Evolving Better Rerouting Surrogate Travel Costs with Grammar-Guided Genetic Programming

Takfarinas Saber* and Shen Wang*
*School of Computer Science, University College Dublin, Ireland
Email: {takfarinas.saber, shen.wang}@ucd.ie

Abstract—**The number of drivers using on-board systems to navigate through urban areas is increasing. Drivers get real time information regarding traffic conditions and change their routes accordingly. Adapting a route clearly enables drivers to avoid closed roads or circumvent major hotspots. However, given the non-linearity of the traffic dynamics in urban environments, choosing a route based only on current traffic load or current average vehicle speed is not a guaranty of a lower overall travel time. In this work, we design an evolutionary system to search for better surrogate travel cost that drivers could optimise in their rerouting to achieve better overall travel times. Our system uses the Grammar-Guided Genetic Programming algorithm to evolve surrogate travel cost expressions and evaluate their performances on a micro traffic simulator. Our system is able to evolve different expressions that meet characteristics of specific urban environments instead of a *one size fits all* expression. We have seen in our experimental study on a traffic scenario representing Dublin city centre that our system is able to evolve surrogate travel cost expressions with ∼34% and ∼10% improvements in average travel time over the no rerouting and the average travel speed based rerouting algorithms.**

Index Terms—**Evolution Computation, Grammar-Guided Genetic Programming, Simulation of Urban MObility, Traffic Rerouting, Surrogate Travel Cost.**

I. Introduction

Vehicle navigation systems have large impacts that go beyond our personal daily lives. Designing a good vehicle navigation system is crucial for enabling an efficient and sustainable development of our cities' mobility and transportation [1]. This importance attracts the attention and the resources of various stakeholders from governments to industrials, researchers, and general public drivers, among others.

An increasing number of drivers are nowadays using on-board on-line and real-time systems to navigate urban environments [2]. Vehicles (represented by either their drivers or their traffic agent systems) plan routes dynamically, and react in real-time to their received traffic information: routing directives, traffic conditions and network congestions [3].

While, there has been a large number of works in the literature that propose different vehicle navigation systems to attempt to improve drivers' travel time and increase the efficiency of city-wide road networks, routing in urban areas is still an ongoing problem [4].

Many works in the literature consider one or multiple perceived traffic network conditions as a cost to inform their routing and rerouting strategies and are called *time-dependent* [5]. For instances, Cao et al. [6] propose a routing algorithm based on a multi-agent approach which considers the real-time travel time. Pan et al. [7] propose a (re-)routing system which considers the real-time traffic load on each road. Despite the sizeable number of time-dependent works, traffic networks are dynamic and it is still not clear what cost function (if it exists) yields better total travel time.

Some more recent works propose to use historical on-road driving performance data to informed the navigation decision-making [8] or attempt to predict traffic conditions using complex features from multiple sources and combined using advance models (e.g., [9]), but this raises several issues related to data collection [10], data storage [11], data quality [12], and data processing [13].

In this work, we propose to use Grammar-Guided Genetic Programming (G3P [14]) to evolve a *time-dependent* cost function that drivers would aim to optimise (i.e., a surrogate travel cost) in their navigation in order to optimise their overall travel time. We make the assumption that drivers only have access to some basic map information and real-time traffic network conditions, and that drivers do not have access to any historical data.

This paper makes the following contributions:

- We show that the choice of the rerouting surrogate travel cost function is of a high importance as it might lead to a worse travel time than with the no rerouting strategy.
- We design a system that evolves better rerouting surrogate travel cost functions and evaluates their performances on a well-known micro traffic simulator.
- The evolved surrogate travel cost functions are in the form simple mathematical formulas which makes them more interpretable than other black/grey box strategies.
- The system could easily be used to evolve surrogate travel cost functions on an urban environment basis (thus better matching their characteristics) rather than having a one size fits all expression. For example, Dublin city centre and Manhattan have different map and traffic load characteristics. Therefore, evolving a different surrogate travel cost function for each of them seems to be more appropriate.

The remainder of this paper is organised as follows: Section II presents the background of our study. Section III defines our problem. Section IV describes our proposed system and approach. Section V presents the design of our experiments. Section VI reports and discusses our experimental results. Finally, Section VII concludes the paper.

II. Background

This section presents the background to our research in four parts: (i) routing, (ii) rerouting, (iii) micro-simulation of urban mobility, and (iv) SUMO.

A. Routing

Routing is also sometimes referred to as Route Assignment or Traffic Assignment.

Let $\mathcal{V}$ be the set of all vehicles in the system. Each vehicle $v \in \mathcal{V}$ departs from its origin $o(v)$ at the time $t_0(v)$ and attempts to arrive to its destination $d(v)$ in a way that minimises its considered total travel cost (often considered to be the travel time).

The travel time on every segment is hard to accurately quantify (as travel time on each road segment is dynamic). Therefore, other more deterministic cost functions are often employed as a surrogate cost function c with more less accuracy (e.g., length of the road segment, current travel time on the road segment, etc).

We define the route $r(v) \in \mathcal{R}$ of the vehicle $v \in \mathcal{V}$ by a list of road segments $\{s_0^v, s_1^v, ..., s_n^v\}$ such that:

- $s_0^v = o(v)$
- $s_n^v = d(v)$
- for every $i \in \{1, ..n\}$, s_{i-1}^v is connected to s_i^v by a road junction.

