# Optimal Ramp Metering with GA based Parameter Optimization

Bhargava Rama Chilukuri (Corresponding author)

Georgia Institute of Technology

Department of Civil and Environmental Engineering

Atlanta, GA 30332-0355

Email: bchilukuri3@gatech.edu

Jorge A. Laval, Ph.D.

Georgia Institute of Technology

Department of Civil and Environmental Engineering

Atlanta, GA 30332-0355

Email: jorge.laval@ce.gatech.edu

Angshuman Guin, Ph.D.

Georgia Institute of Technology

Department of Civil and Environmental Engineering

Atlanta, GA 30332-0355

Email: angshuman.guin@ce.gatech.edu

Prepared for the Transportation Research Board, 2015

Words: 4750 (excluding references)

Figures and Tables: 4 +4

Total: 4750+ 8\* 250 = 6750

# Abstract

This paper presents a Genetic Algorithm based optimization framework to generate optimal parameter values for the ramp metering algorithms. The framework includes a micro-simulation model that replicates freeway capacity drop phenomena and a Genetic Algorithm based optimization module. The framework enables executing multiple instances of the micro-simulation application in parallel, which helps obtaining faster and finer results. This framework was tested on a real-world corridor using SWARM ramp metering algorithm with the objective to minimize total travel time. The analysis indicated that four parameters are critical for SWARM performance and that they are sensitive to geometric and traffic variables on a corridor. The results indicated that the proposed framework successfully determined optimal parameter values that reduced travel time by 11.6%, compared to the default values.

# Introduction

Ramp meters are used to control the vehicular flow entering the freeway. The primary benefits of the ramp metering come from preventing capacity drop and exit blockage [1]. Other benefits of ramp metering include diversion to alternate routes, enhanced safety, and reduced emissions. Over the years researchers have developed several field implemented ramp metering algorithms which can be broadly divided as local ramp metering algorithms and System-wide algorithms. The former optimize the performance at an isolated ramp, while the latter optimizes a group of ramps.

After the pioneering work of Wattleworth and Berry [2] who developed a local ramp control based on static model and historic demands, other researchers developed algorithms with similar assumptions [3, 4]. Later, Papageorgiou et.al [5] developed a dynamic feedback based control called ALINEA that is traffic responsive to local conditions. Stephanedes [6] developed the Zone algorithm that is based on traffic conservation in the vicinity of the ramp. Zhang et al. [7] developed Neural control algorithm that used feedback regulation and kinematic wave theory.

Several system-wide algorithms were developed based on varying philosophies to efficiently manage the freeway-ramp system. Seattle Bottleneck algorithm [8] calculates local metering based on demand and supply at the ramp and the global metering based on the volume reduction needed at the bottleneck. The more restrictive of the two controls is selected. METALINE [9] was developed as an extension of ALINEA and is based on the proportional-integral state feedback. The metering rate is calculated based on the changes in the occupancy. Helper ramp algorithm [10] balances the queues on ramps by distributing the metering rates so that none of ramps have excessively long queues. Linked-ramp algorithm [11] is based on the demand-capacity flow instead of commonly used occupancy/density which results in poor performance during congested conditions. The System Wide Adaptive Ramp Metering (SWARM) [12] control operates at two levels: global control which includes a forecasting and apportionment algorithm based on the predicted density at the bottleneck. Local control determines metering rate based on a local traffic-responsive ramp metering algorithm. Similar to the Bottleneck algorithm the more restrictive metering is selected for each ramp. Stratified Zone Metering [13] was developed to overcome the drawbacks of the Zone algorithm and calculates the metering rates from the freeway conditions as well as the ramp conditions (demand and queue size) with an aim to balance the freeway efficiency and ramp delay.

One common feature of the ramp metering algorithms (both local and system-wide) is the tunable parameters which are critical to their operational performance. Calibration of these parameters generally performed in the field using trial and error method. However, this method does not always ensure global optimal because of the limitation of the resources. To overcome this, Yang et. al. [14] developed a Genetic Algorithm (GA) based parameter optimization framework using PARAMICS [15]. They used the framework to determine optimal values for four parameters of ALINEA. Unfortunately, PARAMICS does not have in-built models to accurately replicate congested traffic dynamics such as capcity drop and relaxation phenomenon. Moreover, the optimization module in that framework performs a serial evaluation which significantly limits its scalability. This is evident from the case study where they limit the population size and number of generations to 10 each. Moreover, this framework also limits the granularity of the results. This is evident from the results that show a large ranges for the optimal values (e.g. KR: 70~200).

To overcome these limitations, we propose a micro-simulation based optimization framework that includes GTsim [16], a micro-simulation application that accurately models capacity drop and other congested conditions and a GA based optimization module that enables executing multiple instances of the micro-simulation application in parallel. The objectives of this paper are: 1) to build an optimization framework that handles large solution spaces, 2) and demonstrate its efficacy by generating optimal parameter values for the SWARM algorithm.

The reminder of this paper is organized as follows: section 2 provides a brief overview of GA optimization method. Section 3 describes the GA based optimization framework and its components. Section 4 describes a case study to demonstrate the efficacy of the proposed framework. Section 5 and 6 describe the results and conclusions respectively.

# Genetic Algorithm

GA belong to a class of evolutionary algorithms that mimic natural selection process. They are popularly used for searching optimal solutions for combinatorial problems. Though these algorithms were used in some elementary form since 1950, they were made popular through the studies of John Holland at University of Michigan during 1970s [17]. Their reduced susceptibility to getting stuck with local minima compared to other solution searching algorithms such as gradient search methods made them very attractive. However, these algorithms are computationally expensive depending on the shape of the fitness landscape.

GA can be broadly described as starting with an initial solution subset (called population), evaluating its performances, and selecting the subsequent subsets (next generation) based on the performance of current solution set. Generally, the initial population set is selected from the solution set using a random process. However, the size of population set is a user defined parameter. Literature qualitatively states that for larger solution space, larger population set is desired. Once the population set of generation I is evaluated, it’s best performing individuals are used to determine the next generation using two process; crossover and mutation.

Crossover is a process of selecting two or more well performing individuals and producing a new individual by selecting the parameter values from the parents. Thus, the crossover mechanism guarantees preserving the good features of the parent. Mutation generates a parameter value that is from neither of the parents. Thus mutation mechanism helps avoid getting trapped in local minima and randomly generates new solutions. The GA process is generally stopped based on a termination criterion or after arriving at a satisfactory solution.

There are a number of specific attributes of GAs that give them an edge over other traditional optimization techniques. These are:

1. GA works with a subset of solution space (not a single point) to avoid being trapped at a local optimum.
2. Working on a pool of solutions allows GA to have a parallel implementation architecture.
3. GA does not need the derivative of the objective function to determine the optimum. It only needs that the objective function be able to be evaluated for any individual point from the solution space. Thus, it is ideal for complex system such as traffic flow where vehicle interactions make it impossible to model it analytically.
4. Abu-Lebdeh and Benekohal [18, 19] showed that as the size of the solution space increases exponentially, the time requirements for the GA grow only linearly.

GAs have limitations as well:

1. Due to the randomness inherent to GA and the dependence of GA parameters on the type of problem, analytical analysis of GA is not possible.
2. GA is slower than methods that work with derivatives.
3. Global optima is guaranteed only with appropriate parameters. Therefore, meta-optimization is recommended.
4. Terminal criteria is not the same for all GA problems. Some of the commonly used criteria are number of generations, generational rate of improvement or computational time.

# Optimization Framework

Figure 1 illustrates the GA based optimization framework developed in this paper. Two main components of this framework are the GA optimizer module and GTsim-Ramp Metering Algorithm (GTsim-RMA) module. During every generation, the GA optimizer module will launch multiple instances of GTsim-RMA modules equal to the size of the population defined by the user. The optimizer will pass a unique individual (from the population) to each of the GTsim-RMA modules, which return the total travel time. Based on this performance, GA optimizer will generate population for the next generation and the process continues till the terminal criteria is met.

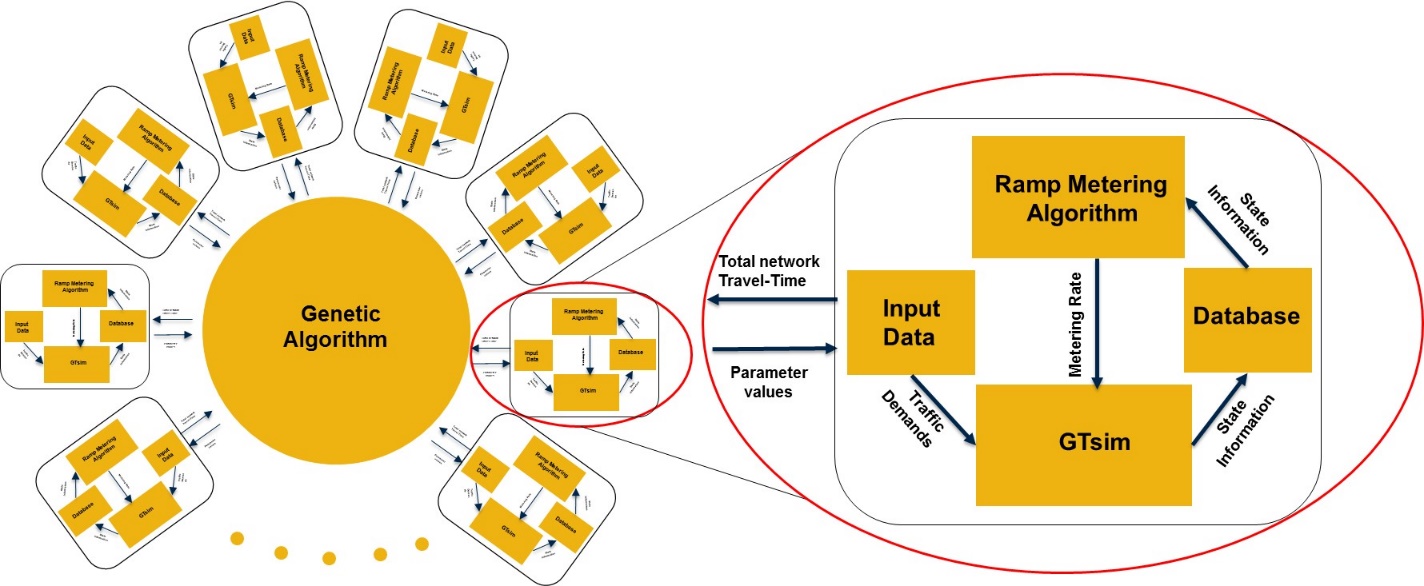


Figure 1: Optimization Framework

On the other hand, GTsim-RMA module includes four sub-modules. The GTsim is a micro-simulation application developed at Georgia Tech [16]. GTsim includes the latest advancements in lane changing models that are capable of explaining congestion dynamics such as capacity drop [20]. GTsim was built in JAVA to perform faster than real-time simulation. For example, a 3-hr simulation of a 5 mile corridor during peak congestion could be simulated in under 3 minutes on a 2.66 GHz processor with 2GB RAM. The time dependent traffic demand as origin-destination flows will be provided by the input data sub-module that feeds GTsim. The loop detector data that collects speed and volume from several loops on the corridor simulated in GTsim is stored in the database. This information is utilized by the Ramp Metering Algorithm module to determine real-time metering rates at different ramps. This information will be transmitted back to GTsim which implements the metering rate on the corresponding ramps in the simulation.

The key to parallel execution of GTsim is the creation of JAR file. It is well known that a standalone executable package, .JAR file, can be created for any Java application [21]. The JAR file aggregates class files, associated metadata, and resources (text, images, etc.) into one file for ease of software distribution. Moreover, each execution of the JAR file creates a separate JAVA runtime environment so that the variables and their values are not shared. This enables GA optimizer to initiate multiple instances of GTsim on any portable computer without any conflicts.

# Case Study

## Study corridor

The I-285 EB/SB corridor between GA-400 and I-85 was selected for this study. It is 6.5 miles long and has five entry locations (called origins for the OD terminology) and eight exit locations (called destinations) as shown in figure 2.

