Hsu-Chih HUANG

National Ilan University

FPGA-Based Hybrid GA-PSO Algorithm and Its Application to Global Path Planning for Mobile Robots

***Abstract**. This paper presents an FPGA-based (field-programmable gate array) hybrid metaheuristic GA (genetic algorithm)-PSO (particle swarm optimization) algorithm for mobile robots to find an optimal path between a starting and ending point in a grid environment. GA has been combined with PSO in evolving new solutions by applying crossover and mutation operators on solutions constructed by particles. This hybrid algorithm avoids the premature convergence and time complexity in conventional GA and PSO algorithms. The initial feasible path generated from the hybrid GA-PSO planner is then smoothed using the cubic B-spline technique, in order to construct a near-optimal collision-free continuous path. Experimental results are conducted to show the merit of the proposed hybrid GA-PSO path planner for global path planning for mobile robots.*

***Streszczenie.** W artykule zaprezentowano algorytm dla mobilnych robotów poszukujący optymalnej ścieżki między punktem startu i końcowym. Algorytm wykorzystuje układy FPGA i bazuje na algorytmach genetycznych i mrówkowych. (**Algorytm sterowania mobilnymi robotami wykorzystujący układy FPGA i bazujący na algorytmach genetycznych i mrówkowych**)*

Keywords: FPGA, Global path planning, GA, PSO.
Słowa kluczowe: mobilne roboty, algorytm genetyczny..

Introduction

Global path planning is an important problem in many disciplines, including very large scale integrated circuits (VLSI) design, global positioning systems (GPS) applications, and autonomous robot navigation. These global path planning problems have been solved using many existing approaches, such as cell decomposition [1,2], skeleton and potential field [3]. In addition, soft computing approaches, such as fuzzy logic, neural networks, and evolutionary algorithm, have been widely used to solve the global path planning problems [4,5,6] .

GA and PSO are both widely used intelligent optimization algorithms, they have their own advantages and disadvantages [7,8]. GA performs crossover and mutation operations to recombine chromosomes. It has strong global search capability, but its convergence is slow because no memory mechanism is used. PSO has much more powerful intelligent background because the knowledge of good solutions is retained by particles. It has constructive cooperation between particles, namely that particles in the swarm share their searching experiences [7,8]. This paper presents an FPGA-based hybrid metaheuristic GA-PSO method to circumvent the problems in conventional GAs and PSOs for global path planning. Furthermore, this FPGA implementation algorithm is more suitable for robot applications in comparison with the PC-based algorithms.

Although the collision-free path can be easily obtained from the proposed FPGA-based hybrid GA-PSO algorithm, the resultant path is composed of a sequence of line segments. These discontinuous segments are smoothed by the B-spline modeling [9,10,11] which is one of the most efficient curve interpolations. The rest of this paper is organized as follows. In Section II, the hybrid GA-PSO algorithm is proposed to resolve the global path planning problem. Section III elucidates the FPGA implementation of the GA-PSO algorithm. Section IV conducts several experiments to show the performance and merit of the proposed methods. Section V concludes this paper.

Hybrid GA-PSO algorithm for global path planning

To circumvent the problems in conventional GA and PSO algorithms, the proposed hybrid GA-PSO algorithm has a standard PSO hybridized with GA operators, including crossover and mutation operations to generate new population. In the following, these two algorithms for global path planning are briefly described and the hybrid GA-PSO approach is then presented.

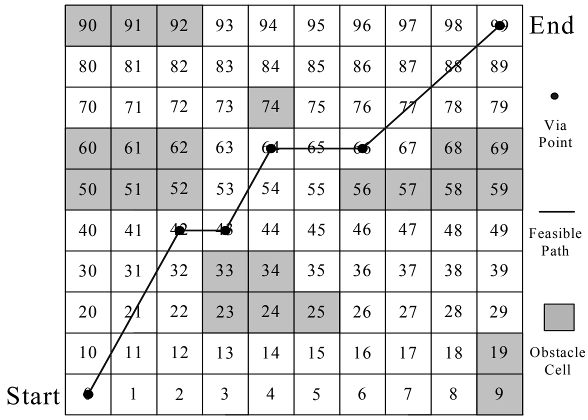

Fig.1. A grid-based environment with obstacles and the planned collision-free path

A. GA for global path planning

GA is an adaptive and heuristic search method. Its main idea is to construct a fitness according to the objective function to evaluate all chromosomes in a population. The optimal collision-free path is evolved by the genetic operators [7].

