Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

A Study on Various Task-Work Allocation Algorithms in Swarm Robotics

Annamalai L, Mohammed Siddiq M, Ravi Shankar S*, Vigneshwar S

Students, Department of Robotics and Automation, PSG College of India, Coimbatore, India

E-mail: *ivarraknash3@gmail.com

Abstract: This paper discusses the various task allocation algorithms that have been researched, analyzed, and used in swarm robotics. The main reason for switching over to swarm robotics from ordinary mobile robots is because of its ability to perform complex tasks co-operatively with other bots rather than individually. Furthermore, they can be scaled to perform any kind of tasks. To carry out tasks like foraging, surveying and other such tasks that require swarm intelligence, task allocation plays an important role. It is the crux of the entire system and plays a huge role in the success of the implementation of swarm robotics. Few algorithms that address this task allocation have been briefly discussed here.

Keywords: swarm robotics, task-allocation algorithm, bio-memetics, fixed threshold response, extreme- conn, co-operative robots.

1. Introduction

Swarm Robots deal with the coordination of multi-robot frameworks which comprise of enormous quantities of generally simple physical robots. It is assumed that the ideal swarm behaviour rises out of the connection between the robots and the environment. Swarm is a bio-inspired idea of utilizing different small robots to perform one large task [1]. It takes its motivation through biomimetics from social orders of insects, birds, etc., which can perform assignments and can't be done by a single individual.

Swarm robotics promotes the development of systems that can cope well with the failure of one or more of their constituent robots: the loss of individual robots does not imply the failure of the whole swarm. Fault tolerance is enabled by the high redundancy of the swarm: the swarm does not rely on any centralized control entity, leaders, or any individual robot playing a predefined role [3].

Task allocation in swarm robotics is an important task as it answers the most important question - how can the robots of the swarm assign themselves to one of the announced tasks in a distributed and efficient way. Here in our work, we have discussed four different task allocation algorithms addressing different ways in which a task or a problem can be solved by a group of robots. These algorithms are already in research and

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

These algorithms were Task allocation is a way that tasks are chosen, assigned, subdivided, and coordinated (here, within a single colony of robots). The remaining of the paper discusses the various algorithms in use now based on biomimetics.

2. Distributed Bee Algorithm

Robot swarms are multi-robot frameworks that normally comprise of a huge group of simple robots cooperating locally with each other and with their surroundings. These frameworks attract motivation from animal swarms however their plan isn't compelled by biological credibility. Their primary element is decentralized coordination which brings about ideal conduct that rises out of the rules of nearby communications. The distributed bees' algorithm (DBA) [3] can be utilized for task designation in a swarm of robots. In the proposed calculation, task allotment comprises in doling out the robots to the discovered targets in a 2-D field. The expected distribution is acquired from the objectives' characteristics or the priority of the task at the objective which can be taken to as scalar qualities.

The goal of the algorithm proposed here is to appoint the robots in a swarm to the discovered targets so that the final dispersion is proportional to the objectives' characteristics/priority. The objectives with related characteristics represent a distributed "food" that requires a distribution of robots generally a non-uniform one in the region.

Despite having many upsides of collaboration, the proposed centralized methodology can't be applied if not all robots can speak with one another. Decentralized coordination of robots has different advantageous circumstances over traditional customary incorporated methodologies. It tends to be applied to lessen the trouble on the multi-robot framework, particularly for enormous groups of robots. In certain applications, communications can be difficult to execute or no communications exist by any stretch. The base up plan topology innate to bio-inspired multi-robot frameworks furnishes them with at least one of the following features, such as being self-sufficient, versatile, hearty and versatile to changes in their environment. Then again, the aggregate conduct has new properties that enable them to create eccentric examples.

Given the depicted scientific classification, the multi-robot framework we proposed can be sorted as homogeneous and circulated, utilizing broadcast communication [4]. We address an issue of instantaneous assignment of single-task robots in multi-robot undertakings. The task allocation situation we study considers the condition that contains various tasks that could be of same or distinctive significance and robots that are similarly equipped for playing out each assignment, however, must be distributed out to one at some random time. The quality of an objective is an application-dependent scalar value that may represent a target's need or priority, where a higher value requires more robots to be apportioned. For

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

instance, it could represent the extravagance of the mineral or water source on a planet that we need to bridle, the measure of trash to be gathered in an open space.

