Multi-Objective Evolutionary Algorithms with Immunity for SLAM

Meiyi Li

College of Information Engineering, Xiangtan University.
Xiangtan China, 4110005
meiy_li@yahoo.com.cn

Abstract. The simultaneous localization and mapping problem with evolutionary computations is translated to a multi-objective optimization problem since it possesses of characters of multi-objective at a certain extent, and in order to efficiently solve the simultaneous localization and mapping problem, a local searcher with immunity is constructed. The local searcher employs domain knowledge, which is named as key point grid which is developed in the paper. The experiment results of a real mobile robot indicate that the computational expensiveness of designed algorithms is less than evolutionary algorithms of single-objection for simultaneous localization and map-ping and accuracy of obtained maps are better.

1 Introduction

SLAM (Simultaneous Localization And Mapping or Concurrent Mapping and Localization) is to acquire a map of an unknown environment with a moving robot, while simultaneously localizing the robot relative to this map[1-2]. The SLAM problem addresses situations where the robot lacks a global positioning sensor, and instead has to rely on sensors of incremental ego-motion for robot position estimation (e.g., odometry, inertial navigation. etc.). Such sensors accumulate error over time, making the problem of acquiring an accurate map into a challenging one. This is a hard problem because noisy sensor data must be simultaneously used for both mapping and localization. Within mobile robotics, the SLAM problem is often referred to as one of the most challenging ones[3-4].

Traditional algorithms are based on extended Kalman filters (EKFs) [5-6]. However, several problems arise when applying the Extended Kalman Filter approach. Especially, this method is not able to deal with uncertainty as follows: the combinatorial labeling problem of data association (e.g., landmark identification, feature recognition and place recognition, etc.) in which a correspondence must be found between sensor measurements and the features already represented in the map.

In order to overcome the difficulty about the data association problem, Reference [7] and [8] employed evolutionary computations to solve SLAM problem. Their investigations indicate that evolutionary computations is a hopeful approach for NP-hard SLAM problem.

© A. Gelbukh, C.A. Reyes-García. (Eds.) *Received 18/06/06*
Advances in Artificial Intelligence. *Accepted 03/10/06*
Research in Computing Science 26, 2006, pp. 27-36 *Final version 09/10/06*

In SLAM occupancy grid present by Moravec and Elfes[9] often is employed, where a map is consisted of grids or cells. A occupancy grid is described as grid[i][j], and every grid[i][j] has a probability or belief occ[i][j] which is occupied and a probability or belief emp[i][j] which is free. The calculations of occ[i][j] and emp[i][j] depended on data from range-finder. Usually, since the reliable degree of data from range-finder is relevant to the distance. The reliable degree of data from range-finder is projected to occ[i][j] and emp[i][j] by means of a sensor fusion approach like theories of evidence. In this paper Dempster-Shafer theory of evidence is employed and works as a sensor fusion.

SLAM possesses of characters of multi-objective at a certain extent since several formulations are combined with weights and the fitness is defined. From Reference [8], we can get that the target of evolutionary computations for SLAM is to solve multi-objective problem(MOP) as follow.

The reliable degree of data from range-finder is projected to occ[i][j] and emp[i][j] by means of Dempster-Shafer theory of evidence. So, the overall consistency of the sensory information contained in the grid-map, which is one target to be optimal, is described as follows:

$$f_1 = \sum \min(1\text{-}occ[i][j],\ 1\text{-}emp[i][j]). \tag{1}$$

The other objectives are to reward the algorithm for producing smaller, more compact maps:

$$f_2 = \sum \delta_1(i,j), \quad f_3 = \sum \delta_2(i,j), \tag{2}$$

where

$$\delta_1(i,j) = \begin{cases} 1, & if \ \ occ[i][j] > 0.5 \\ 0, & other \end{cases}, \quad \delta_2(i,j) = \begin{cases} 1, & if \ \ emp[i][j] > 0.5 \\ 0, & other \end{cases} \tag{3}$$

Objectives or of combining them together as a weighted, linear sum, is the method used in Reference [7] and [8]. This method, as is so often done, will lead bad solutions to since choice of weights is crucial and is difficult to be determined. Even a large change in the weights of a weighted sum scalarization would result in finding a bad solution.

