Improvement of Particle Swarm Optimization in Multi-Robot Trajectory Motion Coordination

Nyaiak Tien Tang 1, Jun Wei Lee 1, Kit Guan Lim 1, Min Keng Tan 1, Tianlei Wang 2, Kenneth Tze Kin Teo 1,*

1 Modelling, Simulation & Computing Laboratory | Artificial Intelligence Research Unit
Faculty of Engineering, Universiti Malaysia Sabah
Kota Kinabalu, Malaysia
Email: msclab@ums.edu.my, ktkteo@ieee.org
2 Department of Intelligent Manufacturing
Wuyi University
Jiangmen, China

Abstract — In swarm robotics, motion coordination is being viewed as a whole system consists of multiple robots each with different velocity and path being optimized. This research focus on improve an optimization process called Particle Swarm Optimization (PSO) by altering the fitness function from static to dynamic. PSO has tendency to being trapped in local optima. By employing dynamic acceleration coefficient, the position of particles will be less dependent on the global best position if the global best fitness is found to be saturated in the region and hence release the swarm to another region for searching in other unexplored region. Nominal, dynamic and extreme scenarios were employed to test the performance between PSO and dynamic coefficient PSO. The different on expected optimization time between DCPSO and PSO for nominal condition is DCPSO lead only 0.26 second. While in dynamic condition the different is DCPSO lead 1.09 second. For extreme condition, the different become significant, which DCPSO lead 3.58 second.

Keywords — swarm robotics, particle swarm optimization, acceleration coefficient, motion coordination, global convergence

I. INTRODUCTION

The beginning of motion coordination in robotics research focused on solving the path trajectory problem for a robot to move from a location to another location without colliding to any obstacle. As the amount of robots being controlled increases, the traffic of robots become increasingly congested, the research focus of the motion coordination on swarm robotics no longer remain in solving path trajectory problem on single robot, but included dynamics of all robots within the system.

As the dynamics being included into the motion coordination, this type of planning is called Kino-Dynamics planning. This type of planning requires higher computation power. The optimum trajectory is normally computed by using metaheuristics optimization techniques.

Metaheuristic optimization are designed to provide a sufficiently good solution to an optimization problem, especially with incomplete information and limited computation capacity. Metaheuristic optimization approach sample a set of solutions which is too large to be completely sampled, such as motion coordination problem where the situation can be dynamic and hard to define a complete analytical model.

Metaheuristics optimization such as Ant-Colony Optimization, Genetic Algorithm and Particle Swarm Optimization are wildly employed to solve many problems faced in swarm robotics. Each of these methods consist its own advantages. Ant-Colony Optimization which mimics the ant’s food gathering through pheromone trails method, is good in dynamic optimization problem [1]. Genetic Algorithm which mimics the evolution theory where only species with best fitness survive, is good in discrete and combinatorial problem [2]. Particle Swarm Optimization mimics the social behavior of bird flying pattern, is great in scheduling optimization problem [3]. Since motion coordination is scheduling optimization problem, where the schedule of each robot is crucial to the final result. Hence in most research, PSO is chosen as the method to optimize the motion coordination in swarm robotics.

The collision avoidance in conventional swarm robot path trajectory optimization only work for static and fixed obstacle, which does not account for the collision among swarm robot during the path intersections. The reason why the displacement and velocity parameter are excluded from the optimization process is due to when these parameters are included, it greatly increments the calculation time of the optimization process. Furthermore, adding displacement and velocity into consideration increases the dimension of the optimization variable which increase chance for particle to trap in local optima.

PSO presented flaws in compute the optimal swarm robot motion coordination. The first flaw is the particle in PSO tends to trap at the local optima if there are lot of local optimum points. This is due to the velocity of the particle is dependent on the global and personal best position. In the context of swarm robot motion coordination optimization, due to there are several optimal trajectory paths each with several optimal speeds, this problem naturally caused the occurrence of large number of local optimums. It is possible to solve this problem by increasing the population size, but by increasing the population size increases the search area, which directly increase the optimization time.