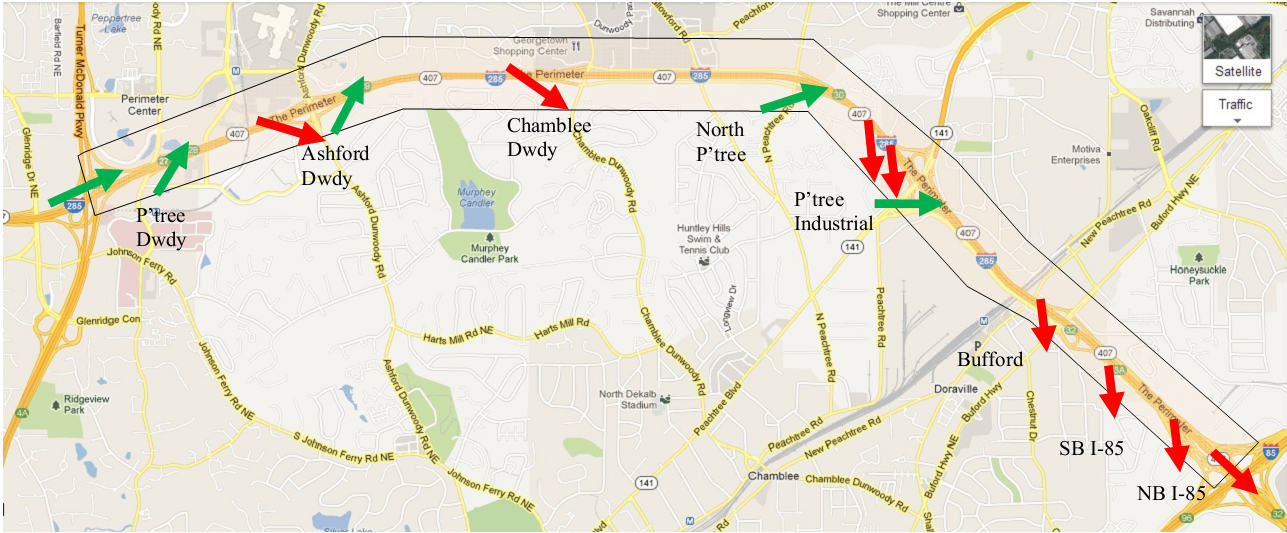


Figure 2: Case study corridor (background: courtesy www.maps.google.com)

It was found that the the quality of data obtained from GDOT’s video detection system (hereafter referred to as the NaviGAtor data) is suitable for use in a micro-simulation based applications. Therefore, the research team manually collecting data. GDOT’s PTZ cameras were used to record all the entry and exit points on the corridor and the videos were processed manually to extract the time-series of traffic volumes on the study corridor. More details on the data collection, congestion characteristics, and calibration of the model can be found in Chilukuri et. al. [16].

The corridor was simulated for 90 minutes (15:00 to 16:30) that included congestion build-up and completely congested conditions. The OD flows were automatically updated every 5 minutes. The flow and velocity information at detector stations prescribed in the model are continuously stored in a databased.

The framework is tested on the SWARM system-wide ramp metering algorithm that had several tunable parameters. A brief overview of this algorithm is presented in the next section.

## SWARM Algorithm

The System Wide Adaptive Ramp Metering (SWARM) control was developed by National Engineering Technology (NET) Corporation [12]. SWARM, a coordinated control algorithm, operates at two levels: global control (SWARM1) and local control (SWARM2). SWARM1 includes a forecasting and system-wide apportioning algorithm that determines metering rate based on the predicted density at the bottleneck. SWARM2 determines metering rate based on a local traffic-responsive ramp metering algorithm.

In SWARM, a freeway network is divided into continuous segments, whereby each segment contains one bottleneck. There may be one or more on-ramps and off-ramps in one segment. Both the local and global metering rates are computed for every time interval for each ramp, and the more restrictive one out of the two is adopted in the field.

The global control algorithm, SWARM1, forecasts future traffic conditions of the bottleneck based on immediate past traffic data. This algorithm focuses on solutions to prevent real-time density from exceeding saturation levels for each segment. To forecast future density around the bottleneck, SWARM1 performs a linear regression on immediate past traffic data [22]. The extent of past data used and time-span into the future forecast, are tunable parameters. Once the future density is obtained, excess density (the portion of future density above the pre-determined threshold saturation density at the bottleneck) can be calculated to get the target density for the next metering cycle.

The volume reduction (positive if local density is greater than target density) is distributed to upstream on-ramps within the system according to the tunable parameters “Intersection Propagation Factor” and “Intrasection Propagation Factor”.

The local control algorithm SWARM2 operates based on local traffic conditions near each ramp. According to the SWARM high level design documentation [22], a predetermined metering table based on density is needed. Since such a table is not available, the research team used ALINEA for local control. Based on the algorithms provided in [22], we developed and integrated a SWARM module with GTsim for this research.

## Meta-Optimization

Since SWARM has several tunable parameters and the solution space is large, the selection of GA parameter values is critical for attaining global optima. Meta-optimization was performed for 5 parameters; Population size, Tournament Size, Crossover probability, Mutation probability, and Termination criteria. The parameter optimization performed in this study involved two steps; identifying high impact parameters and finding optimal values for those parameters. Therefore, two sets of parameter values are developed as shown in Table 1. More details on the two steps of the parameter optimization are discussed in section 4.4

**Table 1. GA parameter values**

|  |  |  |
| --- | --- | --- |
| GA Parameters | Impact Parameters | Optimal Values |
| Population size | 600 | 150 |
| Tournament size | 30 | 10 |
| Crossover probability | 0.5 | 0.5 |
| Mutation probability | 0.4 | 0.1 |
| Termination Criteria | 75 generations | 90 generations |

## Critical Parameters

The SWARM document [22] lists more than 20 tunable parameters that controls the operational performance of the algorithm. While some of the parameters are required for data processing and data smoothing, other parameters are used in the core SWARM algorithms. The “status threshold” parameters that are related to quality of the real-time data are excluded from this study. Tables 2 show a list of 14 parameters examined in this study.

MaxPoints: This is used by the data smoothing algorithm to calculate the average flow and speed at a detector station for a desired time period.

