Artificial Intelligence-Empowered Optimal Roadside Unit (RSU) Deployment Mechanism for Internet of Vehicles (IoV)

Debjani Ghosh[1], Hardik Katehara[2], Oshin Rawlley[3], Shashank Gupta[4*], Naveen Arulselvan[5], Vinay Chamola[6]
[1,2,3,4] *Department of Computer Science & Information Systems*
[6]*Department of Electrical and Electronics Engineering*
Birla Institute of Technology and Science, Pilani, Rajasthan, India
[5] *ARTPARK, Bengaluru, India*
Email: [1]h20200263@pilani.bits-pilani.ac.in, [2]f20190089@pilani.bits-pilani.ac.in,
[3]p20200063@pilani.bits-pilani.ac.in, [4]shashank.gupta@pilani.bits-pilani.ac.in,
[5]naveena@iisc.ac.in, [6]vinay.chamola@pilani.bits-pilani.ac.in

***Abstract*—Currently, the world is witnessing a huge growth in additional computing proficiency and extensive network coverage capability, which resulted in a paradigm shift from VANETs to Internet of Vehicles (IoV). Moreover, enhanced network capabilities facilitate enabling of IoV technology for latency-critical applications in energy-constrained smart IoT devices. However, IoV networks demand energy efficiency due to its dynamic nature for which Roadside Units (RSUs) are critical. However, in cities, huge deployment of RSUs and their maintenance is expensive in IoV infrastructure, requiring a trade-off between the network coverage and installation-related expenses. Also, the latency issues in IoV are highly dependent on the positioning of accessible RSUs. Motivated by the above highlighted issues, we propose an upgraded RSU placement method to boost network efficiency through placement of RSUs in optimal locations in a given road map. The Memetic Framework-based Optimal RSU Deployment (MFRD) algorithm is proposed to maximize the coverage area among the vehicles in an IoV and minimize the overlap in the coverage of the different RSUs. We observed from simulation results based on real-world maps that MFRD yields a significantly higher fitness score as compared to the existing state-of-the-art in terms of optimal positioning of the RSUs.**

***Index Terms*—Roadside Units, IoV, Genetic Algorithms, Memetic Algorithms.**

I. INTRODUCTION

Recently, Internet of Vehicles (IoV) has attracted significant attention from industry and academia alike and has become a promising application of Internet of Things (IoT). Several safety-critical applications and infotainment services are supported by IoT. According to Gartner [1], IoV is expected to contribute between $200-700$ billion to the global economy as it will play an indispensable role in connecting millions of vehicles to Intelligent Transport Systems (ITS). Due to latest advances in IoT technology, vehicles are now equipped with a wide array of devices such as sensors and cameras. These devices collect route information along with location and acceleration of the vehicle which can be exchanged with the neighbouring vehicles or Roadside Units (RSUs). Such a distributed network of intelligent vehicles sharing data among themselves via Vehicle-to-Vehicle (V2V) communication or with RSUs via Vehicle-to-Infrastructure (V2I) communication has led to the advent of IoV. The RSUs are stationary equipment installed adjacent to roads and have greater communication capabilities than the vehicles. They act as communication gateways, thereby, playing a critical role in V2I communication. However, both the installation and maintenance costs of RSUs in an IoV network is high and hence, requires a trade-off between optimal network coverage and economical deployment costs [2]. The situation is critical in urban areas where it can be prohibitively expensive to deploy many RSUs to cover the entire geographical area of a city. Moreover, the location and number of RSUs have an impact on the Quality-of-Service (QoS) parameters such as network delay. Hence, an efficient IoV network demands optimal deployment of RSUs in a given road layout to achieve maximum network coverage with minimal overlap of the coverage areas of each RSU.

