Directed Particle Swarm Optimization Technique for Delivering Nano-robots to Cancer Cells

Doaa Ezzat
Sientific Computing Department.
Faculty of Computer and Information Sciences, Ain Shams University,
Cairo, Egypt
doaa.ezzat@gmail.com

Safaa Amin
Sientific Computing Department. Faculty of Computer and Information Sciences, Ain Shams University,
Cairo, Egypt
ahmed_andeel76@hotmail.com

Howida A. Shedeed
Sientific Computing Department. Faculty of Computer and Information Sciences, Ain Shams University,
Cairo, Egypt
dr_howida@cis.asu.edu.eg

Mohamed F. Tolba
Sientific Computing Department. Faculty of Computer and Information Sciences, Ain Shams University,
Cairo, Egypt
fahmytolba@gmail.com

***Abstract*—Cancer is one of the most killing diseases in this century. The available cancer treatment now is by radio-therapy and chemo-therapy. Both of them cause very harmful side effects on healthy tissues. These side effects lead to decreasing the drug dose and delaying the therapy. Nano-robots can be used to deliver the drug to only tumor cells without harming the healthy tissues. In this paper, a new modification in PSO algorithm is proposed to facilitate Nano-robots delivery to the cancer area. The modified algorithm is Directed PSO (DPSO). It can efficiently deliver all Nano-robots to the target area in a very short time.**

Keywords—Nano-robots; Cancer; PSO; deliver; tumor

I. Introduction

It is observed that the number of cancer deaths increased considerably in the last decades. It is expected that in 2030 this number will be 13.1 million cancer deaths [1]. Until now, there is no proper therapy for treating this disease. The side effects caused by the radio-therapy and chemo-therapy are very harmful on the healthy tissues.

To limit these side effects, scientists proposed using a swarm of Nano-robots to treat cancer. A Nano-robot is a programmable Nano-scale robot. Its size differs from 0.1 to 100 nanometers (a nanometer = 10^{-9} m). A Nano-robot usually has sensors, actuators, power supply, data transmission module and on-board computer. The treatment is done by injecting the Nano-robots in the blood vessel. They can travel through this vessel seeking tumor cells. After reaching their target, they release the drug and destroy only the tumor cells that can be recognized by measuring their pH value. During their journey in the blood vessel, they need to control and optimize their movement using appropriate technique like a swarm intelligence technique.

Particle Swarm Optimization (PSO) algorithm is an optimization technique inspired from the behavior of birds and fish. It was firstly introduced by James Kennedy and Russell Eberhart at 1995 [2]. It uses a swarm of particles spread in the solution space to get the best solution. Each particle represents one solution and has a position and a velocity. The position of each particle is updated using the best position encountered by this particle ($pbest$) and the best position encountered by the whole swarm ($gbest$).

PSO has many advantages like simplicity, insensitivity to scaling, efficiency and the ability to be parallelized. So PSO is a very suitable solution for the problem of delivering a swarm of Nano-robots to the tumor area [3-5]. A lot of modifications have been made to this algorithm in order to enhance its ability to solve this problem and many other problems. Many hybrid techniques that combine PSO with other algorithms like genetic algorithm, simulated annealing, Tabu search, ant colony algorithm and artificial bee colony were also proposed to efficiently solve optimization problems.

However, significant research work is needed to speed up the process of Nano-robots delivery to cancer cells and to make sure that all Nano-robots successfully reach the target area. This is especially because the Nano-robot energy is very limited and this makes the problem of finding the shortest path very significant. Also, the drug dose needed to treat the cancer is related to the number of Nano-robots used in this treatment. So, if some Nano-robots get lost during their journey toward the target region, the drug dose will be not sufficient to treat the complete region. And the remaining cancer cells may reproduce new cells; therefore the treatment process fails.

To overcome these problems, this paper introduces a modification in PSO. This modification makes the algorithm more efficient in delivering Nano-robots to the tumor area. The proposed algorithm is Directed PSO (DPSO) which can deliver the whole swarm of Nano-robots to the target area after a very few iterations.

