Thermodynamic Particle Swarm Optimization for Multi-agent System in Unknown Environment

Kavan Bojappa* and Junsoo Lee*

*Department of Mechanical Engineering, University of South Carolina

South Carolina Honors College
UNIVERSITY OF SOUTH CAROLINA

Particle Swarm Optimization

- Particle Swarm Optimization (PSO)
  - Swarm Intelligence (SI) algorithm [1] to solve optimization problems using particles that represent solutions.
  - Inspired by swarming behavior in nature.

Swarm Behavior in Birds [2] and Drones [3]

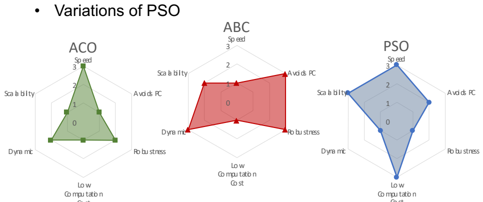
- Each particle uses the best solutions found by itself (pbest) and by the swarm (lbest) to guide itself towards optimal solutions.
- This behavior is controlled by just two equations:

$$v_i^d \leftarrow w v_i^d + c_1 r_1(\text{pbest}_i^d - x_i^d) + c_2 r_2(\text{lbest}_i^d - x_i^d)$$

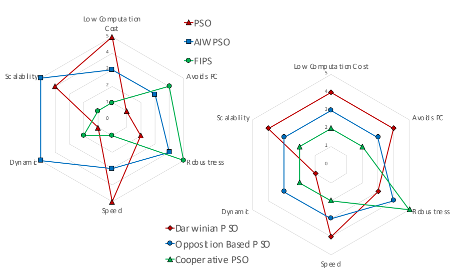
$$x_i^d \leftarrow x_i^d + v_i^d$$

Prior Research

- Comprehensive review of SI and PSO algorithms
  - Compared the performances of SI's
  - Variations of PSO



Behavior Comparison of Swarm Intelligence Algorithms

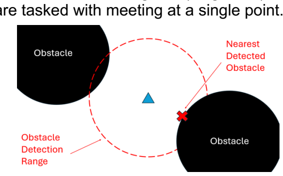


Behavior Comparison of PSO Variants

- Our results show the low computational cost and high scalability of PSO
- We aim to apply agent-based PSO to develop distributed control of an autonomous system.
- This has potential in Swarm Robotics, where many simple robots are used to complete a task faster than a single complex robot would be able to.

Rendezvous Problem

- We will create a framework for autonomous control using PSO by solving the Rendezvous problem of multi-agent system
- Rendezvous Problem
  - Multiple robots (agents) in an unknown environment
  - Each agent can detect nearby obstacles and communicate with some of the other agents (neighbors).
- They are tasked with meeting at a single point.



Representation of Obstacle Detection Capabilities

Thermodynamic PSO (TPSO)

- In TPSO the movement of agents is determined by:

Algorithm 1 PSO Update

$v_x \leftarrow w v_x + c_1 r_1(\text{pbest}_x - x) + c_2 r_2(\text{lbest}_x - x)$
$v_y \leftarrow w v_y + c_1 r_1(\text{pbest}_y - y) + c_2 r_2(\text{lbest}_y - y)$
if $\|v\| > v_{max}$ **then**
  $v = v_{max}$
end if
$x \leftarrow x + v_x$
$y \leftarrow y + v_y$

- The lbest is the average of connected agents, while the pbest is determined by:

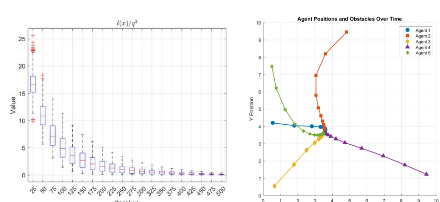
Algorithm 2 pbest Update Algorithm

if Obstacle Detected & $|\theta_{velocity} - \theta_{obstacle}| < 90$ **then**
  if $\theta_{velocity} - \theta_{obstacle} > 0$ **then**
    $\theta_{update} = \theta_{obstacle} + 90$
  else
    $\theta_{update} = \theta_{obstacle} - 90$
  end if
  $mag = min(pbest\_mag, |lbest - position|)$
  $pbest = position + \theta_{update} * mag$
else
  $pbest = position$
end if

- The lbest guides the agents towards one another, while the pbest steers each individual agent around obstacles

Theoretical Analysis

- We created a proof that the algorithm will always converge given infinite time in an obstacle free environment both with and without randomness in the algorithm



Simulation showing convergence for TPSO with randomness (left) and without randomness (right)

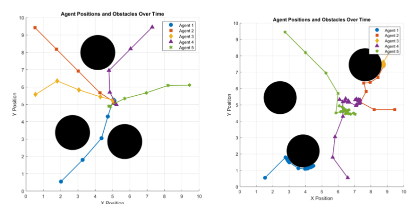
Simulation

- We developed a simulation environment on MATLAB.
- Variable number of circular obstacles.
- Variable number of agents, we used 5 for this research.
- Randomized environment and connectivity matrix.
- We tested a PSO without pbest to our TPSO algorithm in environments with different numbers of obstacles

Number of Obstacles	0	1	2	3
PSO w/o pbest	100	42	19	9
TPSO	100	100	99	97

Success rate (%) for 10,000 simulations at each level of obstacles

- Results show continual effectiveness of the TPSO algorithm even in an obstacle filled environment.
- We tracked the trajectories of the TPSO algorithm in a random successful and unsuccessful run



Sample trajectories of five agent rendezvous in three obstacles environment for successful rendezvous (left) and failed case (right) with TPSO

- Failure occurs in the right figure because Agents 1 and 3 are initialized on opposite sides of the environment, obscured by two obstacles. Agent 2 is drawn towards Agent 3 but is unable to move Agent 3 towards the rest of the swarm.
- This result indicates that the algorithm could perform better removing multiple agents from behind obstacles

Future Study

- Create an analytical framework for TPSO
  - Obstacle avoidance
  - Connectivity condition
- Implement variants to TPSO algorithm to modify parameters
- Investigate real world robotic applications of TPSO algorithm

References

[1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in Proceedings of International Symposium on Micro Machine and Human Science, (Nagoya, Japan), pp. 39–43, 1995

[2]"Marvellous murmurations: why do birds flock together?," *WWT*, Jan. 11, 2021. https://www.wwt.org.uk/news-and-stories/blog/marvellous-murmurations-why-do-birds-flock-together/

[3] "Unleashing the Power of Intelligent Drone Swarms," *www.uh.edu*. https://uh.edu/news-events/stories/2023/june-2023/06052023-becker-drone-swarms-control.php

Contact Information

Kavan Bojappa
(864) 286-1486
Kavanbojappa@gmail.com
Kbojappa@email.sc.edu

Junsoo Lee
(803) 777-2502
junsoo.lee@sc.edu

This work is supported by the Magellan Scholar Award and Honors College Research Grant

Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA