A SURVEY OF COMPARISON BETWEEN VARIOUS META-HEURISTIC TECHNIQUES FOR PATH PLANNING PROBLEM

Toolika Arora, Yogita Gigras,
ITM University, Gurgaon, Haryana, India

ABSTRACT
Path planning is one of the challenging topics of robotics. Meta-heuristics designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. In this paper various meta-heuristic techniques like Ant Colony Optimization, Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing and Tabu Search are compared with each other to obtain solution for path planning problem. The problem is to obtain path from source to destination of an autonomous mobile robot in a moving space with dynamic obstacles. The mobile robot has to reach the target position by avoiding collision with obstacles by finding optimum collision free path from source to destination point.

KEYWORDS: *ACO – Ant Colony Optimization, GA – Genetic Algorithm, SA – Simulated Annealing, PSO – Particle Swarm Optimization, TL – Tabu List, NN - Neural Network, I & D - Intensification and Diversification*

1. INTRODUCTION

1.1 Path Planning

The path planning problem is a very important problem in robotics. The robotic path planning is defined as the determination of a path of robot from a starting point to the destination point while avoiding any collision from obstacles in between. Many algorithms have been proposed for this purpose. The major difficulties for generating path planning algorithms for mobile robots are: efficiency and safety. Efficiency means it should take optimal time and safety means avoiding collision with obstacles. Path planning can be done in static as well as dynamic environment. Static means position of obstacles is fixed and does not changes with time while dynamic means that the obstacles position is not fixed i.e. changes with time. Path planning can be classified as: global path planning and local path planning. If the environment is known to the robot then it is called global path planning while if the environment is unknown to the robot then it has to sense the presence of obstacles and take decisions about its movement towards destination point it is called local path planning. Global path planning is done in offline mode before robot starts to move while local path planning is done in online mode where robot has to avoid obstacles in real time. In this paper meta-heuristic techniques are used to solve this problem. Meta-heuristics are the algorithm that can be used to solve almost all optimization problems. It designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Various meta-heuristics techniques are Ant Colony Optimization (ACO), Simulated Annealing, Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Tabu Search etc. A number of approaches have been proposed for solving this problem like: N.Sadati and J.Taheri presented a combination method consisting of a Hopfield Neural Net (NN) and a genetic Algorithm (GA). Kyung min Han presented a path planning algorithm that uses Ant Colony Optimization (ACO) and Genetic Algorithm (GA). The purpose of this paper is to find a navigational path of a mobile robot in a dynamic environment where the robot has to reach a target point by avoiding collision with obstacles present in its path.

2. META-HEURISTIC TECHNIQUES

For solving optimization problem we have choice from Brute force, approximation and heuristic methods.

1. Brute force method: it provides optimum solution for the problem also called exact method. It has no guarantee on the running time.
2. Approximation method: it provides a certain quality result for a problem. Here the distance of the result from the optimum solution is known. It guaranteed to run in polynomial time.
3. Heuristic method: It provides a result that is "good enough" for a problem. These methods are known for their speed when considering computation resources the other methods are more expensive. So, the running time overcomes the quality of the result. Sometimes this method is not able to find solution for a problem but these situations rarely occur. Heuristic is an approximate solution to a problem. The difference between this and the previous one is that this is about getting a good solution to a problem but you do not know how close it is to the optimal solution. The previous one is about getting a solution which you can determine how close it is to the optimal one.

Another important one is meta- heuristics. Heuristics are problem-dependent, i.e, you can define an heuristic for a specific problem. Meta-heuristics are problem-independent techniques that can provide solution for a broad range of problems. You could say that a heuristic uses problem-related data to find a 'good enough' solution to a specific problem, while meta-heuristics are used to generate general algorithms that can be implemented to a broad range of problems.

Various meta-heuristic techniques are:

Ant Colony Optimization: ACO is a meta-heuristic technique proposed by M. Dorigo et.al in 1992 for solving optimization problems. It is a probabilistic technique that can be applied to optimization problems that itself and their solution can be best described using graphs. It is inspired by the real ants behaviour how they forage for food. Real ants are capable of finding the shortest from a food source to their nest by exploiting pheromone info, where pheromone is a chemical substance deposited by ants. While walking ants deposit pheromone on the ground and follow, probabilistically, paths that have greater amount of pheromone. In ACO a finite size colony of artificial ants is created. Each ant then builds a solution to the problem. While building its own solution each ant collects information based on the problem characteristics and on its own performance and store this information on the pheromone trails associated with the connection of all edges. These pheromone trails play the role of a distributed long term memory about the whole ant search process. Edges can also have an additional associated value η representing a priori information about the problem. Once all ants have computed all their tour ACO algorithm updated the pheromone trail using all solutions produced by the ant colony. Let $\tau_{ij(k)}$ be the pheromone level on edge (i, j). The probability that an ant k chooses j as the next vertex when it is at vertex i at iteration k is given by

$$p_{ij} = \left\{ \frac{\tau_{ij}\eta_{ij}^{\beta}}{\sum_{0,z\in J(i)} \tau_{iz}\,\eta_{iz}^{\beta}}, if\ j\epsilon J(i) \middle| 0, otherwise \right\}$$