The objective from the perspective of every vehicle $v \in \mathcal{V}$ is then to find the route $r(v) = \{s_0^v, s_1^v, ..., s_n^v\}$ that minimises its total surrogate travel cost function $c\,(r(v))$, which corresponds to the sum of the surrogate travel cost function of each road segment $c(s_i^v)$, with $s_i^v \in r(v)$:

$$c\,(r(v)) = \sum_{s_i^v \in \{s_0^v, s_1^v, ..., s_n^v\}} c(s_i) \tag{1}$$

B. Rerouting

Let's consider a vehicle $v \in \mathcal{V}$ with an assigned route $r(v) = \{s_0^v, ..., s_i^v, ..., s_k^v, s_{k+1}^v ..., s_j^v, ..., s_n^v\}$ and located at the road segment s_k^v at the time t_k. The rerouting of the vehicle v with respect to its surrogate travel cost function c at the time t_k is a routing problem for the vehicle v from the origin s_k^v to its destination s_n^v which minimises its same surrogate travel cost function c. The rerouting of v will yield a route $r' \in \mathcal{R}'$, such that $r'(v) = \{s_k^v, s_{k+1}'^v ..., s_j'^v, ..., s_n^v\}$. In total, if we consider no further rerouting, when the vehicle v reaches its destination, it would have travelled the total route $\{s_0^v, ..., s_i^v, ..., s_k^v, s_{k+1}'^v ..., s_j'^v, ..., s_n^v\}$.

C. Micro Simulation of Urban Mobility

Simulators are commonly used in urban road traffic research as it is costly to verify new methodology under the realistic urban infrastructure. In general, there are two types of simulation in road traffic study, microscopic and macroscopic simulation. The underlying simulation unit of macroscopic simulation is the road segment. In contrast, the scope of microscopic simulation can be narrowed down to the behaviour of each different vehicles running on separate lanes of the road segment. Micro-simulation requires a lot more computation resources than macro-simulation due to its much finer granularity. However, the micro-simulation is getting more popular as researchers can have access and control to more details (e.g. vehicle types, the changes of vehicle route, etc.) to study their impact on the traffic.

D. SUMO

Simulation of Urban Mobility (SUMO) [15] is the most widely used open-source microscopic road traffic simulator. It supports various map data formats such as shapefiles and open street map. Based on these map data, SUMO can generate the quasi-realistic traffic from the demographics of a city, or the synthetic traffic from a list of configurable parameters, for instance, the number of vehicles and the distribution of origin and destination locations. The general workflow with SUMO is that researchers prepare required inputs, such as maps, traffic demands, traffic light timing plans, to start running the simulation. When the simulation is finished, the key measurements of this simulation (e.g. vehicle travel time, travel distance, fuel consumption, etc.) can be collected and analysed using tools or APIs provided by SUMO. During the execution of the SUMO simulation, with the help of Traffic Control Interface (TraCI), a Python Library based on SUMO, researchers can retrieve the simulation status on-line (e.g. the current location or speed of a particular vehicle), and apply new control policies (e.g. change the route of a vehicle) before the simulation finishes. In this study, we use SUMO combined with TraCI for evaluating vehicular rerouting strategies with different surrogate travel cost functions.

III. Problem Definition

This section describes the problem we are dealing with in this paper: designing a 'good' surrogate travel cost function that vehicles would use to inform their rerouting algorithm in order to achieve a better (lower) overall travel time.

This problem could be seen as a two level-problem where: (i) in the lower level every vehicle repetitively optimises its own surrogate travel cost function, and (ii) in the upper level we would like to optimise the average travel time.

A. Fitness Function: Upper Level

We aim at optimising the average travel time of all the vehicles in the system. Let $t(v, c)$ be the total travel time (found by SUMO) of the vehicle $v \in \mathcal{V}$ between its origin $o(v)$ and destination $d(v)$, and performing its rerouting in a way that optimises its surrogate travel cost function c. Then the fitness function ($FF(\mathcal{V}, c)$) is:

$$FF(\mathcal{V}, c) = \frac{\sum_{v \in \mathcal{V}} t(v, c)}{|\mathcal{V}|} \tag{2}$$

B. Designing Surrogate Travel Cost Function: Lower Level

Considering that our objective is to optimise the average travel time, it seems intuitive to reroute vehicles to road segments with the lowest travel times (or eventually, to shortest road segments) using the Dijkstra algorithm[16]. However in practice, travel time is constantly varying and road segment size do not inform on their load.

We would like design a better surrogate travel cost function c that vehicles could optimise from their own perspective when performing their rerouting, while improving the global fitness function (i.e., average travel time). The surrogate travel cost function that we are searching for has to be measurable for every road segment. This will enable us to exploit basic rerouting algorithms (e.g., Dijkstra[16], A*, etc.) which find the optimal route for every vehicle though the additive process of the surrogate travel cost on every road segment.

Overall, our problem is to find the best surrogate travel cost function for road segments that vehicles could optimise in their rerouting in order to achieve the lowest average travel time.

IV. Proposed Approach: Evolving Surrogate Traffic Cost Functions

We propose using a Grammar-Guided Genetic Programming (G3P [17], [18]) algorithm to evolve better rerouting surrogate travel cost functions in order to achieve better average travel times.

