OPTIMIZATION OF CYCLE TIME OF PALLETIZATION USING ABB ROBOT THROUGH PARTICLE SWARM OPTIMIZATION ALGORITHM AND TRANSLATIONAL APPROACH

Ameet Singh[1], V. Sugumaran[1,*] , G. Balamuruga Mohan Raj[2]

[1]SMBS, VIT University Chennai, Chennai, Tamil Nadu, India, [2]Dept. of Mechanical Engg., Sri Manakula Vinayagar Engg. College, Madagalipet, Puducherry, India.
E-mail: v_sugu@yahoo.com

ABSTRACT

This paper presents an efficient and reliable evolutionary-based approach to solve palletization problem. The proposed approach employs particle swarm optimization (PSO) algorithm for minimization of cycle time of palletizing robot *i.e.* IRB-1600. The operator of algorithm are reviewed, focusing on how each affects search behaviour in the problem space. This paper first analyzes the impact of the parameters *i.e.* population size and maximum run on the performance of the particle swarm optimizer. Numerical results show that the different algorithm perform optimally for the tested instances in a reasonable computational time. The proposed approach results have been compared to those that reported in the literature recently. The results are promising and show the effectiveness and robustness of the proposed approach.

Keywords: Cycle time, Palletization, Particle swarm optimization, IRB-1600, Population size, Maximum run.

1. INTRODUCTION

In the past two decades, the problem of minimizing the cycle time have received much attention. It is of current interest of many utilities and it has been marked as one of the most operational needs. A wide variety of optimization techniques have been applied in solving the minimization of cycle time problems (Tewolde, 2002). Robotic system plays an important role in automated manufacturing industries. There are various types of robots. In this work, we are interested in a robotic system which realizes pick and place operations. Recently, robotic pick and place operations have been widely applied on automated equipments and press fabrication industries. Two ways of palletization are introduced in the paper *i.e.* online programming and offline programming (OLP). As a tool, robotstudio is used as an interface which is used here to program the robot, as it uses the exact replica of the real robot controller. Overall safety and risk of collision are the main constraints during offline programming. However, it is advantageous to use robotstudio software tool. The programmer discovers his mistakes on PC instead of on the shop floor. One requires 3-D cad models of both parts and apparatuses which will be utilized as a part of cell. These 3-D cad models of parts can be made in any solid modeling software like solidworks or cattia (Andrew, 2012). There are various processes through which products are manufactured. Manual processes involved are been replaced with desired industrial robot and an automated workline is designed for the manufacturing process. (Mirlind Bruqi et al., 2013) introduced the power of simulation tools for solving industrial problems in the paper (Mirlind, 2013). The shift from conventional mass production has accelerated in recent years. In response to continuously varying customer requirements, products are being manufactured in small batches, each with custom features. Taking into consideration the complexity of the manufacturing systems, the performance of the industrial manipulators can be improved by using optimization techniques (Fogel, 1994) and offline programming processes (EberharL 1996) that drives the end product to have high quality and low cost. The optimization of robotic arm trajectory is a frequent design problem. Because of the complexity of the task in the past, many of the proposed approaches entailed only a sub optional solution. Due to that reason, previously, several authors have used evolutionary algorithms (Joshuva and Sugumaran, 2017).

Different from traditional search algori-thms, evolutionary computation techniques work on a population of potential solutions (points) of the search space. Through cooperation and competition among the potential solutions, these techniques often can find optimal solution more quickly when applied to complex optimization problems. The most commonly used population- based evolutionary computation techniques are motivated from the evolution of nature. Four well-known examples are genetic algorithms, evolutionary programming, evolutionary strategies and genetic programming. Different from these evolution motivated evolutionary computation techniques, a new evolutionary computation technique, called particle swarm optimization (PSO), is motivated from the simulation of social behavior.

PSO was originally designed and developed by Eberhart and Kennedy.The proposed approach has been validated on a real life setup, involving a 6-DOFs (degrees of freedom) industrial robot manipulator when performing palletizing task. The obtained results are promising and show the effectiveness of the proposed strategy.

2. Design Work: Palletizing path planning in manufacturing industries is processed in two ways:

- *2.1 Offline and online Programming:* The offline programming is processed using robot studio software where the complete installation of the cell is desired in software. The set of coordinates x, y and z- axis for pallets, sheets and boxes respectively are calculated by using offline simulation in robot studio. The online programming is implemented after the completion of offline programming which is taken as a reference for installation. The robot coordinates *i.e.* (x = 0, y = 0 and z = 0) with reference to other work objects namely, pallet, sheet and in feeder. The electrical conections for powering up the PLC controller and robot controller are to be connected within the reach of operator to operate in critical solutions. The complete station is enclosed in a cell, the offline RAPID program is dumped into flex pendant which is used to operate robot in order to pick and place work objects. The things that have to be considered while installing the station are:
- Height and position of robot
- Wire feeder
- Vacuum gripper
- Reachability to work object

*2.2 Palletizing Power pac: T*he palletizing power pac (PzPP) is a robot studio solution for offline programming and simulation of palletizing application, minimum system requirements needed for palletizing.

3. Procedure: Offline program is simulated in robot studio, the complete station layout is designed and installed, the installation is done with proper positioning of robot and required equipment in positions as mentioned in layout. The main work objects are:

- In feeder
- Pallet feeder
- Sheet feeder

4. Optimization: The objective of optimization is to define the optimal relative position between a robotic manipulator and work objects. Here, for this robotic problem optimization is done using particle swarm optimization algorithm (PSO) tool.

4.1 Optimization using Particle swarm optimization algorithm: Particle swarm optimization is a swarm- intelligence based approximate non-deterministic optimization technique. It maintains multiple potential solutions at one time. During each iteration of the algorithm, each solution is evaluated by an objective function to determine its fitness. Each solution is represented by a particle in the fitness landscape (search space). The particles "fly" or "swarm" through the search space to find the maximum value returned by the objective function. Each particle maintains:

- Position in the search space (solution and fitness)
- Velocity
- Individual best position

In addition, the swarm maintains its global best position. The PSO algorithm consists of just three steps:

1. Evaluate fitness of each particle
2. Update individual and global bests
3. Update velocity and position of each particle

5.2 Model parameters

C.T. - Cycle time of palletizing process
IA1 - Cycle time of an in feeder
PA2 - Cycle time of a pallet feeder
SA3 - Cycle time of a sheet feeder
Ix1, Iy1,Iz1 - x, y and z - coordinates of in feeder respectively
Px2, Py2, Pz2 - x, y and z - coordinates of pallet feeder respectively
Sx3, Sy3, Sz3 - x, y and z - axis of sheet feeder respectively
N - Population size in genetic algorithm
Pc - Crossover rate of genetic algorithm method
Pm - Mutation rate of genetic algorithm method

5.3 Mathematical Formulation

An objective function is developed to minimize the cycle time of palletization process. The objective function has a non - linear form, because the feeder positions in workspace affects the cycle time, which is also known as work object coordinates.

Minimize $C.T. = IA_1 + PA_2 + SA_3.$ (1)

subject to following bounds

$280\ mm \leq Ix_1 \geq 1188\ mm,\ 463\ mm \leq Iy_1 \geq 1150\ mm,\ 50\ mm \leq Iz_1 \geq 630\ mm.$ (2)

Bound (2) represents the range of values for x, y and z - coordinates of in feeder respectively.

$400\ mm \leq Px_2 \geq 1210\ mm,\ -580\ mm \leq Py_2 \geq -1220\ mm,\ 0\ mm \leq Pz_2 \geq 470\ mm.$ (3)

Bound (3) represents the range of values for x, y and z - coordinates of pallet feeder respectively.

