An effective EA for short term evolution with small population for traffic signal optimization

Rolando Armas, Hernán Aguirre, Fabio Daolio and Kiyoshi Tanaka
Faculty of Engineering, Shinshu University. 4-17-1 Wakasato, Nagano 380-8553, Japan.
Email: rolandoarmas@gmail.com, {ahernan, fdaolio, ktanaka}@shinshu-u.ac.jp

***Abstract*—In this work, we study the effects of mutation operators combined with a varying mutation schedule applied to traffic signal optimization. An evolutionary algorithm with specialized mutation operators coupled with a microscopic traffic simulator tackles the optimization of traffic signal settings in different mobility scenarios. Experimental results show that the proposed mutation operators allow for an effective search in large decision spaces, evolving small populations for a short number of generations. The parameters of the evolutionary algorithm are analysed and automatically-generated configurations are discussed, suggesting alternative ways to effectively apply the proposed variation operators for short term evolution.**

I. Introduction

According to UN [1], 54% of the world population currently lives in urban areas. Inhabitants in those cities demand acceptable levels of transportation service to accomplish their activities every day. However, the infrastructure is not growing at the same pace of the demand, which causes severe economic, social and environmental problems. In large urban areas, especially in developing countries, citizens tend to use more their own mobility means, increasing the number of vehicles [2] and contributing to worsen the mobility problem. A proper setting of traffic signals can help to alleviate the traffic congestion with better use of the current transportation network infrastructure. Several parameters define a traffic light control system: first, the cycle length. Next the green times of the different phases or traffic flows, and finally the offset between the beginning of the cycles of consecutive signals to coordinate them and induce continuous flows. The search space is vast if we consider the number of signals, especially in wide geographical areas. Regularly, urban planners deal with the optimal settings and they are restricted to analyze few signals in a small part of the zone of study. Evolutionary algorithms (EA) have become a popular method to optimize large and complex search spaces. In this work, we optimize traffic signal settings to minimize travel time. We explore alternative signal settings under various scenarios of mobility by using activity based micro-simulation, which allows for a detailed analysis of the system. Previous works have been proposed for signal timing optimization. Several techniques ranging from statistical based methods in the 60's [3] [4] to computational intelligence (CI) oriented to implement intelligent transportation systems have been used. In the CI field, in particular, among evolutionary computation (EC) algorithms, genetic algorithms (GA) appear as favored optimizer technique. One early work [5] proposed a GA optimizer and a mesoscopic traffic simulator. This work showed that applying GA allows a reduction in average delay by using a small test scenario consisting of four closely spaced intersection. Another approach [6] considered travel demands on a small network using a macroscopic simulation of traffic flows. The solution representation considers a common network cycle time for all of them. In [7] a network design problem (NDP) bi-level approach is proposed, where one common cycle for each solution was used. The reason is that to maintain signal coordination, each junction in the considered area must operate with a common cycle or a simple multiple of it [8]. This assumption simplifies the complexity of the problem, but it is not applicable to wide areas due to traffic system and network particularities for each neighborhood. We have relaxed the assumption of a common cycle to enable to build microzones. However, the evaluation of several traffic lights is computationally expensive due to the number of signals and also because each solution considers the mobility plan and travel details of a synthetic population of agents which are moving across the area of study. This imposes limits on the number of fitness evaluations, population size, the number of generations, and on the number of times the stochastic evolutionary algorithm can be run. Hence, this work aims to develop an EA to search efficiently in large decision spaces under a small budget of generations, performing a reliable short-term evolution to find high-quality solutions. To achieve this, we design a set of specialized mutation operators combined with a deterministic varying schedule [9] to enable a fast convergence in short-term evolution.

The rest of this paper is organised as follows. In Section II, we describe the simulation method we use. In Section III we detail the evolutionary algorithm components. Section IV is devoted to the experiments and analysis of results. Finally, Section V gives concluding remarks and future work.

II. Method

Figure 1 shows the interaction of the components of the optimization system. The model components are the mobility plan, the transportation network infrastructure and the evolutionary algorithm (EA). The mobility plan is an activity-based model considering three primary activities as destination locations: work, study and other. We use Multi-Agent Transport Simulation (MATSim) [10] to evaluate the mobility scenario. MATSim requires as inputs the initial mobility plans for a set of agents and a model of the road network infrastructure. It simulates traffic following the initial plans of the agents trying various routes and iterates to optimize individual plans and routes for all agents to provide a system in an equilibrium state. Equilibrium state is defined as *a stable condition when no traveler can improve his travel time by unilaterally changing routes* [11].

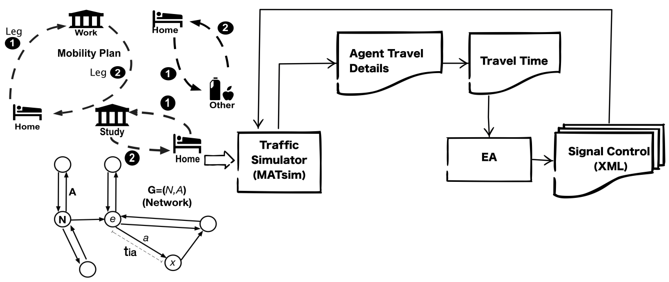

Fig. 1: Model Components

To run a scenario with traffic lights, MATSim simulates traffic lights microscopically using fixed-time controls [12]. That means that traffic lights parameters are set statically beforehand. We use the evolutionary algorithm to find optimal signal settings of a transportation system to minimize average travel time. The EA evolves a population of candidate solutions, each solution represents a configuration of signals (signal control) for the transportation system. At each iteration, the EA calls the transport simulator for each candidate solution in order to evaluate it. Once all solutions are evaluated, the evolutionary algorithm continues to the next iteration.