G3P has been successfully used to evolve algorithms to solve problems from various domains ranging from swarm algorithms design [19], to scheduling wireless communication networks [20], [21], to link allocation in 5G networks [14], to associative classification in Big Data [22].

The fitness function (in terms of average travel time) of every surrogate travel cost function c is evaluated by a SUMO simulation. For every surrogate travel cost function c, vehicles are simulated with the ability to be rerouted at every time t_r using some routing algorithm (in our case Dijkstra[16]) and by evaluating the cost of every road segment based on the surrogate travel cost function c.

A. G3P Algorithm

G3P is an algorithm that belongs to the family of Evolutionary Algorithms. G3P is an extension to Genetic Programming (GP). G3P can evolve programs/functions/expressions in any language/context that is described in a grammar–often in the Backus-Naur Form (BNF). Grammars allow us to specify the correct syntactic structure of the evolved expression. They also allow us to include expert domain-knowledge into the representation (e.g., in our case traffic related statistics).

B. Grammar Design

We would to evolve expressions that represent expressions which could be mapped to surrogate travel cost functions. Therefore, we design a grammar that enables us to formulate expressions: (i) of different sizes, (ii) that are composed of common mathematical operators, and (iii) that include various traffic related statistics.

Designing a suitable grammar is of utmost importance and a hot topic in genetic programming, with researcher proposing meta-grammars [23] and multi-level grammars [17]. In our work, we manually design a single grammar for our evolution.

Figure 1 shows the BNF grammar that we use in our work. In our BFN an expression can be composed of one of the four binary mathematical operators: addition (+), subtraction (-), multiplication (*), and protected division (pdiv, returns 0 in case of division by 0). It can also be composed of one of the three common unary mathematical operators: protected square route (psqrt, square route of absolute value), trigonometric sine (sin), and hyperbolic tangent (tanh). In our expressions, we allow two types of terminals: constants (numbers from 1 to 9) and nine road segment traffic statistics that are deterministic values that could easily be extracted/retrieved from the simulation.

Table I describes each of the nine road segment traffic statistics that are used in out BNF grammar as terminals. These traffic statistics can be extracted from SUMO for every road segment. We distinguish two types of statistics:

- Statistics that are static and do not change during the simulation: length and noNextRoadChoices(i.e. the number of next road choices).
- The rest of the statistics are changing during the simulation.

In our case, we measure the changing statistics in the time interval that starts from the time of the last rerouting (or start of the simulation) and end at the time when we conduct the new rerouting.

Note that G3P's evolved expressions are in the form of mathematical formulas. Therefore, each expression could be directly used as a surrogate travel cost function without performing any complicated mapping. Hence, we could interchangeably refer to a surrogate travel cost function as a surrogate travel cost expression.

C. Sumo-Based Cost Evaluation

Figure 2 shows an overview of our system. It shows on the left side details of our G3P algorithm. G3P receives a grammar in a BNF form. G3P randomly generates an initial population (set) of surrogate travel cost expressions and evaluates their respective fitness. Then, G3P repeatedly iterates through the selection, crossover and mutation operators to evolve newer populations until exceeding its execution time budget (or in our case, the number of generations). G3P uses SUMO for the fitness evaluation of every surrogate travel cost expression. Providing a map, a list of vehicles (with their respective origins, destinations, and starting times), and a surrogate travel cost expression, SUMO simulates the trips of vehicles with their rerouting and returns the overall average travel time to G3P. This average travel time is considered in G3P as the fitness of the given surrogate travel cost expression.