Consider a group of M robots to be assigned among A targets. Let P $\in\{p_1,...,p_A\}$ represent the set of normalized qualities of all available targets. Here the number of robots on the target is denoted as i $\in\{1,...,A\}$ by M_i, a nonnegative integer. The population fraction allocated to target i is $h_i = n_i/M$, which represents the target task's relative frequency, and the vector of population fraction is $h =[h_1,...,h_A]^T$. The distribution expected is the group of desired population fractions for each target, $h^d = [h^d{}_1 ,...,h^d{}_A]^T$, where $h^d{}_i = p_i$. For scaling purposes, the fractions are used rather than integers, but such usage also presents a distribution error. Such errors occur because the fractional values can take an only particular set of values that are defined by the swarm size. However, the decision taken by one robot in the swarm is not known by other robots so in the proposed scenario the optimization of the system based on the maximum utility cannot be applied. Therefore, we propose the algorithm of DBA.

2.1 Algorithm:

The utility value for each target is calculated as soon as the robot receives information about the target. The utility value for each target depends on the target's quality and the relative cost which represents the distance between the robot and the respective target.

2.1.1 Costs: For target i the cost for robot k is calculated as the Euclidean distance between the robot and the target in a 2-D arena

$$(s^k_i)^2 = (x_i - x_k)^2 + (y_i - y_k)^2$$

where (x_i,y_i) represent target's coordinates and (x_k,y_k) represent robot's coordinates in the 2-D arena. However, the reciprocal of the distance is used to calculate the utility represented as the target's visibility.

$$\eta^k_i = (s^k_i)^{-1}$$

2.1.2 Qualities: A particular value must be assigned to each of the targets so that the robots will know the priority or the complexity of particular targets. Normalized qualities are calculated as fractions of the sum of qualities of all available targets.

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

$p_i = P_i / \sum^{A}_{j=1} P_j$

where P_i is a quality of the target i. In real-world scenarios, the quality of a region of interest is an estimated value that is as a result of sensor-readings or a previously acquired knowledge.

2.1.3 Computing Utilities: The utility of a robot depends on both, cost and quality of the chosen target. We define the utility as a probability that the robot k is allocated to the target i, and it is calculated as follows:

$p_i^k = (p_i{}^{\alpha}\, \eta_i^{\beta}) / \sum^{A}_{j=1} p_j{}^{\alpha}\, \eta_j^{\beta}$

where α and β represent the control parameters which is used to bias the decision-making mechanism between the quality of the solution or its cost, respectively (α, β > 0; α, β ∈ R).

$$\sum_{i=1}^{A} P_i^k = 1$$

2.1.4 Decision-Making:

The initial position of robots plays an important role as it defines the distance cost according to all targets before the task allocation so the utility value will vary for the same target since the distance cost varies. The initial position of robots in swarm also determines the end robot distribution. We know that all the target is associated with a probability value which is calculated from the equation in 2.3.1. As the probabilities are calculated, a target is chosen by the robot by 'spinning the wheel'. The utility values cannot be computed for the whole of a swarm as the algorithm proposes a decentralized and distributed method, also the quality value from the targets is used as a quantum of the expected distribution of robots. The equation in 2.3.1 works in such a way that the closer targets have higher attraction than the others.

3. Fixed Response Threshold Model

This is another algorithm [5] which is modelled after the regulation of the division of labour seen in insect societies. This model explains the same flexible and decentralized task allocation found in ants used in

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

SRTA (swarm robot task allocation). Generally speaking, SRTA models have to be robust and scalable which are two features missing in MRTA (multi-robot task allocation). To explain this algorithm again foraging resources has been chosen as the problem but this algorithm can also be extended to search and rescue. This algorithm can be ideally used where we have only local communication and very limited knowledge about the environment.

S- the intensity of stimulus associated with a task

Θ- response threshold, an internal variable that determines the tendency of the robot/ant to react to S

The response function $T_{\Theta}(S)$ used here is given below

$$T_{\Theta}(S) = S^n/(S^n + \theta^n) \quad (n>1)$$

Engaging a task =0 if s<< Θ and vice versa. Thus, among $\Theta_1 < \Theta_2$ the bots with the low value of Θ are more likely to respond when two or more robots encounter the same task.

$$\lim_{n\to+\infty} T_{\Theta}(S) = \lim_{n\to+\infty} \left(\frac{S^n}{S^n+\Theta^n}\right)$$

$$\lim_{n\to+\infty} T_{\Theta}(S) = 0 \; if \; S < \theta$$

$$\lim_{n\to+\infty} T_{\Theta}(S) = 1 \; if \; S >= \theta$$

From the above equation, the entire model is a sigmoid function. In foraging resources, the bots must search for resources only when the resources decline in the base station or home zone. The goal for the swarm in this problem is that the resources must be successfully replenished in the base. This model assumes that all robots have the same value of θ, and n is randomly generated across all bots so that different robots react differently to the stimulus(S). On plotting the sigmoid function using the $T_{\theta}(S)$ it has been observed that when S<θ, the robots with smaller n value have a higher probability to go foraging and same is the case for robots with a higher value of n when S> θ [6].

