A Hybrid Genetic Algorithm for Deploying RSUs in VANETs Based on Inter-Contact Time

Marcelo Fonseca Faraj
Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675, Nova Gameleira
Belo Horizonte, Brazil
marcelofaraj@gmail.com

João Fernando Machry Sarubbi
Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675, Nova Gameleira
Belo Horizonte, Brazil
joao@decom.cefetmg.br

Cristiano Maciel da Silva
Universidade Federal de São João Del Rei
Praça Frei Orlando, 170 - Centro
São João Del Rei, Brazil
cristiano@ufsj.edu.br

Flávio Vinícius Cruzeiro Martins
Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675, Nova Gameleira
Belo Horizonte, Brazil
flaviocruzeiro@decom.cefetmg.br

ABSTRACT

Vehicular ad-hoc networks (VANETs) have potential to ease traffic management, lower accident rates and provide many more benefits. Deploying the infrastructure entities, called roadside units (RSUs), is challenging and purpose-dependent. There are many metrics in literature to evaluate RSU deployment, such as coverage time-based and inter-contact distance-based. In this work, we use a metric called Gamma Deployment, based on the inter-contact time between vehicles and RSUs and on the percentage of vehicles covered. We implement a heuristic approach based on a hybrid of genetic algorithm and local search and compare our results with the algorithm Gamma-G, proposed in literature.

CCS CONCEPTS

• **Computing methodologies** → **Heuristic function construction**; • **Mathematics of computing** → *Evolutionary algorithms*;

KEYWORDS

VANET, Deployment, Genetic algorithm, Memetic Algorithm

ACM Reference format:

Marcelo Fonseca Faraj, João Fernando Machry Sarubbi, Cristiano Maciel da Silva, and Flávio Vinícius Cruzeiro Martins. 2017. A Hybrid Genetic Algorithm for Deploying RSUs in VANETs Based on Inter-Contact Time. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
https://doi.org/http://dx.doi.org/10.1145/3067695.3076032

1 INTRODUCTION

Labeled as a realistic application of mobile ad hoc networks, VANETs allow intelligent transportation systems to provide safer roads, more efficient flow management, *infotainment*, and so forth [1]. In a VANET, communication processes can occur in one or more of the following ways: vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and infrastructure-to-infrastructure (I2I).

In this work, we propose a local search genetic algorithm, called Gamma-LSGA, for deploying RSUs based on the Gamma Deployment (Γ_D) metric [3]. The genetic operators, fitness function, and local search methods were built with abundant problem specific knowledge. We compare our heuristic with the baseline heuristic Gamma-G [3], which uses the same metric and data preparation.

2 GAMMA DEPLOYMENT

Gamma Deployment (Γ_D) is an inter-contact time based metric to assure quality of service (QoS) for a VANET. Assume a deployment R of RSUs in a road network M, and a set V of all vehicles passing through M during a specific period of time. Let also $C \subseteq V$ be the set of vehicles trips meeting the requirement of connecting with at least one RSU during any τ seconds time interval. A given R in M is $\Gamma_D\binom{\tau}{\rho}$ if and only if $\frac{|C|}{|V|} \geq \rho$.

3 HYBRID GENETIC ALGORITHM: GAMMA-LSGA

In this section, we present our algorithm for deploying RSUs based on the Γ_D metric, called Gamma-LSGA. It uses standard concepts and operators of genetic algorithms, such as a population of P individuals passing by G generations with selection, crossover, and mutation. In addition, it includes local search procedures, which makes it a hybrid genetic algorithm, or a memetic algorithm [2].

3.1 Notation and Fitness Function

We encode an individual R as a $\psi \times \psi$ matrix. This represents a discretization of M in ψ columns and ψ rows, adding up to ψ^2 urban cells U_i where RSUs can either be placed or not. If a given U has an RSU, it is set to 1; otherwise, it is set to 0. We represent the number of RSUs of R as $|R|$. For our purposes, an RSU deployed in U assures coverage for all vehicles passing by U's area. Equation (1) shows our fitness function f for an R and a given $\Gamma_D\binom{\tau}{\rho}$.

$$f\left(R, \Gamma_D\binom{\tau}{\rho}\right) = \begin{cases} (1-\rho) \times |V| + \left[\frac{|R|}{\psi^2}\right], & \text{if } \Gamma_D\binom{\tau}{\rho} \text{ is not met} \\ |R| - \psi^2 + [-\rho], & \text{if } \Gamma_D\binom{\tau}{\rho} \text{ is met} \end{cases} \tag{1}$$

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany

ACM ISBN 978-1-4503-4939-0/17/07.
https://doi.org/http://dx.doi.org/10.1145/3067695.3076032

3.2 Recombination and Mutation

We use a *uniform crossover* approach, consisting of a 50% chance for each gene U to be inherited from each parent. A solution R has a mutation rate α. If selected to mutate, one of four different mutation operators is once up to σ times randomly chosen and applied. Each time, the selected operator can be looped once up to δ times. Our mutation has 4 random operators: (1) Insertion of new RSU; (2) Removal of RSU; (3) Move of RSU from U_1 to random U_2; (4) Move of RSU from U_1 to U_2 in U_1's adjacent neighborhood;

3.3 Local Search

We implement 2 local search procedures: Grid Local Search (GLS), and Reduction Local Search (RLS). They are applied with a frequency of $(\theta\ generations)^{-1}$, and for a fraction π of the fittest solutions [2]. In GLS, a loop is run for each RSU in R until no more changes are possible: (1) remove it from U_1; (2) compute the remaining C of R; (3) put it at the U_2 in U_1's $\gamma \times \gamma$ neighboring grid of cells incurring the greatest increase in $|C|$ or do not put it back if C is unchanged. In RLS, we sort all RSUs of R based on the $|C|$ resulting from removing each one. Then, the RSUs are removed orderly while $\frac{|C|}{|V|} \geq \rho$.