Existing works in literature have proposed solutions to the RSU deployment problem. For instance, Gao et al. [3] proposed a density-based Dynamic Limiting as a solution to the RSU deployment problem. However, this technique fails to achieve favourable network performance in urban scenarios. This lacuna is addressed in [4] which uses a utility-based deployment model; however, it is not suitable for road layouts with low vehicle density. Liang et al. [5] proposed a mathematical solution for RSU placement by utilizing the ILP mechanism while the authors in [6] derived a relationship between the delay and distance between the different RSUs. The RSU placement is an optimization problem hence, some of the existing works employ bio-inspired algorithms such as Evolutionary Algorithms to solve it. The authors in [7] and [8] use EAs to optimize the fitness function of the RSU placement problem which is based on message delay. However, these do not take into consideration the cross-layer challenges in a hybrid network. To conclude, the existing state-of-the-art are not applicable in diverse road scenarios and suffer from poor

978-1-6654-0876-9/22/$31.00 ©2022 IEEE
DOI 10.1109/WoWMoM54355.2022.00077

network performance.

Therefore, motivated by the challenges highlighted above, we propose an efficient RSU placement strategy which optimizes network coverage in an IoV network. The contributions of this study are summarized as follows:

- An effective Memetic Framework-based Optimal RSU Deployment (MFRD) algorithm based on the evolutionary computing section of Artificial Intelligence is proposed to maximize the coverage area of each RSU such that there is minimal overlap in their coverage areas.
- The MFRD algorithm increases the fitness score by 15% on an average as compared to the GA algorithm [7] and results in an optimal design of RSU infrastructure.
- The simulated scenarios use real-time information about road traffic such as vehicular mobility and versatile road layouts of both urban and semi-urban areas to obtain realistic results.

II. System Model

The goal of Roadside Units' placement is to create a network of RSUs in an IoV environment on any layout of road map such that the message transmission delay is minimized and the coverage area of each RSU is maximized. Further, the high installation and maintenance cost of RSUs restricts their usage to a limited number. The existing RSU deployment mechanisms fail to achieve network efficiency and are also not applicable to diverse traffic scenarios. In view of these limitations, we propose the Memetic Framework-based Optimal RSU Deployment (MFRD) algorithm which is based on the evolutionary computing section of Artificial Intelligence [9]. For instance, if we consider a 4x4 road network comprising of 16 intersections, then each intersection is a potential location where a RSU can be deployed. We aim to install RSUs at positions which will cover all the intersections with minimal overlap in the coverage areas of the RSUs.

Inspired by the concept of Darwinian evolution, Evolutionary algorithms (EAs) utilize natural selection to solve search and optimization problems. Genetic algorithms (GAs) are a branch of EAs which mimic the natural biological evolution process by subjecting a population of individuals through random processes such as genetic recombination. A selection procedure based on problem-specific criteria is applied on the population to ascertain the individuals best suited to survive. The most fit individuals find their place in the next generation while the lesser fit individuals are pruned from the population. These steps are repeated till the termination condition is achieved [10].

GAs start by establishing a relation between a set of individuals in a natural population and a set of solutions to a problem also called as phenotype. This is done by representing the information of each solution in a string known as genotype. These genotypes are evolved through multiple generations using the genetic operators such as selection of parents, crossover, mutation, and replacement. However, a pure genetic algorithm fails to yield fine-tuned results in a complex combinatorial search space. On the other hand, integration of a local search process resulting in a hybrid GA significantly improves the efficiency of the search process. This combination was termed as Memetic Algorithm (MA) in [11].

