AUTOMATYKA • 2010 • Tom 14 • Zeszyt 3/1

Sławomir Jeżewski*, Maciej Łaski*, Robert Nowotniak*

Comparison of Algorithms for Simultaneous Localization and Mapping Problem for Mobile Robot

1. Introduction

Robotics is an engineering science and technology of robots considered broadly in many scientific papers nowadays. Scientific progress makes them more robust, functional and intelligent all the time. Recent years have witnessed successful applications of robots in medicine [1, 22] and industry [9, 16]. Moreover, robots can be used by people with disabilities or in environments dangerous for human health and life [20, 24].

This paper presents a comparison of selected algorithms for simultaneous localization and mapping (SLAM) problem [19, 21] in mobile robotics [13]. Results of four general metaheuristics, Simple Genetic Algorithm, Particle Swarm Optimization, Quantum-Inspired Genetic Algorithms and Genetic Algorithm with Quantum Probability Representation, have been compared with results of analytic, classical method in this field, Iterative Closest Points algorithm. In the experiments the same objective function, drawn from Iterative Closest Points algorithm, has been used. Two situations have been considered: local and global localization problems of mobile robots. Both problems are import and often critical for successful navigation of robots.

The simultaneous localization and mapping problem is critical for efficient navigation and localization of autonomous robots. In a classical approach, SLAM uses different types of sensors (e.g. sonar, infra red, lasers, laser scanners, cameras) to locate the robot in its knowledge base, i.e. a map of environment. As the robot explores new areas, the knowledge base is updated and the new localization is calculated in the updated map. One of the well-tried methods addressing this problem is Iterative Closest Points algorithm [2]. Other possible approaches in this area include particle filters or geometry based localization [11, 23].

The motivation of our research was to investigate which algorithm performs best and in which variant of different localization problems in different environments.

* Computer Engineering Department, Technical University of Lodz, Poland

This paper is organized as follows. In subsection, 1.1 the localization problem has been described and essential definitions have been given. In subsection, 1.2 the classical method in this field, Iterative Closest Points algorithm, has been outlined. Subsection 1.3 briefly describes selected modern metaheuristics considered in this paper: Simple Genetic Algorithm, Particle Swarm Optimization, Quantum-Inspired Genetic Algorithms and Genetic Algorithm with Quantum Probability Representation. In section 2, parameters and conditions of our experiments have been given and an empirical comparison has been performed. Results of the experiment have been presented in section 3. Section 4 briefly concludes results of this paper.

1.1. Problem statement

The problem of robot localization considered in this paper is based on matching the scan from a laser rangefinder with points of the map. The map is created and updated simultaneously during the localization process. The data flow in this approach has been shown in Figure 1.

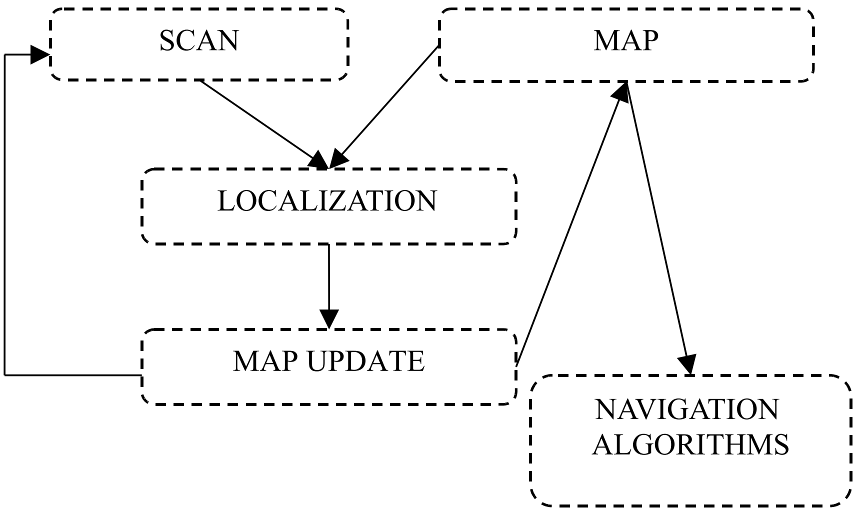

Fig. 1. Data flow in SLAM

A simple illustration of the localization problem has been presented in Figure 2. The arrow in the centre of the picture represents the robot position and orientation. The dark points represent the map of the environment; bright points represent a scan from the laser sensor. The localization problem regards to finding the pose in the space where the scan matches the map precisely.

Let $W = \{(x, y, \alpha) : x, y \in R, \alpha \in [-\pi, \pi]\}$ denote a space of all possible mobile robot poses, which include position (x, y) and orientation α. Let $Q_k \in W$ denote a pose of robot in the step k. Let us denote by $S = \{s_1, s_2, ..., s_n\}$ a *scan*, i.e. a set of points registered by robot sensors and by $M = \{m_1, m_2, ..., m_l\}$ a set of points that make up the *map*, where $s_i = (x_i, y_i)$ and $m_i = (x_i, y_i)$, respectively.

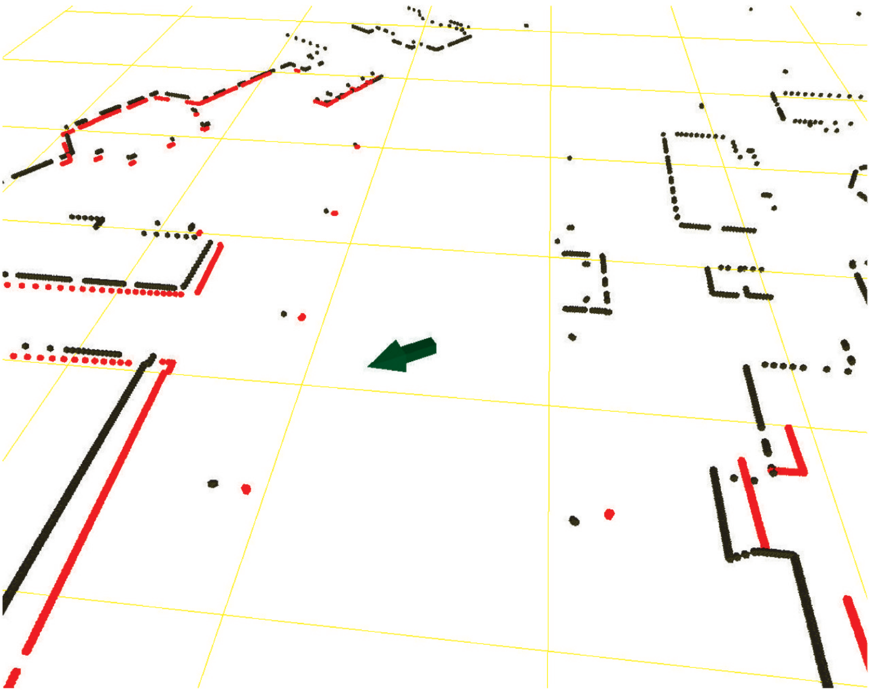

Fig. 2. Illustration of the localization problem

Moving in the environment, the mobile robot changes its pose in the W space. When the robot is in pose Q_k, the scan point s_k is in position (x_k, y_k) in the local coordinate system, while in the consecutive pose Q_{k+1} the same scan point is denoted by the position (x_{k+1}, y_{k+1}). In global coordinate system, points s_k, and s_{k+1} are equal. Assuming that the translation $(\Delta x, \Delta y)$ and rotation $(\Delta\alpha)$ are known, we can express the transformation between Q_k and Q_{k+1}, as shown on Figure 3, by:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\alpha) & -\sin(\Delta\alpha) \\ \sin(\Delta\alpha) & \cos(\Delta\alpha) \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{1}$$

Exact knowledge about the transformation between Q_k and Q_{k+1} is critical for the navigation algorithms. Consequently, localization algorithms have to be accurate and robust.

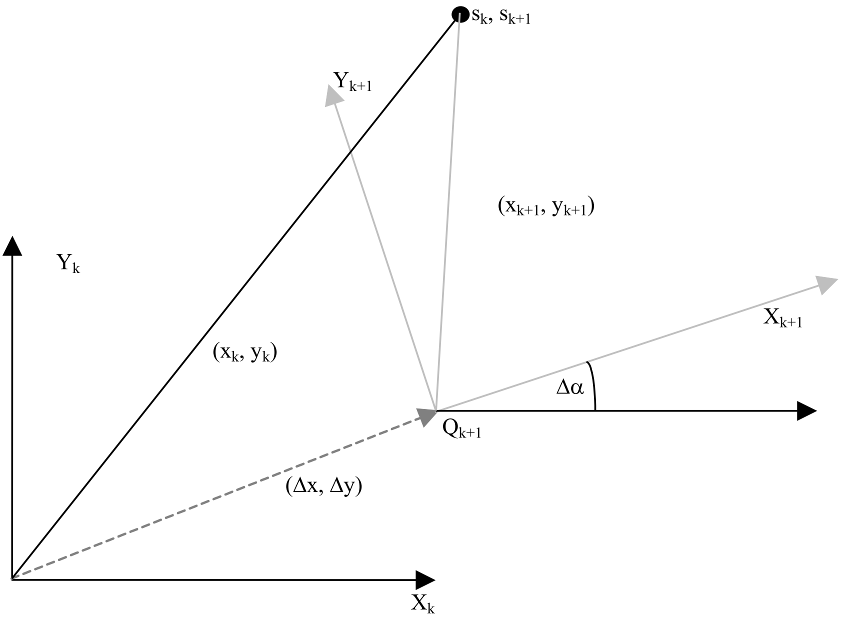

Fig. 3. Transformation between two consecutive poses Q_k and Q_{k+1} of a robot

1.2. Iterative Closest Points

Iterative Closest Points (ICP) algorithm was first presented in [2] and this section outlines the algorithm briefly. ICP is a classical, analytic method for solving the localization problem. The objective of the algorithm is to find the value of (Δx, Δy, $\Delta\alpha$) transformation that identifies current robot pose on the map. ICP algorithm is widely used in mobile robotics to solve the SLAM problem [17, 19, 21]. The algorithm is based on squared Euclidean distances between the scan and map points. This distance function $\delta : S \times M \mapsto R$ between scan and map points is expressed as:

$$\delta(s_i, m_i) = \left(x_i^s - x_i^m\right)^2 + \left(y_i^s - y_i^m\right)^2 \tag{2}$$

where $s_i = (x_i^s, y_i^s)$ and $m_i = (x_i^m, y_i^m)$.