As shown in Fig. 1, a collision-free path (chromosome) contains a set of the grid points (genes of a chromosome), including the start point, via-points and the end point. The length of a chromosome was variable, ranging from two to maximum length $M_{\max}$. A path can be either feasible (collision-free) or infeasible by evaluating the fitness function expressed by

$$F = \sum_{i=1}^{M_{\max}} \left(d_i + \alpha_i T \right) \tag{1}$$

where $M_{\max}$ is the number of line segments of a path, d_i is the distance of the near two nodes forming the line segment, *T* is a constant, α_i is given by

$$\alpha_i = \begin{cases} 0 & \text{if the ith line segment is feasible} \\ \sum_{k=1}^{N} k & \text{if the line segment intersects obstacle(s)} \end{cases} \tag{2}$$

where *N* is the number of obstacles that the line segment intersects.

B. PSO for global path planning

The PSO algorithm is a population based optimization method. This algorithm can be applied to solve for the optimal problems with the multimodal function $f(x) = f(x_1, x_2, ...x_n)$ by using a population of particles [7,8]. The fitness of each particle is given by the fitness function *f(x)*. Each particle represents a feasible path solution of the global path planning problem.

In PSO algorithm, $x_i(t)$ denote the position of particle *i* in the search space at discrete time steps *t*. The position of the particle is changed by adding a velocity, $v_i(t)$ to the current position, given by

(3) $$x_i(t+1) = x_i(t) + v_i(t+1)$$

with a initial position $x_i(0)$. The velocity vector drives the PSO optimization process and reflects both the experimental knowledge of the particle and socially exchanged information from the particle's neighborhood.

The best position reached by the single particle (*gbest*) is responsible for the former type of attraction. On the other hand, the best location found by the rest of the swarm (*pbest*) is the second factor, which indicates the influence of the swarm on the single particle. Each particle *i* moves around the search space, and renew its velocity using its past experience and the population's experience given by

(4) $$v_{ij}(t+1) = wv_{ij}(t) + c_1\varphi_1(p_{best} - x_{ij}(t)) + c_2\varphi_2(g_{best} - x_{ij}(t))$$

where $v_{ij}(t)$ is the velocity of particle *i* in dimension *j* at time step *t*, $x_{ij}(t)$ is the position of particle *i* in dimension *j* at time step *t*, c_1 and c_2 are positive acceleration constants used to respectively scale the contribution of the cognitive and social components. φ_1 and φ_2 are uniform random numbers. These two random values introduce a stochastic element to the PSO algorithm. w is called inertia weight which control the momentum of the particle by weighting the contribution of the previous velocity.

C. Hybrid GA-PSO algorithm for global path planning

This subsection aims to employ the GA-PSO hybrid algorithm to design an optimal path planner for mobile robots. Each PSO particle is composed of the point sequence, including starting point, via point and ending point. The optimal feasible path will be evolved by the efficient GA-PSO hybrid algorithm described by the following steps.

Step 1: Initialize the swarm size, neighbourhood size, search space, acceleration coefficients, and number of iterations.
Step 2: Randomly generates particles and initialize the position and velocity of the particles.
Step 3: Calculate the fitness value for all the particles.
Step 4: GA crossover and mutation operations are executed after all particles have constructed a solution.
Step 5: (1) Search of personal best population.
(2) Search of global best population.
Step 6: (1)Update the velocity using (4).
(2)Update the position using (3).
Step 7: Check the stop criterion. If the stop criterion is not matched, go to Step 3 and set $t = t+1$, otherwise, output the optimal path and its corresponding feasible path and stop the algorithm.