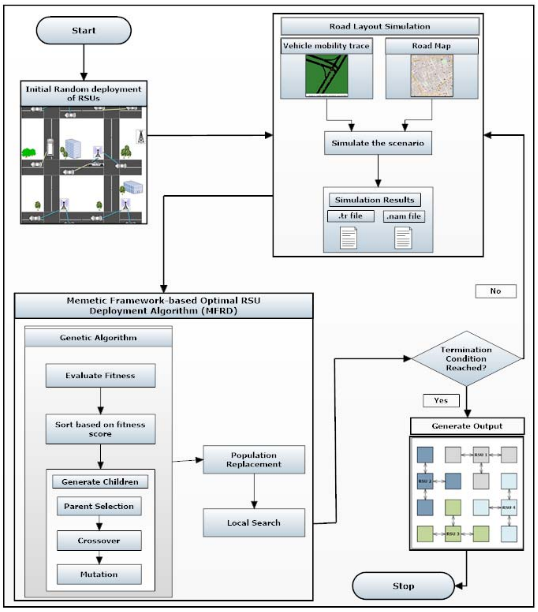

Fig. 1: MFRD algorithm.

The MFRD mechanism employs a memetic framework which is an extension of the traditional genetic algorithms. The Darwinian evolution and the meme concept enunciated by Dawkins [12] both inspired memetic algorithms at the end of the 1980s. A meme according to Dawkins is a basic unit of information that replicates with exchange of ideas between people. In MA, a local search technique is applied to every individual created by the genetic algorithm to improve that individual's fitness as compared to genetic algorithms. The exploration of the solution space combined with the exploitation of the local search space diminishes untimely convergence to local optima [13].

In the case of Roadside Unit deployment problem, the phenotype is the possible locations on which the RSUs can be placed. We have chosen a set of integers to be the corresponding genotype representing the RSU locations. Care is taken so that multiple RSUs are not assigned to a single location. We developed two variations of a hybrid genetic algorithm using hill climbing meta-heuristic. While the Genetic Algorithm with Random Restart Hill Climbing (GARHC) integrates the random restart technique to the local search process, the MFRD algorithm (Algorithm 1) uses a simple hill climbing approach. The input parameters to both the GARHC and the MFRD algorithms include a set of k possible locations, number of RSUs to be deployed R, size of the population n, and the number of generations G. These two algorithms also require as input a map of the target area comprising of the layout of the different streets and intersections and the mobility

Algorithm 1: MFRD algorithm

Input: Possible Candidate Locations L = L_1, L_2, L_3, $\ldots$, L_k, Quantity of RSUs available R, Population size n, Number of generations G
Output: Set of R locations
Procedure `Find_RSU_Location()`:
  Produce an initial population P of size n where an individual has R features;
  P_s, P_m, P_l $\leftarrow$ Probability of selection, mutation, and local mutation respectively;
  I_m, I_c $\leftarrow$ Index of mutation and crossover respectively;
  R_m, R_c, R_l $\leftarrow$ Random numbers between 0 and 1;
  γ_s $\leftarrow$ Fitness score;
  while *G not reached* **do**
    λ_n, ϕ_n $\leftarrow$ New population and an empty list to hold children for every generation respectively;
    calculate γ_s for P and sort P on γ_s;
    for ($i = 0;\ i < n \times P_s;\ i = i + 1$) {
      $\lambda_{ni} \leftarrow P_i$;
    }
    for ($i = n \times P_s + 1;\ i < n;\ i = i + 1$) {
      I_{c1}, I_{c2} $\leftarrow$ Selection(λ_n);
      $\phi_n i$ $\leftarrow$ Crossover(I_{c1}, I_{c2}, I_c);
      if $R_m \leq P_m$ **then**
        ϕ_{n_i}' $\leftarrow$ Mutate(ϕ_{ni}, I_m);
        Replace ϕ_{ni} by $\phi_n i'$;
    }
    Merge the new chromosomes obtained in ϕ_n with the chromosomes in λ_n;
    ϕ_l $\leftarrow$ an empty list to hold the chromosomes chosen by local search process;
    for ($i = 0;\ i < length(\lambda_n)$; *i = i + 1*) {
      $\lambda_n i \leftarrow P_i$;
      if $R_l \leq P_l$ **then**
        Local_search_replacement (λ_{ni}) ;
    }
    $P \leftarrow \lambda_n$;
  end
  return P_0;
Procedure `Local_search_replacement`(*chromosome*):
  currentSolution $\leftarrow$ $chromosome_i$;
  max_local_gen $\leftarrow$ Number of generations to be run for local search process;
  for ($i = 0;\ i < max\_local\_gen;\ i = i + 1$) {
    newSolution $\leftarrow$ Mutate $chromosome_i$;
    if $\gamma_s(newSolution) \geq \gamma_s(currentSolution)$ **then**
      update currentSolution with newSolution ;
  }