The weighted sum is only one possible method in this family of scalarizing methods and has some serious drawbacks. One of others is to consider alternately one objective function then another; and there are various ways this could be organized. Another approach is to use some form of relative ranking of solutions in terms of Pareto dominance. The latter is the most favoured approach in the EA community because it naturally suits population-based algorithms and avoids the necessity of specifying weights, normalizing objectives, and setting reference points.

Mikkel and Jensen[10] used the flowtimes of the individual jobs Fi to build helper-objectives, and optimize the primary objective. The helper-objective simultaneously will be equivalent to simply optimizing the primary objective. Knowles and Corne[11] defined additional objectives with arbitrary sub-tours for travelling salesman problem, and local optima in single-objective optimization problems can be removed. These results enlighten that multi-objective methods can guide the search and deal with sources of difficulty in single-objective optimization. Their investigations also show that the performance of algorithm depends on whether dominant knowledge in problems is used efficiently.

2 Key Point Grid and Its Detection

Sensors like sonar and laser scan, grabbing range-finder data, provide environment information around the robot. From the information the local map can be constructed and a local path for robot can be planned. If there is some structure information like lines in environments, the structure information is used to building a global map more efficiently and exactly. Considering a situation showing as Fig. 1 in which range-finder data for a laser scanner are simulated, it is can seen that a large gap of range-finder data occurred at some point and on left side of which long scan radial line, and on the right side short scan radial line occurred. Furthermore, all long scan radial line forms a continuous sector. We call the occupancy grid that is located by gap point as key-point grid.

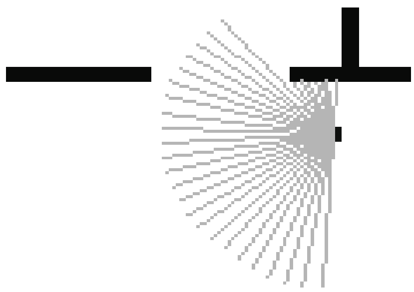

Figure1. Simulations of range- finder data at a key point grid

Fig.2 shows a key point grid in a running for a real robot in our experiment room. Local occupancy/empty grids were calculated by grid range-finder data, so the map in Fig 1 is local where the cells with black are the edges of obstacles calculated from range-finder data gotten from the robot. The ones with white are free of obstacles and gray cells are unknown. It is remarked that these colors in other figures are the same meaning in this paper. On the right side in Fig 1, a large gap in range-finder data occurred, since the obstacle goes to the end. Although range-finder data are not continuous, there is no key point grid since varying scopes of range data are not enough.

In the opinion of the geometry, the concept of key point grid comes form the convex point in polygon obstacle. When the robot samples near a convex point in polygon obstacle, largely discontinuous gap in range-finder data will happen. Because of noise, largely discontinuous gap that is founded haphazard in range-finder data doses not always mean that a key point grid occurs. Therefore, a method for eliminates false key point grid is needed. In this paper if a grid is a candidate key point grid, the grid will be scanning for many times and a key point grid will be determined by means of Dempster-Shafer theory of evidence.

