

A Couple of Algorithms for K-Coverage Problem in Visual Sensor Networks

Mohammad Amin Zare Soltani¹, Abolfazl Toroghi Haghighat², Hassan Rashidi³ and Tahere Godarzvand Chegini⁴

^{1,2}Department of Computer, Qazvin Branch, Islamic Azad University, Qazvin, Iran

³Department of Computer, Allameh Tabataba'i University, Tehran, Iran

⁴Department of Computer, Amir Kabir University of Technology, Tehran, Iran

¹ AminSoltani@qiau.ac.ir, ² Haghighat@qiau.ac.ir, ³ Hrashi@atu.ac.ir ⁴ T.chegini@aut.ac.ir

Abstract— Visual sensor networks (VSNs) are special kind of WSNs employed for monitoring the environment visually and have many military and non-military applications. The problem of covering known location targets in an area with at least k sensors in the networks while keeping energy consumption minimum, is an NP-hard problem. There are several algorithms to solve the problem. This paper does a simulation and compares the performance of the two algorithms, particle swarm optimization (PSO) and harmony search (HS). The first objective is to find a solution that satisfies the k -coverage constraint and the second one is to find optimal solution that forces the minimum degree of rotations in all sensors. The results obtained from the simulations show that HS is faster than PSO and it has a significant convergence rate to reach the optimal solution.

Keywords- Visual Sensor Networks, K-Coverage, Harmony Search, Particle Swarm Optimization

1. Introduction

Wireless sensor networks are a rapidly growing area for research and commercial development [1]. Wireless sensor networks are networks of sensors that cooperate to monitor an environment or some targets. WSNs have challenges like maximum coverage [2] and energy efficiency that should be considered in design. Design problems of WSNs are often of NP-hard order that need to be solved by special kinds of algorithms like metaheuristic methods.

Coverage with energy efficiency is one of main designing problems in WSNs field. The Problem can be solved by employing evolutionary algorithms such as genetic algorithm, ant colony, particle swarm optimization (PSO) or harmony search (HS) algorithms. Lots of papers have offered different solutions for many coverage problems by using the mentioned techniques. Reference [3] is a survey on employing PSO in solving WSN issues such as optimal deployment. Harmony Search is a meta-heuristic algorithm which imitates the improvisation process of a musician to find the perfect state of harmony [4,5]. This algorithm searches for optimal solution vectors using a stochastic derivative [6].

This paper does a simulation and compares the performance of the two algorithms, particle swarm optimization (PSO) and harmony search (HS). The first objective is to find a solution that satisfies the k -coverage constraint and the second one is to find optimal solution that forces the minimum degree of rotations in all sensors. The results obtained from the simulations show that HS is faster than PSO and it has a significant convergence rate to reach the optimal solution.

2. Problem Definition

Visual sensor networks (VSNs) are special kind of WSNs employed for monitoring the environment visually. Consider an area including camera nodes stochastically distributed. The first objective is monitoring existing targets with given locations in a way that each one is covered by at least K sensor. Sensors can be rotated to cover any direction. Sensor rotation consumes energy. So the second objective is to minimize the sum of sensor rotations to keep the energy consumption minimum. Field of view (FOV) and range of all sensors is the same. Each sensor can sense any number of targets in its field of view. The rotation direction can be clockwise or counterclockwise. In brief the problem is a k-coverage problem with minimum sensor rotation.

3. Pso and Hs Metaheuristic Algorithms

PSO and HS are two algorithms we decided to weigh against their performance and evaluate their benefits. PSO has proved its power and speed in solving many optimization problems. HS is also a more recently developed algorithm that offers significant benefits.

Particle swarm optimization algorithm searches the space of an objective function by adjusting the trajectories of individual agents, called particles, as these trajectories form piecewise paths in a quasi-stochastic manner. The movement of a swarming particle consists of two major components: a stochastic component and a deterministic component. Each particle is attracted toward the position of the current global best g^* and its own best location x^*_i in history, while at the same time it has a tendency to move randomly [7].