traces simulating the realistic vehicular traffic. The map of the target area and the mobility traces have been generated through OSM and SUMO.

Steps 1 through 4 of Algorithm 1 indicate the initial solution space comprising of the possible locations, the probability, corresponding indices, and random numbers of mutation and crossover respectively. In Step 5, a variable to store the fitness score is created. Step 7 indicates the variables to store the new population and the children created in each iteration of the MFRD algorithm respectively. Step 8 evaluates the individuals in the population according to the fitness function.

Subsequently, the population is sorted in a non-increasing order based on the fitness score. The fitness function is unique to every problem and indicates how close the generated solution is to meeting the requirements of the problem. The output of this function is a numeral which must be either minimized or maximized depending on the problem. The proposed MFRD algorithm focuses on an objective function – coverage ratio of an RSU denoted by ξ. Our coverage ratio of an RSU aims to find the number of intersections covered by each RSU since these intersections are the possible deployment locations of the RSUs. Let ϱ_i and ϱ_j denote the number of intersections covered by RSUs R_i and R_j respectively. Then, the optimal number of intersections covered by R_i and R_j are $\varrho_i + \varrho_j - (\varrho_i \cap \varrho_j)$. Thus, the coverage ratio ξ of RSU placement in a given map layout is expressed as

$\xi = \sum_{i=1}^{N-1} (\varrho_i + \varrho_j - (\varrho_i \cap \varrho_j))$ (1)

where N denotes total number of RSUs.

We have ensured that once a Roadside Unit's ϱ is calculated, it is not recalculated in subsequent derivations of ξ. The aim of MFRD is to maximize ξ. Thus, the fitness function for MFRD is stated as

γ = Maximum (ξ) (2)

Steps 9 through 11 select the best individuals from the current population based on selection ratio P_s, using an elitist procedure and inserts them into the new population λ_n. Among these best individuals, two parents are randomly chosen to pass their genes to the next generation in step 13. In MFRD algorithm, this is done through single-point crossover in which a point on the chromosomes

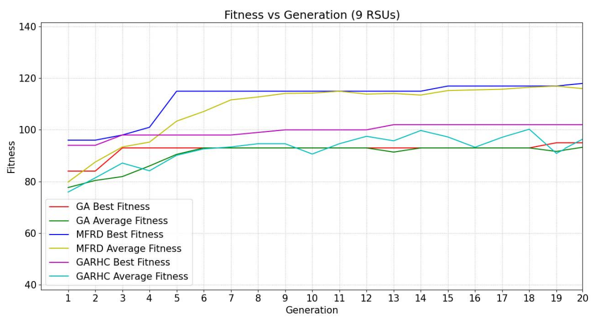

Fig. 2: Comparison of Fitness scores of RSU placement algorithms for 9 RSUs in Murlipura.

(I_c) of both the parents is chosen randomly. This ensures the child contains genes from both the parents. The offspring

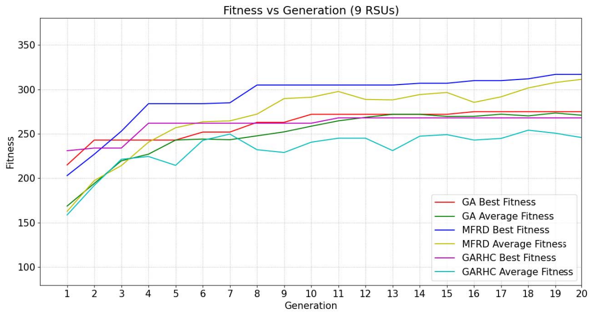

Fig. 3: Comparison of Fitness scores of RSU placement algorithms for 9 RSUs in Connaught Place.

so generated is mutated based on mutation probability P_m in steps 15 through 18 which introduces diversity into the population.