III. Evolutionary Algorithm

The optimizer is an elitist evolutionary algorithm. In the following we detail the representation, main steps, and operators.

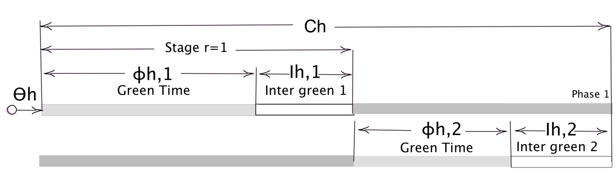

Fig. 2: Traffic light components

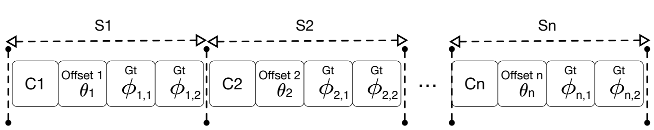

Fig. 3: Chromosome representation

1) Traffic Signals Problem Representation: The principal components of a traffic signal are cycle length, phase, offset, stage, green and inter-green time. *Cycle length* is the time in seconds required for one complete color sequence of the signal. A *phase* is the set of movements that can take place simultaneously. An *Offset* is the time lapse in seconds between the beginning of a corresponding green phase at an intersection and the beginning of a corresponding green phase at the next intersection. One *stage* is a green and inter-green time sequence (see Figure 2). In this work we extend the representation used in [7] to include a cycle per signal. In addition we use integer instead of binary representation. We do not expect meaningful changes in travel time, setting the signals with a fraction of seconds. Thus, a signal *S* in junction *h* is represented by a set of **integer** variables expressed by

$$S_h = (C_h, \theta_h, \phi_{h,1}, \cdots, \phi_{h,r}), \quad (1)$$

where C_h is cycle length, θ_h is the offset, and $\phi_{h,1}, \cdots, \phi_{h,r}$ are the green times for the r phases of the signal. Signal S_h represents one *gene*, and a set of signals constitute the *chromosome* of an individual, i.e. a solution with the complete specification of all signals considered in the system. Figure 3 illustrates the representation of a solution to a system with n signals, each one with two phases. The ranges and constraints of these variables are given in Eq.(2) – Eq.(8), where $I_{h,r}$ is the inter-green time at signal *h* for phase *r* and P_h is the total number of phases at signal *h*. Equations Eq.(2) – Eq.(4) represent the range for cycle length Ch, offset θ_h and green time $\phi_{h,r}$, respectively. C_{hmin} is determined by identifying the signal that needs the longest duration just to accommodate the inter-green times and the minimum green times as shown in Eq.(5). C_{max} is set to 135 seconds. Inter-green per phase is 3 seconds and minimum green time duration is 17 seconds for all signals as shown in Eq.(6). These values imply that the minimum cycle time C_{min} is 40 seconds in two phase signals. Eq.(7) ensures that the sum of the green times in a signal together with inter-green do not exceed the cycle length set for the signal. Eq.(8) establishes the maximum green time for the signal phase based on the cycle time, inter-green and minimum green time.

$$C_{hmin} \quad \leq \quad C_h \quad \leq \quad C_{hmax} \quad (2)$$

$$0 \leq \theta_h \leq C_h - 1 \quad (3)$$

$$\phi_{h,rmin} \leq \phi_{h,r} \leq \phi_{h,rmax} \quad (4)$$

$$C_{min} = Max_{h=1,2...,N} \left\{ \left(\sum_{r=1}^{P_h} \phi_{h,r} + \sum_{r=1}^{P_h} I_{h,r} \right) \right\} \quad (5)$$

$$\phi_{h,rmin} = 17 \quad sec \quad \forall h, r \quad (6)$$

$$C_h = \sum_{r=1}^{P_h} \phi_{h,r} + \sum_{r=1}^{P_h} I_{h,r} \quad \forall h \quad (7)$$

$$\phi_{h,rmax} = C_h - \sum_{r=1}^{P_h} I_{h,r} - \sum_{y=1 y \neq r}^{P_h} \phi_{h,ymin} \quad (8)$$

Table I shows the range of the variables in seconds.