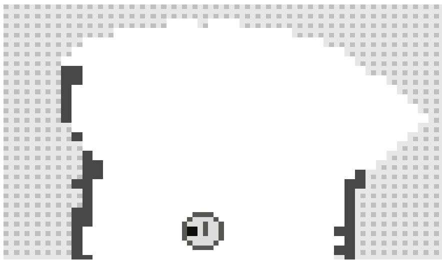

Figure 2. A possible key point grid is found by a real robot

— A robot, — Obstacles, — Unknown area, Space —Obstacle-free area

In the implementation of determining key point grid, when a large gap in range-finder data occurs, the grid which the gap belongs to is consider as a candidate key point grid and the robot samples and takes range-finder data for 3-4 times while it move in a curved path. In Fig. 3 it is showed that a real robot sampled at some key point grid in moving along a curved path. If there is really a key point grid, the grid is marked as K_i=[x_i^b, y_i^b], where x_i^b, y_i^b are coordinates of the grid assigned by the map under global coordinate system. Since K_i depends on the robot's own measurements of its trajectory, these measurements will be corrupted by noise. Hence, real coordinates of Ki can be described as follows:

$$x_i = x_i^b + \Delta x_i \,, \; y_i = y_i^b + \Delta y_i \,, \tag{4}$$

where Δx_i, Δy_i =0, ±1.

The following algorithm provides a method of finding a key point from range-finder data.

(1) Let range-finder data is a set of $\{(r_i, \varphi_i)\}$ in the local coordinate system. (r_i, φ_i) is coordinates under the polar coordinates. In general the polar coordinates (r_i, φ_i) can be translated to the Cartesian coordinates $p_i(x_i, y_i)$ by a simple coordinate transformation:

$$\begin{cases} x_i = r_r \cos\varphi_i \\ y_i = r_r \sin\varphi_i \end{cases}, \tag{5}$$

(2) Let a set A1 ={ p_1 }. The set A_1 is called as a successive section. Let i=2, j=1;

(3) If the point pi in range-finder data satisfies the condition as follows:

$$\frac{\| p_i - p_{i-1} \|}{\min((\| p_i \|, \| p_{i-1} \|)} < \delta_{\text{Key}} \tag{6}$$

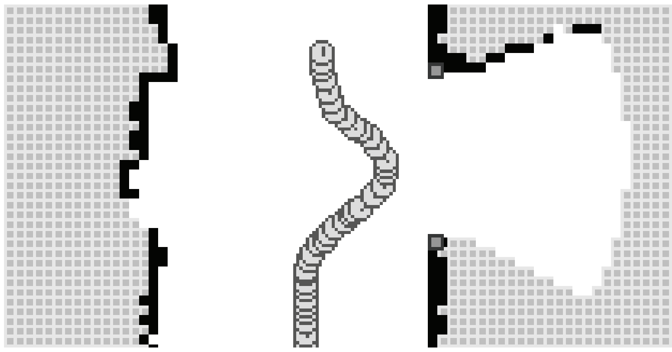

Figure 3. Trajectories and two key point grids.
— Trajectories, — Key point grids, — Obstacles,
— Unknown area, Space — Obstacle-free area

p_i is inserted into successive section A_j. Else, a new successive section A_{j+1}, $j=j+1$, is created, and let $i=i+1$;

(4) The step (3) is repeated until range-finder data is empty.

(5) If the size of successive section A_j is less than 10, this successive section A_j will be deleted.

(6) Let all successive sections can be described as follows:

$$Q_i = \{(x_{i,1}, y_{i,1}), \Lambda\ , (x_{i,N}, y_{i,N})\}., i=1,2,...,N_q. \qquad (7)$$

(i) If $N_q=1$ there is not a c. The algorithm goes to end;

(ii) If $N_q=2$, the last point $p_{1,N}(x_{1,N}, y_{1,N})$ in Q_1 is a key point. The algorithm goes to end;

(iii) If $N_q>2$, successive section pairs $\{Q_i, Q_{i+2}\}$($i=1,2,...,N_q-2$) are constructed, a distance of $\{Q_i, Q_{i+2}\}$ is defined as follows:

$$d(Q_i, Q_{i+2})=\sqrt{(x_{i,N}-x_{i+2,1})^2+(y_{i,N}-y_{i+2,1})^2}\ . \qquad (8)$$

This distance is called as successive section pair distance.

