Incorporation of Swarm Intelligence in Autonomous Cars

Yadhu Prakash, Kahan Prabhu, Shruti Kamtekar, Sainath Gadhe

Computer Department, Fr. C. Rodrigues Institute of Technology,
Vashi, Navi Mumbai, India

***Abstract*—Currently, swarm robotics is one of the most promising technologies in computer science. As technologies become smarter, the potential for swarm robotics improve and the coordination of multi robot system finds applications in various fields. This paper adds new perspective to autonomous cars by proposing to incorporate swarm robotics into it. This paper summarizes various algorithms used in swarm robotics that can also be used in autonomous cars. The current working of autonomous cars would tend to be inefficient in future when streets would be populated with them, by using our idea the efficiency would be improved massively.**

***Keywords*—Autonomous Cars, Swarm Robotics, Swarm Intelligence, Swarm Algorithms, Google cars, Autonomous cars.**

INTRODUCTION

Swarm robotics finds its applications in various fields ranging from space exploration to bomb defusal. The ability to divide tasks and find a collective solution to a particular problem is the highlight of swarm robotics. Various algorithms have been written to improve the collective behavior of a swarm. Autonomous cars have retained the limelight, thanks to Google cars. The success of autonomous cars would result in a large number of self-driven cars on our streets in the near future. The concept of swarm robotics and autonomous cars are closely related and combining them would only result in improved efficiency in the future.

I. SWARM INTELLIGENCE

Swarm Intelligence systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The group of individuals acting in such a manner is referred to as a swarm [1]. The main objective of swarm intelligence is to aggregate the individual behavior, interactions with the neighboring robots and the interactions with the environment to achieve a collective behavior that can be used to solve problems collectively. For the swarm to be more intelligent, the swarm should work more congruently.

A. Introduction to Swarm Intelligence Algorithms

Most well-known algorithms to implement Swarm Intelligence are Ant Colony Optimization and Particle Swarm Optimization.

1) Ant Colony Optimization: The ant colony optimization as the name suggests is based on the collective behavior of the ant colony. Considering the collective work of an ant colony and not an individual ant suggests how problems of survival can be solved. In order to demonstrate this concept Deneubourg conducted the Binary Bridge Experiment [2]. In this experiment ants take two different paths to one particular food source, the ant that returns back first to the colony is the one taking the shorter path. Pheromone (ants deposit pheromone while walking) concentration on the shorter path will be higher since more ants would complete the loop through the shorter path as compared to the longer path. This further encourages the ants to take the shorter path as a result of their social behavior (of following pheromone trail). As a result all ants start to take the shorter route, thus, social interaction and coordination for foraging occurs indirectly through pheromone deposits which modify the environment.

Similar concept is used in Swarm robotic algorithm using ACO, a virtual pheromone is deployed to improve the problem solving efficiency, as the task matures more and more efficient solutions may be observed.

2) Particle Swarm Optimization [3]: Particle Swarm Optimization is mostly associated with the bird flocking analogy. Imagine a flock of birds circling over an area where they can smell a source of food. The bird that is closest to the food source chirps louder and all other birds fly around his direction, as soon as another bird comes even more closer to the target, it will chirp louder that the first bird and hence will result in all the other birds flying in his direction. This pattern continues until one of the birds reaches the target (food). Using this analogy PSO is explained, over a *number of* iterations, a group of variables (birds) have their values adjusted, such that with each adjustment a more efficient solution (target food) is obtained. The adjusted value would be closer to the member whose value is closest to the target.