TABLE I: Variable Range Constraints (s)

Var	min	max
C_h	40	135
$\phi_{h,r}$	17	65
θ_h	0	134
$I_{h,r}$	3	3

2) Main Steps of the Algorithm: Initial Population: We use a population P of size 20. The initial population is created deterministically, covering the full range values of the cycle length, as follows. We prepare 20 cycle lengths in the range [40, 135] seconds in steps of 5. All solutions are set with a different cycle length, but all signals of a solution are set to the same cycle length. The offset times of all signals are set to zero and green times per phase are set to the same value according to the cycle length, i.e. green time = (cycle length - inter-green) /2 for a two phase signal. That is, all signals are synchronized to start at the same time but are not coordinated.
Parent Selection: Individuals are selected for reproduction using binary tournaments among randomly sampled solutions.
Recombination and Mutation: The offspring population Q is created applying crossover to the selected parents with probability P_c followed by mutation. A first configuration of the algorithm uses a step mutation operator per variable of the signal. ${P_m}^{(Ct)}$, ${P_m}^{(Ot)}$, and ${P_m}^{(Gt)}$ are mutation probabilities for cycle, offset and green times, respectively. To mutate, we first decide which mutation operator will be applied using the probabilities of the operators. Then we apply the chosen mutation operator with probability P_m per signal. A second configuration combines a neighborhood mutation operator with step mutation. A third configuration enhances the second one with a deterministic varying schedule for the mutation probability P_m.
Evaluation: To evaluate each individual we first run the traffic simulator. The parameters of the simulation are the signals settings contained in the variables of the individual. The fitness value of the individual is calculated from the output of the simulation using Eq.(10).
Survival Selection: The $nElite$ best individuals from the current population P and the offspring population Q are combined. For the next generation, we select the best $|P|$ individuals from this combined population.

A. Crossover

In this work we implement one and two point crossover taking each signal as an atomic unit. The crossing point is selected randomly with equal probability in the range [1,*n* - 1], where n is the number of signals. Then the crossover operator interchanges complete signals between parents.

B. Step Mutation Operators

Cycle length mutator (CyM): This operator increases or decreases randomly with equal probability the cycle length of a signal using step size $stepCt$. If the new cycle length is out of the specified range, we adjust it accordingly to be either C_{hmin} or C_{hmax}. After that, it is necessary to check whether offset time violates its constraint. If offset is larger than the new cycle length, it is reset to new cycle length - *stepOff*, where *stepOff* is the offset step size. Finally, for each signal phase the green times are adjusted proportionally to the new cycle length. Due to the correlation of offset and green times to the cycle length, this operator may act as a macro-mutation operator.
Green time mutator (GtM): This operator decreases the green time of one phase and adds it to another phase using step size $stepGt$. To determine the phase that will decrease its green time, we randomly visit the phases until we find one in which the decrement does not violate the constraint for minimum green time $\phi_{h,rmin}$. The phase to which the green time is added is also determined randomly among all phases, except the one in which time was reduced.
Offset time mutator (OffM): This operator increases or decreases randomly with equal probability the offset time of a signal using step size *stepOff*. If offset becomes negative, it is reset to 0. Likewise, if offset is greater than the maximum cycle length C_{hmax}, it is reset to C_{hmax} - *stepOff*.
The effects of the above operators tested under several settings were reported in [13], [14].

C. Neigborhood mutation operators

Neighborhood operators aim to improve traffic flow along the two main directions of circulation, namely, South-North-South (SNS) and West-East-West (WEW). The idea is to favor traffic signal coordination along the two directions by simultaneously modifying the parameters of a signal and its neighbors. In particular, cycle times are copied and offsets are adjusted taking into account the distance and the average free speed between the interacting signals. For each signal S_h we pre-calculate the distance d and the average free speed to its two neighbors along each propagation direction. At execution time, when the operator is applied, the propagation direction is stochastically selected with probability 0.85 for NSN direction and 0.15 for WEW direction, respectively, in agreement with the characteristical traffic flow and the most common mobility patterns in the city. Notice that, since in our encoding each parameter of a signal corresponds to one decision variable, the new operators are forcing variable interactions in the decision space. Their implementation is detailed as follows.
Cycle length mutator and neighborhood cycle propagation (NCyP): This operator propagates the cycle length (C_h) of the reference signal (S_h) selected for mutation to its neighborhood. C_h can be either mutated or not previous its propagation. The procedure for this operator is listed in Algorithm 1. There, if the parameter *ctmut* is true, C_h is first mutated with the above operator CyM and then propagated. Otherwise, if *ctmut* is false, C_h is propagated without changing it first.
Cycle length mutator and neighborhood cycle propagation with distance-based offset (NCyOffP): This operator, similar to the above operator NCyP, propagates the cycle length (C_h) of the reference signal (S_h) selected for mutation to its neighborhood. In addition, it sets the offsets of the neighboring signals based on the time required to cover the distance from

S_h to the neighbor traveling at free speed (tff). As in NCyP, C_h can be either mutated or not previous its propagation. The procedure for this operator is listed in Alg. 2.

Algorithm 1: NCyP($S_h, ctmut$)

Data: $S_h = (C_h, \theta_h, \phi_{h,1}, \cdots, \phi_{h,r})\,, ctmut$

phase=getPhase(h);
N=getNeighborhood($h, phase$);
if *ctmut=true* **then**
    CyM(h)
end
for $i = 0$ *to* $|N|$ **do**
    changeSignalSettings($n_{hi}.id, C_h$);
end

Algorithm 2: NCyOffP($S_h, ctmut$)

Data: $S_h = (C_h, \theta_h, \phi_{h,1}, \cdots, \phi_{h,r})\,, ctmut$

offsetprevious=θ_h;
phase=getPhase(h);
N=getNeighborhood($h, phase$);
if *ctmut=true* **then**
    CyM(h)
end
for $i = 0$ *to* $|N|$ **do**
    timefreeflow=$\frac{d(S_h, n_{hi})}{avgfreespeed(S_h, n_{hi})}$;
    newoffset=offsetprevious + timefreeflow;
    changeSignalSettings($n_{hi}.id, C_h$,newoffset);
    offsetprevious=offsetprevious+newoffset;
end