II. Related Work

Researchers made a lot of work in the area of using Nano-robots in detecting and treating diseases [6-11]. Cavalcanti et al. proposed many Nano-robot architectures for medical purposes [7-13]. They presented new hardware architecture for diagnosing brain aneurysm [7] and diabetes control [8]. In [9], the authors proposed the robot architecture and presented a simulation of Nano-robots with sensing capability for identifying medical target. These Nano-robots can move with six-degree-of-freedom, and each Nano-robot has a sensor that can measure chemical concentration without any communications between robots.

Lewis and Bekey [14] used swarm intelligence and chemical signaling techniques for Nano-robots to solve the problem of tumor removal. Chandrasekaran and Hougen [15] proposed control strategies for swarm intelligence in medical Nano-robots; where each Nano-robot can sense the chemical signal in many directions, make a decision for its next move, and then reach the target area, which has the highest or the lowest chemical concentration. Also, there was no communication between robots.

In 2013, S. Zhang, S. Li and Y. Guo [16] presented a cooperative control scheme for Nano-robots that can sense the pH value to deliver drugs to the cancer area. They proposed a distributed control technique where cooperation is done by robot-to-robot communications. In their control technique, the Nano-robots trace the gradient of pH values, and reach the centre of the tumor which has the lowest pH value.

In 2014, S. Ahmed et al. [3, 17-19] introduced a new control strategy for Nano-robots to deliver drugs to the tumor environment. They proposed a modification in the PSO algorithm (MPSO) to deliver a group of Nano-robots to the tumor area. They also made a modification in the obstacle avoidance algorithm (MOA) to enable Nano-robots to avoid collision with blood cells. Finally, they combined the MPSO with the MOA algorithms to control the movement of the Nano-robots.

In 2016, the authors presented an Improved Bacterial Foraging Optimization Algorithm (IBFOA) to reach and kill cancer cells using cooperation between Nano-robots [4]. In 2017, researchers proposed an innovative computing-inspired bio-detection framework called touchable computing for cancer detection [5]. In the same year, the author used Q-learning algorithm for path optimization [20]. This algorithm reduces the path length travelled by Nano-robots to reach their target.

III. The Proposed Algorithm

To apply PSO on the problem of finding the tumor area by a swarm of Nano-robots, each Nano-robot represents a particle in the search space. The fitness value of a Nano-robot at some position is the pH value sensed by this Nano-robot at this position. The pH value equals 7.4 at normal cells. This value decreases gradually until the Nano-robot reaches the tumor region. In this region, the pH value is less than or equals to 5.7 [16].

The classical PSO algorithm works as follows [2]:

1. Initialize particles with random positions and velocities
2. Evaluate the fitness of each particle
3. For each particle, if its fitness value is better than its $pbest$, then: $pbest$ = its current fitness
4. Update the $gbest$ for the whole swarm
5. Update the velocity V of each particle by this equation:

$$V_i^{t+1} = V_i^t + C_1 R_1 (pbest_i^t - P_i^t) + C_2 R_2 (gbest^t - P_i^t) \quad (1)$$

6. Update the position P of each particle by this equation:

$$P_i^{t+1} = P_i^t + V_i^{t+1} \quad (2)$$

7. Calculate the fitness values
8. If target or maximum number of iterations is reached, then terminate
9. Otherwise, go to step 3

Where P_i^t is the position of particle i at iteration t, V_i^t is the velocity of particle i at iteration t, C_1 and C_2 are constants represent the importance of $pbest$ and $gbest$ respectively, R_1 and R_2 are random numbers.

In the proposed algorithm (DPSO), Nano-robots use the classical PSO to update their positions until one of them reaches the target area. Then the other Nano-robots direct themselves toward the location of the one that reached this target area. Then they move in this direction with constant steps until the whole swarm reaches the target.