4 EXPERIMENTS

We base our experiments in the pruned version of a mobility trace from Cologne, Germany [1]. We use the following parameters: P=40; G=600; ψ=100; α=3%; σ=10; δ=5; θ=10; π=5%; γ=5; $|V| \in$ {500, 1000, 1500, 2000, and 2500}; $\tau \in$ {40s, 80s, 120s}; and $\rho \in$ {0.6, 0.8, 1.0}. We run the Gamma-LSGA 33 times for each instance, since it is a stochastic algorithm. The two only solutions in which we apply GLS and RLS are the best one at the beginning of the generation and the best among the mutated ones. To verify our parameters for the local search, we have tested some experiments with θ=1 and π=100%. The solutions had up to 0.5% fewer RSUs, but the running time was was up to 100 times higher. We also implement and run the algorithm Gamma-G [3] for the same instances to compare the results. It is run only once, since it is a deterministic heuristic.

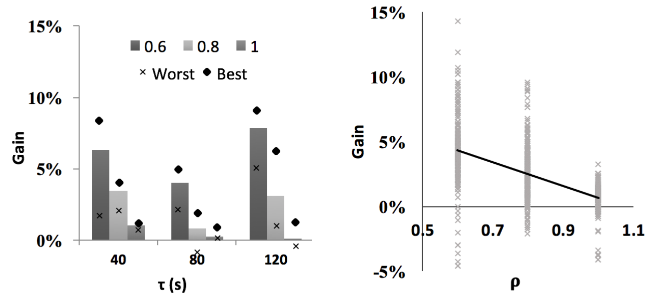
4.1 Number of RSUs

Figure 1a summarizes the gain of Gamma-LSGA over Gamma-G for all instances with $|V|$=1000. This set of experiments exemplifies the pattern of gains we obtain for the given combinations of τ and ρ. In total, Gamma-LSGA outperforms Gamma-G in 93% of the experiments. In figure 1b, we plot the gain of all the 1485 executions of Gamma-LSGA over Gamma-G as a function of ρ. As the trend line shows, the gain tends to be higher for lower ρ values.

4.2 Convergence

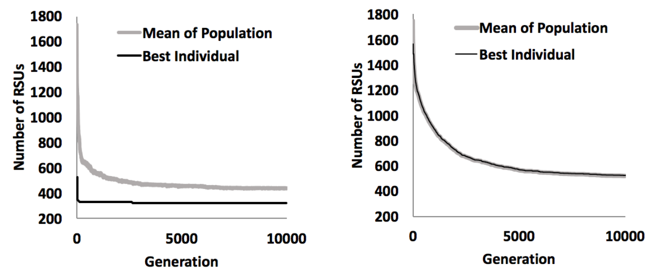
Figure 2a presents the convergence of Gamma-LSGA for a G=10000, $|V|$=1000, ρ=0.6, and τ=80s sample. In Figure 2b, we plot the same situation, but using a traditional genetic algorithm (TGA), consisting in a Gamma-LSGA without GLS and RLS. In both figures, the gray curve shows the average number of RSUs of the whole population, and the black curve presents the number of RSUs of the best solution. The number of RSUs in the best solution and

[1] available at http://kolntrace.project.citi-lab.fr/



(a) (b)

Figure 1: Graph (a) shows the gain of Gamma-LSGA over Gamma-G for $|V|$=1000. The columns represent the median. The balls and X's mark the best and the worst results. Graph (b) plots the gains of all the experiments and a trend line.



(a) Gamma-LSGA (b) Genetic Algorithm

Figure 2: The graphs (a) and (b) show, respectively, the behavior of Gamma-LSGA and a TGA for τ=80s, ρ=0.6, V=1000.

in the population average are very distant for Gamma-LSGA and pretty close for TGA. Furthermore, the best solution evolves earlier and achieves lower marks in Gamma-LSGA. The results show the highly positive impact of including local search procedures.

5 CONCLUSION

In this work, we propose a hybrid genetic algorithm, called Gamma-LSGA, for deploying RSUs in VANETs. We guarantee QoS using Gamma Deployment [3], a metric based on the inter-contact time. We present some results of our experiments, in which Gamma-LSGA outperforms Gamma-G for almost all instances, showing gains up to 15%. We also find that the presence of local search procedures in Gamma-LSGA impacts the algorithm execution by providing a faster convergence and better results.

REFERENCES

[1] Saif Al-Sultan, Moath M Al-Doori, Ali H Al-Bayatti, and Hussien Zedan. 2014. A comprehensive survey on vehicular Ad Hoc network. *Journal of network and computer applications* 37 (2014), 380–392.
[2] Tarek A El-Mihoub, Adrian A Hopgood, Lars Nolle, and Alan Battersby. 2006. Hybrid Genetic Algorithms: A Review. *Engineering Letters* 13, 2 (2006), 124–137.
[3] Cristiano M Silva, Daniel L Guidoni, Fernanda S Souza, Cristiano G Pitangui, João FM Sarubbi, Andre LL Aquino, Wagner Meira, Jose Marcos S Nogueira, and Andreas Pitsillides. 2016. Using the inter-contact time for planning the communication infrastructure in vehicular networks. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2089–2094.