D. Varying mutation schedule

Varying mutation operators allow to modify strategy parameters during the run of the algorithm. The combination of a high selection pressure introduced by elitism together with varying mutation operators have been shown to improve the convergence speed of genetic algorithms [9], [15]. In this work we implement a time-dependent schedule that deterministically varies mutation rate P_m in a hyperbolic shape [9] expressed by

$$P_m{}^{(t)} = \left(\frac{1}{P_m{}^{(0)}} + \frac{\frac{1}{P_m{}^{(T)}} - \frac{1}{P_m{}^{(0)}}}{T-1} t \right)^{-1}, \qquad (9)$$

where T is the maximum number of generations, $t \in \{0, 1, ..., T-1\}$ is the current generation, $P_m{}^{(0)}$ and $P_m{}^{(T)}$ are the desired mutation probabilities per signal at time 0 and T, respectively. This approach has been used to reduce mutation rates, i.e. $P_m{}^{(0)} > P_m{}^{(T)}$. In our case, we also use it to investigate the effect of increasing mutation rate by setting $P_m{}^{(0)} < P_m{}^{(T)}$.

E. Fitness Function

MATSim sets the configuration of the signals with solution passed by the evolutionary algorithm, simulates the movements of the agents following the plans in the equilibrium state, and outputs the time it takes each agent to travel each link in its route. We minimize average travel time expressed by

$$\min \quad \frac{\sum_{i=1}^{V} \sum_{l=1}^{L} t_{il}}{V} \quad , \qquad (10)$$

where t_{il} is the travel time on link l for vehicle i, V is the number of vehicles being simulated and L is the number of links in the network, subject to signal timing design and feasibility constraints shown in Eq.(2)- Eq.(8) [7].

IV. Simulation Results and Discussion

A. Experimental Setup

For each type of experiment, we conduct ten runs of the algorithm, setting the number of generations to 50, using different random seeds but starting from the same initial population. Population size is 20 and the number of elite individuals is $nElite$=10. We set crossover rate to $P_c = 1.0$. When constant mutation is applied, mutation rate per signal is set to $P_m = 4/n$, where n is the number of signals. For varying mutation we set its range to $[20/n, 4/n]$. In our experiments $n = 70$. For neighborhood operators, a neighborhood per direction (SNS and WEW) for each signal is set in advance according to its geographical location. The neighborhood includes the next and previous signal to the reference signal. Step sizes for mutation operators are $stepCt = 5$, $stepGt = 3$ and $stepOff = 10$. Mutation probabilities per operator are detailed in Table II. The simulation in MATSim takes approximately 5 hours to reach the equilibrium state and 3 minutes to evaluate each solution.

B. Initial Mobility Plans

We model the initial mobility plans based on activities [16]. In this work the agents perform two trips or legs. One from home to their activity destination and then the return trip from their activity to home, as illustrated in the top left side of Figure 1. The number of simulated agents with their home located in a given district is proportional to the actual population of that district. The home coordinates of the agents are assigned sampling uniformly these areas. We consider three types of activity: *work*, *study*, and *others* such as leisure, business, shopping, etc. The proportion of agents for each activity is 32%, 33% and 36% respectively according to mobility survey data [17]. The location coordinates for each type of activity are determined using different procedures.

We have defined a range of starting times and activity durations according to the type of activity. The activity duration and start times are assigned sampling uniformly from these ranges. In this work, the simulation scenario considered a large part of the city (5x8 Km^2) where 70 traffic signals were allocated. We simulate the mobility of 20.000 agents. This

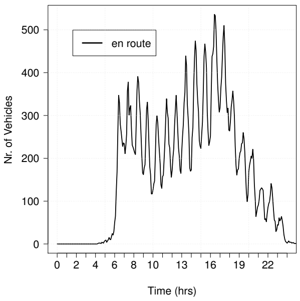

Fig. 4: Mobility Scenario S124h

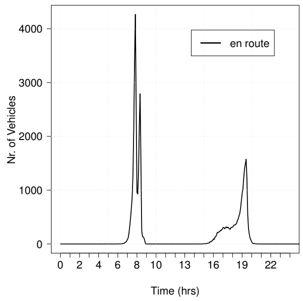

Fig. 5: Mobility Scenarios S2M and S2A

number represents approximately the 30% of the estimated number of vehicles for the inhabitants of the area of study [17] and allows to perform the simulation in a reasonable computation time. With this number of agents, MATSim requires approximately 8 hours to reach the equilibrium state and around 2.5 minutes per individual to compute its fitness. We create three different scenarios. In the first scenario the trips are distributed from early morning to late evening, i.e. 06:00h to 24:00h, as illustrated in Figure 4. In the following we refer to this scenario as *S124h*. In the second and third scenarios the trips are distributed from 06:00h to 09:00h and from 15:00h to 20:00h, respectively, as illustrated in Figure 5. These mobility plans aim to study the system under congested scenarios during morning and afternoon peak hours. In the following we refer to those scenarios as *S2M* and *S2A*, respectively. Since all scenarios simulate the mobility of 20,000 agents, the less congested scenario is *S124h* and the most congested one is *S2M*.