The second flaw is that large population size on PSO causes slow progress in terms of fitness function, caused slow speed of converge to target location. By reducing the population size will decrease processing time but increase the chance for particles to converge into false global optima. Hence, the number of population should be fixed, but more efficiently utilize each of the particle.
To tackle these two flaws, fix social and cognitive component acceleration coefficient must be modified from constant to variable depending on the iteration number. This modification prevents the particle to be trapped into local optimum. By employing dynamic acceleration coefficient, the position of particles is hypothesized be less dependent on the global best position if the global best fitness is found to be saturated in the region and hence released the swarm to another region for searching in other unexplored regions.

This paper is organized as follows: Section II briefly describes the review of topics such as swarm robotics overall structure, motion coordination and metaheuristic optimization technique particular in PSO. Section III explains the framework of the swarm robotics and alteration of acceleration coefficient function in calculating next iteration velocity of each particle. Section IV presents the experimental result of the swarm motion coordination. Lastly, the conclusion of the DCPSO algorithms for this paper is included.

II. SWARM ROBOTICS FRAMEWORK

Overall Swarm Robotics researches consist of Signal Interface, Control Approach, Mapping and Localization, Object Transportation and Manipulation and Motion Coordination as discussed in [4].

A. Signal Interface

Signal Interface among robot is important when certain form of cooperation is required by the specific task. In the past, there is much debate on what type of level of Signal Interface should be used among the swarm robotics. In the past literatures review, two categories can be distinguished which are implicit as known indirect and explicit as known as direct Signal Interfaces.

B. Control Approach

Iocchi concluded in [5] that there are two main type of control approach which are centralized control and distributed. For centralized control, the system consists of a leader within the system which then only distribute the work to the members among the system. The decision is made by the leader while the member’s only job is to follow the instruction decided by the leader. The system with all fully autonomous robot, where the decisions are made within each robot is called distributed control, this type of control doesn’t require leader.

In 1993, Parker [6] research on pros and cons of the decentralized control. The conclusion shows that in order for swarm robotics to achieve desired emergent group behavior, the control approach should be balance between centralized and distributed control.

C. Mapping and Localization

In swarm robotics, it is important to represent the physical environments surrounding the agents. Mapping allow the data gathered by swarm robots’ on-board sensors to be map into spatial models. Besides mapping, it is important to know the real time location of the agents, where localization technique is used to keep track the exact or absolute location of agent within the generated spatial data. Mapping method can be categorized into two main approaches, which are topological mapping and geometric mapping. Topological mapping approach utilize the encoding of the surrounding’s structural characteristics in order to construct the map [7, 8]. Normally, the topological mapping approach only encode the data gathered surrounding the swarm robotics as within vector form, where each point represents a distinctive target. On the other hand, geometric mapping approach encode the surrounding with much detailed data, such data can be used to represent the floor plan of the map.

D. Motion Coordination

In the past two decades, motion coordination or path-planning is one of the hot topics in swarm robotics research. Most of the research are focusing on providing the path for the given robot according to the environment description, where the obstacles must be avoided and having highest fitness possible simultaneously. Path-planning is normally categorized into global and local path-planning.

Path-planning for local situation, the robot normally plans the path though data sensed by the on-board transducer, where allow the swarm robotics to avoid the obstacle. In other hand, the global path-planning required the precisely defined surrounding model in order to successful plan [8].

There are static environments and dynamics environment type path-planning. The static environments path-planning’s workspace is comprised of stationary obstacle, which the shape of the obstacle is known and fixed [9]. For dynamic environment path-planning, the obstacle can be either stationary or moving, sometimes the environment can be both static and dynamic.

The hardware framework methodology includes the task Signal Interface, control approach, mapping and localization, task allocation and characterization of swarm robotics motion. All data within the system will be stored at google spreadsheet through "Google API". The computer on board swarm robot access the google spreadsheet to gather data.

---

Fig. 1. General flow chart describes about swarm robotics framework.

As shown in Fig. 1, this system consists of main processor, server and swarm robots. Main processor will first obtain input from camera image to detect the location of obstacle and robot. The destination will be pre-set into the system by user.

III. PSO ON MOTION COORDINATION

The metaheuristics approach optimization, PSO was first proposed by [1]. This optimization method able to handle several optimizing variables in different dimension. The theory is based on the social cooperation among each
particle, the particle moves with a velocity toward new position. The motion of particle is inside the search space, where the dimensions are the optimizing parameters, the position is used to determine the fitness value.

