Mixed-Integer Programming for a New Bus-Lane Reservation
33 Problem. *Presented in IEEE 18th International Conference on Intelligent Transportation*
34 *Systems*, Las Palmas, 2015, pp. 2782–2787.
- 35 30. Mesbah, M., M. Sarvi, and G. Currie. Optimization of Transit Priority in the Transportation
36 Network Using a Genetic Algorithm. *IEEE Transactions on Intelligent Transportation*
37 *Systems*, Vol. 12, No. 3, 2011, pp. 908–919. <https://doi.org/10.1109/TITS.2011.2144974>.
- 38 31. Mesbah, M., M. Sarvi, and G. Currie. New Methodology for Optimizing Transit Priority at
39 the Network Level. *Transportation Research Record: Journal of the Transportation*
40 *Research Board*, Vol. 2089, No. 1, 2008, pp. 93–100. <https://doi.org/10.3141/2089-12>.
- 41 32. Bagherian, M., M. Mesbah, and L. Ferreira. Using Delay Functions to Evaluate Transit

- 1 Priority at Signals. *Public Transportation*, Vol. 7, 2015, pp. 61–75.
2 <https://doi.org/10.1007/s12469-014-0090-3>.
- 3 33. Bayrak, M., and S. I. Guler. Optimizing Bus Lane Placement on Networks While
4 Accounting for Queue Spillbacks. No. 2018-November, 2018, pp. 920–925.
- 5 34. Bayrak, M., and S. I. Guler. Determining Optimum Transit Signal Priority Implementation
6 Locations on a Network. *Transportation Research Record: Journal of the Transportation*
7 *Research Board*, 2020, p. 036119812093479. <https://doi.org/10.1177/0361198120934792>.
- 8 35. Ahn, K., and H. Rakha. System-Wide Impacts of Green Extension Transit Signal Priority.
9 2006.
- 10 36. Tang, Q., and B. Friedrich. Minimization of Travel Time in Signalized Networks by
11 Prohibiting Left Turns. *Transportation Research Procedia*, Vol. 14, No. 14, 2016, pp.
12 3446–3455. <https://doi.org/10.1016/j.trpro.2016.05.306>.
- 13 37. Hajbabaie, A., J. C. Medina, and R. F. Benekohal. Effects of ITS-Based Left Turn Policies
14 on Network Performance. *13th International IEEE Conference on Intelligent*
15 *Transportation Systems*, 2010, pp. 80–84. <https://doi.org/10.1109/ITSC.2010.5625269>.
- 16 38. Chowdhury, M., N. Derov, P. Tan, and A. Sadek. Prohibiting Left-Turn Movements at Mid-
17 Block Unsignalized Driveways: Simulation Analysis. *ASCE Journal of Transportation*
18 *Engineering*, Vol. 4, No. 279, 2005, pp. 279–295.
- 19 39. H. Poorzahedy, and O. M. Rouhani. Hybrid meta-heuristic algorithms for solving network
20 design problem. *European Journal of Operational Research*, vol. 182, 2007, pp. 578–596.
- 21 40. Q. Chen. An optimization model for the selection of bus-only lanes in a city. *PloS one*, vol.
22 10, 2015.
- 23 41. J. Yao, F. Shi, Z. Zhou and J. Qin. Combinatorial optimization of exclusive bus lanes and
24 bus frequencies in multi-modal transportation network. *Journal of Transportation*
25 *Engineering*, vol. 138, 2012, pp. 1422–1429.
- 26 42. X. Sun and J. Wu. Combinatorial optimization of bus lane infrastructure layout and bus
27 operation management. *Advanced Mechanical Engineering*, vol. 9, 2017.
- 28 43. D. E. Goldberg. Genetic algorithms and Walsh functions: Part I, a gentle introduction.
29 *Complex systems*, vol. 3, 1989, pp. 129–152.
- 30 44. D. E. Goldberg. Genetic algorithms and Walsh functions: Part II, deception and its analysis.
31 *Complex systems*, vol. 3, 1989, pp. 153–171.
- 32 45. Baluja, S. Population-Based Incremental Learning: A Method for Integrating Genetic
33 Search Based Function Optimization and Competitive Learning. *Report No. CMS-CS-94-*
34 *163*, 1994.
- 35 46. Pelikan, M., D. E. Goldberg, and E. Cantú-Paz. *BOA: The Bayesian Optimization*
36 *Algorithm*.
- 37 47. Barceló, J., and J. Casas. Dynamic Network Simulation with AIMSUN. In *Simulation*
38 *Approaches in Transportation Analysis*, Springer, pp. 57–98.
- 39 48. Daganzo, C. F., V. V. Gayah, and E. J. Gonzales. Macroscopic Relations of Urban Traffic
40 Variables: Bifurcations, Multivaluedness and Instability. *Transportation Research Part B:*

- 1 *Methodological*, Vol. 45, No. 1, 2011, pp. 278–288.
- 2 49. Miller, B. L., and D. E. Goldberg. *Genetic Algorithms , Tournament Selection, and the*
3 *Effects of Noise*. 1995.
- 4 50. Schwarz, G. Estimating the Dimension of a Model. *Annals of Statistics*, Vol. 6, No. 2, 1978,
5 pp. 461–464. <https://doi.org/10.1214/AOS/1176344136>.
- 6 51. Henrion, M. Propagating Uncertainty in Bayesian Networks by Probabilistic Logic
7 Sampling. In *Machine Intelligence and Pattern Recognition*, North-Holland, pp. 149–163.
- 8 52. Deprator, A. J. Dynamic Traffic Control: The Treatment of Left Turns in an Urban Grid
9 Network. *MS Thesis at the Pennsylvania State University*, 2016.

10

11