C. Effects of Operators on One Day Scenario

In this section, we analyze the effect of step mutation operators, neighborhood operators, and varying mutation on mobility scenario S124h. We perform five different experiments and group them in three sets using the settings shown in Table II.

Figure 6 shows the transition of mean travel time in all runs over the generations for all experiments. Figure 7 shows box plots of the best solution at the last generation and Table III shows their mean, standard deviation and inter-quartile range.

First, we look at the first set of experiments E0 and E1. E0 is a reference experiment where step mutation operators are used for cycle, offset, and green time [14]. E1 uses the neighborhood cycle propagation operator $NCyP$ instead of the step cycle operator CyM. These experiments have the same settings of mutation operators probability (see Table II). These operator probability settings gave the best travel time when only step mutation operators were used [14]. Note that the travel time transition by E0 and E1 are similar as shown in Figure 6. However, in experiments E1, where cycle propagation is applied, a considerable reduction of solutions variance can be achieved as shown by the inter-quartiles in

TABLE III: Mean travel time, standar deviation and interquartile range of best solutions in experiment E0-E4 and E4DVM.

Exp.	Travel Time		
	avg	std	iqr
E0	810.83	77.42	107.31
E1	811.89	107.75	51.37
E3	776.44	30.64	34.04
E4	749.34	20.36	26.53
E4DVM	**746.50**	**17.43**	**21.20**

Figure 7 and Table III. Yet, some outliers are still observed and their effect can be noticed in the standard deviation.

In the second set of experiments E3 and E4 we include the NCyOffP operator, instead of the step cycle and step offset mutation operators, and analyze the effect of cycle propagation and offsets defined by the distance between neighbors. E3 emphasizes cycle and offset propagation over green time step mutation and E4 vice versa, as shown by the operator probability for cycle and green time in Table II. Note from E3 and E4 in Figure 7 that when cycle propagation with offsets is included, variance reduced further and better travel times can be achieved. Emphasis on green times (E4) eliminates outliers at the end of the run giving even better results than emphasis on cycle propagation (E3), marked in bold in Table III. Looking at Figure 6 we note that emphasis in mutation of green times (E4) also allows for faster convergence. This is because the cycle propagation rapidly finds appropriate values for cycles, but reduces diversity in the population and loses its effectiveness to create new solutions in later stages of the search. Thus, a smaller rate for cycle propagation combined with a relatively larger constant rate for exploration of green times leads to faster and better convergence.

In the final experiment E4DVM we apply configuration E4 with a deterministic varying mutation schedule in the range $P_m = [20/n, 4/n]$ instead of using constant mutation rate $P_m = 4/n$. Figure 8 shows the expected number of mutated signals by $P_m^{(Ct)}$ and $P_m^{(Gt)}$ when constant and varying mutation are applied. From Figure 6 note that a very fast convergence is achieved when the deterministic varying mutation schedule is applied to E4. Note also that travel time, standard deviation and inter-quartile range reduce further compared to E4.

These results show that the combination of cycles propagation, setting of offsets based on distance, emphasis on mutation of green times, and the application of a varying mutation schedule works well to explore a large space and quickly finds good solutions for the problem at hand.

D. Varying Mutation on Denser Scenarios

In this section we analyze the behavior of the algorithm in scenarios where there is a higher volume of traffic in shorter periods of time compared to the scenario used in the previous section. Here we focus on E4 and E4DVM configurations to verify the search ability and convergence of the algorithm under congested situations. Figure 9 shows the

TABLE II: Algorithm experimental setup

Exp.	Op.Prob. $P_m^{(Ct)}, P_m^{(Ot)}, P_m^{(Gt)}$	Operators Ct	Off	Gt	Observations
Constant mutation $P_m = 4/70$					
E0	0.5,0.3,0.2	CyM	OffM	GtM	Step operators
E1	0.5,0.3,0.2	NCyP	OffM	GtM	Cycle propagation, step offset and green time
E3	0.7, – ,0.3	NCyOffP	–	GtM	Cycle prop. with offsets, step green time
E4	0.3, – ,0.7	NCyOffP	–	GtM	
Deterministic varying mutation $P_m^{(t)} = [P_m^{(0)} = 20/70, P_m^{(T)} = 4/70]$					
E4DVM	0.3, – ,0.7	NCyOffP	–	GtM	Cycle prop. with offsets, step green time

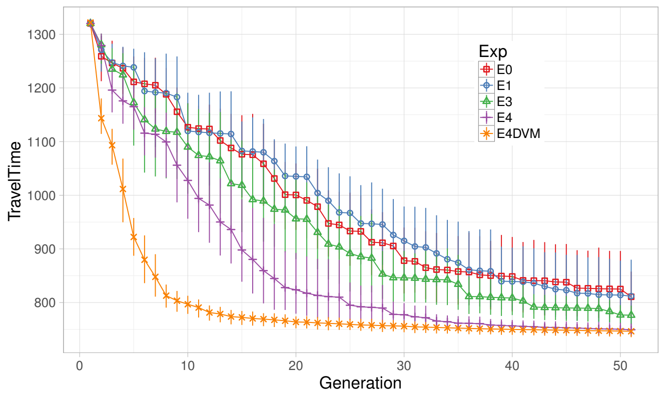

Fig. 6: Mean travel time over generations in experiments E0 - E4 and E4DVM.

