Mobile Robot Localization Using Multi-Objective Optimization

Sabita Mali

Department of Electronics and communication Engg., ITER,SOA University, Bhubaneswar, Orissa

***Abstract* - The Swarm Intelligence-based Reinforcement Learning (SWIRL) method is proposed in this paper to efficiently generate Artificial Neural Network(ANN) based solutions to various problems. An aritificial neural networks learning method for mobile robot localization, which combines the two popular swarm inspired methods in computational intelligence areas: Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) to train the ANN models. The most used artificial neural networks(ANNs) models is the well-known multi-layer perceptron (MLP). Training neural networks is a complex task for reinforcement learning methods .The training process of MLPs for pattern classification problems consists of two tasks , the first one is the selection of an appropriate architecture for the problem , and the second is the adjustment of the connection weights of the network. Recently artificial neural networks based methods are applied to robotic systems. An ANN was trained to estimate a robot's position relative to a particular local object and to correct the pose estimates from odometry using ultrasonic sensors.**

***Keywords:* ANN, ACO,PSO,MLP,SWIRL**

I. INTRODUCTION

Two well-known approaches among many successful bio-inspired swarm intelligence based computational paradigms are known as Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). The ACO algorithm is essentially a system that simulates the natural behavior of ants, including mechanisms of cooperation and adaptation. The involved agents are steered toward local and global optimization through a mechanism of feedback of simulated pheromones and pheromone intensity processing. Particle Swarm Optimization(PSO) is a biologically-inspired algorithm motivated by a social analogy, such as flocking, herding, and schooling behavior in animal populations. Both algorithms have been applied to solve problems in various areas, such as clustering, data mining, dynamic task allocation, and optimization. The basic idea of the SWIRL method is that ACO is used to optimize the topology structure of the ANNs, while PSO is used to adjust the ANN connection weights within a given topology structure. This is designed to split the problem such that ACO and PSO can both operate in the environment they are most suited for. ACO is ideally applied to finding paths through graphs. One can treat the ANN's neurons as vertices and its connections as directed edges, thereby transforming the topology design into a graph problem. PSO is best used to find the global maximum or minimum in a real-valued search space. Considering each connection weight plus one associated fitness score as orthogonal dimensions in a hyperspace, each possible weight configuration is merely a point in that hyperspace. Finding the optimal weights is thus reduced to finding the global maximum of the fitness function in that hyperspace.

II. OVERVIEW OF NEURAL NETWORK IN ROBOTICS

An Artificial Neural Network (ANN) is an information processing paradigm which is composed of a large number of highly interconnected processing elements (neurons) and the weighted connections between them which is shown in fig 1. Learning in biological systems involve adjustments to the synaptic connections that exist between the neurons.

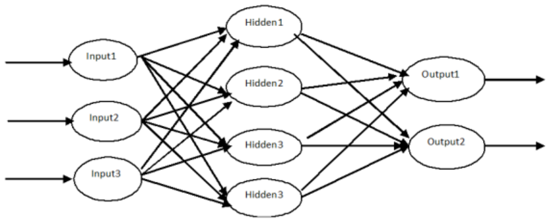


Fig. 1 Diagram of an Artificial Neural Network

An individual neuron cannot accomplish much, but the cumulative effect of many neurons connected together is effectively unlimited in complexity. A multilayer perceptron is a feed forward neural network with one or more hidden layers . The network consists of an input layer of source neurons ,at least one middle or hidden layer of computational neurons, and an output layer of computational neurons. A hidden layer "hides" it's desired output .Neurons in the hidden layer can not be observed through the input/output behavior of the network

We report on the objective of the motion planning problem well known in robotics. Given an object with an initial location and orientation, a goal location and orientation, and a set of obstacles located in workspace, the problem is to find a continuous path from the initial position to the goal position, which avoids collisions with obstacles along the way. In other words, the motion planning problem is divided into two sub-problems, called **'Findspace'** and **'Findpath'** problem. The findspace problem is used in construction the configuration space of a given object and some obstacles. The findpath problem is in determining a collision-free path from a given start location to a goal location for a robot. Various methods for representing the configuration space have been proposed to solve the findpath problem. The major difficulties in the configuration space approach are: expensive computation is required to create the configuration space from the robot shape and the obstacles and the number of searching steps increases exponentially with the number of nodes. Thus, there is a motivation to investigate the use of parallel algorithms for solving these problems, which has the potential for much increased speed of calculations. A neural network is a massive system of parallel distributed processing elements connected in a graph topology. Several researchers have tried to use neural networks techniques for solving the find path problem .