Where $J(i)$ is the set of vertices that remain to be visited by the ant and β represent relative influence of pheromone and heuristic information. After all ants have completed their tours the pheromone level is updated by

$$\tau_{ij} = (1-\rho)\tau_{ij} + \Delta\tau_{ij}^{k}$$

Where $\rho \in (0,1)$ represent pheromone evaporation coefficient. And $\Delta\tau_{ij}^{k}$ is the amount of pheromone deposited on (i,j) by best ant k and is given by

$$\Delta\tau_{ij}^{k} = Q/L_k$$

Where Q is constant and L_k is length of path travelled by k^{th} ant.

Genetic Algorithm: The genetic algorithm is a well-known technique for optimization, intelligent search and machine learning. For example, the GA provides feasible solution to the travelling salesman problem, job scheduling, etc. These problems bear a strong similarity in that the main objective of these problems is that of optimizing or selecting the best solution out of a number of possible solutions. Thus, GA has a reasonable motive of being employed in a path optimizing problem. In this an initial set of points (chromosome) called population is used to start the process. An objective function is used as the fitness function. The genetic algorithm has 3 main steps: natural selection, crossover, and mutation.

Natural Selection: At each generation or iteration the chromosomes are updated using their fitness value. If a chromosome has more fitness value than other chromosome than that chromosome win the competition and has the higher probability to appear in the next generation.

Crossover: It is defined as the exchange of bits between the parents (chromosome) to generate new offspring. A group of chromosomes usually undergoes crossover at each generation. It is controlled by P_c crossover rate. It is for diversity in the population.

Mutation: It is defined as the random change of one or more bits in each chromosome. It is controlled by P_m mutation rate.

Simulated Annealing: The SA algorithm is based on the principle of solid annealing where a highly heated molten metal is slowly cooled to obtain the lowest minimum energy. The SA algorithm for solving optimization problems uses the initial solution *i* and initial control temperature like parameter *t* by repeating the iteration of "generate new solution → calculate the difference of objective function → accept or discard" to the current solution, and gradually reduce the value of control parameter *t*. At the terminating point the current solution is the approximate to the optimal solution. The process is controlled by *Cooling Schedule,* which consist of initial control parameter *t*, change in parameter Δt, iteration number for each *t a*nd the termination condition.

Particle Swarm Optimization: PSO is based on the collective behavior of a colony of insects, such as ants, bees etc; a flock of birds; or a school of fish. The word *particle* represent, for example, a *bee* in a colony or a *fish* in a school. Each individual in a swarm behaves using its own intelligence and the group intelligence of the swarm. As such, if one particle discovers a good path the rest of the swarm will also be able to follow the good path instantly even if their location is far away in the swarm. The PSO firstly generates a random initial population of particles, each particle represents a potential solution of system, each particle is represented by three indexes: position, velocity, fitness. Initially each particle is assign a random velocity, in flight, it dynamically adjusts the velocity and position of particles through their own flight experience (personal best position), as well as their neighbour's (global best position). Thus, the whole group will fly to the search region with higher fitness through continuous learning and updating. This process is repeated until reach the maximum iterations or the predetermined minimum fitness. The PSO is therefore a fitness based and group-based optimization algorithm, whose advantages are simplicity, easy implementing, fast convergence and less parameters. The entire particle swarm performs a strong "convergence", and it may be easily got trapped in local minimum points, which makes the swarm lose diversity.

Tabu Search: Tabu Search is a local search algorithm used to solve optimization problem. It uses local search to iteratively move from one solution to another solution which is a neighbour of the present solution until some stopping criteria has met. It may stuck to local optima during the local search and to avoid this it uses a memory structure which is called the TL. There are two components in the TS algorithm, the TL and the aspiration criteria. TL consist of Tabus which are those solution which causes sticking in local optima or cycling in the search space so as to avoid those solutions. Tabus are the short-term memory that help the search to move away from previously visited search space.

Aspiration criteria: Tabus are sometimes so powerful that they prevent a solution even if there is no danger of being cycling with that solution. In that case algorithm must be given some devices tha cancel the tabus. These are called aspiration criteria. The most commonly used aspiration criteria is to allow a move if it result in a objective value better than the current solution even if it is in the TL.

3. COMPARISON OF META-HEURISTIC TECHNIQUES

Here comparison is made on various characteristics of different meta-heuristic techniques like parameters whose values should be initialized before starting the implementation, convergence i.e. how the algorithm get trapped in local optima, I&D Component I means Intensification i.e. the exploitation of search space and D means Diversification i.e. reaching the unexplored portion of the search space, CPU Time It is the running time of the implementation of the algorithms and In the last Path length is the length of the path find by these for a common problem. On observing various papers the values are calculated for path length and CPU time taken by these techniques for finding path of a travelling salesman problem containing 20 nodes for a tour.