Harmony search is an optimization algorithm that has an initial population of solutions like the genetic algorithm (GA). HS uses three operations to generate new solutions in each iteration. The operations are: usage of harmony memory, pitch adjusting, and randomization. The usage of harmony memory is similar to choose the best fit individuals in the genetic algorithms. This will ensure the best solutions will be carried over to the new population. Pitch adjustment is similar to the mutation operator in genetic algorithms. The use of randomization can drive the system further to explore various regions with high solution diversity so as to find the global optimality. While in GA all the population of solutions replaced by the new solutions in each iteration, in HS algorithm just one of the individuals (the worst one) will be replaced by the new generated solution.

4. Solution for Min-Rotation Coverage Problem

PSO and HS search the solution space iteratively and randomly. So spending more time will lead to better results. To compare these algorithms fairly, we set the parameters of them in a way that they have roughly equal execution time. The parameter that has most significant effect on execution time of algorithms is the number of evaluation function calls which is as follows:

The number of evaluation function calls in PSO = Population size * Number of iterations

The number of evaluation function calls in HS = Population size + Number of iterations

We used different population size and different number of iterations for both algorithms in a manner that the number of evaluation function calls are the same and equal to a predetermined value (100,000 times). Table 1 shows the parameter values of PSO and HS algorithms that led to our best experimental results.

TABLE I. PARAMETER VALUES OF PSO AND HS ALGORITHMS

PSO	HS
Population size: 100 Iterations: 1000 $\alpha_0=2, \gamma=1-1/\text{iterations}, \alpha = \alpha_0\gamma^t$ $\beta_0=2, \gamma=1-1/\text{iterations}, \beta = \beta_0\gamma^t$	Population size: 100 Iterations: 99,900 Accepting rate: 0.99 Pitch Adjusting rate: 0.7

Each iteration in PSO leads commonly to a result that is closer to optimal. So the movement of particles in next iterations should be in a way that changes are decreased gradually. This could be achieved by decreasing α and β values gradually during the execution of algorithm. We used $\alpha = \alpha_0\gamma^t$ to decrease α in each iteration. The parameter t is the number of iterations till now and γ is a parameter between 0 and 1. If γ is close to 1, α will be decreased slowly and if γ is close to 0, α will be decreased more rapidly. Changing the value of α

should have a trend towards the best effect on results. In this paper we used $\gamma=1-1/\text{iterations}$ achieved by experience. Decreasing α like this will cause it to have good effect on particles movements in last iterations. Parameter β is also set in the same manner. The pseudo codes of two algorithms are listed in Fig. 1 and Fig. 2.

5. Initialize Population

For both algorithms, the initial population is generated in a same manner. Firstly, the targets that exist in each sensor range are determined and then, each sensor is rotated randomly towards one of the targets in its field.

```
[Initialize HS solutions]
[Evaluate solutions (Fig. 3)]
for i = 1 : numIterations
    % generate new solution x by HS operations
    for j = 1 : numSensors
        if (rand > acceptRate) % new search via
            randomization
                x(j) = [generate a random value between 0
                    and 360];
            else % Harmony Memory Accepting Rate
                x(j) = [select one of the current solutions
                    randomly and return its j-th element]
            if (rand < PARate)
                % Pitch Adjusting in a given range
                pa = 0.5;
                x(j) = x(j) + pa * (rand - 0.5);
            end end end
        % check range of directions in x to be in 0-360
        degree
        [Evaluate the new solution]
        [Replace the worst solution by the new solution if
            it is better]
    end
```

Figure 1. Pseudo code of Harmony Search

```
[Setting the parameters: alpha=2, beta=2]
[Initialize particles]
for i = 1 : numIterations
    [evaluate all particles (Fig. 3)]
    alpha = alpha * (1 - 1/numIterations);
    beta = beta * (1 - 1/numIterations);
    bestParticle = [find the best particle];
    % Move all the particles to new locations
    particles = pso_move(particles, bestParticle, alpha,
        beta);
end
```