$-350\ mm \leq Sx_3 \geq -750\ mm,\ -450\ mm \leq Sy_3 \geq -1050\ mm,\ 0\ mm \leq Sz_3 \geq 510\ mm$ (4)

Bound (4) represents the range of values for x, y and z - coordinates of sheet feeder respectively.
Velocity of robot = *10000 mm/s.*

5. RESULTS AND DISCUSSION

Cycle time reduction involves several steps and it can be achieved by following methods

- Speed control
- Reducing the air position
- Path position planning
- Fine tuning
- Configuration settings

5.1 Optimization using translational approach for position of work objects

The values of x, y and z axes for all four feeders work objects after optimization are:
In feeder: *(900, 463, 150) mm*
Pallet feeder: *(800, -600, 100) mm*
Sheet feeder: *(-400, -650, 100) mm*

5.2Optimization using particle swarm optimization algorithm tool: PSO is initialized with a group of random particles and then searches for optimal by updating generations. Particles move through the solution space, and are evaluated according to some fitness criterion after each time step. In every iteration, each particle is updated by follow-ing two "best" values.

- The first one is the best solution (fitness) it has achieved so far (the fitness value is also stored). The value is called pbest
- Another best value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population.

Following are the plots which are tracked by particle swarm optimizer:

- *Population size:*

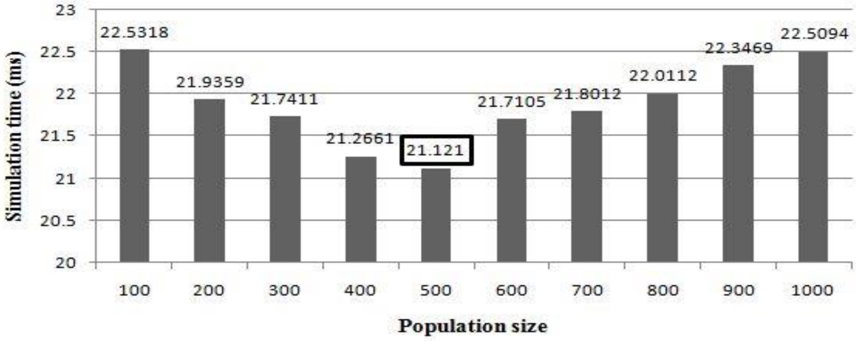

Figure 3: Population Size vs. Simulation time

- *Maximum run:*

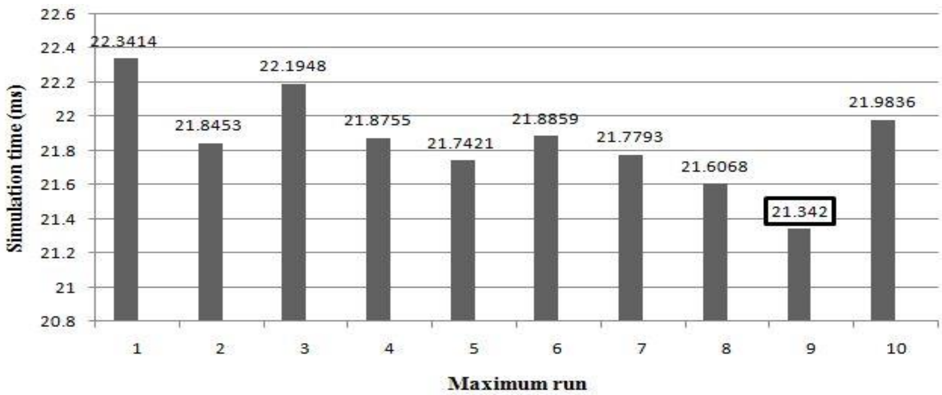

Figure 4: Maximum run vs. Simulation time

- *Maximum number of iterations:*

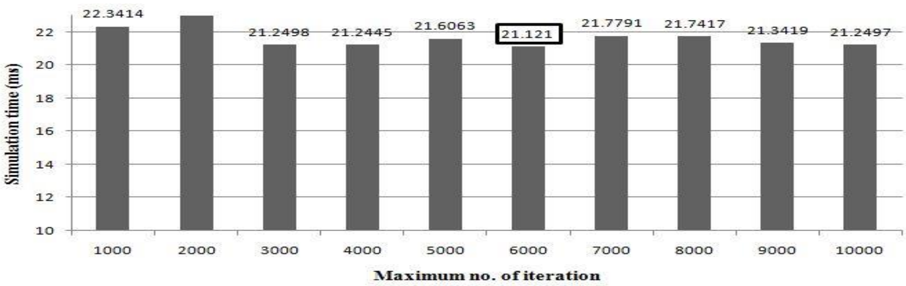

Figure 5: Maximum no. of iteration vs. simulation time

- *Best point:*

It plots the best location to date as shown in Figure 6.

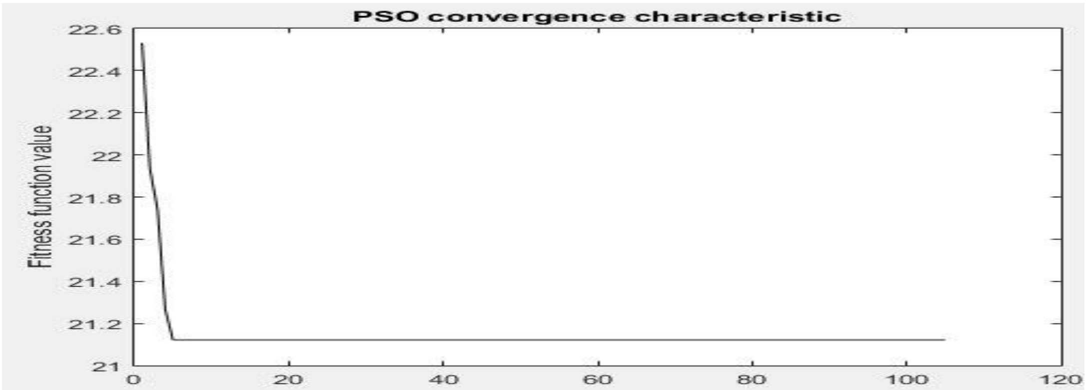

Figure 6: Best point or PSO convergence characteristics

The cycle time of *21.12 milliseconds* is the nearest optimal solution provided by the algorithm with corresponding model parameters and hybrid function parameters as it works as an alternate solver that runs at specified times.

7. Conclusion

In food packaging industries and other manufacturing industries the palletizing process is mostly online. Time taken for installation and teaching the path for palletizing directly over a station results in the reduction of cycle time and increase in productivity. Hence, path generation of any product using offline programming, would increase the productivity and reduce the cycle time as discussed earlier in results and discussion section i.e. *27.4 milliseconds* reduced to *23.4 milliseconds.* In conclusion, PSO approach is found to be an effective and efficient method for determining near optimal solution for palletizing process. PSO approach have a least cycle time of *21.1210 milliseconds.* Considering the future research work, obstacle avoidance can be taken into account by extending the proposed algorithm.

References

Andrew Swary, Offline Programming and Simulation in Robotic Welding Applications Speeds Up programming Time, Reduces Robot Down time, Practical Welding Today March 9, (2012).

EberharL R.C., R.W. Dobbins and P.K. Simpson, Computational Intelligence PC Tools, Boston: Academic Press (1996).

Fogel, L.J., Evolutionary programming in perspective: the top-down view, in Computational Intelligence: Imitating Life, J.M. Zurada, R.J. Marks II, and C.J. Robinson, Eds., IEEE Press, Piscataway, NJ (1994).

Joshuva A. and V. Sugumaran, A data driven approach for condition monitoring of wind turbine blade using vibration signals through best-first tree algorithm and functional trees algorithm: A comparative study. ISA transactions. 2017 Mar 31; 67:160-72 (2017)

Mirlind Bruqi, optimizing of flexible palletizing lines using simulation tools, Trends in the development of machinery and associated technology, TMT, 24(9): (2013).

Tewolde, G.S. and W. Sheng, Robot path integration in manufacturing process; Genetic algorithm versus Ant colony optimization. IEEE Transaction on Systems, Mass and Cybernetics 38(2): 278- 287 (2002).