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

At time t, the probability that a robot in a swarm will decide to forage for resources or stay back is given by the probability P_f.

$$P_f = 0 \text{ if } N_t >= N_I$$

$$P_f = (N_I - n_p)^n / (N_I - n_p)^n + \theta^n \quad \text{if } N_t < N_I$$

Where

N_I – the number resources in home station, initially

N_t- stimuli

n_p- the resources left in the home at time t.

When this algorithm is implemented the bots go through four stages namely Wait, Search, Return, Deliver.

Wait: At the beginning of the mission, all robots will start a timer the delay equally distributed in a fixed range. The timer will control the intensity of N_t. When the timer expires the robots calculate P_f and N_t.

Search: The transition from Wait to Search happens based on the values of P_f. Another timer for searching the resource is started and within the limited time, if the robot finds any such resource it moves to the Deliver state or else Return state

Return: Failed to find any necessary resources the robot returns to the base for charging. By chance, if it finds any resources it moves to Deliver state

Deliver: Deliver the resource to base and re-enter Wait state.

The individual robot only must sense the stimuli N_t which is randomly generated in the Wait state. Even if there is an increase in the number of robots in the swarm it will not affect the foraging strategy of the individual bots.

4. Novel Communication Methods: Communication through light signals, Gossip based communication and Leader follower approach

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

Many methods exist to tackle the problems due to the high variability in swarm robotics. Each has their own merits and are suited to particular problem statements. In swarm robotics, the market-based strategy is the most preferred because of high efficiency. It follows a centralized approach discussed as algorithm [7] below wherein it requires a central master and a long communication range which might not be feasible in all cases.

There are also threshold-based systems which are used frequently. They bear a striking similarity to biological methods of task allocation. The threshold variation is calculated and depending on the intensity of the task signal, task allocation is done. The methods which are discussed below provide a new formulation for task allocation.

4.1 Light Signal Based:

This method is highly useful in the case of heterogeneous swarms where different swarms have different functionalities. In communication by light signals, depending on the light's colour, the other robots may get attracted to the light or get repelled by it. Depending on the total number of required bots, the intensity of the light differs. In contrast to the attraction light, the repulsion light covers only a short region. Each robot is influenced by both the attractive and repulsive lights, which in turn provides appropriate light signals according to the requirement. There is also a set internal frustration level for each robot, which occurs when both attraction and repulsion is experienced. At this instance, the robot moves a certain distance opposite to the direction of the attractive signal and finds other tasks. If the robot experiences no light-based signals, it just exhibits random movements and keeps searching for tasks.

Each robot calculates the cost required to reach the destination zones. If destination zones are more than one, the robot calculates the cost to traverse to each one and starts moving to the nearest goal. An appropriate heuristic is chosen in the graph-based approach for path planning to calculate the cost.

4.2 Gossip Based Approach:

This method uses wireless technology like an infrared module, Bluetooth to transmit necessary information. Each robot is assigned a specific identification code and receives the number of tasks from a common master.

The task information is given in the following format:

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

1) Task ID - This corresponds to the ID of the supervisor robot announcing the task.
2) Required workers - The number of robots the task requires.
3) Hops - The number of hops to the task. (in terms of communication)
4) Route length - The distance to the task following the hops.
5) Age - The age of the information about the task.

The hops are the primary constraint in moving to the next task. The hops are nothing but the successive positions of the robot which emits the signal. The robot moves in a hop-hop trajectory rather than the straight line. The attraction to a task is active only if the required number of robots is greater than one. Here too, the robot has an internal frustration level, this increases as the distance from the destination increases. In case this value crosses a level, the robot stops moving towards this task. Similar to the light signal-based approach, if no active task is present, the robot exhibits random movements. Along with this, there is an obstacle avoidance movement which is done in the presence of obstacles. If near a task destination, this obstacle avoidance is overridden to focus on the task as the task might be an obstacle.

Compared to the light-based approach gossip-based approach is more complex and more efficient in case of a smaller number of robots. In the case of a large number of robots, the difference in time to do the tasks is negligible. The gossiping can take place with the help of local communication between a particular group of neighbour robots, thereby eliminating the major disadvantage of auction-based market strategy. Also, here the robots entirely take the decisions on their own, contributing to more autonomy to individual robots.