Consecutive steps of the algorithm are given as follows:

1) First, the initial values of Δx, Δy, $\Delta\alpha$ need to be selected. The values can be taken from the previous execution of the algorithm, a priori knowledge can be used or the values can be set to zero in the first run.
2) Next, each point s_i of the scan S is transformed to obtain the scan S', as follows:

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} \cos(\Delta\alpha) & -\sin(\Delta\alpha) \\ \sin(\Delta\alpha) & \cos(\Delta\alpha) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{3}$$

In the last iteration the scan S' should fit to the map M precisely.

3) The minimum distance to the points of the map M is calculated for each point of the scan S'. Let us denote by $F_{S',M} : S' \mapsto M$ the closest point on the map M for the given scan point $s' \in S'$ i.e.:

$$F_{S',M}(s') = \underset{m \in M}{\arg\min}\, \delta(s', m) \tag{4}$$

4) Then, the error value E is calculated as the sum of all minimum distances between S' and M for each s_i':

$$E = \sum_{i=1}^{|S'|} \delta(s_i', F(s_i')) \tag{5}$$

When the value of the error function between current and previous iteration is smaller than a specified threshold, the algorithm is finished.

5) Finally, the new values of Δx, Δy, $\Delta\alpha$, which minimize the error value E, are calculated as given in (6)–(8). Then, the next iteration is started at the second point.

$$\Delta x^{new} = \sum_{i=1}^{|S|} \left(\left(F(s_i')\right)_x - \cos(\Delta\alpha) \cdot (s_i)_x + \sin(\Delta\alpha) \cdot (s_i)_y \right) \tag{6}$$

$$\Delta y^{new} = \sum_{i=1}^{|S|} \left(\left(F(s_i')\right)_y - \cos(\Delta\alpha) \cdot (s_i)_y - \sin(\Delta\alpha) \cdot (s_i)_x \right) \tag{7}$$

$$\Delta\alpha^{new} = \mathrm{arctg}\left(\frac{\sum_{i=1}^{|S|} \left((s_i)_y \cdot \Delta x - (s_i)_x \cdot \Delta y - \left(F(s_i')\right)_x \cdot (s_i)_y + \left(F(s_i')\right)_y \cdot (s_i)_x \right)}{\sum_{i=1}^{|S|} \left(-(s_i)_x \cdot \Delta x - (s_i)_y \cdot \Delta y + \left(F(s_i')\right)_x \cdot (s_i)_x + \left(F(s_i')\right)_y \cdot (s_i)_y \right)} \right) \tag{8}$$

1.3. Evolutionary computing methods

Evolutionary computing [4] is a branch of artificial intelligence that concerns optimization and search problems. In general, evolutionary techniques are metaheuristic optimization methods. Main subfields in this area are evolutionary algorithms [7, 18] and swarm intelligence methods [15]. Contrary to analytic or enumerative methods, in evolutionary computing, the algorithms maintain a population of candidate solutions. The algorithms use iterative process, such as growth and development of a population. Solutions in the population are evaluated according to their fitness. Knowledge extracted from the solutions, that with high fit, is exploited in subsequent iterations of the algorithm to explore the promising regions of the search space. Various random factors present in this approach allow the algorithms to perform very well in highly multimodal, nonlinear optimization problems.

From a wide range of evolutionary computing techniques, the following algorithms have been considered in this paper:

1) **Simple Genetic Algorithm** (SGA) – a classical evolutionary algorithm which mimics the process of natural evolution [10]. In Simple Genetic Algorithm solutions are represented as binary chromosomes and two genetic operators are applied: crossover and mutation. The algorithm is based on concepts drawn from biological evolution: adaptation, feature inheritance, selection pressure and survival of individuals that fit best in the environment.
2) **Particle Swarm Optimization** (PSO) – a method located in the subfield of swarm intelligence [6, 15]. In the used algorithm, a swarm of particles move in the search space. The algorithm draws its inspiration from collective behaviour of decentralized systems such as birds flocking, fish schooling or animals herding. This method is particularly suitable for numerical optimization problems as solutions, i.e. positions of particles, are encoded in real numbers.
3) **Quantum-Inspired Genetic Algorithm** (QIGA) – an evolutionary algorithm, proposed in [8], that draws inspiration from both: biological evolution and unitary evolution of quantum systems. The algorithm is based on concepts and principles of quantum mechanics such as qubits or superposition of states. Genes in the algorithm are modelled upon qubits, two-level quantum systems, which brings additional element of randomness and a "new dimension" into the algorithm. Genetic operators are based on quantum rotation gates that modify probability distributions of sampling the search space, encoded in quantum genes.
4) **Genetic Algorithm with Quantum Probability Representation** (GAQPR) – a relatively novel algorithm, proposed in [3]. This algorithm is an extension to Quantum-Inspired Genetic Algorithm. To prevent premature convergence of the evolutionary process an additional genetic operator is employed, exchanging information between quantum chromosomes.