II. EXISTING ALGORITHMS IN SWARM ROBOTICS:

A. Shape formation Algorithm in a swarm:

Situations might arise where individual agents in a swarm might want to align themselves in a line, or aggregate themselves into shapes, of varying forms and sizes. The SHAPEBUGS Algorithm employs a decentralized approach to achieve swarm formations using local interactions. The algorithm is flexible in that it continues to work even in the face of an unprecedented influx or exodus of agents. The first process of the SHAPEBUGS Algorithm uses trilateration coupled with a gradient algorithm to help the agent locate itself in the arena. A successful instance of execution of the Gradient Algorithm depends on the initial starting position, an approximation of which is provided by the process of trilateration. It is important to take into account the likelihood of errors, on the part of the proximity sensors and the algorithm. Thus an arithmetic mean of the trilaterations needs to be taken. The entire arena is divided into two parts, one being the area within the desired shape, the other being outside it. An agent that is lost automatically assumes it is outside the shape, and hopes to find a way into it. Meanwhile agents that are already inside the shape aren't allowed to get out. Fluctuations of agent density inside the shape formation are handled using the concept of gas particles in a container. Agents move away from regions of higher concentrations to regions of lower concentrations, till equilibrium is achieved, thus making the swarm capable of sustaining loss of agents provided there are enough left to generate a sensible equilibrium. The goal of equalizing pressure at any agent density can be achieved by making the repulsive force decay in proportion to the distance between agents. The SHAPEBUGS Algorithm can thus satisfactorily form arbitrary shapes without compromising performance to the various sources of error and agent influx and death.

B. Traffic Collision Avoidance in swarms [4]

This algorithm was developed for collision avoidance for the delivery robots of a warehouse. This approach is efficient because it focuses on how to solve the problem in a dynamic, structured and crowded environment. All robots follow paths assigned to them dynamically using the A* algorithm. The collision avoidance algorithm defines a front area as shown in figure 1. The origin of front area is the current position of the robot and its orientation depends on the current position and the next waypoint in the path to be reached.

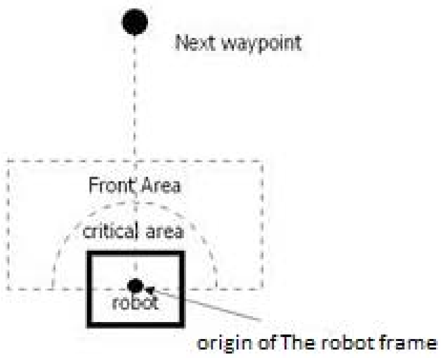

Fig 1. Definition of the front area and the critical area.

In case of collision detection, specific behavior is automatically selected and the automatic selection of the behavior is based on certain traffic rules.

Eight behaviors of robots as defined by the algorithm are as follows

- In the **FollowWayPoint** behavior, the robot follows the next waypoint until the robot reaches the final station. This is the robot's default operation.
- In the **Avoid** behavior, the robot moves around a robot which is its partner, or an unknown obstacle treated like its partner. Once the partner leaves the front area, the behavior is completed and FollowWayPoint is continued.
- The **Exchange** behavior is applicable for a head-on collision, and the two robots pass each other and become each other's partners. Once the partner leaves the front area, the behavior is completed and FollowWayPoint is continued.
- The **GoThrough** behavior is used to deal with a side collision, when the robot is going through an intersection, the other has to wait until this robot has passed.
- The **Dock** behavior of the robot is initiated when it reaches the docking region of a station and starts to dock at the station.
- In the **WaitKeepDistance** behavior, the robot waits for a partner to move and keeps a certain distance from it. Once the partner leaves the front area, the behavior is completed and FollowWayPoint is continued.
- For the **WaitForGoThrough** behavior the robot must wait and if needed, make the intersection free for the partner to pass through.
- In the **WaitForDocking** behavior the robot must wait until another finishes docking at the same station.

The traffic rules for executing these behaviors are written in the form of algorithms, taking into consideration all the possibilities of collisions in the given circumstance. For each collision scenario one of the above said behaviors will be triggered to avoid collision. Once a new possible collision is found in the front area, the robot will select a new behavior using the traffic rules.

C. Dynamic Task Assignment Algorithm:

The objective of task assignment is to divide a swarm into a number of functional subgroups, such that each subgroup is capable of performing a task on its own. Dynamic task assignment ensures that each agent in a swarm gets a predefined share of the total workload. This process is continuously adjusted in response to changes at runtime thereby assuring a fair distribution of tasks.

1) Card Dealer's Algorithm: In this algorithm, communication is minimal which we can be considered as an advantage. It divides the task assignment problem into a series of stages [5]. At each stage, the task is considered a function of that stage number, and a card is dealt to a person as a function of his position around a gaming table. Before focusing or say reaching on to the next stage, the Card Dealer's algorithm completes the current stage. This delays the entire procedure and eventually slows down the operation. But here memory requirements are minimum with minimal agent interactions as it neither calculates nor stores any global quantities.