In this paper we introduce a neural networks-based approach for planning collision-free paths among known stationary obstacles in structured environment for a robot with translational and rotational motion. Our approach basically consists of two neural networks to solve the findspace and findpath problems respectively. The first neural network is a modified principal component analysis network, which is used to determine the "free space" from ultrasound range finder data. Moving robotics modeled as a two-dimensional object in this workspace . The second one is a multilayer perceptron , which is used to find a safe direction for the next robot step on the collision-free path in the workspace from start configuration to a goal configuration while avoiding the obstacles.

Essentially, neural network deal with cognitive tasks such as learning, adaptation ,generalization and optimization. To solve these problems fuzzy logic and neural networks are used.. The processing of imprecise or noisy data by the neural networks is more efficient than classical techniques because neural networks are highly tolerant to noises. Learning in the neural network can be supervised or unsupervised. Supervised learning uses classified pattern information, while unsupervised learning uses only minimum information without pre-classification. Unsupervised learning algorithms offer less computational complexity and less accuracy than supervised learning algorithms. Then former learn rapidly, often on a single pass of noisy data. A mathematical expression of a widely accepted approximation of the Hebbian learning rule is

$$W_{ij}(t+1) = W_{ij}(t) + \eta x_i(t) y_i(t)$$

where x_i and y_j are the output values of neurons i and j, respectively, which are connected by the synapse w_{ij} and η is the learning rate (note that x_i is the input to the synapse).

2.1 The basic motion planning problem and algorithm

Let A be a rigid object, a robot, moving in a workspace W, represented as a subspace of R^N, with N=2 or 3. Let O_1 ,...,O_m be fixed rigid objects distributed in W called obstacles. Assume that both the geometry and the location of A, O_1 ,...,O_m, in W is known. The problem is: Given an initial position and orientation of A in W, generate a path specifying a contiguous sequence of positions and orientations of A avoiding collision with o_i's, starting at the initial position and orientation, and terminating at the goal position and orientation. Report the failure if no such path exists.

In general, we consider the case when A is a two dimensional object that translates and rotates in $W = R^2$. A grid map will represent the environment. The grid map is an M x N matrix with each element representing the status $s_{i,j}$ of an individual grid $s_{i,j} = k$ if its interior intersects with the obstacle region and $s_{i,j} = 0$, if its interior does not intersect the obstacle region. A configuration of an arbitrary object is a specification of the position of every point in this object relative to a fixed reference frame. In addition, let F_A and F_W be Cartesian frame embedded in A and W, respectively. Therefore, a configuration of A is a specification of the position (x,y) and orientation θ of FA with respect to FW . The configuration space of A is the space of all the configurations of A. Let the resolution in x-axis, y-axis and orientation is M, N, and K respectively. A rectangle $r_{i,j,k}$ is model of the object A located by (x_i, y_j, Θ_k) and it represents the region $\left[x_i - \frac{w_x}{2}, x_i + \frac{w_x}{2}\right]$, $\left[y_j - \frac{w_y}{2}, y_j + \frac{w_y}{2}\right]$, $\left[\Theta_k - \frac{\Delta\Theta}{2}, \Theta_k + \frac{\Delta\Theta}{2}\right]$where w_x is the width, w_y is the height and $\Delta\Theta = \Pi/k$

2.2 Motion planning algorithm

1. Specify the object, environment information and the start and goal configurations.
2. Set the current object orientation equal to the goal orientation.
3. Activate range finder via control unit to determine the local part of the map of the workspace.
4. Initialize the first neural network, which will use the measured data from range finder. The neural network is iterated until the weights and the outputs converged to the returned one free space segment.
5. Activate the second neural network. It returns the direction Θ_k of next robot motion step.
6. Generate the robot motion path in the direction Θ_k and go to the step 3.