A. Principle of Particle Swarm Optimization

The position of each particle, \((x_i[n])\) consist of the information of the optimizing parameter, where each dimension describes one optimizing parameter. The subscript \(i\) indicate the \(i^{th}\) particle where \(i \in [1, n_{\text{max}}]\). The \(n_{\text{max}}\) is the population number. \(n\) is the number of iterations, where each iteration the position of the particle may changes depending on the situation. The particle consist of a velocity, \((v_i[n])\) which allow the position \((x_i[n])\) to move throughout the optimizing plane. The movement of particle can be described in (1).

\[
x_i[n+1] = x_i[n] + v_i[n+1]
\]

Initially, the velocity of each particle is uniformly random distributed. After first iteration, the velocity of each iteration changes according to personal best position, global best position and the inertia factor. The velocity changes can be expressed in (2).

\[
v_i[n+1] = \omega v_i[n] + c_1(P_i[n] - x_i[n]) + c_2(G[n] - x_i[n])
\]

Where \(\omega\) is the inertia coefficient, \(c_1\) is the cognitive component acceleration coefficient, \(c_2\) is the social component acceleration coefficient, \((P_i[n])\) is the personal best position and \((G[n])\) is the global best position. In order for the swarm to converge, the inertia coefficient \(\omega\), which control the last velocity has to be reduced each iteration. The damping factor is added into the model to make sure the particles converge toward the global best position. Hence, the inertia coefficient is expressed in (3).

\[
\omega = \eta \mu \omega_0
\]

Where \(n\) is the number of iterations, \(\omega_0\) is the initial inertia coefficient and \(\mu\) is the damping factor; Where \(\mu\) is normally set between [0.95,0.99].

B. Fitness Function

Fitness of each particle is determined by the fitness function. The fitness function in this research is the simulator of robot motion. Since the goal of this optimization is to determine the shortest distance, but at the same time the robot must complete the task given. Hence, the fitness function is defined as shown in (4), (5) and (6).

\[
d_{\tau}, \quad \tau, \quad \Delta \tau = F(x_i[n])
\]

\[
y = \varepsilon_1 d_{\tau} + \varepsilon_2 \frac{\tau}{\Delta \tau}
\]

\[
\sigma = \frac{1}{y^{\tau}}
\]

Where \(F(x_i[n])\) is the simulation which the input is the optimizing parameter, or the position of the particle, the output of the simulation is the total distance travelled by the swarm robots \(d_{\tau}\), the total assigned linear distance ignoring obstacle \(\tau\) and the distance between respective robot and its assigned task \(\Delta \tau\). The cost index, \(\gamma\) is calculated through (5). Where the \(\varepsilon_1\) is the primary cost weightage, \(\varepsilon_2\) is the secondary cost weightage, which the weightage can be adjusted. The fitness index, \(\sigma\) is just reciprocal of \(\gamma\), where \(\eta\) is accelerate factor of fitness function.

C. Dynamic Acceleration Coefficient Deviations

The flaw noticed on the ordinary PSO is being trap into the local optimum if the optimization plane consists of many local optima. Where if one of the particles enter the local optima, all the others particle will easily be attracted toward the wrong searching area. Hence, there must be an escape mechanism for the particle to jump out from the local optima. The solution is to allow the social acceleration component to be dynamic, where if the global best position, \((G[n])\) does not change for a certain amount of iteration. Indicate that the surrounding search area of the global best position consist no better fitness index. Hence, the others particle should start to search toward another search space.

The social component acceleration coefficient \(c_s\) should be a function of iteration, which decrease as the global best position remain unchanged for a certain iteration. Hence, the model of the social component acceleration coefficient should be inversely proportional to number of iterations since last change in global best position, \(n_{G}\). Besides, if the global best position consists low fitness index overall, the particle should not be attracted much toward it compared to better best position. Therefore, the social component acceleration factor should be proportional to the change in global best fitness, \(\Delta \sigma_{G}\).