Min/Max Vol, Min/Max Speed, Min/Max Den: These parameters are used as the thresholds to update the saturation density of the bottleneck.

Saturation Density: This is used to calculate “excess” flow forecasted at the bottleneck.

Q: Bottleneck saturation density is updated using a weighted average of “old” saturation density and “new” density that satisfies the Min/Max Volume, Min/Max Speed, and Min/Max Density thresholds.

DenSampSize: SWARM algorithms use linear regression to forecast density at the bottleneck. Historic data is used for the prediction model. This parameter describes the number of historic data points that needs to be used for forecasting. This is an important parameter since an excessively large sample size will result in smoothing out the short-term patterns, and an excessively small sample size will magnify the impact of local fluctuations.

ForecastLeadTime: The forecasting model estimates the density at the bottleneck and determines the metering rate based on the predicted traffic congestion levels. The amount of time into the future that the algorithm needs to forecast is an important parameter, since an excessively short time period into future will result in frequent rate changes, and an excessively long time period will be inefficient.

Intersection propagation and Intrasection propagation factors: Once the density at the bottleneck is forecasted and required density is determined, SWARM calculates metering rate starting at the downstream ramp closest to the bottleneck. Excess values are calculated for each ramp and are transferred to the upstream ramp using the Intersection propagation factor. Similarly, excess is transferred across section using the Intrasection propagation factor.

UpRate and DownRate: The metering rate determined every 20 seconds by SWARM algorithms are bounded by these values.

Table 1: SWARM Parameters studied for optimization in this study

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Variable Name** | **Description** | **Range** | **Default Value** |
| Data Smoothing | MaxPoints | Sample size of points to average | 3-15 | 6 |
| Saturation Density Calculation | Min Vol | Minimum volume threshold | 4-6 veh | 5 veh |
| Max Vol | Maximum volume threshold | 8-12 veh | 10 veh |
| Min Speed | Minimum speed threshold | 40-56 kph | 48 kph |
| Max Speed | Maximum speed threshold | 58-81 kph | 72 kph |
| Min Density | Minimum density threshold | 9-28 vpkm | 22 vpkm |
| Max Density | Maximum density threshold | 28-56 vpkm | 50 vpkm |
| Saturation Density | Initial saturation density | 22-41vpkm | 55 vpkm |
| Q | Sat. Density Smoothing parameter | 0-1 | 0.02 |
| SWARM1 Forecasting and Required Density | DenSampSize | Number of polls in the past to use for the forecast | 3-51 | 30 |
| ForecastLead Time | Number of polls into the future to forecast | 3-42 | 15 |
| SWARM1 Apportionment | Intersection propagation factor | Propagate “excess” between ramps | 0-1 | 0.85 |
| SWARM2 algorithm | MaxUpRate | Maximum rate at which the metering rate can increase per lane | 20-900 vph | 300 vph |
| MaxDownRate | Maximum rate at which the metering rate can decrease per lane | 20-900 vph | 60 vph |

It can be seen that with large ranges of acceptable values for each of the 14 parameters, the solution space becomes extremely large (~ 212,468,465,664,000,000 combinations). Determining the optimal set of parameter values from this huge set is extremely time consuming even with any sophisticated search algorithm. Therefore, parameters that have a significant influence on the SWARM performance, here after called impact parameters, have to be identified and separated from the rest (here after called no-impact parameters).

Separating impact parameters from no-impact parameters was done using GA parameter values shown in the second column of Table 1. GA optimizer was run for 75 generations that converged to a local minima. It was observed that out of the 45,000 individuals evaluated, 28,713 individuals (parameter value combinations) produced the same travel time. Then, those 28,713 individuals were examined for the repeating parameter values. It was observed that only 4 parameters had one value each for all the 28,713 individuals and the other 10 parameters accepted all possible values. Based on this observation, it was hypothesized that the following 4 parameters are the impact parameters:

1. UpRate,
2. DownRate,
3. SatDen,
4. InterSectionPropagationFactor

The other 10 parameters are hypothesized to be non-impact parameters. A closer look at these 10 parameters indicated that this may be an artifact of two factors; the simulation outcomes and traffic patterns on the corridor. The inherent outcomes/assumptions of simulation such as good quality data, continuous stream of data, homogenous traffic conditions across lanes, comparable driver behaviors and vehicle characteristics, etc. may be the reason for max points, max speed, min speed, and Q to end up as non-impact parameters. Similarly, steady change in traffic conditions on this corridor may be responsible for the other 6 parameters to end up as non-impact parameters. This analysis can be performed again once the simulation is made more realistic and/or traffic patterns change to ensure these parameters are truly non-impact parameters.