2.3 The findspace problem using neural network

we use the sensor data from the environment and the robot has in any position in workspace information about its distances to the all objects in this workspace. We use this information in first neural network that learns these situations and in any position gives the free segment of space for safe path as output. The neural network using for the findspace problem is principal component analysis network (PCA). Principal component analysis networks combine unsupervised and supervised learning in the same topology (see Fig. 2).

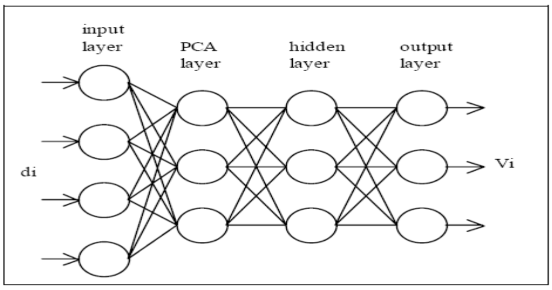


Fig. 2. PCA neural network topology

This neural network uses as inputs the data measured by the range finder. The output is free segment of the robot workspace. Principal component analysis is an unsupervised linear procedure that finds a set of uncorrelated features from the input. A feed-forward network is used to perform the nonlinear classification from these components. PCA is a data reduction method, which condenses the input data down to a few principal components.

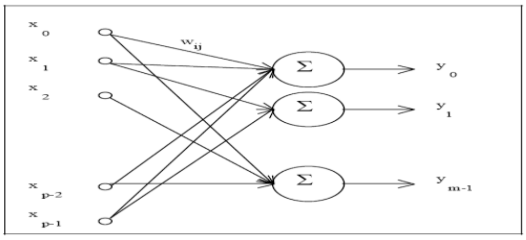


Fig. 3. Detail in PCA network

The network has four layers - input, PCA, hidden and output layer. The learning is realized in two phases. In the first place an unsupervised linear procedure gets a set of uncorrelated features from the inputs and selects a few principal components. These components in hidden layer feed-forward supervised part gives the output. The PCA updated the synapse weights $W_{i,j}$ (see Fig. 3) to a small random number and learning parameter η to a positive small number.

The neural network uses the normalized data from ultrasound range finder as inputs. There are distances di, ranging from 20 to 250 cm, to the all objects in the front robot space from 0° to 180°. From input layer of the network we obtained information about free segments V_i. Each of the output neurons "represent" particular segment of the workspace as is depicted on the Fig. 4.

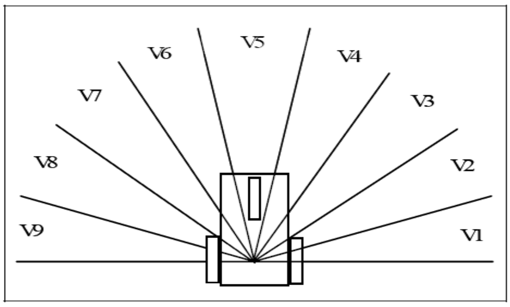


Fig. 4 The workspace segments

2.3 Solving the find path problem

For the solving of the find path problem we use neural network, a multilayer perceptron (MLP). Multilayer perceptrons are layered feed-forward networks typically trained with static back propagation .Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key Dis advantages are that they train slowly ,and require lots of training data. Aim of this network is determining of the robot azimuth Θ_k for the next robot motion step from the output of first network and from the goal coordinates. The topology of this network is depicted on the Fig. 5. The network contains a three layer – input, hidden and output. This is a layered feed-forward network typically trained with static back-propagation. Here are updating the synapse weights between neurons and threshold. The process is similar to above described for the second phase of the learning process for find space problem . On the input of this neural network in our case we give the known free space segment V_i as the output of the first neural network and the goal segments s_i in which the coordinates of the robot goal position should be situated. The choice of the goal segments is the same as is depicted on the Fig. 4. From output layer of this neural network we obtained information o_j about robot motion direction (azimuth) in the next step. This information is given to the control unit. It manages this information into the robot command for the robot motor control.