The idea of the DPSO comes from the fact that the search space has one target area. We can exploit this fact when at least one Nano-robot reaches the target; it drags all other Nano-robots to its location. By this way, we can considerably decrease the number of iterations needed to deliver all Nano-robots to the tumor area. In other words, we need only one or two PSO iterations for delivering at least one Nano-robot to the target. Then we need some few direct steps to deliver all other Nano-robots to this target. Fig. 1 shows the steps of DPSO.

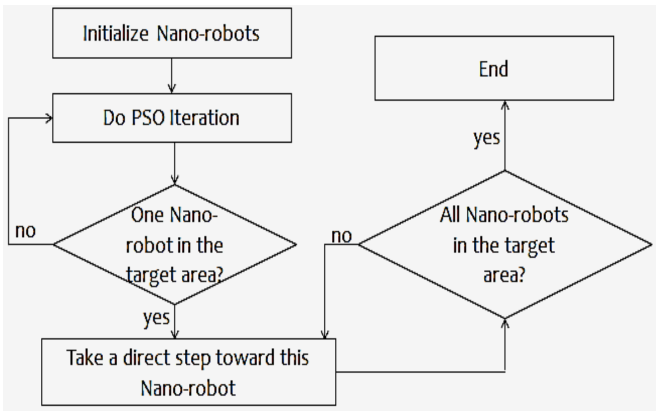

Fig. 1. DPSO Algorithm

Experiments have shown that, in some cases, some Nano-robots get lost when using PSO. DPSO outperforms PSO in this point because it guarantees the reachability of the target area by all Nano-robots. The next section of this paper explains our experimental results in more details.

IV. Simulation and Results

We implemented a simulator using visual C++ to simulate the Nano-robots movements after locating them in the deployment area nearby the tumor region. The simulation is done with 10, 20 and 30 Nano-robots and repeated 10 times each. We applied simulation using four different algorithms. To be unbiased, we used the same starting locations of Nano-robots for the four algorithms in each run. Fig. 2 shows the initial state of the Nano-robots.

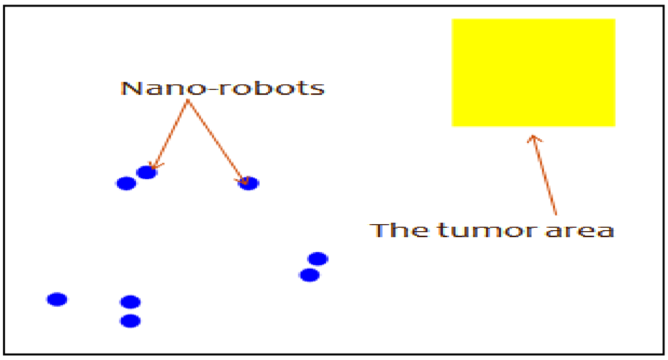

Fig. 2. The Simulator in the Initial State

The four algorithms used in the simulation are as follows:

1. PSO: the classical PSO algorithm (discussed above)
2. BFOA: the Bacterial Foraging Optimization Algorithm, which simulates bacteria. In this algorithm, each Nano-robot moves a unit step toward a random direction, then moves toward the same direction if the fitness value is improved; otherwise it will take another random step [4]
3. MPSO: the Modified PSO algorithm, in which each Nano-robot moves with its initial velocity for a certain period. In this period, Nano-robots just share the information about the target area. After this period, MPSO behaves like the traditional PSO [3]
4. DPSO: The Directed PSO (proposed in this paper)

Firstly, we conducted 10 experiments using 10 Nano-robots, and the averaged results are shown in Table I. Another 10 experiments were conducted using 20 Nano-robots, and the averaged results are shown in Table II. Finally, Table III shows the averaged results of another 10 experiments using 30 Nano-robots. For each algorithm, these three tables show the number of Nano-robots in the target area after 10, 20 and 30 iterations.

