

# Application and Comparison of Metaheuristic any Colony Optimization and Particle Swarm Optimization Algorithms in Manufacturing Automation

RHYTHM SUREN

NTNU Valgrinda, Inst. For Produksjons-og kvalitetstek.  
7491 Trondheim

**Abstract.** The paper presents an application and comparison of Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) for object recognition problem in a robot material handling system. To enable robust pick and place activity of metalcasted parts by a six axis industrial robot manipulator, it is important that the correct orientation of the parts is input to the manipulator, via the digital image captured by the vision system. This information is then used for orienting the robot gripper to grip the part from a moving conveyor belt. The objective is to find the reference templates on the manufactured parts from the target landscape picture which may contain noise. The Normalized cross-correlation (NCC) function is used as an objection function in the optimization procedure. The ultimate goal is to test improved algorithms that could prove useful in practical manufacturing automation scenarios.

**Keywords.** Foundry Automation, Particle Swarm Optimization, Ant Colony Optimization

## 1. Introduction

The presented work, compares two evolutionary computation techniques Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). The goal of the paper is not to declare one of the techniques as better than the other, but to test their applications after modification to suit the manufacturing scenario discussed, as well as their limitations. The case study is a small-to-medium batch manufacturing foundry and we intend to test the suitability of the algorithms for the purpose of lean workflow and reducing machine starvation in the manufacturing facility.

## 2. Problem description

Recognizing orientation of objects is a challenging task due to constant changes in images in the real world. The most straightforward technique for part orientation recognition is called template matching (2). It is the process of determining the optimal matching between the same scenes taken at different times or under different conditions and the template known according to some similarity measure. (26). In other words, the basic idea is to find a match of the pattern in some part of the landscape image. Normalized Cross Correlation (NCC) is the most robust correlation measure for determining similarity between points in two or more images providing an accurate foundation for motion tracking images (17). This technique has been used on several works. Cole (6) used this technique to reduce the size of a set of images to which new images were compared. Modegi (31) proposed a structured template matching technique for recognizing small objects in satellite images. There are other methods of tracking that do not use NCC, including Gradient Descent Search (GDS) and Active Contour Matching (1).

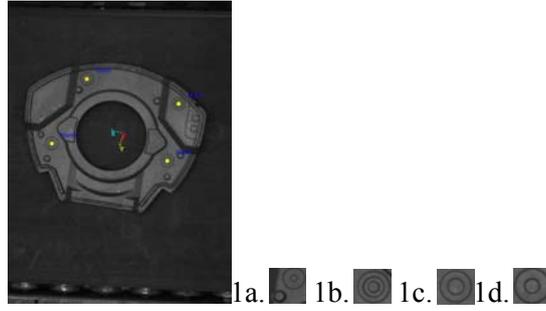


Fig. 1 Image of the sample part on the assembly conveyor belt, as seen from the overhead camera image. 1a 1b 1c 1d  
The templates to be detected on the part to predict its orientation for handling by the robot gripper.

In this work, we want to find a reference image in the target landscape image. When the pattern is found in the target image, its rotation angle is determined. To evaluate a candidate solution, the measure of similarity  $\gamma$  between the reference and target landscape image has been proposed. Several similarity measures have been proposed in the literature, such as mutual information and sum of square of differences between pixels (2)(6). In this work, we used the relation in equation 1, considering the degree of similarity between the images.

$$\gamma = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [F(x+i, y+j) - \bar{F}_{i,j}] \cdot [T(i,j) - \bar{T}]}{\left\{ \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [F(x+i, y+j) - \bar{F}_{i,j}]^2 \cdot \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [T(i,j) - \bar{T}]^2 \right\}^{1/2}} \quad (1)$$

In the equation (1),  $F(x,y)$  is the landscape image,  $\bar{F}_{i,j}$  is the grey-scale average intensity of the captured image in the region coincident with the template image,  $T(x,y)$  represents the template image and  $\bar{T}$  is the average intensity of the template image. We have to address that the dimensions of the matrix  $F$  is  $M \times N$  and the size of the template  $T$  is  $m \times n$ . The maximum value of  $\gamma$  is 1, will say that the match between the landscape and the template is perfect.

### 3. Proposed Methodology

#### 3.1. Ant Colony Optimization

Ant colony optimization was formalized into a metaheuristic for combinatorial optimization problems by Dorigo and co-workers [27], [28]. One can find ACO metaheuristic application to real-world applications mentioned in the literature such as by Price et al. [29], who have applied ACO to an industrial scheduling problem in an aluminum casting center, and by Bautista and Pereira [18], who successfully applied ACO to solve an assembly line balancing problem with multiple objectives and constraints between tasks. Our primary goal in this work is to analyze the manufacturing related application capabilities of ACO, hence in this first investigation we do not use local search.

#### 3.2. Particle Swarm Optimization

The initial ideas on particle swarms of Kennedy and Eberhart were essentially aimed at producing computational intelligence by exploiting simple analogues of social interaction, rather than purely individual cognitive abilities. The first simulations [20] were influenced by Heppner and Grenander's work [16] and involved analogues of bird flocks searching for corn. These soon developed [9][10] into a powerful optimization method— Particle Swarm Optimization (PSO). Because of the characteristics of the flexible manufacturing environment, PSO needs to be discretized. The PSO was modified in order to improve the optimization effect. Therefore, an improved discrete PSO (IDPSO) was applied in this case.

## 4. Results and Comparison

### 4.1. Improved Discrete Particle Swarm Optimization (IDPSO)

#### 4.1.1 With 150 particles

To implement IDPSO, a population size of 150 particles was chosen to provide sufficient diversity into the population taking into account the dimensionality and complexity of the problem. This population size

ensured that the domain was examined in full but at the expense of an increase in execution time. The other parameters of DPSO and IDPSO were:  $c1 = c2 = 2.0$ ,  $\omega = 1.2 - 0.8$  with linearly decreasing, total iteration = 300 and  $V \in [-3, 3]$ .