FPGA implementation

This section presents the FPGA implementation of the proposed GA-PSO using hardware/software co-design technique. Fig. 2 depicts the architecture of the FPGA implementation for the proposed GA-PSO. The user IP cores (custom logic) for this GA-PSO operators have been developed by VHDL (VHSIC Hardware Description Language), including random number generator (RNG) module, particle updating module, GA operations, and fitness module. Note that the soft-core processor Nios II is embedded as path smoother to smooth the discontinuous path from the GA-PSO path planner. The software-based path smoother and hardware-based custom logics for the GA-PSO are connected to the system interconnect fabric via Avalon memory-mapped interface in one FPGA chip.

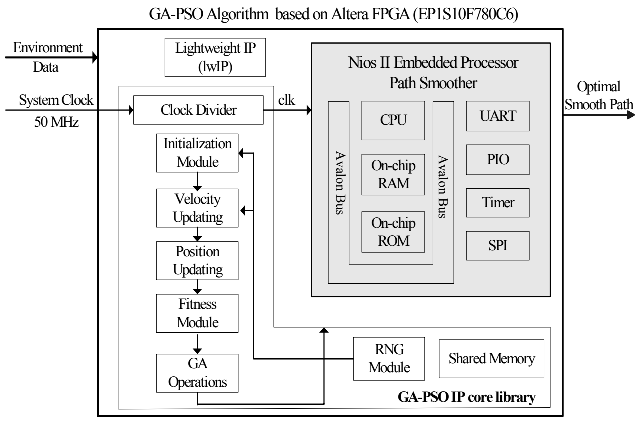

Fig. 2. FPGA implementation of the proposed hybrid GA-PSO algorithm for global path planning

In VHDL GA-PSO processor, the particle position and velocity are continuously updated according to the p_{best} and g_{best} information, thereby moving the particles toward the optimal solution. The update process can be implemented by using a simple adder and a multiplier to update the position and velocity of the particles in the proposed GA-PSO algorithm. This high-performance hardware module is implemented by VHDL to efficiently update the particles. The two random numbers, φ_1 and φ_2 are generated from the linear feedback shift register (LSFR) RNG module. The initial particle data stored in the swarm memory is also obtained from the LSFR-based RNG module.

As mentioned above, the fitness function can be defined for its corresponding optimal problem. The purpose of this fitness evaluation module is to evaluate the particles after position and velocity update and find out the p_{best} and g_{best} in the GA-PSO algorithm. By taking the advantage of high-performance of hardware implementation, the fitness evaluation module has been efficiently implemented by VHDL, thus significantly improving the execution performance. After the evaluation process, the p_{best} and g_{best} for the PSO algorithm will be stored in the swarm memory. The FPGA-based GA-PSO for solving the global path planning problem with the fitness function in (1) is described by the following steps.

Step 1: Use the LSFR-based RNG to randomly generate the initial particles from the start to the goal.
Step 2: In the VHDL-based GA-PSO, execute the procedure of particle updating and also check whether new particles are acceptable. If the new particles do not satisfy the requirement, repeat this procedure until acceptable particles are obtained.

Step 3: Evaluate the fitness values of the particles.
Step 4: Check the stopping conditions. If the stopping criterion is not met, go to Step 2.
Step 5: Smooth the discontinuous path using B-spline.

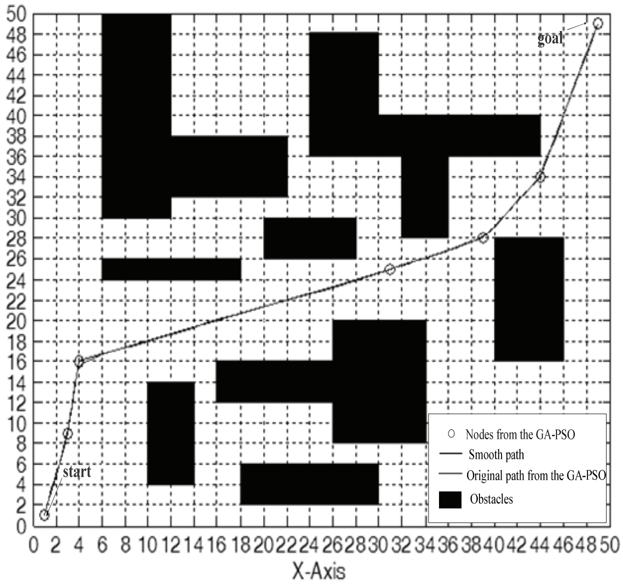

Fig. 3. Experimental results of the proposed GA-PSO and GA path planning in a complex environment