(7) If $d(Q_i, Q_{i+2})> d_{\text{Key}}$, $i=1,2,...,N_q-2$, points $p_{i,N}(x_{i,N}, y_{i,N})$ and $(x_{i+2,N}, y_{i+2,N})$ are considered as key points.

In the implementation of the algorithm in this paper key point grids were stored in a database, so computations of extracting key point grids from range-finder data were carried out only once.

3 Multi-Objective Evolutionary Algorithms with Immunity for SLAM

3.1 Multi-Objective Algorithms (MOA) with Immunity

Multi-objective algorithms with immunity for SLAM proposed in this paper are described as follows.

(1) Let G and P' empty
(2) Generate an initial population P
(3) Calculate the fitness of each individual
(4) **repeat**
  (5) Let G and P' empty
  (6) **for** j =1 to #recombinations **do**
    (7) Select a set X_{par} in $P \cup G$ to be parents with ranking-based selection[12]
    (8) Perform recombination on X_{par} with probability p_R to generate x_{off1}
    (9) Perform vaccination on x_{off1} with probability p_I
    (10) Perform immune selection to generate x_{off} from x_{off1}
    (11) Replace P' with $P' \cup \{x_{off}\}$
  (12) **endfor**
  (13) **for** j =1 to #mutations **do**
    (14) Select $x \in P$ for mutation
    (15) Perform Mutate with probability p_M to generate x'
    (16) Perform vaccination on x' with probability p_I
    (17) Perform immune selection to generate x"
    (18) Replace P' with $P' \cup \{x''\}$
  (19) **endfor**
  (20) Replace P with Select from($P \cup P'$) using ranking-based selection[12]
  (21) **if** P has converged **then** replace P with an initial population P
(22) **until** termination condition is TRUE

In the implementation of the algorithm, both termination conditions are used.

3.2 Chromosome Encoding, Recombination and Mutation

SLAM is treated as a continuous global optimization problem where the search is carried out in the space of possible robot trajectories. A trajectory can be defined as a vector $[T_1, T_2, ..., T_N]$, where T_j=[d_j, θ_j], d_j and a_j are the relative distance and rotation that are traveled by the robot in one small step j, and there are N steps in total.

The robot's own measurements of its trajectory are used to generate candidate solutions by applying different correction factors, which are described as follows, to the measured values of d_j and a_j.

$$d'_j = d_j + \Delta d_j,\ \theta'_j = \theta_j + \Delta\theta_j. \quad (9)$$

Each chromosome is encoded as a string of floating point numbers $[X_1, X_2, \ldots, X_N, K_1, K_2, \ldots, K_{Nk}]$ corresponding to the correction factors and the key point grids, where $X_i = [\Delta d_j, \Delta\theta_j]$, and $-d_{max} \le \Delta d_k \le d_{max}$, $-\theta_{max} \le \Delta\theta_k \le +\theta_{max}$, $k = 1, 2, \ldots, N$. d_{max} and θ_{max} are real positive integers.

Pairs of selected strings are then combined by recombination. Recombination is carried out with probability p_c. Mutation is carried out by picking single value within the strings with very low probability p_m and replacing those values with randomly generated values, as upon initialization.

3.3 Vaccination Operator

A local exploration process, named as a vaccination operator, is constructed by means of the feature of key point grid. A key point grid K_i is selected uniformly, and a trajectory T_j is found such that at T_j the robot can detect the key point grid K_i by means of laser scanner. If the point where a large gap occurs in range-finder data at T_j does not belong to K_i, the correction factors Δd_j applied to the distance will be adjusted. This proccess will be performed with probability p_l.

3.4 Immune Selection Operator

If the correction factors $\Delta\theta_j$ and $\Delta\theta_{j+1}$ applied to the angle measurements at T_j and T_{j+1} are adjusted in the vaccination operator, an immune selection operator, which will be described as below, will be performed.