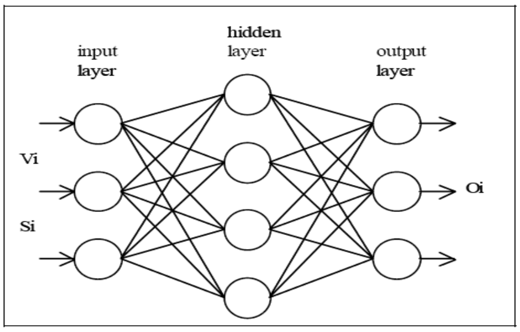


Fig. 5. Topology of MLP network

III. SWARM INTELLIGENCE

3.1 The SWIRL System

The objective of the SWIRL system is to achieve an optimal model for the problem of interest through re-inforcement learning, which is analogous to a school system. Therefore, the SWIRL system can be modeled as a school system, where the ACO module, the PSO modules, and the untrained ANNs taking on the roles of administrator, teachers, and students, respectively. In a school system , students learn from teachers, teachers train students, and administrators allocate resources to teachers. In the SWIRL system , the ACO algorithm (the administrator) allocates training iterations to the PSO algorithms (teachers). The PSO algorithms (teachers) then run for their allotted iterations to train their ANNs (students). The global best score for all the

ANNs trained by a particular PSO instance is then used by the ACO algorithm to allocate the next set of training iterations . Fig. 6 gives a high-level overview of this SWIRL system.

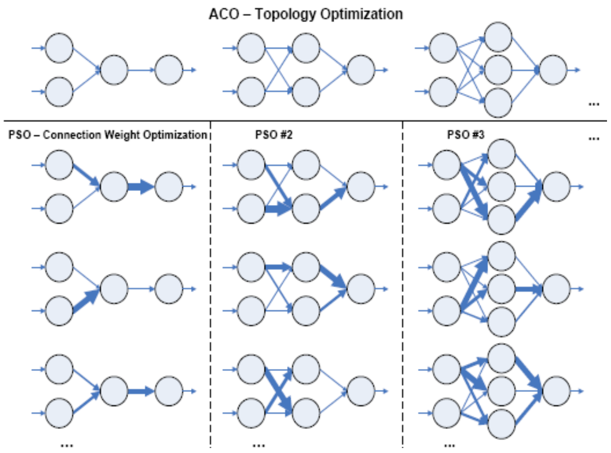


Fig-6 SWIRL Overview

This composition of the ACO and PSO algorithms is designed to best leverage their inherent strengths. The ACO algorithm excels at finding the optimum path through a graph. The optimization of an ANN topology can be treated as a graph problem where the neurons are vertices and the connections are directed edges. The value of a path is the best possible performance of the ANN whose topology is defined by that path. At first, this value is completely unknown, but as training progresses it can be estimated with gradually increasing accuracy. This makes the graph extremely dynamic, with edge values changing every iteration.

3.2 PSO(Particle Swarm Optimization:

Particle Swarm Optimization (PSO) is applied to adjust ANN connection weights within a given topology structure.PSO is best suited to find the global maximum or minimum in a real valued search space. The PSO algorithm is essentially a stochastic search for the maximum or minimum value of a function in a real valued hyperspace. The goal is to find the global maximum of the reinforcement (i.e fitness) function ,so PSO algorithm can be used to optimize the ANN connection weights. PSO is tested in noisy and continuously changing environments, where the function values are imprecise and the global minimize moves within the search space. PSO is used to solve norm errors-in-variables problems, determining the best model for fitting sets of data.