4.3 Leader-Follower Approach:

A simple yet effective approach to do small scale tasks is the bi/multi-directional leader-follower approach which takes the cost to reach the destination [8]. Based on the distance to the goal, the master bot is chosen. The slave bots follow the master to reach the goal. Here, the number of required bots per task information is provided to each robot. In case for the next task, one of the slave bots which follow is nearer compared to others, it is given the master position while the others are supposed to follow the master based on their distance from it and also based on the required number. This is most suitable for a small number of tasks with a limited number of robots as it is less complex. A major disadvantage in this method includes high computational power required to calculate the cost to the goal each time for every robot. This also lacks the autonomy which the above methods provide. Also, this is not effective in case of a large number of robots and a large number of tasks due to the limitations in assigning leaders to each task and requiring a wide network with a massive range.

5. Extreme-Comm Algorithm

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

This can be used in a particular case where a large swarm of homogeneous robots are divided into subgroups, with each of them assigned a specific task. These dynamic task assignment in swarm systems can be seen naturally in ants when ants forage different groups of workers must parallelly scout for food, lay trails, and carry their food back to their nest. A typical modern world application can be compared to a search and rescue mission, wherein the case of 100 homogenous bots are deployed an ideal task allocation would be 75 robots work to explore the environment and 5 robots mark the resources and 20 robots maintain the communication network. This distribution, (75%,5%,20%) should be maintained even when reinforcements arrive, or robots leave the area to charge their batteries, or in case some robots malfunction and gets out of the system. When an entire subgroup is removed, the remaining robots should still be able to reallocate themselves to maintain a global distribution. The system should also be alert and reactive to any new global distribution inputs from a user or a task-allocation algorithm. This task allocation problem, therefore, can be considered as two components, the first component to decide an optimal number of subgroups, with each subgroup performing a different task. The second to achieve the robot join a subgroup so that they can achieve a desired global task distribution. The first component is task-dependent, so assuming we have a "black-box" solution to it we will focus on the second component.

At a particular moment t, the task distribution vector possessed by the system is dv(t) = (n1/n,. . . , nm/n) in which the number of robots in a given task-group referenced as i is given by ni. The problem is finding a distributed algorithm for a global task assignment is that given a target distribution vector p = (p1, . . . , pm), the distribution dv(t) should converge close to p.

That is,

$$\lim_{t\to\infty} dv(t) = argmin_v\{\|v - p\|_2 \qquad \text{for} |v| = n\}.$$

The algorithm should also be unresponsive to any uninformed changes in the system like change in the number of robots in the network n, or the topology of graph G, while the network remains connected.

The Extreme-Comm [9] algorithm is a suitable solution to this problem. In this algorithm, each robot uses local communications to build a broad list of all other bots in the swarm and uses this new list to determine its task. This algorithm is fast and accurate but requires an extreme amount of inter-robot communications.

5.1. Execution Cycle

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

This algorithm is robust to change in the number of robots at any time, which is vital in any real-time multi-robot system. After each robot creates a list of IDs, it selects its task based on its relative position in this list. Though this algorithm runs quickly it requires a large amount of inter-robot communications. During each execution cycle, the robot transmits a Robot-ID message to its neighbours. This message is a tuple consisting of its ID and timestamps, i.e., its present clock execution value. In each cycle, the robot obtains a new Robot-ID message from its neighbours and complies a list from it. In the next cycle, it rebroadcasts this message list, along with its Robot-ID message and an updated timestamp. The algorithm runs uninterruptedly so that Robot-ID messages from each robot circulate throughout the network. After T = Diam(G) cycles, each robot will be having an accurate list of all the robots in G, where G is a directed graph in which each node represents corresponding robot and their representing the corresponding communication links between them. If any new robot is added to the network a message will be propagated throughout the system, so its addition will be known within T execution cycles by all robots. Once a message has been transmitted, it gets appended to a separate list for a "refractory period" P before being deleted. There is no rebroadcast feature available for robots to transmit messages from this list, or any copies of them received from neighbours, a second time. When a robot is removed from the network, its removal will also be known to all other robots in at most T execution cycles, since it is no longer possible for it to broadcast its ID in the subsequent executions.

In this algorithm since each of the robot being able to maintain an IDs list of all the robots present in the network, it is easy to find the relative position of the robot itself from that list. Based on its position from the list each robot takes on a task and discards it with respect to p. This algorithm doesn't need to use an unbounded timestamp in the Robot-ID messages, it can reset it to its default value 0 after reaching some maximum value, given that the maximum value is larger than the refractory period P. Diam(G) is the largest value that P could take, but it is also possible for P to take values as short as 2 cycles, but by giving smaller values, the algorithm becomes less robust when a change in network topology occurs. The algorithm is self-stabilizing and converges deterministically to the correct task distribution within T cycles of initialization or T + P cycles of any unexpected deviations. The communications complexity per-robot per-cycle scales as n, since each robot must send one Robot-ID message for every robot in the network. The total number of messages sent by all robots during convergence scales as $\text{Diam}(G) \times n^2$, which could make this algorithm impractical for a swarm with moderate and limited communication bandwidth. Though this algorithm seems aesthetically unattractive because of the large amount of data being collected about global information on each robot this is one of the fastest and more accurate algorithms.