Steps 20 through 26 in the MFRD algorithm employ a local search process on the children created as a result of the genetic algorithm. The local search aids in the discovery of the local optimum faster and helps in quicker convergence of the search process. We have used the simple hill climbing algorithm as the local search process which is carried out on every individual obtained after merging new chromosomes with the existing ones.

The new population so generated is once again evaluated for fitness. This goes on until the termination condition is reached. MFRD returns a set of R locations as output which achieves minimal transmission delay with maximum RSU coverage.

Fig. 1 demonstrates the steps followed by MFRD. Initially, RSUs are randomly deployed across various intersections. This scenario is simulated using a network simulator. The fitness of the current population is determined using the fitness function of MFRD. This is followed by the application of genetic operators: selection of parents, crossover, and mutation processes. A local search process is applied on the offspring generated. Subsequently, the algorithm replaces a fraction of the current population with the offspring generated and checks if the termination condition is reached. In case of MFRD, the termination condition equals a maximum number of generations reached. If the termination condition is not satisfied, the process is repeated.

III. Implementation

This section presents the simulation environment that we used to examine the performance and viability of our proposed framework. Deploying a vehicular network in real time is not feasible with respect to scalability of vehicles and the expense involved. Hence, we have opted for simulation of an IoV network. We have selected two real-world locations, Murlipura and Connaught Place, as the target areas to simulate and test the RSU deployments. The maps of these two locations have been sourced from an open-source software OSM (OpenStreetMap). The mobility of vehicles has been obtained from yet another open-source software SUMO (Simulation of Urban MObility) which is a realistic traffic generator and aids in route and road network visualization.

These two locations have been chosen for their diversity. Murlipura area in Jaipur represents a scenario with lesser number of road intersections. On the other hand, Connaught Place, New Delhi represents a layout with dense road network comprising of large number of road intersections, thus, providing more possibilities for locations to deploy the RSUs. The number of road intersections and other route information about the two chosen locations have been obtained from SUMO and summarized in Table I. Using two different road maps helps us confirm the potential of MFRD algorithm in suggesting the best possible RSU locations under different circumstances. The Wireless Access in Vehicular Environments (WAVE) standard has been simulated by using the IEEE 802.11p standard, while a data rate of 11 Mbits per second has been chosen for transmitting the data packets.

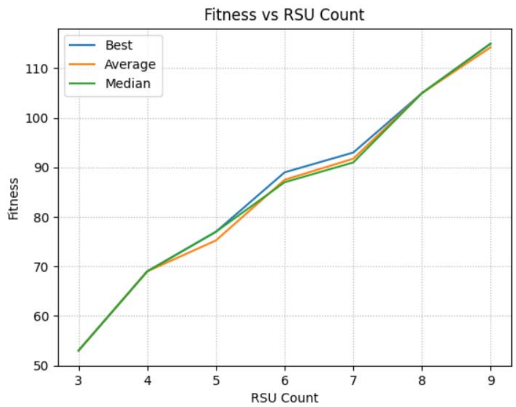

Fig. 4: Fitness score versus number of RSUs deployed in Murlipura.