Besides weaken the attraction of particle toward the global best position, the personal best position should be strengthened if several iterations have pass and the global fitness is yet to improve. Hence, the cognitive component acceleration factor, \(c_p\) should be proportional to \(n_{G}\). Another benefit on strengthen the cognitive component acceleration factor is to prevent the particle to stop moving due to weaken social component velocity and damped previous velocity. Hence, combining all the model discussed above, the social and cognitive acceleration factor can be expressed as (7), (8) and (9).

\[
n_{G} = \sum_{i=0}^{n} i^{\gamma}
\]

\[
c_1' = \frac{\rho_1 a \Delta \sigma_{G}}{\rho_2 n_{G} c_1}
\]

\[
c_2' = \rho_2 \beta \gamma c_2
\]

Where \(n_{G}\) is the damping coefficient after last change of global best position, \(\gamma\) is the acceleration factor of the damping coefficient, \(c_1\) is the modified social and cognitive component acceleration coefficient respectively. \(\Delta \sigma_{G}\) is the change in global best fitness, \(\rho_{AI}\) is the weight factor for \(\Delta \sigma_{G}\) component, \(\rho_{AC}\) is the weight factor of \(n_{G}\) component for social component acceleration factor and \(\rho_{BC}\) is the weight factor of \(n_{G}\) for cognitive component acceleration factor. The \(\gamma\) directly influence the power of increment on \(n_{G}\), where larger \(\gamma\) value will cause the \(c_2\) decrease exponentially each iteration, while increase \(c_1\) exponentially at the same time. The greater \(\Delta \sigma_{G}\) indicate that the newly found best position consist greater chance on increase the global fitness, hence the greater \(\Delta \sigma_{G}\) should allow the \(c_1\) to increase better and hence attract more particle toward the region.

IV. EFFECT OF DYNAMIC ACCELERATION COEFFICIENT

The model of social and cognitive acceleration factor can be expressed as (8), (9) and (2) as discussed in section III. The modified model of velocity can be expressed in (10).
\[
\begin{align*}
\nu[n+1] &= \omega\nu[n] + \frac{\rho_{A1}}{\rho_{A2} + c_2 r[n]} c_1 (g[n] - x[n]) + \\
&\quad \rho_2 \sum_{i} c_2 (r[n] - x_i[n])
\end{align*}
\]  

(10)

As shown in equations above, \(\rho_{A1}\) and \(\rho_{A2}\) the weights of the dynamic cognitive velocity coefficient, while \(\rho_2\) is the weight of the dynamic social velocity coefficient. The value of \(\rho_{A1}\;\rho_{A2}\) and \(\rho_2\) directly affect the rate of divergence and convergence toward local optimal point. Altering \(\rho_{A1}\) affect the dependency of \(\Delta g\), where the steeper improvement causing greater convergence toward global best position. Altering \(\rho_{A2}\) affect he dependency on \(r[n]\) of social velocity component, where the longer unchanged in global fitness, increase divergence from global best position. Similar to altering \(\rho_B\) affect the dependency on \(g[n]\) of cognitive velocity component, where longer unchanged in global fitness increase convergence toward personal best position.

A. Effect of Global Convergence weightage

By allowing the social velocity coefficient to be altered according to model shown above, the particle should be able to escape the local optima. But if the coefficient is too sensitive, it will cause the particle to unable to converge to any optima since the particle will never able to converge toward global or personal best position. If too insensitive cause the particle unable to escape from the local optima. Therefore, the optimization time required should reduce initially, then over an optimal value, the optimization time will increase again.

As shown in Fig. 2, the data are not consistence due to the uncertainty nature of PSO. Hence, a regression method is used to estimate the relationship between \(\rho_{A1}\) and iteration required. The regression is done through Microsoft Excel regression tool. The 4th order polynomial regression is used to compute the estimated model.

![Fig. 2. Result of \(\rho_{A1}\) versus average iteration required for PSO to achieve minimum fitness.](image)

B. Effect of Local Divergence Weightage

Another similar experiment as part A is carried out to determine the effect on changing \(\rho_{A2}\). The only different is the manipulated variable \(\rho_{A1}\) is switch with \(\rho_{A2}\) in this experiment, while \(\rho_{A1}\) and \(\rho_B\) set as unity. The result is shown in Fig. 3.