*3.2.1 PSO-based Weight Adjustment***:**

A configuration for an ANN with n connections can be considered as a point in an n+1 dimensional space ,where the extra dimension is for the reinforcement score .After every round of testing , the teacher updates the connection weights of the student ANNs according to the following equations:

$$V_{r+1} = c_{inr} .r_1 .v_r + c_{cgn} .r_2.(x_{pb} - x_r) + c_{scl} .r_3.(x_{gb} - x_r)$$

$$X_{r+1} = x_r + V_{r+1}$$

where the position and velocity vectors are denoted by x and v, respectively .The big dot symbol is for Hadamard multiplication. The r_i represents vectors where each element is a new sample from the unit-interval uniform random variable. Personal best, X_{pb}, is the point in the solution-space where that particular student received it's highest score so far. Global best, X_{gb}, is the point with the highest score achieved by any student of this PSO teacher. The three constants, c_{inr}, c_{cgn}, c_{scl} allow the adjustment of the relative weighting for the inertial, cognitive and social components of the velocity, respectively.

3.3 ACO(Ant Colony Optimization):

ACO algorithm can be used to allocate training iterations among a set of candidate network topologies. ACO is essentially a system that simulates the nature behavior of ants, including mechanisms of co-operation and adaptation. The involved agents are steered towards local and global optimization through a mechanism of feedback of simulated pheromones and pheromone intensity processing. It is based on the following ideas. First, each path followed by an ant is associated with a candidate solution for the target problem. Third, when an ant has to choose between two or more paths with a larger amount of pheromone are more attractive to the ant. After some iterations , the ants will converge to a shorter path which is expected to be the optimum or near optimum solution solution for the target problem.

3.3.1ACO-based Topology Optimization **:**

ACO algorithm can be used to allocate training iterations among a set of candidate network topologies. The corresponding desirability in the ACO algorithm can be defined as:

$$d(i) = \frac{1}{h+1}$$

where h is the number of hidden nodes in ANN i. ANN i represents ith topology of a set of ANNs with different number of hidden nodes.τ (pheromone concentration) is initialized to 0.1, τ is then updated according to:

$$\tau(i, t+1) = \rho.\tau(i,t) + n_a(i).g(i)/g_{sum}$$

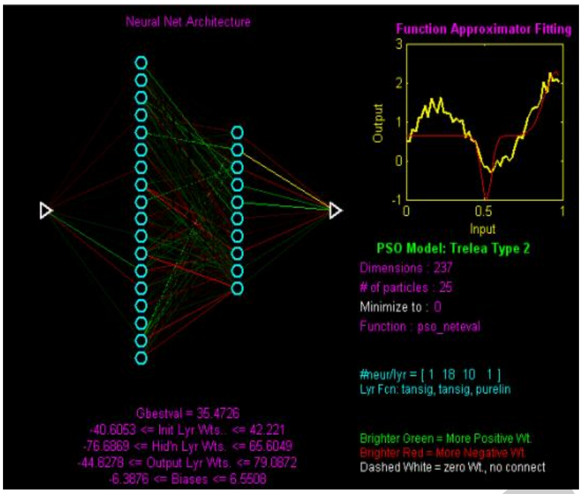
where ρ is the rate of evaporation , n_a is the number of ants returning from the neural network i, g(i) is the global best for i, and sum g_{sum} is the sum of all the current global bests. Each ant represents one training iteration for the PSO teacher. During each major iteration (i.e. ACO step), the ants go out into the topology space. The probability ant *k* goes to neural network *i* is given by:

$$p(i) = \frac{[d(i)]^\beta [\tau(i,t)]^\alpha}{\sum_{j=1}^{i}\{[d(i)^\beta][\tau(i,t)]^\alpha\}}$$

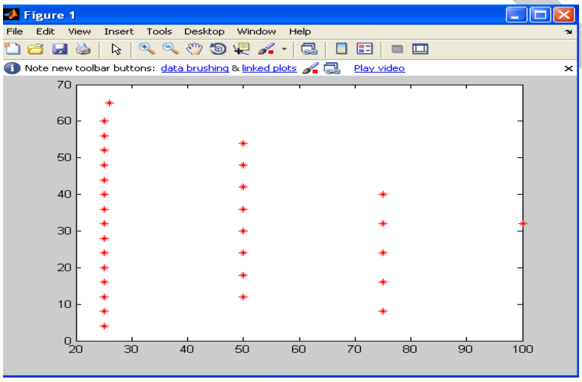
where α and β are constant factors that control the relative influence of pheromones and desirability, respectively.

