Ant Colony Optimization algorithm design and its FPGA implementation

Shih-An Li, Min-Hao Yang, Chung-Wei Weng, Yi-Hong Chen,
Chia-Hung Lo, Ching-Chang Wong
Department of Electrical Engineering,
Tamkang University,
New Taipei City, 251, Taiwan
lishyhan@gmail.com

Abstract—**In this paper, a Hardware/Software (HW/SW) co-design method of ant colony optimization (ACO) algorithm is proposed to implement on the FPGA chip. In this paper, the software is designed with C language and hardware is designed with Verilog hardware description language (HDL). The HW/SW co-design method is a technique based on a SOPC (System on a Programmable Chip). In this paper, the path selecting and path analysis are designed in SOPC. The path selecting belongs to the pre-processing of the ACO algorithm and it cost a longer computing processing time. Therefore, a hardware circuit is designed to speed up processing. The path analysis will be designed by the C language within the NIOS II processor. In the experimental results, the processing time can be reduced by the proposed method.**

Keywords- Ant Colony Algorithm; FPGA; Hardware/Software Codesign; SOPC;

I. INTRODUCTION

Research topics in robotics Consists of image recognition, robot positioned, move control, path planning, and arms control and other related research. In these topics, Path planning becomes very important, regardless of the route of robot moves, the robot moves, and during the move to avoid obstacles or move to specific points to complete the required tasks. Human beings in determining the destination, through thinking and learning to change the path and speed up the arrival time, the robot is also optimized path planning through reflection and learning, in path planning often used genetic algorithms, PSO algorithm, ant colony algorithm and other methods to achieve the effect of optimization. In optimization problem, the ant colony algorithm has the advantage is that the choice of the location of a mobile for the decision easier to meet the requirements of the transition probability rules, and also have a more detailed consideration, so it can produce a good solve. Therefore, this paper will be designed of ant colony algorithm by travelers to find the optimal path planning. And how to effectively allocate the limited space within the robot mechanism, does not affect the performance has been a very important issue. Therefore, the operation of the FPGA chip to quickly deal with the robot is a very viable approach. In addition, Nios II processor is designed by Altera Corporation. It can be programmed to the 32-bit embedded soft core processor (Embedded Soft-Core Processor). The Nios II processor is designed by the VHDL or Verilog language and it's a reduced instruction set compute (RISC) architecture. Therefore, the FPGA chip is a flexibility chip and integrated the circuit easily for robot design. And in order to achieve the above factors, this paper will use the SOPC [1-2] technique and HW/SW co-design [3-4] to achieve the ant colony algorithm.

The main frameworks of this paper are described below. First, in Section II, describes the SOPC system. In Section III, the principle of ACO algorithm and the choice of path selection method will be introduced. In Section IV, the actual design of the design method of ant colony algorithm, filed in Section II. The experiment results are presenting in section V, and finally section VI is conclusions.

II. SOPC SYSTEM

At the high level application, it's more and more important to combine the microprocessor into FPGA. Therefore, the SOPC will provide a platform to integrate a programmable system into a FPGA chip. A design flow of traditional system is shown in Figure.1. The system designers must consider the design procedure around the all processor hardware and add the peripherals to integrate into the required system. The design flow of hardware circuit will cause great difficulties to system designers. The system designers must both have the development of hardware and software experiments to develop the corresponding to the function, and it will be greatly increased the difficulty on hardware and software integration.

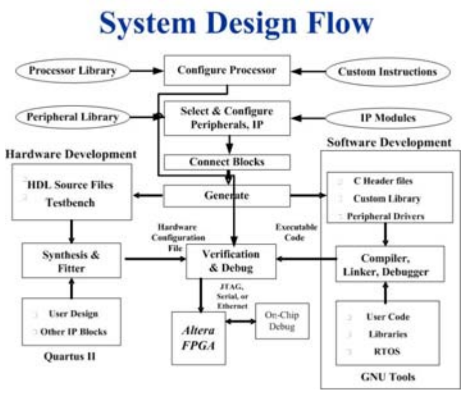

Figure 1. Traditional hardware and software co-design process

his research was supported in part by the National Science Council (NSC) of the Republic of China under contract NSC 100-2221-E-032-010.

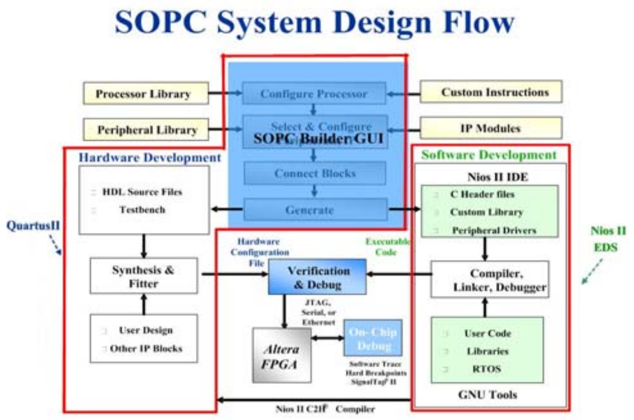

Figure 2. SOPC system hardware and software co-design framework

Altera Corporation presents the SOPC Builder tool to integrate the software system architecture shown in Figure.2. It provides a good platform for the development environment for common hardware and software designers. It incorporates a soft-core processor and digital hardware circuit development environment. The designers can easy to integrate the complex system and IP library with a fast and stable embedded system platform.

III. ANT COLONY OPTIMIZATION ALGORITHM

The ACO algorithm is based on the mobile and behavior of the ants. Under normal circumstances, the ants will release the pheromones to the movement path. Once the ants find food, the ants will carry some food to the nest. The other ants will follow the trajectory of pheromones to handle the foods at the same location. Therefore, the pheromone can achieve the indirect communication.

We will introduce the formula applied to the ACO algorithms for path planning. The formula is set by simulating the pheromones growth or decline in the environment. In the natural environment, it should be no pheromones left by the ants. How the ants to find food? In addition to rely on pheromones to move, the ants also have the ability of exploring to explore the unknown place. Besides the pheromones, the other parameters must be added to the ability of exploration. Based on the pheromones, the exploratory behavior can derive the formula to a transition probability equation (1).

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}\times[\eta_{ij}(t)]^{\beta}}{\sum\limits_{k\in allowed_k}[\tau_{ik}(t)]^{\alpha}\times[\eta_{ik}(t)]^{\beta}} & if \quad j\in allowed_k \\ 0 & others\end{cases} \tag{1}$$

where the $\tau_{ij}(t)$ is the pheromone concentration between node i to node j ,and $\eta_{ij}(t)$ represents the visibility between node i to node j , that is $\eta_{ij}=\dfrac{1}{d_{ij}}$, d_{ij} is the distance between node i to node j . The formula is shown in equation (2). The $P_{ij}^{k}(t)$ is a transition probability of choosing this path.

$$d_{ij}=[(x_i-x_j)^2+(y_i-y_j)^2]^{1/2} \tag{2}$$

The α is the pheromone parameter and β is the absolute distance parameters. They were used to control the $\tau_{ij}(t)$ and $\eta_{ij}(t)$, t is the time parameter. When the α is set to great, it will have a great the proportion to choice the mobile path. The large β indicates that it have higher exploration ability.

When the ants on the move, pass through the route will leave some pheromones to record the route and the other ants will follow the route. The pheromones will increase by more ants to pass through the route. Or it will reduce by the environmental factors. Therefore, development of the pheromone update formula can be described as equation (3)

$$\tau_{ij}(t+1)=\rho\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij} \tag{3}$$

ρ is behalf of the pheromones disappeared coefficient. The coefficients range between 0 and 1. If the ρ is too small (close to 0) then it can not converge. Because the pheromones are evaporating and it can not be accumulated over time. If the ρ is too big (close to 1), will be early convergence. $\Delta\tau_{ij}$ is the pheromones between node i to node j is shown in equation (4). Q is the secrete pheromones of the ants. When an ant pass through the route， $\tau_{ij}(t+1)$ will equal to $\rho\tau_{ij}(t)$ plus Q . If some route have no ants pass through, $\tau_{ij}(t+1)$ will only equal to $\rho\tau_{ij}(t)$, over time the $\tau_{ij}(t)$ will become zero.

$$\Delta\tau_{ij}=\begin{cases}Q\\0\end{cases} \tag{4}$$

The flowchart of the ant ACO algorithm is shown in Figure 3. A first, the basic parameters and the starting position of the ants will be set. After then, the program will be begun to calculate the transition probability to choose the path and the start trip.

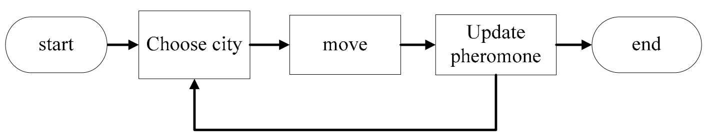

Figure 3. basic flow chart of ant colony optimization

When the completion of the first searching trip, the path length of their journey will be calculated, and update the pheromones on the route. Then put the ants from the start location and begin searching the foods. Finally, the each of the journey will be compared and found out the shortest path, and outputs the result.

When we calculate the transition probability rules, the following manner will be used to select a mobile node. The path diagram and flowchart are shown in Figure 4. The point S is the origin, and point A to point G are the choice points of all the nodes. $P_{SA}(t)$ 、 $P_{SB}(t)$ 、 $P_{SC}(t)$ 、 $P_{SD}(t)$ 、 $P_{SE}(t)$ 、 $P_{SF}(t)$ 、 $P_{SG}(t)$ are the transition probability between point S to point A ~ G transition probability. A random value R and the sum of the value of SUM are given. First, the $P_{SA}(t)$ have put to the SUM and compared SUM and R. If SUM is greater than R, we will choose A be the next node. Otherwise, if SUM is less than R, We will proceed to the next steps. In the next step, the sum of $P_{SA}(t)$ plusing $P_{SB}(t)$ will be put to SUM and compared with R. If SUM is greater than R, we will choose B be the next node. Otherwise, if SUM is less than R, We will proceed to the next steps, put the sum of the $P_{SA}(t)$ plusing $P_{SB}(t)$ and $P_{SC}(t)$, and compare SUM and R. If SUM is greater than R, we will choose C be the next node. Otherwise, if SUM is less than R, We will proceed to the next steps. By repetitive sequence to each transition probabilities to SUM, and compare the value of SUM and R to determine the mobile node to the next is our path selection method.

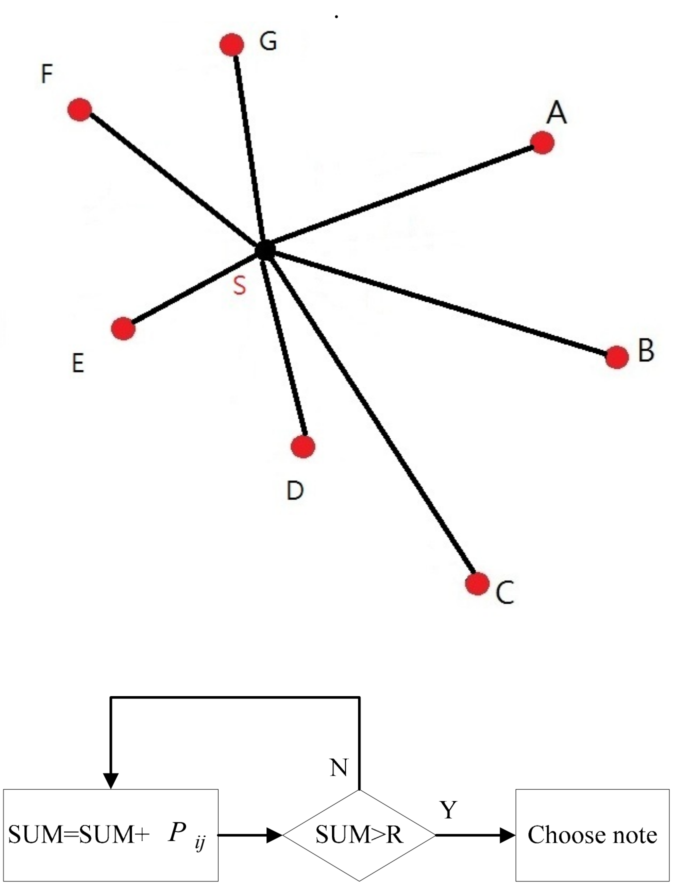

Figure 4. Path selected schematic and flowchart

IV. HARDWARE/SOFTWARE CO-DESIGN

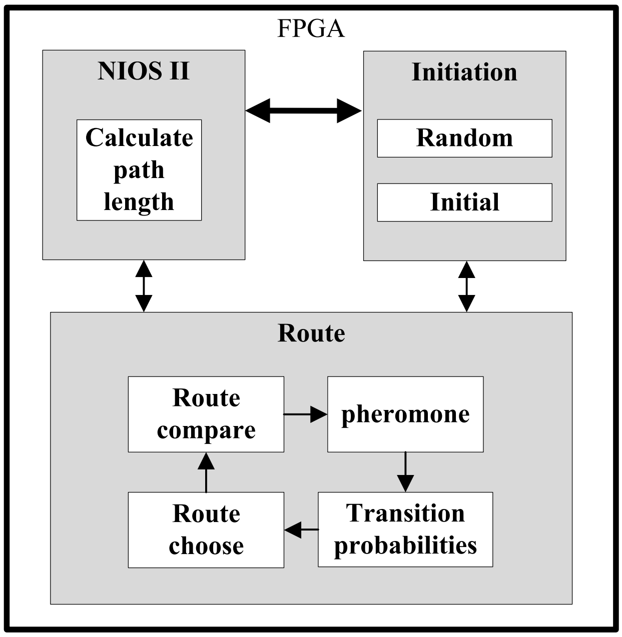

Figure 5. Ant colony algorithm circuit chart

Hardware and software co-design the ACO algorithm circuit architecture is shown in Figure 5. The initiation block will initial the signals and data, and send the data to the NIOS II processor as well as the path selection module. Then, the path selection module will begin the path of choice. Finally, when the path selection is completed after the path data output to the NIOS II processor. The NIOS II processor will calculate the path length. Finally, the path length will send back to the path selection module to inform its update signal to update the pheromones.

Below we will introduce ant colony algorithm circuit of each module, the initialization module of the hardware side, the path selection module and software side of computing the path length. Respectively function as follows:

A. Initiation module

This module is used to initiate the parameterized variables which the required number of iterations, the number of ants, as well as the definition of ants, ρ value(pheromones disappeared coefficient), α value(pheromone parameter), β value(absolute distance parameters), and $\eta_{ij}(t)$. Random number circuit is used as a random value provided to the path selection module to use.

B. Route module

In this module, a pheromone matrix circuit will be built which contains the pheromone concentration values $\tau_{ij}(t)$ of all paths between nodes, the intention in the use of look-up table to increase our speed of operation. If the module receives a update signal then a pheromone update function will be started. The transition probability rule circuit is used to calculate the pheromone matrix. This module achieves the equation (1) to calculate the probability of the conversion rules between each node. The route choose circuit receives a probability value and selects the next path to go. The route compare circuit to decide and update the best path and create a matrix to store the best path.

C. Nios II processor

The all path length and the best path are calculated by this processor. The all calculate results will send to the route circuit module and updated data.

V. EXPERIMENTS

The hardware and software co-design method for ACO algorithm design have realized the pure software design and hardware circuit. In this paper, the simulation parameters are set the total nodes to 31, ants to 31, α equal 1, β equal 5, ρ equal 0.8, iteration 200 times. The experimental results are shown in the following.

Design approach	Initialization time (S)	Path choose time (s)
software design	0.7380	0.4369
Hardware/software design	0.01440	0.0549

REFERENCES

[1] Altera Corporation, SOPC Builder User Guide, 2003.

[2] Altera Corporation, Designing With Nios & SOPC Builder, 2003.

[3] Shih-An Li, Chen-Chien Hsu*, Ching-Chang Wong, and Chia-Jun Yu, “Hardware/Software Co-design for Particle Swarm Optimization Algorithm,” submitted to Information Sciences, Aug. 15, 2009. (SCI)

[4] Dorigo M., V. Maniezzo, and A. Colorni, “Positive feedback as a searchstrategy,” Technical Report no.1-016 Revised, Dip. Elettronica, Politecnico di Milano,1991a.

[5] Dorigo M., V. Maniezzo, and A. Colorni, “The ant system: An autocatalytic optimizingprocess,” Technical Report no. 91-016 Revised, Politecnico di Milano, Italy,1991b.

[6] Dorigo M., V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol.26, no. 1, pp.29-41, 1996.

[7] Dorigo M., and L.M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Transitions on Evolutionary Computation*, vol. 1, no. 1, pp.53-66, 1997a.

[8] Dorigo M., and L.M. Gambardella, “Ant colonies for the traveling salesman problem,” BioSystems, 43, 73-81,1997b.

[9] L.M. Gambardella, E. Taillard, and Dorigo M., “Ant colonies for quadratic assignment problem, ” *Journal of the Operational Research Society*, vol.50, no.2, pp.167-176, 1999.