TABLE I. THE AVERAGE NUMBER OF DELIVERED NANO-ROBOTS USING 10 NANO-ROBOTS

Number of Iterations	Algorithm			
	PSO	***BFOA***	***MPSO***	***DPSO***
10	7	2	7	9
20	9	3	9	10
30	9	4	9	10

TABLE II. THE AVERAGE NUMBER OF DELIVERED NANO-ROBOTS USING 20 NANO-ROBOTS

Number of Iterations	Algorithm			
	PSO	***BFOA***	***MPSO***	***DPSO***
10	12	3	13	19
20	16	7	17	20
30	17	10	18	20

TABLE III. THE AVERAGE NUMBER OF DELIVERED NANO-ROBOTS USING 30 NANO-ROBOTS

Number of Iterations	Algorithm			
	PSO	***BFOA***	***MPSO***	***DPSO***
10	20	6	20	29
20	24	12	25	30
30	26	17	26	30

From these results, we can conclude that after the first 10 iterations:

1. PSO and MPSO can deliver from 65% to 70 % of the Nano-robots to the tumor area
2. BFOA can deliver from 15% to 20% of the Nano-robots to the tumor area
3. DPSO can almost deliver all the Nano-robots to the tumor area

Also, the average number of iterations, at which all Nano-robots reach the target area using DPSO, is: 10 for 10 Nano-robots, 11 for 20 Nano-robots and 12 for 30 Nano-robots. This means that only one extra iteration is needed for delivering 10 extra Nano-robots.

The experiments showed that, sometimes, some Nano-robots get lost and can't reach the target area during the first 100 iterations. This phenomenon was observed in PSO, BFOA and MPSO experiments, but it was not observed in DPSO experiments. Table IV summarizes the comparison of the four algorithms on the basis of number of experiments without loss in Nano-robots, using 10, 20 and 30 Nano-robots. We assume that, if a Nano-robot doesn't reach the target area before the completion of the first 100 iterations, this Nano-robot will be considered lost.

TABLE IV. EXPERIMENTS WITHOUT LOSS IN NANO-ROBOTS

Number of Nano-robots Used	Algorithm			
	PSO	***BFOA***	***MPSO***	***DPSO***
10	7	0	8	10
20	3	0	8	10
30	2	0	4	10

In Table IV, we can notice that:

1. MPSO outperforms PSO in the capability of conserving Nano-robots
2. BFOA is the worst because some Nano-robots are always lost in each experiment
3. DPSO is the best because none of the Nano-robots is lost in all experiments

Fig. 3 shows a detailed view on one simulation of the four algorithms using 30 Nano-robots. The figure describes the number of the Nano-robots that reached the target after certain number of iterations. We can see in this figure that BFOA is the slowest algorithm in delivering Nano-robots to the target area because of lack of communication between Nano-robots. PSO is faster than BFOA in delivering Nano-robots. MPSO is faster than both BFOA and PSO. The fastest algorithm is DPSO.

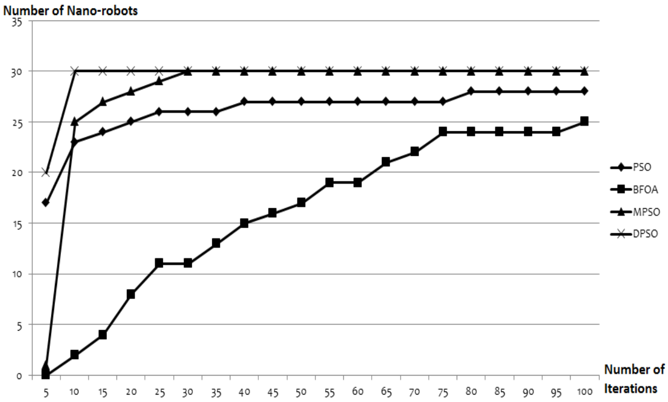

Fig. 3. A Detailed View using 30 Nano-robots

Fig. 4 describes the final state of the simulator after 15 iterations using 10 Nano-robots. This figure shows that when using PSO, 5 Nano-robots can reach the target area. But if we use BFOA, only 3 Nano-robots can reach the target area. MPSO can deliver 8 Nano-robots to the target area. The proposed algorithm (DBSO) can deliver all the 10 Nano-robots to the target area.