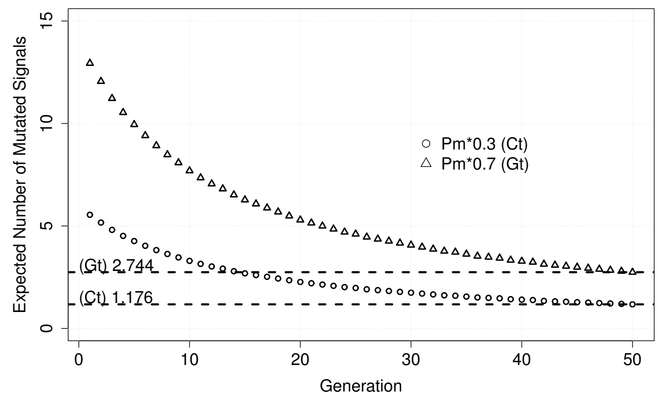

Fig. 8: Expected number of mutated signals with constant and varying mutation schedules (E4,E4DVM)

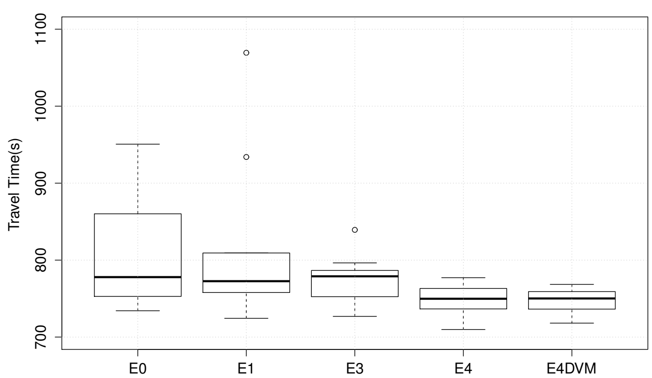

Fig. 7: Mean travel time of best solutions in experiments E0 - E4 and E4DVM.

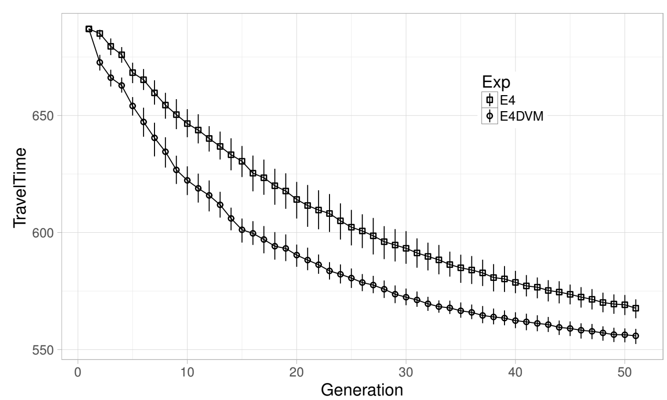

Fig. 9: Mean travel time over generations S2M scenario (E4, E4DVM).

transition of mean travel time over the generations by the E4 configuration with constant and deterministic varying mutation (DVM) schedules on scenario S2M. Here 20,000 vehicles move from home to the activity destinations, all trips take place from 06h00 to 09h00 with a high peak between 07h30 and 08h30, as shown in the left side of Figure 5. Similarly, Figure 10 shows results on scenario S2A, where the same vehicles move back home, but the trips are distributed in a larger period of time from 15h00 to 20h00, as shown in the right side of Figure 5. Note that E4DVM shows faster and better convergence results than E4 with constant mutation. Also, note that a larger performance difference in terms of travel time is seen for congested scenarios.

E. Crossover Effect

In this work we focus mostly on the design of appropriate mutation operators and their schedule. However, the effect of crossover should also be considered. In all previous experiments we use one-point crossover. It is well known that one-point crossover could be too disruptive, particularly in

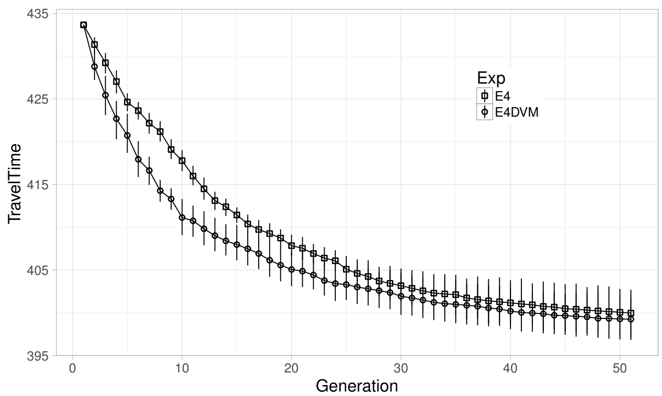

Fig. 10: Mean travel time over generations S2A scenario (E4, E4DVM).