Figure 2. Pseudo code of PSO

6. Evaluation Function

Two objectives are aimed in this problem including: k-coverage and minimum sensor rotation. The first objective is essential. Solutions that do not meet the first requirement are not acceptable. In this paper we used an evaluation function with two criteria. Fig. 3 shows the function code written in MATLAB.

```

function value = evaluate(coverage, sumRotations)
% coverage: is a vector that contains the coverage of
each target
% sumRotations: sum of rotations of all sensors (from
their initial
% directions to the changed directions)
minCoverage = min(coverage);
if (minCoverage>=k)
value = 1 + 1/sumRotations; % value>1
else
value = (minCoverage -
(1/numTargets)*sum(coverage==minCoverage)) /
k; % value<1
end

```

Figure 3. Evaluation function that used in PSO and HS

Where coverage is a vector of length numTargets that is the number of targets. The i-th element of this vector shows the number of sensors that see the i-th target. The function checks the primary objective first. If the k-coverage requirement is met, returns a value more than 1, and if not, returns a value less than 1. Thus, the first objective has priority and the second objective will be followed provided that the first objective is completed. If k-coverage requirement is met, the return value is just affected by sum of sensors rotations and if not, the value is just affected by coverage vector.

7. Experimental Results

Before comparing the performance of the algorithms, the population size and number of iterations parameters should be adjusted such that the best result will be achieved by each algorithm. As mentioned before, these parameters need to be set in a way that the number of evaluation function calls for each algorithm be the same that we considered it 100,000 in our experiments. Keeping constant other parameters of algorithms, we tested different values for above two parameters. The results are listed in Tables 2 and 3.

Since both algorithms move randomly towards the optimal solution, we repeated the experiments 10 times for each and recorded the results as two distinct values. The first value shows the percent of successful efforts of algorithms to find the solution that satisfies the k-coverage constraint. The second value is the average of “sum of rotations” parameter for those successful efforts. As can be seen in tables, the best result for PSO is achieved when the population size and number of iterations are 100 and 1000 respectively. Also, the best result for HS is achieved when the population size and number of iterations are 100 and 99,900 respectively.

TABLE II. EFFECT OF POPULATION SIZE AND NUMBER OF ITERATIONS ON PSO

Population	10	100	1000	10000
Iteration	10000	1000	100	10
Meet k-Coverage	100%	100%	100%	100%
Sum of Sensor Rotations	1982	1974	3498	6704

TABLE III. EFFECT OF POPULATION SIZE AND NUMBER OF ITERATIONS ON HS

Population	10	100	1000	10000
Iteration	99990	99900	99000	90000
Meet k-Coverage	100%	100%	100%	100%
Sum of Sensor Rotations	1157	1047	1356	5887

After finding the best values for algorithms parameters, we can compare them fairly. To compare the algorithms we considered an area size of 50*50 including 100 visual sensors and 10 targets that randomly distributed in the area. The FOV of visual sensors is assumed 90 degrees and their initial directions assigned by random values between 0 to 360 degrees. Different values have been used for K and sensor ranges to do the experiments for various conditions.

Table 4 shows the results. We repeated the algorithms executions 10 times for each parameter set and recorded the results as two values. The first one shows the percent of successful attempts of algorithm to meet the k-coverage requirement. The second value shows the mean for sum of sensor rotations in successful k-coverage attempts.

Regarding to Table 4, the HS algorithm successfully complete k-coverage objective in all experiments, while the PSO failed to achieve 100% for this criteria in some situations. For the parameter set of Range=10 and K=10, just 2 attempts of 10 runs of PSO successfully fulfilled the objective. The other difficult situation for PSO was the last one i.e. Range=20 and k=20. In that situation, neither of 10 attempts of this algorithm found a coverage better than k=18. The harmony search algorithm also outperforms particle swarm optimization in the second objective that was minimizing sum of sensors rotations.