Simple Genetic Algorithm and Particle Swarm Optimization are classical techniques in this field, while Quantum-Inspired Genetic Algorithm and Genetic Algorithm with Quantum Probability Representation are considered to be state-of-the-art methods in this area. Each of these methods has been successfully applied in variety of problems, ex. [3, 4, 15].

2. Empirical comparison of the algorithms

In our research, performance of ICP algorithm has been compared with four metaheuristics presented in the previous subsection. This section describes how the study was done.

The experiments have been conducted on two maps:

1) **Localization on artificial map** – one of the environments provided by Microsoft Robotics Developer Studio has been used. The map reflects the virtual urban environ-

ment. Nvidia PhysX engine has been used which allows collecting data that is similar to real signals [19]. The size of this artificial map is approximately 70×60 meters and the map has been presented in Figure 4. In the experiment, the correct pose of robot was located at $Q^* = (18.06, -0.84, 2.21)$.

2) **Localization on real map** – the map has been created for the Computer Engineering Department building. The real map has been generated by six-wheeled mobile robot platform which is being developed at the Technical University of Lodz in grant founded by The Ministry of Science and Higher Education. The robot is equipped with laser rangefinder that is able to scan environment in 270^{o} horizontally with the resolution of 0.25^{o}. The maximum visibility range of the laser rangefinder is 30 meters. The size of the map is approximately 20×20 meters. The map has been presented in Figure 5. Much more irregularities and noise from the laser scanner are visible in comparison to the artificial map. In the experiment, the correct pose of the robot was located at $Q^* = (2.89, -0.62, 0.75)$.