2) Extreme Communication Algorithm: In Extreme Communication algorithm a large amount of inter agent communication (local interaction) is needed to build a complete list of all other agents in the swarm. This list is then used to determine the task. This is a fast running algorithm, with a large amount of inter agent communication required to achieve this. All agents need to form a list of the IDs of all the other agents in the network. At every execution cycle each machine sends its ID and timestamp to all the agents and thereby collects similar messages from others too. Basically it collects information about the number of agents active and its own position relative to the others. On the basis of relative position, the task is distributed among all the agents. For the next cycle, the agent updates its list with the received IDs and sends the updated timestamp and list. This needs a robust network. The prominent feature of this algorithm is that the task is given on the basis of 'Relative position' which is helpful in our scenario but drawback is that unwanted information is collected.

III. . How Self Driven Cars Work:

The much awaited Google Car shows promising results as an autonomous vehicle especially with regards to passenger safety. It has now been driven over 300,000 miles accident free. Mr. Urmson[6](tech lead for the project) accurately attributed a Velodyne 64-beam laser as the heart of the project. Along with this the car also hosts various sensors such as 4 radars on the front and rear bumpers, a camera near the rear view mirror, a GPS system and an inertial measurement unit and a wheel encoder. The Velodyne 64-beam laser is used to generate a three dimensional mapping of the environment. Meanwhile, the other sensors and radars are used to determine the vehicle's location and track its movement.

IV. Problem Statement:

With gaining popularity of autonomous cars, it is only a matter of time before which hundreds of self-driven cars will flock our roads. The current system associated with such cars would drastically reduce the efficiency, as each car would be required to compute its own course. With safety being one of the highest priority a single wayward car could compromise on security of all cars on the roads.

V. Our Application Of Swarm Robotics In Autonomous Cars:

We propose that autonomous cars incorporate the concept of swarm robotics into the self- driving vehicles. An autonomous vehicle incorporating such a concept would still require a GPS system that could break a car from the swarm to guide it to its destination. Once broken from the swarm the car may join any swarm that is moving in its desired direction or navigate autonomously to become leader of a newly developed swarm.

A. . *Advantage of such an application would be the following:*

- As autonomous cars move in swarms, constant average speed would be maintained, thereby ensuring constant traffic.
- Higher safety, in case of failure of individual sensor, probability of collision both within the swarm and outside with pedestrians or obstacles reduces.
- Higher efficiency as workload of collision detection and navigation is divided among large number of autonomous cars.
- Higher efficiency as independent navigation and creation of 3D data models is required individually, only when car is not part of a swarm.
- Better response to traffic situations as cooperation between cars would result in faster clearing of roads with traffic.

B. *Disadvantages of such an implementation:*

- The upper limit on the number of robots in a swarm could be a disadvantage
- If at all any of the robots in the swarm cause a deadlock, it may pose a threat.
- The security mechanism in a swarm is inefficient
- Multiple swarms coming in contact with each other may create problems

VI. Conclusions

As mentioned earlier, automated cars are surely going to dominate global road traffic in the years to come. With the proposed integration of swarm behavior in such cars, we eliminate all limitations faced by them when on their own. The repercussions of individual entities interacting and coordinating with one another to achieve feats of massive proportions has profound implications on the technology of tomorrow. If exploited judiciously, the concept of autonomous cars in a swarm is revolutionary and is here to stay.

References

[1] http://en.wikipedia.org/wiki/Swarm_intelligence

[2] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels. The self-organizing exploratory pattern of the argentine ant. Journal of Insect Behavior, 3:159–168, 1990.

[3] http://mnemstudio.org/particle-swarm-introduction.htm

[4] Dali Sun, Alexander Kleiner, Bernhard Nebel Behavior-based Multi-Robot Collision Avoidance, 2014 IEEE International Conference on Robotics & Automation (ICRA)

[5] Daniel Yamins , James McLurkin. Dynamic Task Assignment in Robot Swarms

[6] http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works