TABLE IV. RESULT FOR PSO AND HS

		Range=5	Range=10		Range=20			
		K=1	K=2	K=5	K=10	K=5	K=10	K=20
PSO	Meet k-Coverage	100%	100%	100%	20%	100%	100%	0%
	Sum of Sensors Rotations	733	406	1974	6491	265	1428	Best K=18
HS	Meet k-Coverage	100%	100%	100%	100%	100%	100%	100%
	Sum of Sensors Rotations	383	104	1047	4450	18	528	4614

8. Convergence Speed

Harmony search algorithm rapidly completes the first objective in first iterations and does more attempt to minimize the sum of sensors rotations in next iterations. Particle swarm optimization algorithm also show a similar behavior in cases that we used low values for K. Here, we will compare the convergence trend of these algorithms for the sum of sensors rotations.

While the PSO algorithm calls the evaluation function by the number of its population size, HS calls this function just one time for each iteration. Therefore, the time consumption of 100 HS iterations is equivalent to one PSO iteration with a population size of 100. In order to compare the convergence trend of HS with PSO convergence trend, we used the number of evaluation function calls instead of the number of iterations. Fig. 4 shows the convergence trend of both algorithms for Range=10 and k=5. According to this figure, HS has faster convergence rate than PSO.

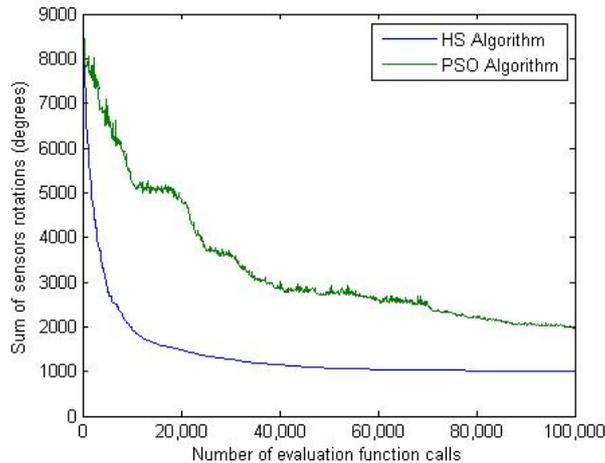


Figure 1. Convergence trend of HS versus PSO

9. Conclusion and Future Works

The problem of covering known location targets in area with at least k sensors while keeping energy consumption minimum, is an NP-hard problem used in this paper to compare the performance of two algorithms of particle swarm optimization (PSO) and harmony search (HS). The first objective was to find a

solution that satisfies the k-coverage constraint and the second one was to find optimal solution that forces the minimum degree of rotations in all sensors. Multiple K value and sensor ranges are assumed to repeat the experiments in different parameter sets. As the results showed, Harmony Search has more power either in achieving k-coverage or minimizing the sum of sensors rotations. According to results, HS has faster convergence rate than PSO.

10. References

- [1] C. Y. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.
- [2] Mulligan, Raymond, "Coverage in Wireless Sensor Networks: A Survey," *Network Protocols and Algorithms*, ISSN 1943-3581, Vol. 2, No. 2, 2010.
- [3] Raghavendra V. Kulkarni and Ganesh Kumar Venayagamoorthy, "Particle Swarm Optimization in Wireless Sensor Networks: A Brief Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 2, March 2011.
- [4] Nezhad, S.E. et al., "Solving K-Coverage Problem in Wireless Sensor Networks Using Improved Harmony Search," International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010.
- [5] Geem, Z. W., Kim, J. H., and Loganathan, G. V., "A New Heuristic Optimization Algorithm: Harmony Search", *Simulation*, 76(2), pp. 60-68, 2001.
- [6] Zong Woo Geem and Justin C. Williams, "Harmony Search and Ecological Optimization", *International Journal of Energy and Environment*, Issue 2, Volume 1, 2007.
- [7] Xin-SheYang, *Engineering Optimization: An Introduction with Metaheuristic Applications*, Hoboken, New Jersey, USA, John Wiley & Sons, 2010.