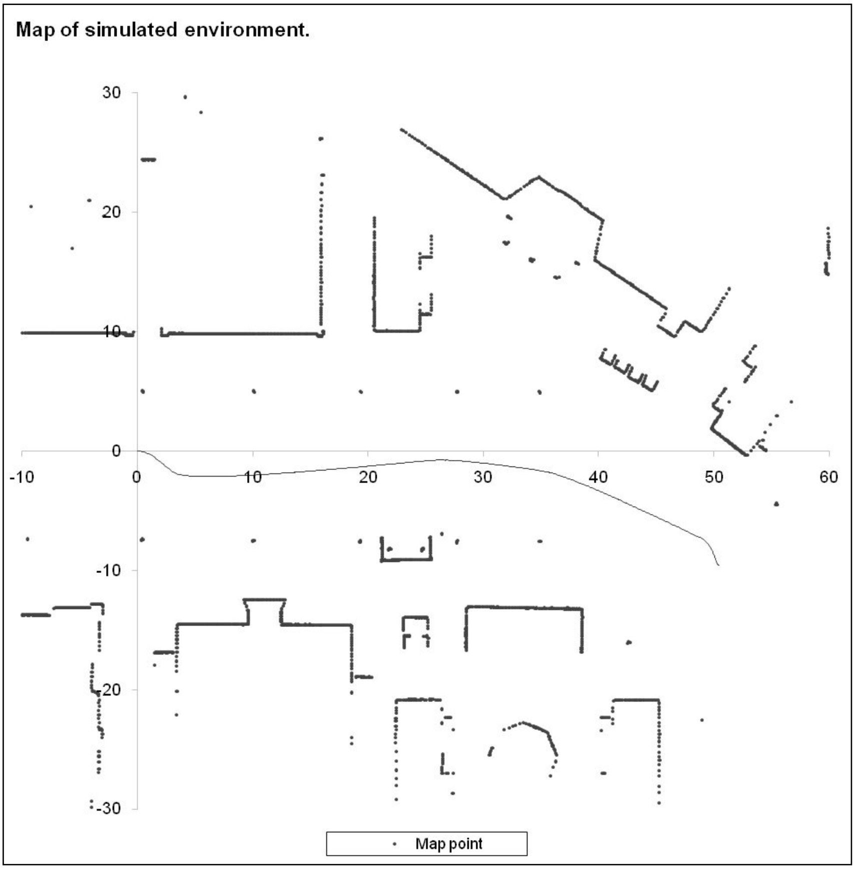

Fig. 4. Map generated from simulated data

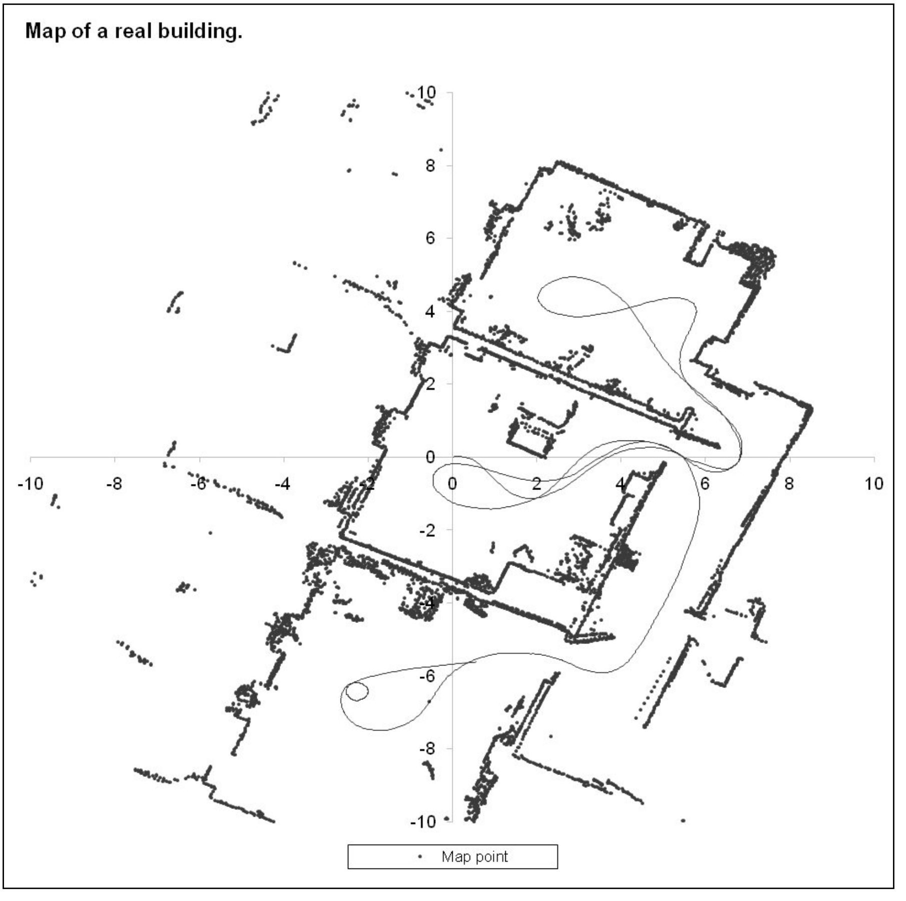

Fig. 5. Map of real building

For each map, two variants of the localization problem have been considered:

1) **Local localization problem** – assuming that previous localization and maximum speed of the robot is known, the feasible region in the search space can be reduced greatly. We have assumed that the maximum translation between consecutive poses is ±30 cm in each direction and maximum rotation is $\pm 30^{o}$. In this case:
   - The search space is highly reduced.
   - Usually, only one optimum in the feasible region exists, i.e. it is a uni-modal optimization problem.
2) **Global localization problem** – this situation occurs when the robot is turned on at an unknown location of the map. The objective is to find the robot's pose on the whole map. Size of the map is at least several dozen meters. Consequently, in this case:
   - Usually, plenty of local optimums exist in the search space (multi-modal problem).

Therefore:

– Greedy or deterministic algorithms can prematurely converge to local optimums, corresponding to wrong locations on the map, easily.

In each variant of the experiment, the same population size, 30 individuals, has been used. The fitness function has been based directly on the error value (5) used in Iterative Closest Points algorithm. The objective of the algorithms was to minimize the error value, i.e. find the pose $Q^* \in W$ such that $Q^* = \arg\min_{q \in W} S^q$, where S^q denotes the scan in the pose q. Neither scaling nor any other modification of the fitness value has been applied. Other parameters of the algorithms were set to highly typical values given in literature, namely:

1) Simple Genetic Algorithm – fitness-proportionate (roulette wheel) selection method has been used with no elitism. Two genetic operators applied were: one-point crossover with probability $P_c = 0.9$ and uniform mutation with probability $P_m = 0.001$, which results in mutation of one gene in population per generation on average. The chromosomes length was set to 30 genes.
2) Particle Swarm Optimization – learning factors were set to $c_1 = 2$, $c_2 = 2$, maximum velocity was set to about 1 meter/generation in the global problem and about 1 centimetre/generation in the local problem. In the beginning of the algorithm particles were scattered in the feasible region of the search space randomly.
3) Quantum-Inspired Genetic Algorithm (QIGA) – parameters given originally in [8] have been used. The crucial parameters of this algorithm are rotation angles given in so-called lookup Table. The rotation angles were also taken directly from [8]. Despite the fact the genetic operators in these algorithms were originally adapted for another type of problem, combinatorial optimization, the algorithm performs also very well on numerical optimization. The lookup table is the only special parameter for the QIGA algorithm, apart from the number of chromosomes and their length. The chromosomes length was set to 30 genes. Initial population consisted of quantum chromosomes that sample each point of the search space with equal probability.
4) Genetic Algorithm with Quantum Probability Representation (GAQPR) – the parameters that are in common with QIGA were set to the same values. Additional genetic operator, exchanging information between quantum individuals, presented in [3] has been applied with the probability $P_c = 0.7$.

The termination criterion in each case was simply the maximum number of generation, 130.

3. Results

Figures 6–9 in following subsections present results of the experiments. The plots present fitness of the best solutions in populations per generation number. The plots have been created for an average over 10 runs of the algorithms. Single iteration of Iterative Closest Points has been considered as a generation for comparison with evolutionary algorithms. However, it needs to be taken into account that in single iteration of Iterative Closest Points the objective function is evaluated only once. On the contrary, in evolutionary algorithms, the number of evaluations equals to the number of individuals in the populations. Since fitness evaluation is a dominant operation, it has a significant impact on real execution time of the algorithms.