The change of the correction factors $\Delta\theta_j$ and $\Delta\theta_{j+1}$ will lead to improvements of the consistency and compactness of the map. Measure of improvements will be used to evaluate the performance of a vaccination operator. In detail, either of both trajectories T_j and T_{j+1} is evaluated by constructing a local occupancy map using the recorded range-finder data of the robot along the path, and a value f_1 is calculated according to formation (1) within the local occupancy map. If T_j and T_{j+1} are replaced with T'_j and T'_{j+1}, as the same as above calculations, a value f'_1 is calculated. If $f'_1 < f_1$, T_j and T_{j+1} are replaced with T'_j and T'_{j+1} in the individual in which the vaccination operator has been performed, else, the operator of replacement will be done[13] with probability $\exp(-(f'_1 - f_1)/\ln(k+1))$, where k is the evolutionary generation in evolutionary process.

4 Implementation of the Algorithm and Experiment Results

The algorithm was tested using data recorded by an AmigoBOT mobile robot produced by ActivMedia Robotics, LLC with the addition of a SICK laser scanner showed at the Intelligence Control Lab of the Central South University in China. The odometer trace was divided into segments of about from 1 to 2 meters in length and was 0.2 meters at every key point grid. For the environment of Fig. 4, there were 26 segments corresponding to the about 30 meters traveled by the robot. Because movable scope for robot is not large enough, all range-finder data are truncated such that the lengths of range-finder data are less than 3 meters, which means that if r_i>3 in range-finder data (r_i,φ_i), r_i=3.

In our experiments D_{max}=0.32, δ_{Key} =1, d_{max}=20cm, θ_{max}=10, β_0=1meters, integer $d_k \in$[-20, 20], integer $\theta_k \in$[-10, 10]. In implementation of the algorithm the population size is 50, p_I=0.3, p_C =0.9 and p_M=0.08.

The algorithms proposed in this paper and in Reference [7] and [8] run 10 times for the same test case (environment, trajectories, and range-finder data) in order to prove that the algorithm proposed in this paper outperforms other approaches. Running results of algorithms are list in Table 1, and a grid-map gained through the algorithm in this paper is shown in Fig. 4, where the terminate condition is that running generation is 300. From Table 1 it can be seen that the convergence rate is higher than the algorithms in Reference [7]. Similar experiments were conducted for the traditional multi-objective algorithm without vaccination operator and immune selection operator, the results are shown in table 2, where the terminate condition is if the best fitness values in the population are not improved in N_ς =20 generations, the algorithm will go to the end, which is often the convergence critical. The table holds one column for every algorithm. The first column reports the average total FS of the traditional algorithm, while the remaining columns report the performance for the multi-objective algorithm. The experiments reveal that the multi-objective algorithm with immunity performs better than the traditional algorithm without immunity.

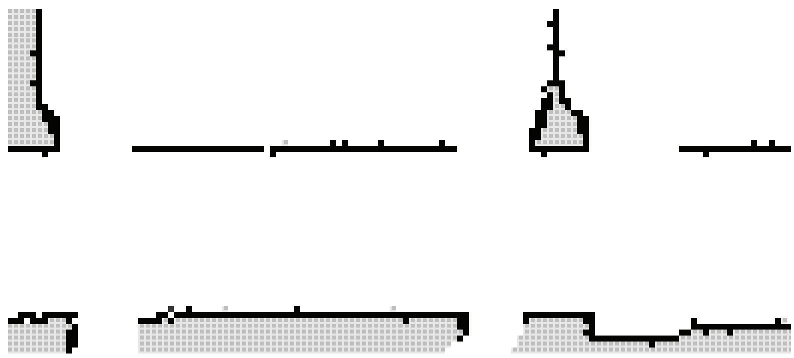

Figure 4. A grid-map was gotten with proposed algorithm when the robot ran in our experiment Lab.

In order to compare the accuracy of maps achieved by several algorithms, manual build a map using stored data captured by the mobile robot.