# Results and Discussion

## Optimal Parameter Values

The optimal values of the 4 impact parameters were determined using the GA optimizer by using default values for no-impact variables. The 15th percentile values of all the generations are plotted in Figure 3 that shows the convergence of GA over generations.

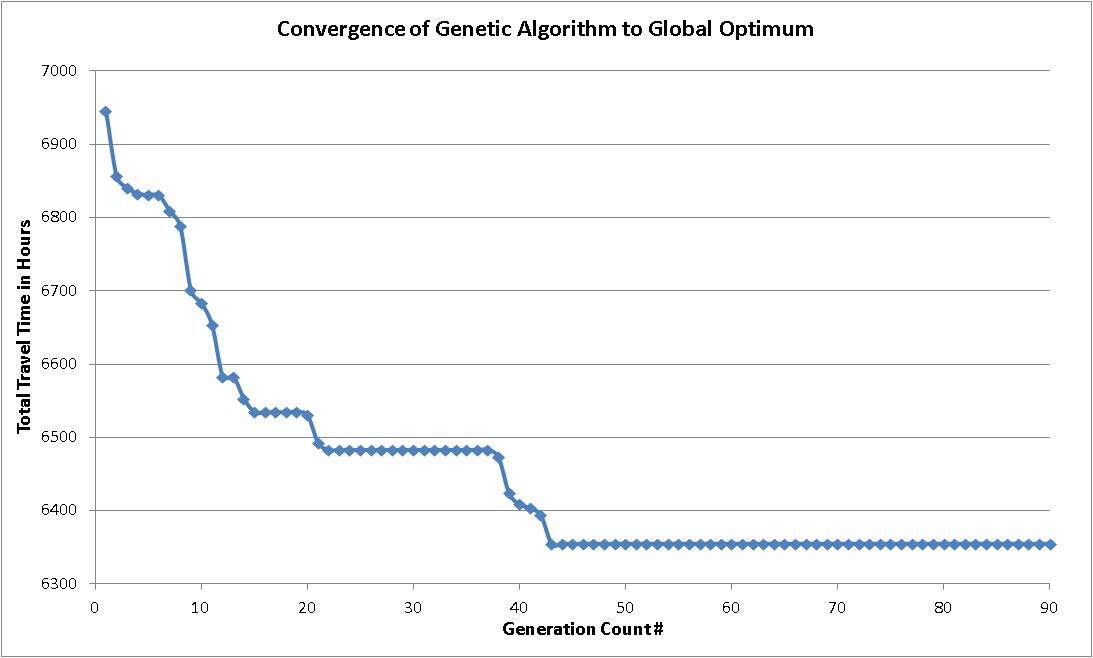


Figure 3: GA Convergence to global optimal over successive generations

It can be seen that the algorithm converged after 42 generations. To confirm that it is the global minima, all the combinations of allowable values of impact parameters were simulated. Table 3 shows the optimal values found for the 4 impact parameters.

Table 3: Optimal SWARM Parameter values with queue flush

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Max  points | Den  SampSize | Forecast  LeadTime | UpRate | Down  Rate | InterSection  Propagation  Factor | MinVol |
| Optimal Value | 6 | 30 | 15 | **780** | **720** | **0.1** | 5 |
| Parameter | MaxVol | MinSpeed | MaxSpeed | MinDen | MaxDen | q | SatDen |
| Optimal Value | 10 | 48 | 72 | 22 | 50 | 0.02 | **22** |

Note that the four impact parameters are sensitive to geometric and traffic patterns on the corridor. Therefore the parameter optimization process should be repeated when traffic patterns change.

Figure 4 and Table 4 show a comparison of performance of various strategies. The results validate that both ALINEA and SWARM improve both system and freeway travel times compared to a no-metering scenario. However, due to the queue flush mechanism implemented in this study (using field thresholds), ramp travel times do not increase significantly in spite of ramp metering. The results indicate that SWARM performs better than ALINEA even with default values. The total travel time reduction could be as much as 13%. Finally, optimal parameters result in travel time reduction of 11.6% compared to default parameters, in this case study.