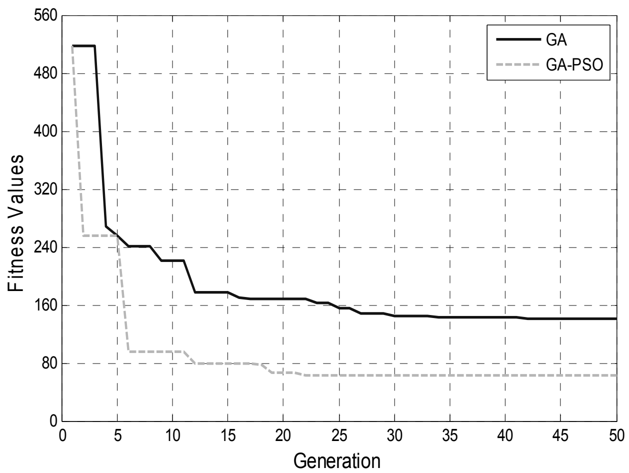

Fig.4. Fitness values of the GA-PSO and GA planners to achieve path planning in a complex environment

Experimental results

The following experiments are conducted to illustrate the feasibility and merit of the proposed hybrid GA-PSO path planner together with the B-spline path smoother in different environments. In addition, the resultant fitness values are presented for illustration of effectiveness of the proposed GA-PSO path planner. These simulations are performed with the parameters: $w = 0.8,\ c1 = c2 = 1.5$. The crossover probability is 0.7, and the mutation probability is 0.1. Worthy of mention is that the fitness values of hardware-based GA-PSO are obtained from the FPGA.

A. Path planning in a complex environment

The first experiment was conducted to present the path planning and smoothing results for mobile robots in a complex environment. Fig. 3 depicts the discontinuous feasible path from the proposed hybrid GA-PSO algorithm and the smooth collision-free path using the B-spline modelling. Fig. 4 presents the fitness value of the GA-PSO for solving the global path planning problem. In order to exploit the merit of the proposed hybrid GA-PSO, the conventional GA used to solve the same problem is also presented in Fig. 4. As shown in Fig. 4, the hybrid GA-PSO has better convergence behaviour than GA does.

B. Path planning in a double U-shape environment

Fig. 5 presents the path planning result for mobile robots in a double U-shape environment. This result has shown that the proposed GA-PSO is capable of evolving optimal collision-free path in this environment. Fig. 6 depicts fitness values for solving the global path planning in a double U-shape environment using the proposed GA-PSO and conventional GA. As shown in Fig. 6, the proposed GA-PSO algorithm converges to the optimal collision-free path with better fitness value, namely that the proposed GA-PSO outperforms conventional GA to solve the optimal problem.

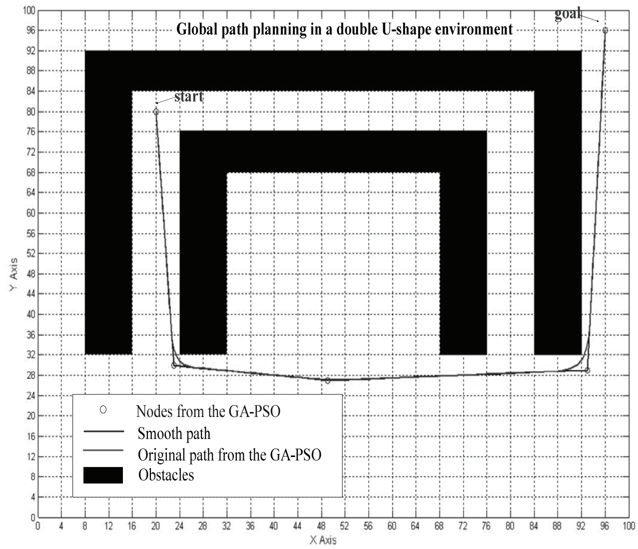

Fig.5. Experimental results of the proposed GA-PSO and GA path planning in a double U-shape environment