3.1. Local localization problem

In Figures 6 and 7 results for the local problem on artificial and real map, respectively, have been presented. Not surprisingly, Iterative Closest Points algorithm performed absolutely best for the local problem. Moreover, number of fitness measure evaluations in ICP is 30 times lower in comparison to other methods. Therefore, ICP is completely incomparable with heuristic methods in local optimization. Out of the evolutionary algorithms, PSO performed best for both, artificial and real map. The reason of this result can be discerned the in representation of solutions. PSO is the only algorithm with solutions coded in real numbers, out of the considered methods.

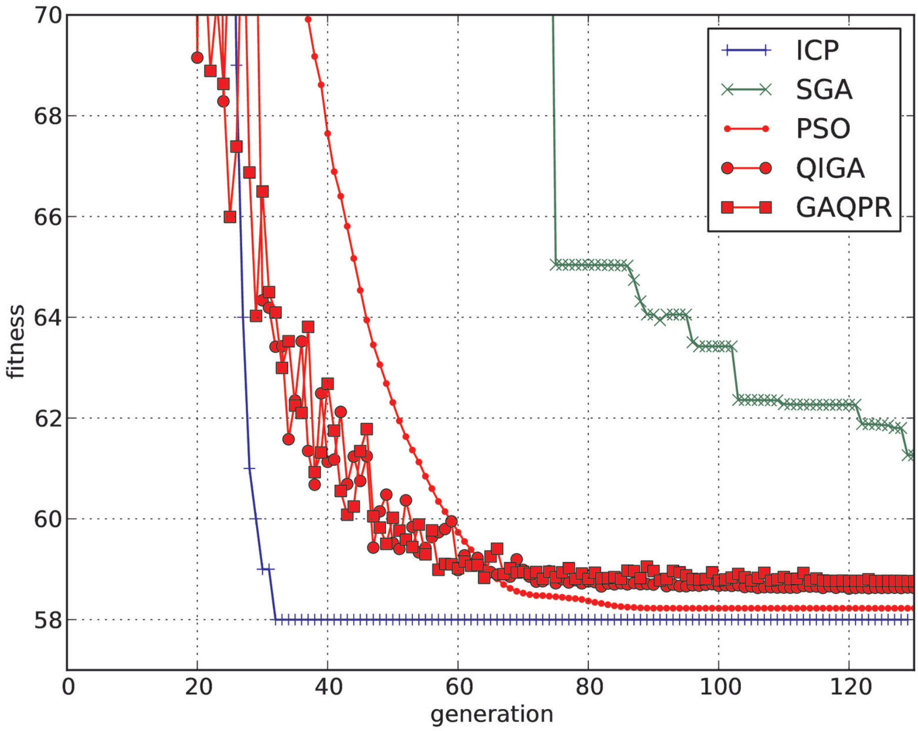

Fig. 6. Comparison for local problem on artificial map. Fitness/generation number