In particular, the interaction between G3P and SUMO is described as follows: when a new surrogate travel cost expression is evolved by G3P, it is sent to SUMO, along with the map and traffic demand definition to simulation the scenario. Every fixed time interval (e.g., in our case every 60 seconds), our strategy will use TraCI to retrieve the up-to-date traffic metrics shown in Table I. These metrics will be used to (i) compute the travel cost with the evolved expression, and (ii) to reroute all vehicles based on these updated travel costs. When the simulation is finished, the average travel time of the whole

```
<e>   ::= <e>+<e> | <e>-<e> | <e>*<e> | pdiv(<e>,<e>) | psqrt(<e>) | sin(<e>) |
          tanh(<e>) | <terminal>
<terminal> ::= <constant><constant>.<constant><constant>*<stat> | <stat>
<stat> ::= avgSpeed | maxSpeed | roadOccupancy | minSubsequentOccupancy |
          avgSubsequentOccupancy | maxSubsequentOccupancy | avgTravelTime |
          noNextRoadChoices | length | <constant>
<constant>  ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Fig. 1. Grammar in the Backus-Naur Form used in our Grammar-Guided Genetic Programming algorithm. The grammar include common mathematical unary and binary operators, in addition to some useful traffic related statistics as a mean to introduce expert domain-knowledge.

TABLE I
DESCRIPTION OF THE ROAD SEGMENT TRAFFIC STATISTICS THAT ARE USED AS TERMINALS IN THE GRAMMAR OF OUR GRAMMAR-GUIDED GENETIC PROGRAMMING ALGORITHM.

Traffic Statistic	Description
avgSpeed	The average speed of all vehicles that travel the road segment in the considered time period.
maxSpeed	The maximum speed of all vehicles that travel the road segment in the considered time period.
roadOccupancy	The percentage of the total length of all vehicles on the road divided by the length of that road.
minSubsequentOccupancy	The minimum occupancy of all subsequent roads to the current road.
avgSubsequentOccupancy	The average occupancy of all subsequent roads to the current road.
maxSubsequentOccupancy	The maximum occupancy of all subsequent roads to the current road.
avgTravelTime	The length of the road divided by the average speed of the vehicles running on the same road.
noNextRoadChoices	The number of down stream road segments that are adjacent to the current one.
length	The length of the road segment.

simulation is obtained and used as a fitness for expression. This cost will inform which travel cost expression is better and worth keeping for the evolution of future expressions.

At the end of this process, G3P returns the expression with the best found fitness. This expression could then be used for the rerouting in other similar situations (similar maps and vehicle lists) without a need to re-run the evolution.

V. EXPERIMENT DESIGN

In this section, we describe our experimental design in three parts: (i) the data set of traffic scenarios on which we are basing our experiments, (ii) the algorithms we are comparing against, and (iii) the setup of our system and the values defined for the parameters of our algorithms.

A. Data Set

We evaluate our proposed method on two data sets: a randomly generated map and an urban area in Dublin city centre as shown in Figure 3. The key features/characteristics of the two traffic scenarios used in our experimental evaluation are summarised in Table II.

Instead of using a standard grid map, we are using a randomly generated grid map [1] that is widely used by the SUMO community. The Random Grid traffic scenario avoids perfectly shaped/angled roads and symmetrically distributed number of junctions and roads. Therefore, it enlarges the travel cost difference for various routing choices and makes it more reasonable for us to study the impact of different rerouting strategies. In addition to the random grid map, we also use a realistic traffic scenario which encompasses a subset map of Dublin city centre that is obtained from OpenStreetMap.

[1]https://github.com/lcodeca/PyPML/tree/master/examples/random_grid

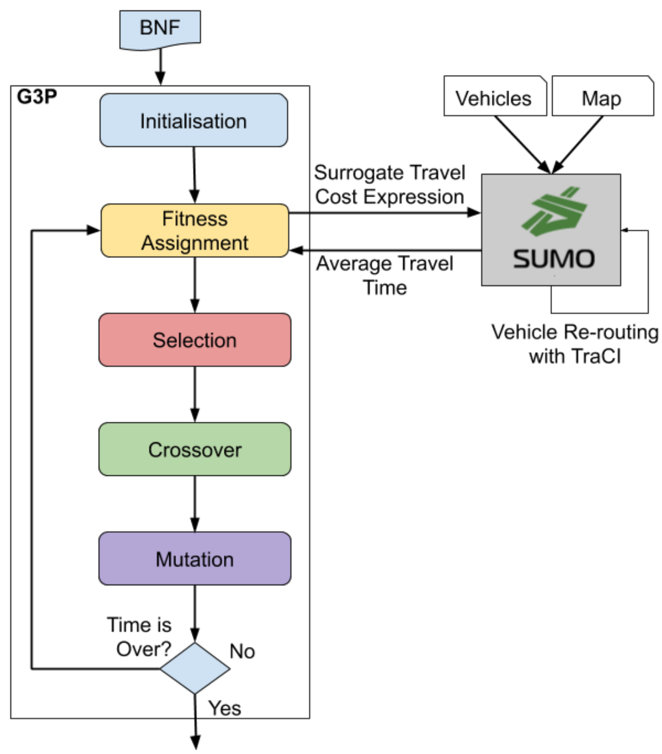


Fig. 2. Overview of our evolutionary system. On the right is the G3P algorithm which evolves surrogate travel cost expressions. On the right is the SUMO simulation process which takes in a list of vehicles (with their origins, destinations and starting travel times) and a surrogate travel cost expression for G3P, and returns the average travel time.

The traffic demand generation of these two traffic scenarios are uniformly distributed over 3600 simulation seconds. The generated traffic is neither too low nor too high, as we expect some congested roads, but not all roads are congested.

Fig. 3. Maps of the two traffic scenarios used in our experimental evaluation (left: a randomly generated Grid map; right: a subset map of Dublin city centre).

TABLE II