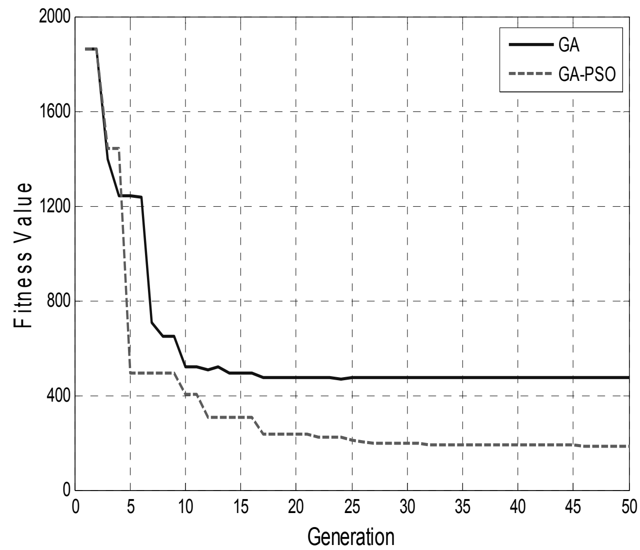

Fig.6. Fitness values of the GA-PSO and GA planners to achieve path planning in a double U-shape environment

Conclusion

This paper presents an FPGA-based hybrid metaheuristic GA -PSO algorithm for mobile robot to find an optimal path between a starting and ending point in a grid environment. The proposed GA-PSO has been efficiently resolved for the global path planning problems in a structure environment with obstacles. In order to smooth the planned paths from the proposed GA-PSO planner, the path smoother has been proposed using the B-spline smoothing technique. Through experimental results, the proposed GA-PSO has been shown to find the feasible paths in different environments. These experimental results clearly indicate that the proposed FPGA-based GA-PSO outperforms conventional GAs.

Acknowledgments

The authors gratefully acknowledge financial support from the National Science Council, Taiwan, R.O.C., under grant NSC100-2221-E-197-032.

REFERENCES

[1] A. Hourtash and M. Tarokh. Manipulator path planning by decomposition: algorithm and analysis. IEEE Transactions on Robotics and Automation, 17(6)(2001) 842-856.

[2] C. Cai and S. Ferrari. Information-driven sensor path planning by approximate cell decomposition. IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics, 39(3)(2009) 672-689.

[3] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. IEEE Transactions on Robotics and Automation, 8(5)(1992) 501-518.

[4] I. K. Jung, K. B. Hong, S. K. Hong, S. C. Hong. Path planning of mobile robot using neural network. Proceedings of the IEEE International Symposium on Industrial Electronics, 3(1999) 979-983.

[5] H. Surmann, J. Huser, J. Wehking. Path planning for fuzzy controlled autonomous mobile robot. International Conference on Fuzzy Systems, 3(1996) 1660-1665.

[6] C. Hocaoglu, A. C. Sanderson. Planning multiple paths with evolutionary speciation. IEEE Transactions on Evolutionary Computation, 5(3)(2001) 169-191.

[7] R. Nie, J. Yue. A GA and particle swarm optimization based hybrid algorithm. IEEE Congress on Evolutionary Computation, (2008) 1047-1050.

[8] S. W. Chen, J. S. Chiou, Y. C. Hu, K. Y. Wang, M. Y. Shieh. Application of the GA-PSO with the fuzzy controller to the robot soccer. Proceedings of SICE Annual Conference, (2010). 2083-2087

[9] C. G. Johnson and D. Marsh. Modelling robot manipulators with multivariate B-splines. Robotica, 17(3)(1999) 239-247.

[10] Q. Huang, S. Hu. and R. Matin. Fast degree elevation and knot insertion for B spline curve. Computer Aided Geometric Design, 22(2)(2005) 183-197.

[11] Z. Wu, H. S. Seah and M. Zhou. Skeleton based parametric solid models: ball b-spline surfaces. IEEE International Conference on Computer-Aided Design and Computer Graphics, (2007) 47-52.

***Hsu-Chih Huang**: Assistant Professor, Department of Electrical Engineering, National Ilan University, Taiwan, E-mail: hchuang@niu.edu.tw.*