Although several techniques have been proposed to suggest best possible locations of RSUs in a map of a given area, we chose to compare our approach against the GA algorithm [7] as it provides enough details to undertake a fair comparison. We then utilized a random restart hill climbing algorithm (GARHC) as our local search process to develop the memetic framework. It was further improved by using a simple hill climbing approach to fine tune the search process resulting in the MFRD algorithm. We analyse the performance of the memetic algorithm utilizing the simple hill-climbing as local search proposed in the MFRD approach to find the best possible locations of the RSUs among all the available locations. An average of five runs spanning twenty executions ensures that the results obtained a representative value. Next, the performance of MFRD is compared against GA and GARHC algorithms.

The performance of MFRD was analysed for the layouts of Murlipura and Connaught Place for which an appropriate RSU

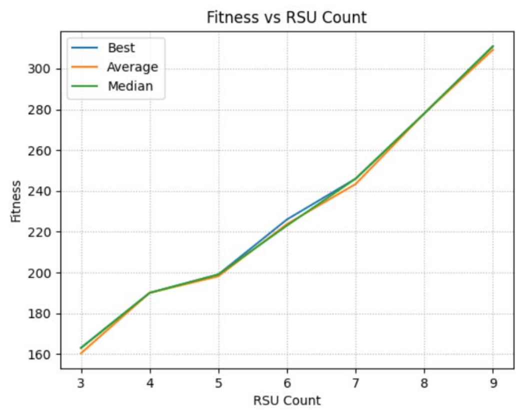

Fig. 5: Fitness score versus number of RSUs deployed in Connaught Place.

TABLE I: Details of the Area Layouts.

Name of the Area	**Number of Intersections**	**Number of Roads**
Murlipura, Jaipur	193	553
Connaught Place, New Delhi	1069	1890

deployment was conceived. This was compared to the performance of GA and GARHC. The MFRD algorithm follows the evolutionary process which is presented in terms of fitness scores of the best and average chromosomes (symbolizing the best deployments of the RSUs) in two different road layouts using 9 RSUs in Figures 2 and 3. The objective of the MFRD algorithm is to maximize the fitness function which is related to the number of intersections covered by an RSU (as represented in Equation 1). As is depicted, the MFRD algorithm fine tunes the search process resulting in significant improvement in the fitness score as the number of generations increase. For both the locations chosen, the MFRD performs the best by providing the highest fitness score. The GARHC performs better than the GA in all the cases. However, it performs equivalent to GA in two cases. These are when the least number of RSUs are chosen (in our case 3) for both the locations and when the maximum number of RSUs are chosen for Murlipura. Initially, an increase can be observed in the fitness score which gradually converges towards the global optimum value as the number of generations increase. Moreover, in some cases sudden improvements in the fitness score can be witnessed which can be credited to the mutation operator as well as the local search process that allow searching for new maximums in the solution space. While the GARHC algorithm performed better than the GA, it was observed that the proposed MFRD performed even better than the GARHC algorithm. The GA produces inefficient results as compared to the proposed MFRD technique in terms of a poor

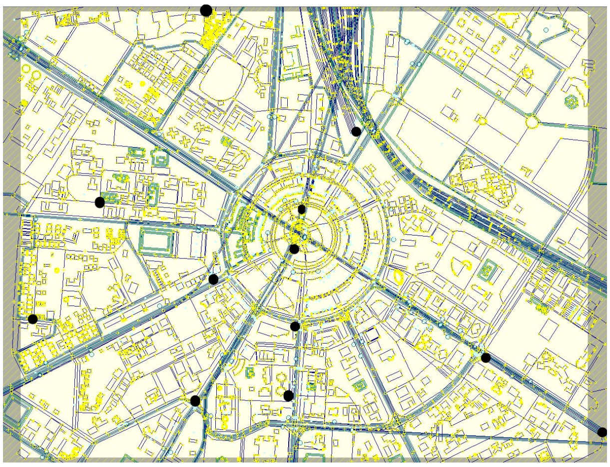

Fig. 6: Evolution of the deployment of 12 RSUs in Connaught Place after Generation 0.