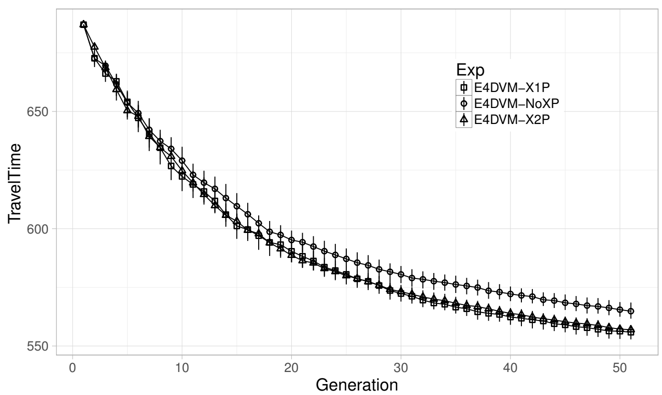

Fig. 11: Effect of crossover operator on mean travel time.

rugged landscapes [18], [19]. In this section we verify the performance of the algorithm with no crossover, one-point crossover and a less disruptive two-point crossover. Figure 11 shows the transition of the average travel time over generations for configuration E4DVM changing the crossover operator on the densest scenario S2M. From this figure note that the use of crossover helps to achieve better convergence, however, there is no significant difference between one-point and two-point crossover. It could be interesting in the future to study specialized crossover operators for this problem.

F. Algorithm's Parameters Analysis

The proposed algorithm combines a strong selection pressure given by elitism with higher varying mutation. In the following we verify if the parameter settings of the algorithm are appropriate and whether there are other configurations that can lead to similar or better results.

We analyze the parameters settings of configuration E4DVM with respect to algorithm performance using Sequential Model-based Algorithm Configuration (SMAC) [20]. SMAC constructs a model to predict algorithm performance based on supervised machine learning and random forests. The analyzed parameters are the initial $P_m^{(0)}$ and final mutation probability $P_m^{(t)}$ of the deterministic varying mutation schedule, cycle mutation probability P_m^{Ct}, and crossover rate P_c. The green time mutation probability is determined by $1-P_m^{Ct}$. The algorithm's observed performance is travel time.

We performed three independent runs, 50 evaluations each. The number of training instances was set to 10 and the number of threes on random forest to 100. Table IV shows the values of the parameters corresponding to the SMAC solutions

Figure 12 shows the average travel time transition of SMAC solutions compared with E4DVM on scenario S2M . As can be observed, convergence is similar in all experiments.Figures 13 and 14 show the expected number of mutated signals in cycle and green time, respectively. That is, $P_m^{(t)} \times P_m^{(Ct)} \times 70$ and $P_m^{(t)} \times (1 - P_m^{(Ct)}) \times 70$.

There are three schedule trends. Sol1 decreases the expected number of mutations, Sol2 increase them and Sol3 keep them almost constant. In all cases, the number of expected cycle time mutations are lower than the expected number of green time mutations. Although Sol1 and E4DVM reduce mutations with time, mutations in Sol1 are much higher than in E4DVM. However, note that crossover rate is lower in Sol1 than in E4DVM.

When deterministic varying mutations are used, typically the schedule reduces mutations with time [9], [15], such as E4DVM and Sol1. On the contrary, Sol2 is a search strategy with a schedule that increases mutations with time, giving at the end of the run relatively more emphasis to cycle time than to green times compared to the other SMAC solutions. However, note also that crossover rate in Sol2 is the smallest.

Sol3 uses a schedule in which the expected number of cycle mutations approaches 1, the expected number of green time mutations is high, almost half the number of signals, and crossover rate is the second lowest. This suggests that in this problem constant mutation with small rate for cycle time and very high rate for green times is an alternative to varying mutation.

Regarding crossover, note that SMAC favors configurations that apply crossover with rate in the range [0.5, 0.6], which is different to the rate 1.0 used in E4DVM. Since the proposed algorithm applies crossover with probability P_c followed by mutation, in E4DVM with $P_c = 1.0$ offspring is always created by crossover followed by mutation. On the other hand, in SMAC's suggested configurations only 50–60 percent of the offsprings are created applying varying mutation after crossover. The rest of the offsprings are created by the varying mutation operator alone. To summarize, E4DVM always applies crossover with relatively smaller range of varying mutations, whereas SMAC suggests configurations with higher varying mutations but smaller rates of crossover. These are scenarios that help reduce interference between crossover and high mutation [15].

V. Conclusion

This work presented an evolutionary algorithm for short-term evolution with small populations for a traffic signal

TABLE IV: SMAC solutions: EA parameters

	P_c	P_m^0	P_m^T	$P^{(Ct)}$	tt avg.
Sol1	0.60	0.93	0.63	0.19	564.10
Sol2	0.50	0.28	0.61	0.29	554.52
Sol3	0.55	0.47	0.52	0.027	558.83
E4DVM	1.00	0.29	0.06	0.3	555.90

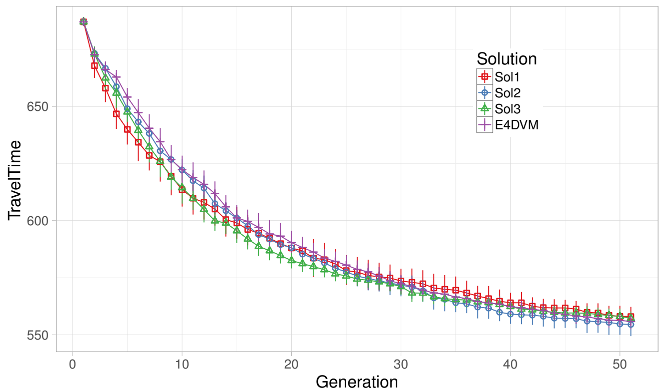

Fig. 12: Mean travel time, scenario S2M.