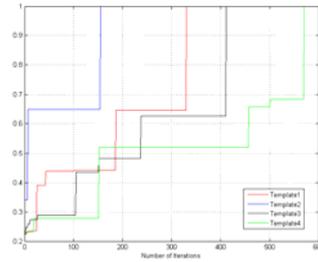


Fig. 9. Testing with 150 particles

The average NCC value of the templates obtained in the experiments was .998 or greater in less than 1000 iterations.

#### 4.1.2 With 500 particles

The average NCC value of the templates obtained in the experiments was .998 or greater in less than 1000 iterations.

### 4.2. Ant Colony Optimization (ACO)

With ACO we chose the following settings  $m = 10, \beta = 2, q_0 = 0.98, \alpha = \rho = 0.1$  and

$\Delta \tau_{ij}^k = \frac{Q}{L_k}$ . (27)(28) The location of the four templates/markers (Figure 11) by the three algorithms is shown

in Figure 12. It took an average 8.86 seconds for ACO to find the four templates, and hence the fastest of the three algorithms. Results showed the limits of robustness of the Bee Colony Algorithm, for different food sources. When compared with the results obtained by a particle swarm algorithm for the same problem, they are generally equivalent. The average time taken by ACO, was the closest match to the robot assembly cell takt time of 9 seconds that would be required to establish a lean workflow and reduce machine starvation at the manufacturing facility.

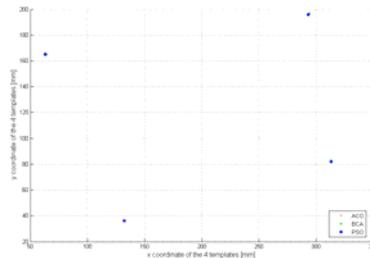


Fig. 11. Coordinates of the 4 templates as solved by ACO, BCA and modified PSO

## 5. Conclusions

In this paper, a variant of Particle Swarm Optimization (PSO) was tested with and Ant Colony Algorithm, and a combination of different strategies, such as generation of scout bees, varying the number of food sources, and explosion of stagnated population. The performance of the PSO is good when dealing with images without scaling factor, but this wasn't necessary for our particular manufacturing case study scenario. The choice of algorithms for a manufacturing assembly scenario could vary with the required tact times in the assembly cell, and the production environment such as vibration, dust etc. It was observed that the computational cost effectiveness of the PSO varies according to the initial starting point. The algorithm can still offer good solutions in the presence of noise within reasonable ranges. Future work will focus on improving the robustness of the algorithm in such situations.

## 6. References

- [1] Brunelli, R. *Template Matching Techniques in Computer Vision: Theory and Practice*. New York: John Wiley & Sons, 2009.
- [2] Chan, T.S., Pak, H.A., Heuristic job allocation in a flexible manufacturing system. *Int J Adv Manuf Technol* 1(2):69-90, 1986.
- [3] Chidambaran, C. and Lopes, H. S., (2010) An Improved Artificial Bee Colony Algorithm for the Object Recognition Problem in Complex Digital Images Using Template Matching, *International Journal of Natural Computing Research*, Vol1, Issue2, pp.54-70.
- [4] Curkovic, P. and Jerbic, B. (2007) Honey-Bees Optimization Algorithm applied to path planning problem, *International Journal of Simulation Modeling*, Vol 3, pp. 154-164
- [5] Eberhart, R. C., and Kennedy, J., (1995) A new optimizer using particles swarm theory. *Proceedings of Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43.
- [6] Eberhart, R. C., Simpson, P. K., and Dobbins, R. W., (1996) *Computational intelligence PC tools*, Boston: Academic Press.
- [7] Evans, H., & Zhang, M. Particle Swarm Optimization for Object Classification. In *Proceedings of the 23<sup>rd</sup> International Conference on Image and Vision Computing* (pp. 1-6), 2007.
- [8] Hackel, S. Dippold, P. The bee colony inspired algorithm (BCiA): a two-stage approach for solving the vehicle routing problem with time windows. In *Proceedings of the 11<sup>th</sup> Genetic and Evolutionary Computation Conference* (pp. 25-32), 2009
- [9] Hii, A. J. H., Hann, C. E., Chase, J. G. and Van Houten, W. E. W., (2006) Fast normalized cross correlation for motion tracking using basis functions. *Computer Methods and Programs in Biomedicine*. Vol. 82, No. 2, pp. 144-156.
- [10] J. Bautista and J. Pereira, "Ant algorithms for assembly line balancing," in *Proc. ANTS2002*, ser. LNCS, M. Dorigo et al., Eds., Springer Verlag, vol. 2463, pp. 65–75, 2002.
- [11] Karaboga, D., Akay, B., A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation*, 214 (1), 108-132, 2009.
- [12] Kennedy, J., and Eberhart, R. C. (1995) Particle swarm optimization. *Proceedings of the IEEE international conference on neural networks IV*, pp. 1942–1948.
- [13] Koren, Y.U.A., *Reconfigurable manufacturing systems: Key to future manufacturing*. *Journal of Intelligent Manufacturing*, 2000. 11(4).
- [14] L. M. Gambardella and M. Dorigo. Solving symmetric and asymmetric TSPs by ant colonies. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, pages 622–627. IEEE Press, Piscataway, NJ, 1996.
- [15] M. Dorigo, V. Maniezzo, and A. Colomi. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1):29–41, 1996.
- [16] Modegi, T. Small object recognition techniques based on structured template matching for high-resolution satellite images. *SICE Annual Conference*, 2168-2173