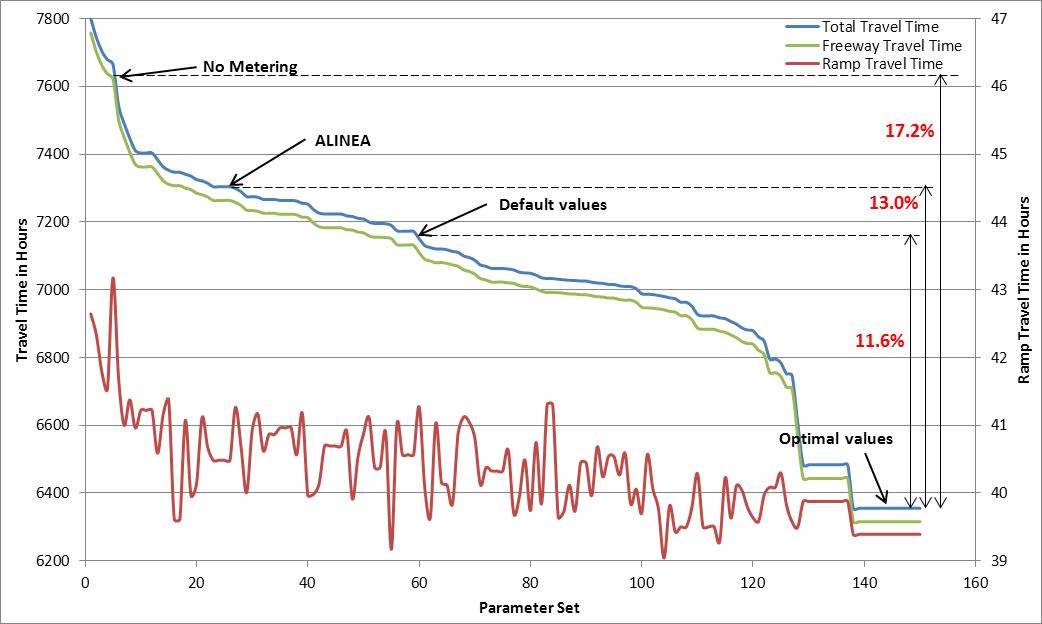


Figure 4: Sample results of the 45th generation of GA optimization

Table 4: Travel time reduction (veh. hrs) comparison of SWARM (with queue flush) with other strategies

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Comparison with No-Metering | | | Comparison with ALINEA | | | Comparison of SWARM with default values | | |
|  | System | Ramp | Freeway | System | Ramp | Freeway | System | Ramp | Freeway |
| ALINEA | 375 (4.9%) | -14.2 (-54.3%) | 389 (5.1%) | - | - | - | - | - | - |
| SWARM w/default parameters | 491 (6.4%) | -13.7 (-52.2%) | 505 (6.6%) | 116 (1.6%) | 0.5 (1.3%) | 115 (1.6%) | - | - | - |
| SWARM w/optimal parameters | 1324 (17.2%) | -13.2 (-50.1%) | 1337 (17.5%) | 949 (13.0%) | 1.1 (2.7%) | 948 (13.1%) | 833 (11.6%) | 0.6 (1.4%) | 832 (11.7%) |

## Sensitivity Analysis

Non-impact parameters

Since it was hypothesized that there are only 4 impact parameters, it was verified using sensitivity analysis as follows. Multiple simulation runs were performed using optimal values for four impact parameters and perturbing the no-impact parameters on either side of their default value, one parameter value at a time. Thus, total of 206 runs were performed to check the sensitivity of non-optimized parameters. As expected, it was found that that the total travel time remained same for all the runs. This showed that default values perform adequately for the 10 non-impact parameters.

Random number variation

GTsim uses several random numbers for vehicle destination determination, mandatory and discretionary lane changes, determine vehicle acceleration/deceleration etc. To verify if the global optimal values still hold true for different set of random numbers the random number seeds were changed and GA based optimization was performed using the GA parameter values used earlier to obtain global minima. After 50 generations, it was found that the global optimal parameters still performed the best compared to other parameter combinations. However, the total travel time values varied in the range of + 0.3%. Therefore, the global optimum obtained is resistant to change in the random number seed and the results are stable.

Traffic flow variation

The sensitivity of the optimal parameters to the day-to-day flow variations is also studied. The OD flows are changed randomly by + 15% and GA based optimization was performed with the GA parameters used earlier. The GA was run for 50 generations and the results indicated that for some flow conditions GA converged to a different set of optimal parameter values. However, it was observed that the performance of the optimal values obtained earlier is comparable. If one were to use the earlier optimal values for various flow conditions, the elasticity of the performance degradation for flow variation was found to be 0.3% for every 1% change in flow.

# Conclusions and Recommendations

This paper presented a GA based parameter optimization framework ramp metering algorithms. As a part of this framework, the microsimulation model (GTsim), and a GA based optimization module were integrated. It was found that parallel implementaion of micro-simulation has significant advantages over serial computation. Parallel implementation enables obtaining faster results. Moreover, since GA generations can be evaluated quickly larger population sizes for larger search spaces can be accomodated. Incase of large number of tunable parameters, the methodology used in section 4.4 could be used to separate impact parameters from non-impact parameters to make the search space manageable.