As predicted in hypothesis, it can be observed that the average iteration, \(N\), initially decrease as the \(\rho_{A1}\) decrease from 0 to around 0.5, then it increase as \(\rho_{A1}\) increase from 1.50 to 3.80, then \(E[N]\) saturated at \(\rho_{A2}\) greater than 3.80. Hence, the minimal iteration laying between \(\rho_{A1}\) equal 0.50 to 1.50.

![Fig. 3. Result of \(\rho_{A2}\) versus average iteration required for PSO to achieve minimum fitness.](image)

C. Effect of Local Convergence weightage

Another similar experiment is carried out to determine the effect on changing \(\rho_B\). The variable \(\rho_B\) is altered from 0.02 to 4.98, while \(\rho_{A1}\) and \(\rho_{A2}\) set as unity. The result is shown in Fig. 4. A 6th order polynomial regression is used to compute the estimate model.

![Fig. 4. Result of \(\rho_{B1}\) versus average iteration required for PSO to achieve minimum fitness.](image)

As the effect of each weightage of dynamic parameter shown, each weightage \(\rho_{A1}\), \(\rho_{A2}\), and \(\rho_B\) affect the average optimization iteration differently. Therefore, it is hard to obtain the optimal weightage for dynamic acceleration coefficient, ordinary PSO is employed to determine the best result. The vector gradient can be express as shown in (11).

\[
T_f = \\
\begin{bmatrix}
0.278p_0 - 3.32i + 11.928p_0 - 11.399 \\
1.4752 - 15.0356p_0 + 48.448p_0 - 32.6213 \\
0.7290p_0 - 12.744p_0 + 50.976p_0 - 119.085p_0 + 122.007p_0 - 36.1360
\end{bmatrix}
\]  

(11)

V. ANALYSIS OF PARTICLE SWARM FITNESS FUNCTION

The nominal, dynamic and extreme condition will be chosen as the comparison condition. The dynamic coefficient PSO and ordinary PSO will be employed to perform swarm robotics motion coordination. The gradient descendent method is also employed as a control condition. As gradient descendent method is wildly used in simple multi-dimensional optimization problem.

A. Performance Distribution in Nominal Condition

In nominal condition, there are six static obstacle and two swarm robots. Each robot with two cascaded destinations, the map of the simulation is shown in appendix C2. All three-method described above are employed to compute the
optimal motion, so that both the swarm robots are able to move from initial position to destination. Notice that there will be large amount of different route for the robot to complete the task assigned, hence it indicates that the optimal point which satisfy the minimal fitness will be abundant.

Since there will be a lot of optimal point which satisfy the minimum fitness value, the ability to escape from local optimal is not significant. Because it is highly possible the local optimal is one of the many solutions of motion coordination in this condition. Therefore, the hypothesis is the dynamic coefficient PSO will perform similarly to static coefficient PSO.

The frequency of PSO and DCPSO rise rapidly as optimization time increased from 0 second to around 3 second, then decrease inverse exponentially as optimization time increases. This type of distribution can be model as log-logistic distribution, this type of distribution is used in survival analysis as a parametric model for events whose rate increases initially and decreases later.

For Gradient Descendent Method, the distribution is very similar to normal distribution, as it presented a bell shape distribution of frequency as shown in fig. 5. Hence, it can be model as a normal distribution.

It can be observed that the mode frequency of PSO is 214, located at 3.00 second optimization time. The mode frequency of DCPSO is 216, located at 2.20 second optimization time. The mode frequency of Gradient Descendent Method is 210 located at 15.4 second optimization time. Notice that the mode frequency does not represent the mean value of the distribution because the left skew on log-logistic.

The rapid increase on frequency in Fig. 5 is because in optimization time shorter than 0.3 second is very hard to be done. Hence, at short optimization time the frequency is very low because it is very rare the PSO and DCPSO complete the motion coordination in such short time. But, as the optimization time increase, the chance for PSO and DCPSO increase rapidly, hence the frequency shows rapid increase in the beginning in Fig. 5. As the optimization increases, the frequency decrease inverse exponentially because the chance for PSO and DCPSO decrease rapidly as the optimization time kept increasing. Although it is rare, but it does happen sometime the PSO or DCPSO require long time to complete the motion coordination.