THE KEY CHARACTERISTIC OF TWO INSTANCE TRAFFIC SCENARIOS: A RANDOM GRID, AND A PART OF DUBLIN CITY CENTRE. WE REPORT THE NUMBER OF ROADS (# ROADS), THE AVERAGE ROAD LENGTH (AVG. ROAD LENGTH), AND THE NUMBER OF JUNCTIONS (# JUNCTIONS) IN THE MAP OF EACH TRAFFIC SCENARIO. WE ALSO REPORT THE NUMBER OF VEHICLES SIMULATED IN EACH TRAFFIC SCENARIO.

Characteristic	Random Grid	Dublin City
# roads	536	6297
avg. road length (m)	82.35	37.00
# junctions	200	3129
# vehicles	2048	1755

B. Algorithms

To evaluate the performance of the rerouting algorithm with our evolved surrogate travel cost expressions, we compare it against three other algorithms: No Rerouting, Travel Time Based Rerouting, and Occupancy Based Rerouting.

1) No Rerouting (NRR): This algorithm is the default algorithm provided by SUMO (i.e. duarouter). We consider it as a baseline as it considers no rerouting for any vehicle during the simulation. Vehicles are assigned the route with the least possible *estimated* travel time, which is calculated as the road length divided by the maximum allowed travel speed limit on this road. Once the route is determined for each vehicle, this algorithm offers no possibility to change any vehicle's route during its trip.

2) Travel Time Based Rerouting (TTBRR): This algorithm is also provided within SUMO as the default rerouting mechanism. TTBRR enables the rerouting of all the vehicles at every fixed time period (we set this period to 60 seconds similarly to our proposed approach). Vehicles are assigned their initial route in a similar way as in the NRR algorithm. Every 60 seconds, all vehicles that started their trip and did not reach their destination are rerouted. This algorithm routes the vehicles using the Dijkstra algorithm [16] with the *estimated* travel time on each road segment in the last simulation step taken as its cost. SUMO estimates this travel time using the road length divided by the average vehicle speed on this road. The possible range of this average speed value is between 0.1 m/s when all vehicles on this road are standing still, and the maximum allowed speed limit on this road when there is no vehicle. As shown in the previous work [24], this default estimation used in SUMO can lead to excessive errors when a small number vehicles waiting for red traffic light in a long road. Our system will overcome this limitation that directly lead to reduction of total travel time.

3) Occupancy Based Rerouting (OBRR): This algorithm follows the same principal as TTBRR. However, OBRR bases its rerouting on road segment occupancy to avoid overloaded roads. Therefore, the objective of every vehicle is to minimise the total route occupancy. A reminder here that a road occupancy generally means the percentage of a road that is occupied by vehicles.

Note that TTBRR and OBRR could be considered as instances (special cases) from our system. Both road travel time and road occupancy could be expressed by the grammar designed to our G3P algorithm. Therefore G3P is able to evolve them as surrogate travel cost expressions in its search process.

C. Setup

In our experiment, we use the implementation of G3P that is provided in the PonyGE 2 framework [25] with a tree representation for the evolved surrogate travel cost expressions. We also define the evolutionary parameters of G3P as shown in Table III. We generate an initial population of expressions with a maximum tree depth of 10 using the Ramped Half-Half (RHH [26]) algorithm, and evolve expressions with that same maximum tree depth (i.e., 10). For the evolutionary process, we use the sub-tree crossover with a probability 0.7 and the sub-tree mutation with a probability 0.3.

In our experiments, we set the size of the population to 30 and evolve it for 30 generations. This limited population size, number of generations and number of runs is mainly motivated by the fact that simulating the traffic with a surrogate travel cost expression in SUMO requires a non-negligible execution time (between 5 and 10 minutes). Therefore, limiting these parameters enables us to conduct our experiments in a timely manner.

However, it deserves to be noted that if there is a need to increase the population size and the number of generations, it is possible to parallelise our evolutionary process to keep the execution time reasonable (several SUMO simulations could be run at the same time as they are not inter-related within the same generation). Furthermore, in order to handle larger traffic scenarios, it is also possible the use some distributed implementation of SUMO (e.g., dSUMO [27]).

TABLE III
EVOLUTIONARY PARAMETERS USED FOR OUR G3P ALGORITHM.

Parameter	Value
Initialisation	Ramped Half-Half
Max initial tree depth	10
Overall max tree depth	10
Population size	30
Number of generations	30
Selection	Tournament
Tournament size	2
Replacement	Generational
Crossover type	Sub-tree with a 70% probability
Mutation type	Sub-tree with a 30% probability

In our experiment, we also define some simulation parameters. Particularly, we set the rerouting period to 60 seconds. Therefore, all vehicles that are travelling undergo a rerouting with the provided surrogate travel cost expression.

To limit the number of varying variables in our work, we have chosen to use the same rerouting algorithm as the one provided in SUMO, thus only leaving travel cost of the road segments as the variable element.

Rerouting algorithms such as Dijkstra[16] are known to not behave properly in presence of negative costs (the algorithms suffer from the non-halting problem). Therefore, to avoid negative surrogate travel costs affecting affecting our rerouting algorithm, we put a lower bound on the travel cost of each road segment as 0. Therefore, the surrogate travel cost of a road segment is the maximum between 0 and the result of its surrogate travel cost expression.

VI. EVALUATION

This section reports our experimental results in terms of overall performance (average travel time for all vehicles) and performance per vehicle.

A. Improvement in Terms of Average Travel Time

In this section, we would like to study the ability of the G3P approach to evolve more efficient surrogate travel cost expressions.

Table IV compares the performance in terms of average travel time (in seconds) achieved by the best surrogate travel cost expression found by G3P and other algorithms described in Section V (i.e., NRR, TTBRR, and OBRR).

We see from Table IV that G3P finds surrogate travel cost expressions that achieve the best average travel time on both traffic scenarios. G3P finds surrogate travel cost expressions that achieves 4.12% and 9.85% improvement over the best rerouting algorithm (i.e., TTBRR) on Random Grid and Dublin City respectively. G3P also finds surrogate travel cost expressions that achieve 11.99% and 34.75% improvement over NRR on Random Grid and Dublin City respectively.

TABLE IV
COMPARISON OF THE PERFORMANCE IN TERMS OF AVERAGE TRAVEL TIME (IN SECONDS) ACHIEVED BY THE BEST SURROGATE TRAVEL COST EXPRESSION FOUND BY G3P AND THE ALGORITHMS NRR, TTBRR, AND OBRR. WE PUT IN BOLD THE BEST PERFORMANCE ACHIEVED ON EACH TRAFFIC SCENARIO.

Traffic Scenario	NRR	TTBRR	OBRR	G3P
Random Grid	1632.39	1498.41	2035.18	**1436.59**
Dublin City	426.25	308.50	882.62	**278.11**

We also see from Table IV that the choice of the surrogate travel cost function has a drastic impact on the average travel time in comparison with NRR. We clearly see that using travel time as the surrogate travel cost function in TTBRR achieves a better performance than using occupancy in OBRR (26.37% on Random Grid and 65.04% on Dublin City). Furthermore, the rerouting algorithm TTBRR achieves a better performance than NRR (8.20% on Random Grid and 27.62% on Dublin City) whereas OBRR achieves a worse performance than NRR (19.79% on Random Grid and 51.70% on Dublin City).

Figures 4 show the evolution of performance in terms of average travel time (in seconds) of the best surrogate travel cost expression at every generation of the G3P algorithm on both considered traffic scenarios.

We see that G3P successfully evolves surrogate travel cost expressions with better fitness (average travel time in second) over generations on both traffic scenarios.

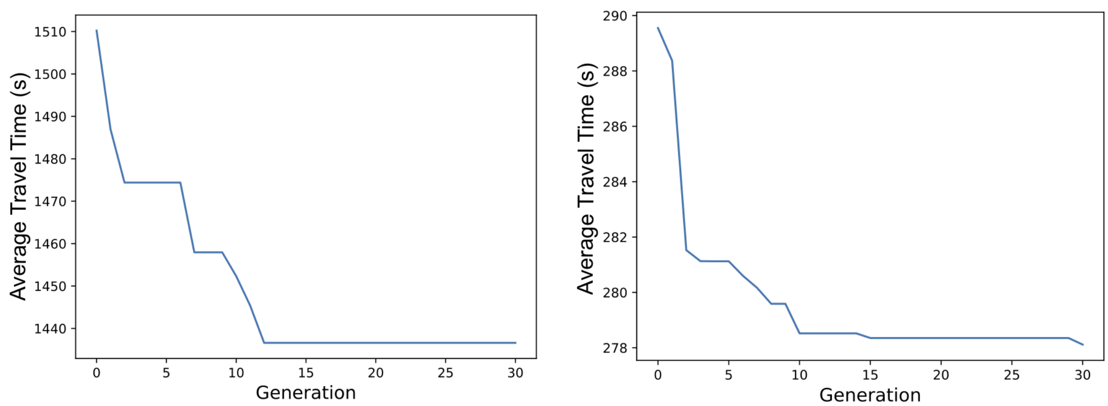
We also see that G3P generates initial populations which contain at least one surrogate travel cost expression with a better performance than NRR and OBRR on both traffic scenarios. These initial populations also contain surrogate travel cost expressions that are better than TTBRR on the Dublin City traffic scenario and expressions with a performance close to TTBRR's performance on Random Grid one. This clearly shows that outperforming the standard (re-)routing algorithms is not a very hard task and even a random algorithm is able to achieve it.

After the initialisation, G3P achieves a steep improvement of the fitness function over the first third of the generations. Then, the performance stabilises over the following two thirds of the generations. However, on the Dublin City traffic scenario, we see that G3P continues to find some improved surrogate travel cost expressions even at the end of the evolution, which might indicate that the algorithm did not fully converge on this traffic scenario.

B. Impact on Vehicle Travel Time

Beyond improving average travel time, we would like to improve travel time for the largest number of vehicles and not drastically worsen the travel time for the rest (in comparison to the no rerouting algorithm). In this section, we would like to evaluate the evolution of travel time on a vehicle basis.

For each traffic scenario, we compare the changes of trip time (or travel time) for every vehicle under different rerouting strategies. In particular, we choose NRR as the baseline, then present the results of the trip time difference for each vehicle



Random Grid Dublin City

Fig. 4. Evolution of performance in terms of average travel time (in seconds) of the best surrogate travel cost expression at every generation of the G3P algorithm on both considered traffic scenarios.

between each rerouting strategy (i.e., TTBRR, OBRR, and G3P's best evolved expressions) and the baseline NRR.

As shown in Table V, in the Random Grid traffic scenario, our G3P rerouting system has improved most of vehicles' trip time under all main statistics, though a little less trip time improvement (i.e. 1.68%) than TTBRR in terms of the largest trip time reduction. The maximum trip time sacrifice for a single vehicle using G3P is 16.11% less than TTBRR, and 52.24% than OBRR. OBRR is the worst rerouting system that can only improve trip time for a small portion of vehicles while a lot other vehicles have to increase their trip time considerably. Similar conclusions can be drawn from the results obtained on the Dublin City traffic scenario shown in Table VI, on which our G3P rerouting system demonstrates its superiority over all main percentile statistics, including the largest trip time reduction.

Our findings can also be confirmed by the kernel density distribution of the trip time difference for each vehicle between each rerouting system and baseline system NRR. As shown in Figure 5, on both traffic scenarios, our G3P system (in blue color) is the most left-skewed and has the shortest right tail, which implies that it improves trip time for the majority of vehicles, while only increases trip time for a small number of vehicles.

TABLE V

ON RANDOM GRID TRAFFIC SCENARIO, THE COMPARISON OF THE TRIP TIME DIFFERENCE (IN SECONDS) FOR EACH VEHICLE UNDER TTBRR, OBRR, AND G3P, IN WHICH NRR IS THE BASELINE. A NEGATIVE VALUE INDICATES THAT VEHICLE TRIP TIME IS REDUCED UNDER A CERTAIN REROUTING STRATEGY. A POSITIVE VALUE INDICATES THAT THE TRIP TIME FOR A SPECIFIC VEHICLE IS INCREASED AFTER BEING REROUTED.

Statistic	TTBRR-NRR	OBRR-NRR	G3P-NRR
mean	-133.98	402.80	**-195.68**
min	**-2257.00**	-2044.00	-2219.00
25%	-294.25	-12.00	**-355.00**
median	-86.00	280.50	**-134.00**
75%	8.25	771.25	**-20.00**
max	2291.00	4024.00	**1922.00**

TABLE VI

ON DUBLIN CITY TRAFFIC SCENARIO, THE COMPARISON OF THE TRIP TIME DIFFERENCE (IN SECONDS) FOR EACH VEHICLE UNDER TTBRR, OBRR, AND G3P, IN WHICH NRR IS THE BASELINE. A NEGATIVE VALUE INDICATES THAT VEHICLE TRIP TIME IS REDUCED UNDER A CERTAIN REROUTING STRATEGY. A POSITIVE VALUE INDICATES THAT THE TRIP TIME FOR A SPECIFIC VEHICLE IS INCREASED AFTER BEING REROUTED.

Statistic	TTBRR-NRR	OBRR-NRR	G3P-NRR
mean	-117.75	456.37	**-148.14**
min	-2960.00	-2359.00	**-2960.00**
25%	-82.00	6.00	**-99.50**
median	0.00	176.00	**-1.00**
75%	1.00	624.50	**1.00**
max	721.00	4321.00	**209.00**

VII. CONCLUSION AND FUTURE WORK

In absence of large data sets of traffic historical data and with the arduousness of designing efficient prediction algorithms, leveraging time-dependent traffic metrics is the solution for optimising drivers' travel time. However, choosing a surrogate travel cost function that combines available traffic metrics for travel time is still an open challenge for the traffic rerouting in urban environments, especially given that the urban environments differ drastically from a city to another. Therefore, designing a surrogate travel cost function on a city basis seems to be more appropriate.

We designed a G3P system to evolve surrogate travel cost functions on the urban environment basis. For more accuracy between in-lab and applied results, G3P evaluates the performance of its surrogate travel cost expressions on a well-known simulator for urban mobility.

We evaluated our evolved surrogate travel cost functions on two traffic scenarios. We showed that our system is able of evolving surrogate travel cost functions with ~34% improvements in average travel time over the no rerouting algorithm on a Dublin city traffic scenario. We also showed that our system able to evolve a ~10% better surrogate travel cost function than travel time for the rerouting algorithm on the same Dublin city centre traffic scenario.

Additionally, when compared with TTBRR and OBRR, we have seen that G3P rerouting system was able to improve the

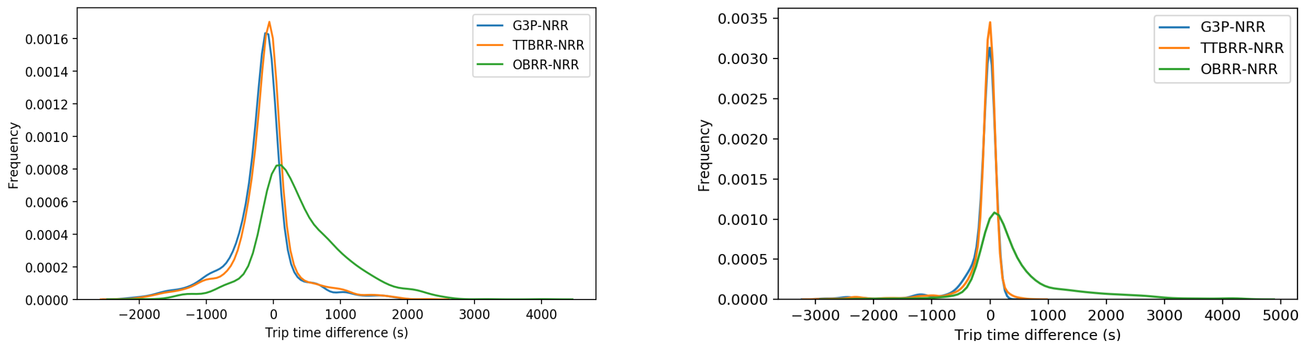


Fig. 5. The distribution of trip time difference for each vehicle between each rerouting strategy (G3P, TTBRR, OBRR) and the baseline NRR.

travel time for the largest group of vehicles, and only a small number of vehicles needed to sacrifice a little of their travel time to achieve this overall gain.

Our future work will investigate means to include a temporal dimension into the grammar to inform our routing algorithm about the pertinence of cost value at various stages of the trip.

References

[1] Z. Cao, H. Guo, J. Zhang, and U. Fastenrath, "Multiagent-based route guidance for increasing the chance of arrival on time," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[2] T. Saber, A. Ventresque, and J. Murphy, "Rothar: Real-time on-line traffic assignment with load estimation," in *Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*. IEEE Computer Society, 2013, pp. 79–86.

[3] C. Caprioli, M. Bottero, and M. Pellegrini, "An agent-based model (abm) for the evaluation of energy redevelopment interventions at district scale: An application for the san salvario neighborhood in turin (italy)," in *International Conference on Computational Science and Its Applications*. Springer, 2019, pp. 388–403.

[4] A. Paricio and M. A. Lopez-Carmona, "Urban traffic routing using weighted multi-map strategies," *IEEE Access*, vol. 7, pp. 153 086–153 101, 2019.

[5] A. Agafonov and V. Myasnikov, "Efficiency comparison of the routing algorithms used in centralized traffic management systems," *Procedia engineering*, vol. 201, pp. 265–270, 2017.

[6] Z. Cao, H. Guo, and J. Zhang, "A multiagent-based approach for vehicle routing by considering both arriving on time and total travel time," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 3, pp. 1–21, 2017.

[7] J. S. Pan, I. S. Popa, and C. Borcea, "Divert: A distributed vehicular traffic re-routing system for congestion avoidance," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 58–72, 2016.

[8] J. Liu and A. Khattak, "Informed decision-making by integrating historical on-road driving performance data in high-resolution maps for connected and automated vehicles," *Journal of Intelligent Transportation Systems*, pp. 1–13, 2019.

[9] L. Li, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Traffic speed prediction for intelligent transportation system based on a deep feature fusion model," *Journal of Intelligent Transportation Systems*, pp. 1–12, 2019.

[10] J. Handscombe and H. Q. Yu, "Low-cost and data anonymised city traffic flow data collection to support intelligent traffic system," *Sensors*, vol. 19, no. 2, p. 347, 2019.

[11] H. E. Ciritoglu, T. Saber, T. S. Buda, J. Murphy, and C. Thorpe, "Towards a better replica management for hadoop distributed file system," in *IEEE BigData Congress*, 2018, pp. 104–111.

[12] M. Bendechache, S. Limaye Nihar, and R. Brennan, "Towards an automatic data value analysis method for relational databases," in *ICEIS*, 2020.

[13] Z. Ullah, F. Al-Turjman, L. Mostarda, and R. Gagliardi, "Applications of artificial intelligence and machine learning in smart cities," *Computer Communications*, 2020.

[14] D. Lynch, T. Saber, S. Kucera, H. Claussen, and M. O'Neill, "Evolutionary learning of link allocation algorithms for 5g heterogeneous wireless communications networks," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019, pp. 1258–1265.

[15] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 2575–2582.

[16] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[17] T. Saber, D. Fagan, D. Lynch, S. Kucera, H. Claussen, and M. O'Neill, "A multi-level grammar approach to grammar-guided genetic programming: the case of scheduling in heterogeneous networks," *Genetic Programming and Evolvable Machines*, vol. 20, no. 2, pp. 245–283, 2019.

[18] T. Saber, D. Fagan, D. Lynch, S. Kucera, H. Claussen, and M. O'Neill, "Multi-level grammar genetic programming for scheduling in heterogeneous networks," in *European Conference on Genetic Programming*. Springer, 2018, pp. 118–134.

[19] P. B. Miranda and R. B. Prudêncio, "A novel context-free grammar for the generation of pso algorithms," *Natural Computing*, pp. 1–19, 2018.

[20] T. Saber, D. Fagan, D. Lynch, S. Kucera, H. Claussen, and M. O'Neill, "Hierarchical grammar-guided genetic programming techniques for scheduling in heterogeneous networks," in *CEC*, 2020.

[21] T. Saber, D. Fagan, D. Lynch, S. Kucera, H. Claussen, and M. O'Neill, "A hierarchical approach to grammar-guided genetic programming: the case of scheduling in heterogeneous networks," in *International Conference on Theory and Practice of Natural Computing*. Springer, 2018, pp. 225–237.

[22] F. Padillo, J. Luna, and S. Ventura, "A grammar-guided genetic programing algorithm for associative classification in big data," *Cognitive Computation*, pp. 1–16, 2019.

[23] M. O'Neill and A. Brabazon, "mgga: The meta-grammar genetic algorithm," in *European Conference on Genetic Programming*. Springer, 2005, pp. 311–320.

[24] S. Wang, S. Djahel, J. McManis, C. McKenna, and L. Murphy, "Comprehensive performance analysis and comparison of vehicles routing algorithms in smart cities," in *Global Information Infrastructure Symposium - GIIS 2013*, Oct 2013, pp. 1–8.

[25] M. Fenton, J. McDermott, D. Fagan, S. Forstenlechner, E. Hemberg, and M. O'Neill, "Ponyge2: Grammatical evolution in python," in *GECCO*, 2017, pp. 1194–1201.

[26] C. Ryan and R. M. A. Azad, "Sensible initialisation in grammatical evolution," in *GECCO*, 2003, pp. 142–145.

[27] Q. Bragard, A. Ventresque, and L. Murphy, "Self-balancing decentralized distributed platform for urban traffic simulation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1190–1197, 2016.