The performance of the framework was tested using the SWARM ramp meterig algorithm. It was found that the framework was successful in estimating optimal values for the 4 tunable parameters. When compared to the SWARM performance default values, optimal values resulted in a reduction of travel time by 11.6%. It was found that the optimal parameter values derived in the case study are location sensitive and need to be optimized for other locations which can be seamlessly done using the proposed framework.

This research was supported by the GDOT’s research project TO 02-49; RSCH PROJ 07-22: Optimal Ramp Metering Strategies. The authors wish to thank the GDOT for its support, in particular Binh Bui, David Jared, Mark Demidovich and W. Grant Waldrop.

# References

1. Papageorgiou, M., Kotsialos, A., 2002. Freeway ramp metering: an overview. IEEE Transactions on Intelligent Transportation Systems 3, 271–281.
2. Wattleworth, J.A., Berry, D.S., 1965. Peak-period control of a freeway system-some theoretical investigations. Highway Research Record 89 (1), 1–25.
3. Iida, Y., Hasegawa, T., Asakura, Y., Shao, C., 1989. A formulation of on-ramp traffic control system with route guidance for urban expressway. In: Proceedings of the IFAC/IFIP/IFORS Symposium, Paris, September 19–21, pp. 229–236.
4. Yang, H., Yagar, S., 1994. Traffic assignment and traffic control in general freeway areterial corridor systems. Transportation Research Part B 28 (6), 463–486.
5. Papageorgiou, M., Habib, H.S., Blosseville, J.M., 1991. ALINEA: a local feedback control law for on-ramp metering. Transportation Research Record 1320, 58–64.
6. Y. Stephanedes, Implementation of on-line Zone Control Strategies for optimal ramp metering in the Minneapolis Ring Road,7th International Conference on Road Traffic Monitoring and Control, 1994.
7. Zhang, M., Ritchie, S., Recker, W., 1996. Some general results on the optimal ramp control problem. Transportation Research Part C 4 (2).
8. Jacobsen, L., Henry, K., and Mahyar, O., 1989. Real-time metering algorithm for centralized control, Transportation Research Record 1232.
9. Diakaki, C., Papageorgiou, M., 1994. Design and simulation test of coordinated ramp metering control (METALINE) for A10-West in Amsterdam. Internal Report 1994–2, Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Greece
10. Lipp, L., Corcoran, L., and Hickman, G., 1991. Benefits of central computer control for the Denver ramp metering system, Transportation Research Record 1320.
11. Banks, J., 1993. Effect of response limitations on traffic-responsive ramp metering, Transportation Research Record 1394.
12. G. Paesani, J. Kerr, P. Perovich and E. Khosravi, 1997. System wide adaptive ramp metering in southern California, ITS America 7th Annual Meeting.
13. Xin, W., Michalopoulos, P. G., Hourdakis, J., and Lau, D., 2004. Minnesota’s New Ramp Control Strategy: Design Overview and Preliminary Assessment, 83rd Transportation Research Board Annual Meeting, Washington D.C.
14. Yang, X., Chu, L., Recker, W., 2002. GA-based Parameter Optimization for the ALINEA Ramp Metering Control. In: Proceedings of the Fifth IEEE Intelligent Transportation Systems Conference, Singapore
15. Quadstone Ltd., Paramics Modeller V.6.9 users guide, 2012.
16. Chilukuri B.R., Laval, J.A, and Guin, A., 2014. Micro-Simulation Based Framework for Freeway Travel-Time Forecasting. Forthcoming in the Transportation Research Record.
17. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Holland, John H. Oxford, England: U Michigan Press. (1975).
18. Abu-Lebdeh, G., and R. F. Benekohal. Convergence Variability and Population Sizing in Micro-Genetic Algorithms. Computer-Aided Civil and Infrastructure Engineering, Vol. 14, 1999, pp. 321–334.
19. Abu-Lebdeh, G., and R. F. Benekohal. Genetic Algorithms for Traffic Signal Control and Queue Management of Oversaturated Two-Way Arterials. In Transportation Research Record: Journal of the Transportation Research Board, No. 1727. TRB, National Research Council, Washington, D.C., 2000, pp. 61–67.
20. J.A. Laval and C.F. Daganzo. (2006) Lane-changing in traffic streams. Transportation Research Part B, 40 (3), pp. 251-264.
21. <http://en.wikipedia.org/wiki/JAR_(file_format)> visited 8/1/2014
22. Ramp Metering Navigator High Level Design, GDOT– NAV01-122, Georgia Department of Transportation, Atlanta, May 1, 2006.