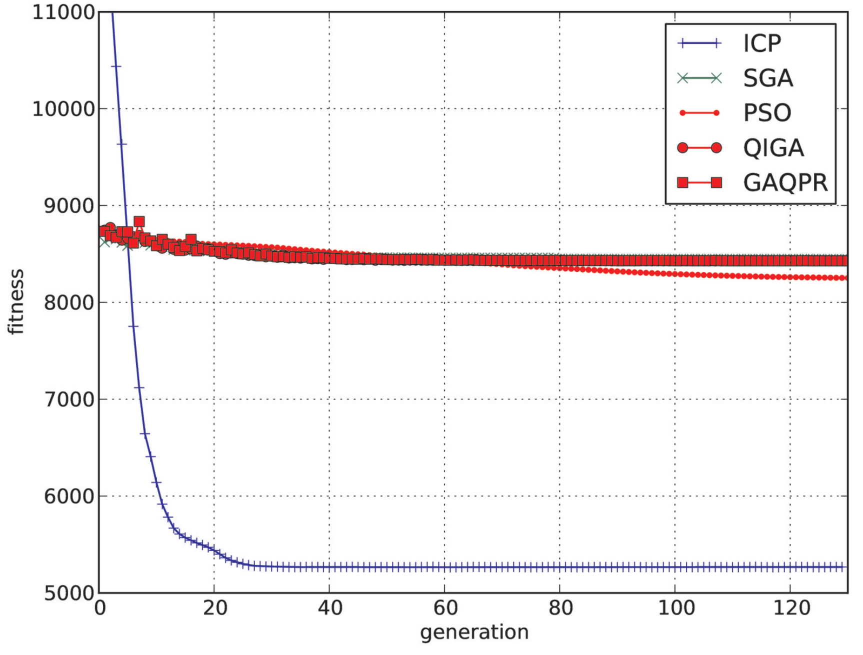

Fig. 7. Comparison for local problem on real map. Fitness/generation number

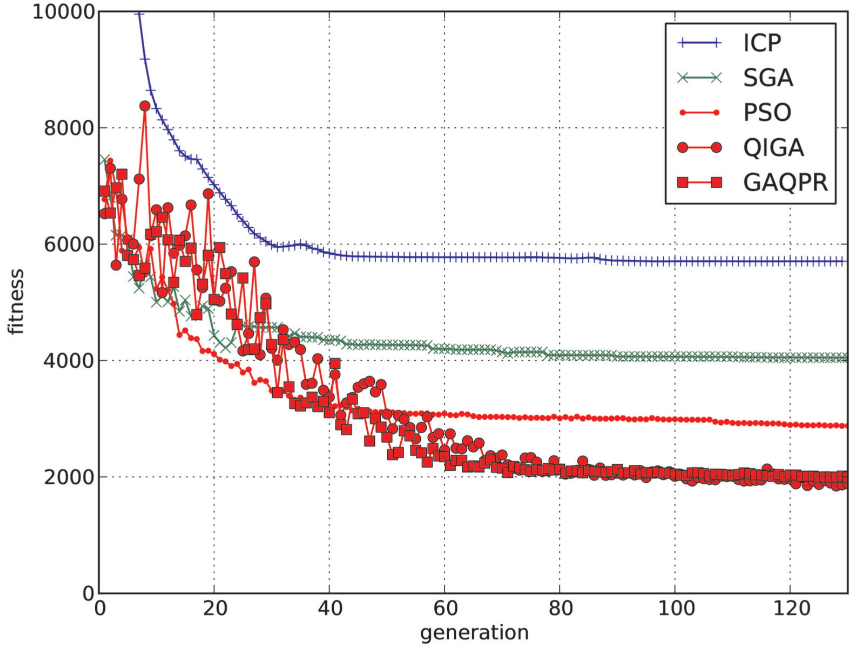

Fig. 8. Comparison for global localization problem on artificial map. Fitness/generation number

3.2. Global localization problem

In the global problem, ICP is the worst algorithm out of the considered group. This deterministic algorithm is suited for local localization problem. Therefore, in this case it gets trapped in local optimums, easily. The best algorithms were, respectively: QIGA, GAQPR, PSO. Surprisingly, the quantum-inspired genetic algorithms, QIGA and GAQPR, were the best methods in this case, despite the fact that they use binary coding and originally they were invented for another type of problem, the combinatorial optimization. Moreover, in 60 generations, they outperformed the PSO algorithms which suits perfectly for problems encoded in real numbers perfectly.

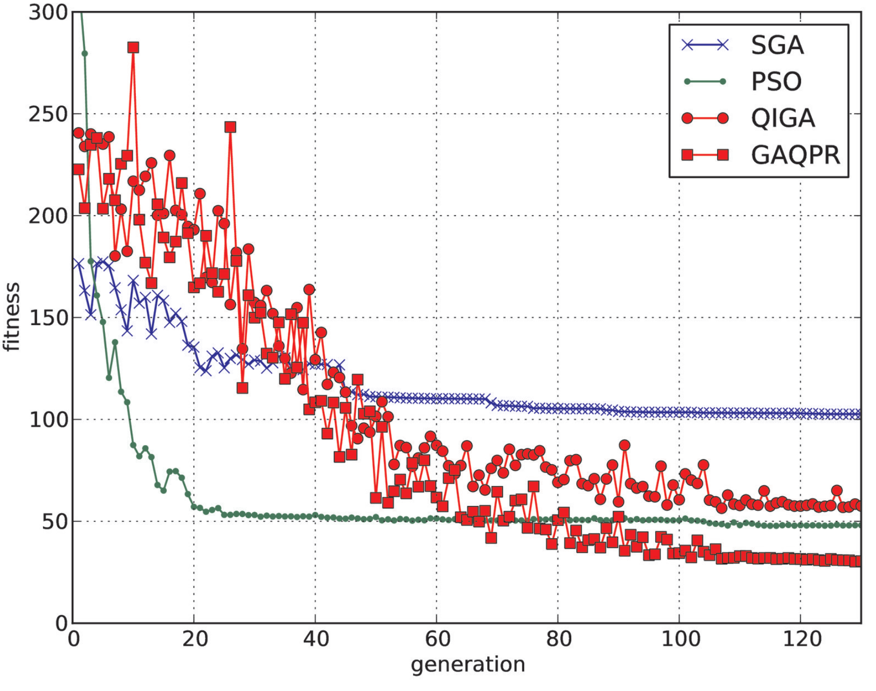

Fig. 9. Comparison for global localization problem on real map.
Fitness/generation number

In Figure 9, results for the real map have been presented. In 130 generations GAQPR algorithm performed best. However, it is easy to see that performance of PSO was also very good. Moreover, PSO performed extremely well in the first 20 generations of the algorithm, achieving the best convergence rate. Results of ICP algorithms, which performed terrible during the experiment on the real map, have been excluded from Figure 4 to keep readability of the Figure.

4. Conclusions

In this paper a comparison of selected evolutionary techniques for the simultaneous localization and mapping problem has been presented. The techniques were Simple Genetic Algorithm, Particle Swarm Optimization, Quantum-Inspired Genetic Algorithms and Genetic Algorithm with Quantum Probability Representation, for the simultaneous localization and mapping problem has been presented. The results have been compared with Iterative Closest Points algorithm. Local and global localization problems have been considered. Numerical experiments have been conducted for artificial environment and real map generated by laser rangefinder.