Table 1. Comparisons of running results for two algorithms

Algorithms	Mean value of best fitness values	Standard deviation of best fitness values
MOA with immunity	510.4	22.7
MOA without immunity	697.3	30.1
Algorithm in Ref.[7]	631.8	37.5
Algorithm in Ref.[8]	582.0	29.8

Table 2. Comparisons of running results for two algorithms

Algorithms	Mean number of fitness function evaluations
MOA with immunity	2074.8
MOA without immunity	3893.6

In the implementation of the algorithm in this paper, computations of extracting line segments and key point grids from range-finder data are carried out only once, hence in a vaccination operator and an immune selection, major computation time is spent on the computation of values f_1 in a local grid-map that was constructed by only both of trajectories. The average ratio of the computation time of values f_1 in a local grid-map to the one in a global grid-map is about $4/N$ (N is the total of trajectories). So, its computation time is more less than computation of the fitness value f in formulation (1). To sum up, the algorithm proposed in this paper can increase the convergence rate of SLAM based on evolutionary algorithms, and the larger the scope is for robot to travel the higher the convergence rate of our algorithm is, since the total of trajectories will increase.

5 Conclusions

(1) Multi-objective algorithms with immunity for SLAM have bee proposed, which are combined with feature of key point grids in range-finder data in order to increase the convergence rate of SLAM based on evolutionary algorithms.

(2) The feature of large gap in range-finder data at a convex vertex in polygonal obstacle is employed, and the feature of key point grids is extracted and used to construct a local search operator of key point grid with immunity.

(3) Experiments results showed that multi-objective evolutionary algorithms with immunity could improve optimization for SLAM in some cases.

Acknowledgements. This work was supported in part by the Crossover Item of XTU (No. 05IND04). Thanks to Prof. Zixing Cai & other fellows in Central South University for help in experiments.

References

1. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M Csorba. A solution to the simultaneous localization and map building (SLAM) problem. Transactions of Robotics and Automation, 2001, 17(3), pp. 229-241.
2. J. J. Leonard and H. F. Durrant-Whyte. Directed Sonar Sensing for Mobile Robot Navigation. Kluwer, 1992.
3. Thrun S. Robotic mapping: A survey. In: Lakemeyer G, Nebel B eds, Exploring Artificial Intelligence in the New Millenium. Kaufmann M, 2002, pp. 1-35
4. Wang L and Cai Z X. Progress of CML for mobile robots in unknown environments. Robot. 2004,26(4), pp. 380-384(in Chineses)
5. Smith RC and Cheeseman P. On the representation and estimation of spatial uncertainty. International Journal of Robotics Research, 1986, 5(4): 56-68
6. Smith R, Self M and Cheeseman P. Estimating uncertain spatial relationships in robotics. In: Wilfong G T and Cox I J, eds, Autonomous Robot Vehicles, Springer-Verlag, 1990:167-193
7. Duckett T. A genetic algorithm for simultaneous localization and mapping. In: IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003: 434-439
8. Meiyi Li, Zixing Cai. Immune Evolutionary Computations for Concurrent Mapping And Location. Fuzzy Sys-tems and Knowledge Discovery, Springer, 2005.8. Volume(3), pp.1308-1311,
9. Moravec H, Elfes A. High resolution maps from wide angle sonar. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'85), 1985, pp. 116-121
10. EE, Mikkel T. Jensen: Guiding Single-Objective Optimization Using Multi-objective Methods. EvoWork-shops 2003, pp. 268-279.
11. J. Knowles and D. Corne. Reducing local optima in single-objective problems by multi-objectivization. In First International Conference on Evolutionary Multi-criterion Optimization (EMO'01), Springer-Verlag, 2001, pp. 269–283.
12. David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
13. Jiao L C, Wang L. A novel genetic algorithm based on immunity. IEEE Trans. Syst., Man, Cybern. A. 2000, 30(5), pp. 552-561.