IV. EXPERIMENTAL RESULTS:

We have implemented SWIRL algorithm using PSO and ACO, We used PSO algorithm to adjust the connection weights of the selected topology and The ACO algorithm is used to select the neural network topology .The experimental results using PSO is given below.



In ACO we have taken 30 number of input ants which are targeted at positions [25 4;25 8;25 12;25 16;25 20;25 24;25 28;25 32;25 36;25 40;25 44;25 48;25 52; 25 56;25 60;26 65;50 12;50 18;50 24;50 30;50 36;50 42;50 48;50 54;75 8;75 16;75 24;75 32;75 40;100 32].The figure given below shows ants positions.



and the resource for the network by using ACO is

Net_Res =

-0.4326	1.0668	0.8156	-2.1707
-1.6656	0.0593	0.7119	-0.0592
0.1253	-0.0956	1.2902	-1.0106
0.2877	-0.8323	0.6686	0.6145
-1.1465	0.2944	1.1908	0.5077
1.1909	-1.3362	-1.2025	1.6924
1.1892	0.7143	-0.0198	0.5913
-0.0376	1.6236	-0.1567	-0.6436
0.3273	-0.6918	-1.6041	0.3803
0.1746	0.8580	0.2573	-1.0091
-0.1867	1.2540	-1.0565	-0.0195
0.7258	-1.5937	1.4151	-0.0482
-0.5883	-1.4410	-0.8051	0.0000
2.1832	0.5711	0.5287	-0.3179
-0.1364	-0.3999	0.2193	1.0950
0.1139	0.6900	-0.9219	-1.8740

V. CONCLUSION

In this paper SWIRL algorithm is proposed to achieve an optimal model for the problem of interest through reinforcement learning. The ACO algorithm is applied to select the neural network topology, while the PSO algorithm is utilized to adjust the connection weights of the selected topology. The paper presents our first results that we obtained making use of the proposed path planning algorithm working with the neural network and sensor data. The simulation examples of the generation of the collision free path for point robot and for two-dimensional robot show that designed strategy are acceptable for solution of this problem. We played the role of the supervisor to learn the robot to make it's way intelligently toward its target and to avoid obstacles.

REFERENCES

[1]. Bekey, G. A. & Goldberg, K.Y. (1993). *Neural Networks in Robotics*. Kluwer Academic Publishers, ISBN 0-7923-9268-X, Boston

[2]. Chohra, A.; Sif, F. & Talaoubrid, S. (1995). Neural Navigation Approach of an Autonomous Mobile Robot in a Partially Structured Environment.

[3]. J.Kennedy and Eberhart R.C., *"Particle swarm optimization"*, Proc. IEEE International Conference on Neural Networks, IV, Piscataway, NJ, pp. 1942- 1948,1995.

[4]. N.M. Kwok et al., *"Evolutionary computing based mobile robot localization"*, Engineering Applications of Artificial Intelligence 19 (2006) 857–868

[5]. P. Tournassoud, *"A strategy for obstacle avoidance and its application to multi-robot systems"*, Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 1986, pp. 1224-1229.

[6]. J. Latombe, —*Robot Motion Planning"*, Kluwer Academic.

[7]. Kurt Derr and Milos Manic, *Multi-Robot, Multi-Target Particle Swarm Optimization Search in Noisy Wireless Environments"*, Catania, Italy, May 21-23, 2009.

[8]. Racz. J. & Dubrawski, A. (1994). Mobile Robot Localization with an Artificial Neural Network, International Workshop on Intelligent Robotic Systems IRS '94, Grenoble, France.