optimization problem. We used the multi-agent transport simulator MatSim to study the optimization of 70 traffic signal controls spread over a wide area using three different mobility scenarios with 20.000 agents moving in the city. The proposed algorithm combines a strong selection pressure given by elitism with crossover followed by varying mutation. The combination of step mutations, propagation of cycles to neighboring signals, setting of offsets based on distance between adjacent signals, and a varying mutation schedule works well to explore a large search space and converge quickly to good solutions. The crossover was shown to help convergence of the algorithm, but no significant difference between one and two point crossover was found. We conducted an analysis of parameters of the algorithm using sequential model based algorithm configuration (SMAC) that showed that alternative configurations applying relatively higher varying mutations but smaller rates of crossover lead to similarly good results. In the future, we would like to include multi-objective formulations for the multi-modal transport network (car, bike, public transportation) and study ways to improve the sustainability of transportation and mobility systems.

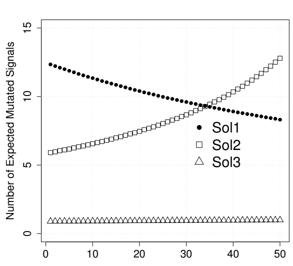

Fig. 13: Mutation schedule for cycle time $P_m^{(t)} \times P_m^{(Ct)} \times 70$

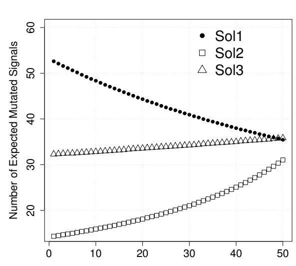

Fig. 14: Mutation schedule for green time $P_m^{(t)} \times (1 - P_m^{(Ct)}) \times 70$

Acknowledgment

Rolando Armas gratefully acknowledges the support of National Secretariat of Higher Education, Science, Technology and Innovation of Ecuador.

References

[1] Department of Economic United Nations and Population Division Social Affairs. World urbanization prospects: The 2014 revision, (2014).

[2] R. Gakenheimer. Urban mobility in the developing world. *Transportation Research Part A: Policy and Practice*, 33:671–689, 1999.

[3] F. V. Webster. Traffic signal setting. *Road Res. Lab., HMSO, London, U.K., Tech.*, (Paper 39):1–44, (1958).

[4] A. J. Miller. Settings for fixed-cycle traffic signals. *Operational Research Quarterly*, vol. 14(no. 4):373–386, (1963).

[5] B. Park, C. Messer, and T. Urbanik. Traffic Signal Optimization Program for Oversaturated Conditions: Genetic Algorithm Approach. *Transportation Research Record*, 1683(1):133–142, (1999).

[6] H. Ceylan and M. Bell. Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing. *Transportation Research Part B: Methodological*, 38(4):329–342, (2004).

[7] F. Teklu. A Genetic Algorithm Approach for Optimizing Traffic Control Signals Considering Routing. *Computer Aided Civil and Infrastructure Engineering*, 22:31–43, (2006).

[8] R. P. Roess, Prassas E., and McShane W. *Traffic Engineering*. Pearson Education, fourth edition, (2011).

[9] T. Back and M. Schutz. Intelligent mutation rate control in canonical genetic algorithm. In *Lecture Notes on Artificial Intelligence*, volume 1079, pages 158–167, Springer,(1996).

[10] Nagel K Horni, A and K W Axhausen, editors. *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 2016.

[11] J G Wardrop. Road paper. some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, (1952).

[12] D. Grether and A. Neumann. Traffic light control in multi-agent transport simulations. Technical report, Transport Systems Planning and Transport Telematics, Technical University Berlin, (2011).

[13] R. Armas, H. Aguirre, and K. Tanaka. Effects of mutation and crossover operators in the optimization of traffic signal parameters. In *The Tenth International Conference on Simulated Evolution and Learning (SEAL)*, volume 8886 of *Lecture Notes in Computer Science*, pages 167–179, Springer,(2014).

[14] R. Armas, H. Aguirre, S. Zapotecas, and K. Tanaka. Traffic signal optimization: Minimizing travel time and fuel consuption. In *Proc. in 12th International Conference, Evolution Artificielle (EA'2015), Lyon, France*, volume 9554 of *Lecture Notes in Computer Science*, Springer,(2015).

[15] H. Aguirre and K. Tanaka. A study on parallel varying mutation in deterministic and self-adaptative gas with 0/1 multiple knapsack problems. *IPSJ Transactions on Mathematical Modeling and Its Applications*, 45:77–91, 2 (2004).

[16] Castiglione J., Bradley M., and Gliebe J. *Activity-Based Travel Demand Models: A Primer*. Transportation Research Board, (2015).

[17] DMQ. Encuesta domiciliaria de movilidad (edm11) del distrito metropolitano de quito (dmq), (2012).

[18] W. Spears. *Evolutionary Algorithms: The Role of Mutation and Recombination.* Springer-Verlag Berlin, (2000).

[19] H Aguirre and K . Tanaka. A study on the behavior of genetic algorithms on nk-landscapes: Effects of selection, drift, mutation, and recombination. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E86-A(9):2270–2279, 09 (2003).

[20] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proc.of LION-5*, pages 507–523, (2011).