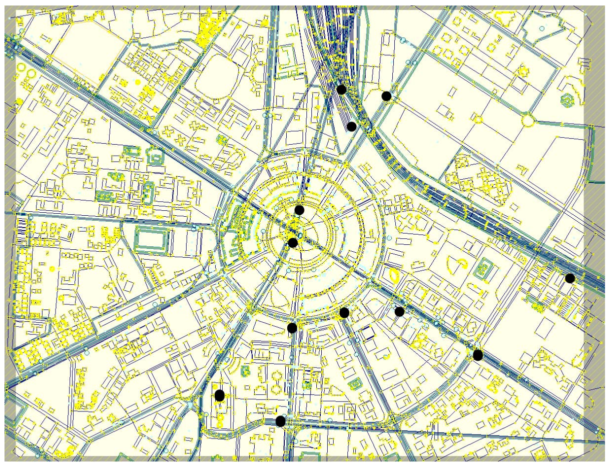

Fig. 7: Evolution of the deployment of 12 RSUs in Connaught Place after Generation 15.

fitness score. Although the GA converges faster than both the GARHC and the MFRD, however, both GARHC and MFRD achieve significantly better fitness scores. This is because the local search technique applied in the GARHC and the MFRD approaches help in finding the local optimum more efficiently than the GA.

Further, it can be clearly inferred from Figures 4 and 5 that the increase in the number of Roadside Units results in an improvement in the fitness score for both the locations. This observation is more pronounced in case of Connaught Place which has a denser road network, hence, comprises of larger number of intersections. This provides the possibility of covering more intersections using fewer RSUs and less overlap in the coverage areas of the RSUs.

To better illustrate the evolutionary process of the MFRD algorithm, we chose to simulate two different scenarios involv-

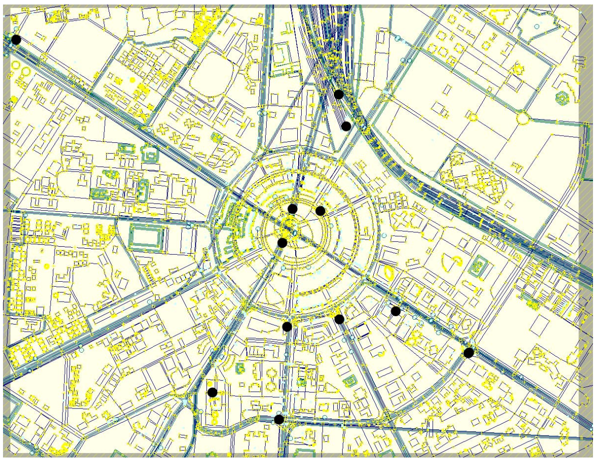

Fig. 8: Evolution of the deployment of 12 RSUs in Connaught Place after Generation 30.

ing 12 RSUs for Murlipura and Connaught Place respectively. Moreover, three different stages of the algorithm (after Generation 0, after Generation 15, and after Generation 30) were chosen to demonstrate how the MFRD computes the optimal locations over subsequent generations. Figures 6-8 exhibit the determination of best positions for 12 RSUs over multiple generations in Murlipura and Connaught Place respectively.

It was observed that the number of potential deployments increases considerably when 12 RSUs are considered as compared to 6 RSUs, hence, requiring a greater number of generations to converge. As the count of Roadside Units increases, so does the search space. It is possible to find new optimal solutions during the later generations due to the random nature of the algorithm. This is illustrated in Figures 6-8 since the best possible locations to deploy the RSUs keep changing through the generations as the MFRD algorithm tries to find the best possible deployments till a later generation. The MFRD and the GARHC find better positions for RSUs to be deployed by emitting a higher fitness score, however, the same is obtained at the cost of a higher response time. It was observed that both the MFRD and the GARHC consumed more time to generate solutions as compared to GA which has been documented in Table II.

TABLE II: Response time (RT) of the RSU placement algorithms in seconds.

No. of Generations	**GA**	**GARHC**	**MFRD**
20	5.041	22.383	38.478
40	10.242	44.413	83.181
60	13.263	77.441	133.346
80	18.992	93.428	164.584

IV. Conclusion

In this paper, we proposed the MFRD algorithm, a memetic-algorithm-based algorithm, to deploy Roadside Units in an efficient manner. It finds the optimal location where an RSU can be deployed such that maximum coverage and minimal overlap among the coverage areas of the neighbouring RSUs can be achieved. The simulation results based on real-world map layouts ranging from normal to complex road networks and realistic vehicular traces show that the MFRD algorithm performs significantly better than the existing state-of-the-art technique. Further, our proposed approach also provides better results for varied number of RSUs. In the future, we plan to evaluate the performance of the MFRD technique in a real environment by deploying a testbed.

Acknowledgment

The authors acknowledge the support of ARTPARK (AI and Robotics Technology Park), India for support of this research.

References

[1] M Rimol. Gartner forecasts more than 740,000 autonomous-ready vehicles to be added to global market in 2023. *URL: https://www. gartner. com/en/newsroom/pressreleases/2019-11-14-gartner-forecasts-more-than-740000-autonomous-ready-vehicles-to-be-addedto-global-market-in-2023 (: 09.04. 2021)*, 2020.

[2] Zisang Xu, Wei Liang, Kuan-Ching Li, Jianbo Xu, and Hai Jin. A blockchain-based roadside unit-assisted authentication and key agreement protocol for internet of vehicles. *Journal of Parallel and Distributed Computing*, 149:29–39, 2021.

[3] Zhenguo Gao, Danjie Chen, Shaobin Cai, and Hsiao-Chun Wu. Optdynlim: An optimal algorithm for the one-dimensional rsu deployment problem with nonuniform profit density. *IEEE Transactions on Industrial Informatics*, 15(2):1052–1061, 2018.

[4] Yuanzhi Ni, Jianping He, Lin Cai, Jianping Pan, and Yuming Bo. Joint roadside unit deployment and service task assignment for internet of vehicles (iov). *IEEE Internet of Things Journal*, 6(2):3271–3283, 2018.

[5] Yunyi Liang, Zhizhou Wu, Yu Tian, and Huimiao Chen. Roadside unit location for information propagation promotion on two parallel roadways with a general headway distribution. *IET Intelligent Transport Systems*, 12(10):1442–1454, 2018.

[6] Yu Wang, Jun Zheng, and Nathalie Mitton. Delivery delay analysis for roadside unit deployment in vehicular ad hoc networks with intermittent connectivity. *IEEE Transactions on Vehicular Technology*, 65(10):8591–8602, 2015.

[7] Manuel Fogue, Julio A Sanguesa, Francisco J Martinez, and Johann M Marquez-Barja. Improving roadside unit deployment in vehicular networks by exploiting genetic algorithms. *Applied Sciences*, 8(1):86, 2018.

[8] R Gunasekaran, S Siddharth, P Krishnaraj, M Kalaiarasan, and V Rhymend Uthariaraj. Efficient algorithms to solve broadcast scheduling problem in wimax mesh networks. *Computer Communications*, 33(11):1325–1333, 2010.

[9] Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003.

[10] Colin Reeves and Jonathan E Rowe. *Genetic algorithms: principles and perspectives: a guide to GA theory*, volume 20. Springer Science & Business Media, 2002.

[11] Pablo Moscato et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826:1989, 1989.

[12] Robert Trivers and Richard Dawkins. The selfish gene, 1976.

[13] Sudha Anbalagan, Ali Kashif Bashir, Gunasekaran Raja, Priyanka Dhanasekaran, Geetha Vijayaraghavan, Usman Tariq, and Mohsen Guizani. Machine learning-based efficient and secure rsu placement mechanism for software defined-iov. *IEEE Internet of Things Journal*, 2021.