MEASURES	ACO	GA	SA	PSO	Tabu Search
Parameters	• Pheromone evaporation rate.	• Crossover rate. • Mutation rate. • Population size.	• Annealing rate. • Initial temperature like parameter.	• Population size. • Velocity of each particle.	• TL length.
Convergence	Slow due to pheromone evaporation.	Rapid	It avoid trapping by assigning probability to deteriorating moves.	Rapid but less than GA	By using TL it avoid trapping in local search.
I & D Component	Pheromone update, probability of selecting next vertex	crossover, mutation, natural selection.	cooling schedule, solution Acceptance strategy	Local search, fitness	TL for neighbour selection, aspiration criteria.
CPU Time (s)	250	200	101	220	140
Path Length	300	200	99	250	97

REFERENCES

[1] Youmei Li and Zongben Xul."An Ant Colony Optimization Heuristic for solving Maximum Independent Set Problem".IEEE, 0-7695-1957-1,2003.
[2] Yang XIAO, Xuemei SONG and Zheng YAO. "Improved Ant colony optimization with particle swarm optimization operator solving continous optimization problems". IEEE, 978-1-4244-4507-3,2009.
[3] M.Clerc and J.Kennedy, "The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space", IEEE Trans. on Evolutionary Computation, Vol.6, 2002, pp.58-73.
[4] Eberhart, R. C. & Kennedy, J. (1995). A New Optimizer Using Particles Swarm Theory. Proceedings of the 6th Int'1 Symposium of Micro Machine and Human Science, pp. 39-43, 0-7803-2676-8, Nagoya, Oct. 4-6
[5] Yuan, Z. L.; Yang, L. L. & Liao, L. (2007). Chaotic Particle Swarm Optimization Algorithm for Traveling Salesman Problem. Proceedings of the IEEE International Conference on Automation and Logistics, pp.1121-1124, 978-1-4244-1531-1, Jinan, China, Aug. 18-21
[6] Stützle, T. & Hoos, H. H.(1997). The MAX-MIN Ant System and local search for the traveling salesman problem. In T.Bäck, Z.Michalewicz, & X.Yao (Eds.), *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97),* pp.309-314, Piscataway, NJ, IEEE Press
[7] Stützle, T. & Hoos, H. H. (2000). MAX-MIN Ant System. *Future Generation Computer Systems, Vol. 16, No. 8: pp. 889-914*
[8] Dorigo, M. & Gambardella, L. M. (1997). Ant colonies for the traveling salesman problem. *BioSystems, Vol. 43, No.2: pp. 73-81*
[9] Chia-Feng Juang and Chun-Ming Lu. "Ant Colony Optimization Incorporated with Fuzzy Q-Learning for Reinforcement Fuzzy Control". IEEE Transactions on systems,man, cybernetics Vol. 39,No. 3, 1083-4427, May 2009 .
[10] Youmei Li and Zongben Xul."An Ant Colony Optimization Heuristic for solving Maximum Independent Set Problem".IEEE, 0-7695-1957-1,2003.
[11] Yang XIAO, Xuemei SONG and Zheng YAO. "Improved Ant colony optimization with particle swarm optimization operator solving continous optimization problems". *IEEE,* 978-1-4244-4507-3,2009.
[12] Marco Dorigo, Mauro Birattari, and Thomas Stutzle "Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique," IEEE Computational Intelligence Magazine, 2006.
[13] Y.T. Kao, E. Zahara, I.W. Kao, A hybridized approach to data clustering, Expert Systems with Applications 34 (3) (2008) 1754–1762.
[14] D.N. Cao, J.C. Krzysztof, GAKREM: a novel hybrid clustering algorithm, Information Sciences 178 (2008) 4205–4227.

[15] K.R. Zalik, An efficient k-means clustering algorithm, Pattern Recognition Letters 29 (2008) 1385–1391.
[16] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering, Analytica Chimica Acta 509 (2) (2004) 187–195.
[17] M.K. Ng, J.C. Wong, Clustering categorical data sets using tabu search techniques, Pattern Recognition 35 (12) (2002) 2783–2790.
[18] C.S. Sung, H.W. Jin, A tabu-search-based heuristic for clustering, Pattern Recognition 33 (5) (2000) 849–858.
[19] T. Niknam, J. Olamaie, B. Amiri, A hybrid evolutionary algorithm based on ACO and SA for cluster analysis, Journal of Applied Science 8 (15) (2008) 2695–2702.
[20] T. Niknam, B. Bahmani Firouzi, M. Nayeripour, An efficient hybrid evolutionary algorithm for cluster analysis, World Applied Sciences Journal 4 (2) (2008) 300–307.