6. Conclusion

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

The several methods used here correspond to different applications in the field of robotics. It entirely depends on the requirements and no method is inferior to others though there may be updated on each algorithm. With the advent of heavy computing infrastructure and depleting workforce to do heavy jobs, the developments of swarm robots in the coming years is expected and largely beneficial to industries. There are also new algorithms being proposed based on Artificial Intelligence leveraging Neural Networks which prove to be increasingly effective in task allocation.

References

[1] X. Yi, A. Zhu, S. X. Yang and C. Luo (2017). A Bio-Inspired Approach to Task Assignment of Swarm Robots in 3-D Dynamic Environments, in IEEE Transactions on Cybernetics, vol. 47, no. 4, pp. 974-983.

[2] Winfield A F T and Nembrini Julien (2006). Safety in Numbers: Fault Tolerance in Robot Swarms, in International Journal on Modelling Identification and Control, vol. 1, no. 1, pp. 30-37.

[3] Aleksandar Jevtic, Alvaro Guti errez, Diego Andina, and Mo Jamshidi (June 2012). Distributed Bees Algorithm for Task Allocation in Swarm of Robots. IEEE Systems Journal.

[4] Pham, D.T., Eldukhri, E.E., Soroka, A.J., Phama, D.T., & Ghanbarzadeha, A. (2009). Multi-Objective Optimisation using the Bees Algorithm.

[5] Yongming Yang, Changjiu Zhou, Yantao Tian (Feb 10-12, 2009). Swarm Robots Task Allocation Based on Response Threshold Model. Proceedings of the 4th International Conference on Autonomous Robots and Agents.

[6] Eduardo Castello, Tomoyuki Yamamoto, Yutaka Nakamura, Hiroshi Ishiguro. "Foraging optimization in swarm robotic systems based on an adaptive response threshold model", Advanced Robotics, 2014

[7] Ducatelle, Frederick & Förster, Alexander & Di Caro, Gianni and Gambardella, Luca Maria (2009). New task allocation methods for robotic swarms. Proceedings of the 9th IEEE/RAS Conference on Autonomous Robot Systems and Competitions.

[8] Annamalai L, Ravi Shankar S, Mohammed Siddiq M, Vigneshwar S. "Implementation of a multi-agent mobile robot system to perform co-operative tasks", 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019

[9] James McLurkin and Daniel Yamins. Dynamic Task Assignment in Robot Swarms.

Authors Biography

ISSN: 2582-418X (online)

Journal of Information Technology and Digital World (2020)
Vol.02/ No. 02
Pages: 123-134
https://www.irojournals.com/itdw/
DOI: https://doi.org/10.36548/jitdw.2020.2.006

Ravi Shankar. S Senior Undergraduate, BE - Robotics & Automation, PSG College of Technology. An unremitting team player, who expands his bailiwick constantly. He is a Machine Learning fanatic and likes to manipulate and play with data, finding evidence that facilitates in creating statistical models in forecasting future results. He is looking forward to pursuing higher in robotics.

Annamalai. L,Senior Undergraduate, BE - Robotics & Automation, PSG College of Technology.He is currently working as a Research Intern in the development of Exoskeleton at IIT Bombay. He is bestowed with the Summer Research Fellow award by Indian Academy of Sciences in the field of Control Systems. He is primarily concerned with Manipulation, Navigation and Perception for Robots.

Vigneshwar S, Senior Undergraduate, BE - Robotics & Automation, PSG College of Technology.His projects and research areas include Co-operative Robotics, Applied Artificial Intelligence and Self Driving Cars. He has visited Festo India, Bosch India, CVRDE-DRDO as an Intern and studied the various systems implemented there and has taken up a project at Bosch India focusing on AI in engine calibration.

Mohammed Siddiq M, Senior Undergraduate, BE - Robotics & Automation, PSG College of Technology. His projects focus on Applied Robotics in Defense and Medical applications. His primary interests include Haptics and Bio-Medical Robotic Technologies. He is currently working as a Research Intern in Robert Bosch Engineering and Business Solutions Private Limited in a project involving Robot Process Automation (RPA).

ISSN: 2582-418X (online)