For Gradient Descendent, it is obvious that the distribution is located at far-right hand side of the graph where it took averagely more time to complete motion coordination. For PSO and DCPSO, the different in mode optimization time is 0.8 second. But once again, mode optimization time does not indicate the performance. The performance can be determined through the expected value but not mode of the data

B. Performance Analysis in Dynamic Scenario

In dynamic condition, there are six dynamic obstacles and three swarm robots. The obstacles’ motion will be patrolling motion between two fixed point. Each robot with two cascaded destinations. All three-method described above are employed to compute the optimal motion, so that both the swarm robots are able to move from initial position to destination. Notice that there will be lesser amount of different route for the robot to complete the task assigned in dynamic scenario compared to nominal condition, hence it indicates that the optimal point which satisfy the minimal fitness will be harder to obtain.

From Fig. 6, blue dots are the frequency of dynamic coefficient PSO with respective optimization time, similar to orange dots are result gather from optimization through PSO method, grey dots are result gather from Gradient Descendent Method.

From Fig. 6, blue dots are the frequency of dynamic coefficient PSO with respective optimization time, similar to orange dots are result gather from optimization through PSO method, grey dots are result gather from Gradient Descendent Method. Performance Analysis in Dynamic Scenario The frequency of PSO and DCPSO rise rapidly as optimization time increased from 0 second to around 6.80 second, then decrease inverse exponentially as optimization time increases. Similarly, this type of distribution also modelled as log-logistic distribution.

For Gradient Descendent Method, the distribution is similar to normal distribution, as it presented a bell shape distribution of frequency as shown in Fig. 6. Hence, it can be model as a normal distribution. It can be observed that the mode frequency of PSO is 81, located at 6.80 second optimization time. The mode frequency of DCPSO is 70, located at 7.80 second optimization time. The mode frequency of Gradient Descendent Method is 66 located at 53.60 second optimization time. Notice that the mode frequency does not represent the mean value of the distribution because the left skew on log-logistic.
C. Performance Distribution in Extreme Scenario

In extreme condition, there are six static obstacles, ten dynamic obstacles and two swarm robots. The dynamic obstacles’ motion will be patrolling motion between two fixed point, each obstacle velocity is different. Each robot with two cascaded destinations, the map of the simulation is shown in appendix D2. Both methods described above are employed to compute the optimal motion, so that both the swarm robots are able to move from initial position to destination. Notice that there will be significantly lesser amount of different route for the robot to complete the task assigned in dynamic scenario compared to nominal condition, because the robots are required to avoid static obstacles and dodge moving obstacle at the same time. Hence it indicates that the optimal point which satisfy the minimal fitness will be significantly harder to obtain.

The rapid increase on frequency in Fig. 7 is because in optimization time increased from 0 second to around 17.60 second, then decrease inverse exponentially as optimization time increases similar to Fig. 5 and 6. It can be observed that the mode frequency of PSO is 9, located at 18.00 second optimization time. The mode frequency of DCPSO is 12, located at 16.80 second optimization time. Notice that the mode frequency does not represent the mean value of the distribution because the left skew on log-logistic.

The frequency of PSO and DCPSO rise rapidly as optimization time increased from 0 second to around 17.60 second, then decrease inverse exponentially as optimization time increases similar to Fig. 5 and 6. It can be observed that the mode frequency of PSO is 9, located at 18.00 second optimization time. The mode frequency of DCPSO is 12, located at 16.80 second optimization time. Notice that the mode frequency does not represent the mean value of the distribution because the left skew on log-logistic.

V. Conclusion

Despite all finding and contribution claimed in this project, it is always a vision for the author to provide a research in a different set-up compares to conventional method. The distribution of optimization time for PSO and DCPSO in nominal condition, dynamic scenario and extreme scenario is summarized in Fig. 8.

As shown in Fig. 8, the different on expected optimization time between DCPSO and PSO for nominal condition is DCPSO lead only 0.26 second, while in dynamic condition, the different is DCPSO lead 1.09 second. For extreme condition, the different become significant, which DCPSO lead 3.58 second.

Fig. 7. Graph of distribution of optimization time for PSO and DCPSO in extreme scenario.

Fig. 8. Graph of distribution of optimization time for PSO, DCPSO and gradient descent method in all scenario.

REFERENCES