In the local problem, ICP is definitely the best algorithm, out of the compared methods. However, it does not work for the global problem. Best results for the global problem have been achieved by quantum-inspired genetic algorithms and PSO algorithm. PSO performs definitely best in the first dozens of generations. This was visible particularly in experiments performed on the real map with irregularities. Promising approach for further research is hybridization of the algorithms for global localization problem, i.e. development of a method consisting of two stages: rough selection of the feasible region by evolutionary technique followed by efficient and precise localization by Iterative Closest Points algorithm.

Other possibilities of further research include: analysis of the influence of parameters on the performance of the algorithms, implementation of the best methods in the real mobile robot and comparison of the algorithms for localization in three-dimensional space. Also, a comparison with respect to real time execution of the optimized algorithms execution is an interesting approach for further research.

Acknowledgement

Co-author Maciej Łaski is participating in the programme: "Innowacyjna dydaktyka bez ograniczeń – zintegrowany rozwój Politechniki Łódzkiej – zarządzanie uczelnią, nowoczesna oferta edukacyjna i wzmacnianie zdolności do zatrudniania, także osób niepełnosprawnych".

References

[1] Abolmaesumi P., Salcudean S.E., Zhu W.H., *Image-Guided Control of a Robot for Medical Ultrasound.* IEEE Transactions on Robotics and Automation, 18(1), 2002, 11–23.

[2] Besl P.J., McKay N.D., *A Method for Registration of 3-D Shapes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 239–256.

[3] Bin L., Junan Y., Zhenquan Z., *GAQPR and its application in discovering frequent structures in time series.* Neural Networks and Signal Processing, 2003.

[4] De Jong K.A., *Evolutionary computation: a unified approach*. MIT Press, Cambridge, MA, 2006.

[5] Dung-Ying L., Waller S.T., *A quantum-inspired genetic algorithm for dynamic continuous network design problem*. The International Journal of Transportation Research, vol. 1, Issue 1, January 2009.

[6] Eberhart R., Kennedy J., *A new optimizer using particle swarm theory.* Micro Machine and Human Science, 1995. MHS'95., Proc. of the Sixth International Symposium on, Anonymous, 1995, 39–43.
[7] Goldberg D.E., *Genetic algorithms in search, optimization and machine learning*. Addison Wesley, 1989.
[8] Han K., Kim J., *Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem* Proc. of the 2000 Congress on Evolutionary Computation, Anonymous, IEEE Service Center, Piscataway, NJ, 2000, 1354–1360.
[9] Hirschfeld R.A., Aghazadeh F., Chapleski R.C., *Survey of Robot Safety in Industry.* International Journal of Human Factors in Manufacturing, 3(4), 2007, 369–379.
[10] Holland J.H., *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.
[11] Howard A., *Multi-robot Simultaneous Localization and Mapping using Particle Filters.* In Robotics: Science and Systems, 2005.
[12] Jeżewski S., Łaski M., *Przegląd i porównanie środowisk symulacji robotów mobilnych*. ISSN: 1429-3447, 2009.
[13] Jeżewski S., Sankowski D., Dadan W., *Koncepcja autonomicznego robota pola walki przeznaczonego do zadań zwiadu i wykrywania min*. ISSN: 1429-3447, 2009.
[14] Jopek Ł., Nowotniak R., Postolski M., Babout L., Janaszewski M., *Zastosowanie kwantowych algorytmów genetycznych do selekcji cech*. Automatyka (półrocznik AGH), t. 13, z. 3, 2009, 1219–1232.
[15] Kennedy J., Eberhart R., *Particle swarm optimization.* 1995.
[16] Lopes A., Almeida F., *A Force-Impedance Controlled Industrial Robot using an Active Robotic Auxiliary Device.* Robotics and Computer-Integrated Manufacturing, 24(3), 2008, 299–309.
[17] Martínez J.L., González J., Morales J., Mandow A., García-Cerezo A.J., *Mobile Robot Motion Estimation by 2D Scan Matching with Genetic and Iterative Closest Point Algorithms*. Wiley InterScience, Journal of Field Robotics 23(1), 2006, 21–34.
[18] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1999.
[19] Nuchter A., *The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom.* ISSN 1610-7438, 2009.
[20] Speich J.E., Rosen J., *Medical Robotics*. Encyclopedia of Biomaterials and Biomedical Engineering, 2004, 983–993.
[21] Stachniss C., *Robotic Mapping and Exploration.* ISSN 1610-7438, 2009.
[22] Stan S.D., Balan R., Matieş V., *Kinematics and Fuzzy Control of ISOGLIDE3 Medical Parallel Robot.* Mechanika, 75, 2009, 62–66.
[23] Wang C., Thorpe C., *Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects*. Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2003, 842–849.
[24] Zhuang, F., Zupan C., Chao Z., Yanzheng Z., *A cable-tunnel inspecting robot for dangerous environment*. International Journal of Advanced Robotic Systems, vol. 5, No. 3, ISSN 1729-8806, 2008, 243–248.