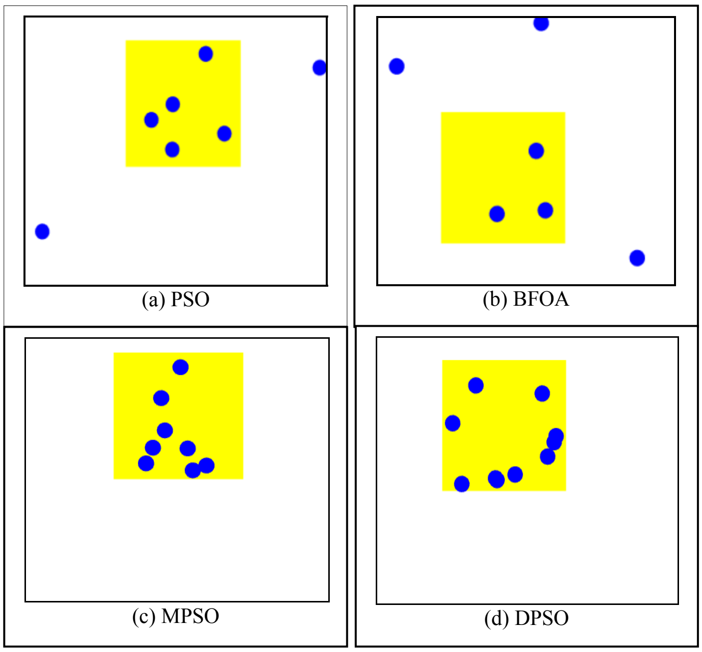

Fig. 4. The Simulator in the Final State

DPSO has only one parameter that controls the direct steps of the algorithm. This parameter is the step size. We should select this parameter carefully because if the step size becomes too large, then the Nano-robot will oscillate around the target area, and if the step size becomes too small, then the algorithm will be very slow.

V. Conclusion

Cancer treatment is very challenging because drugs must be delivered to tumor cells without affecting healthy ones. Using Nano-robots is a perfect way to achieve this goal. There are many proposed algorithms to deliver Nano-robots to the tumor area. This paper proposes a new technique (DPSO) to efficiently deliver Nano-robots to the tumor region. Experiments have proved that DPSO is faster than PSO, BFOA and MPSO in delivering Nano-robots to the target. Also, DPSO guarantees the reachability of the target area by the whole swarm of Nano-robots.

VI. Future Work

We need to work on three points. Firstly: After reaching the target area, Nano-robots must be distributed over this area to sufficiently cover the whole tumor for accurate treatment. Secondly: During Nano-robots journey, they encounter obstacles of the immune system. So, we need to find an efficient way to avoid these obstacles. Thirdly: DPSO is very efficient if used with one target area, but we need to put an appropriate strategy for dealing with two or more target areas.

References

[1] R. Devasena Umai, P. Brindha Devi, R. Thiruchelvi, “A Review on DNA Nanobots - A New Technique for Cancer Treatment,” Asian J Pharm Clin Res, Vol 11, Issue 6, 2018, pp. 61-64.

[2] R. Eberhart and J. Kennedy, “A New Optimizer Using Particle Swarm Theory,” in Proceedings of the Sixth International Symposium on Micro machine and Human Science, pp. 39–43, 1995.

[3] S.Ahmed (2014). Nano-robotics Control For Biomedical Applications (Unpublished doctoral dissertation). Ain Shams University, Cairo, Egypt.

[4] J. Cao, M. Li, H. Wang, L. Huang, and Y. Zhao, “An Improved Bacterial Foraging Algorithm with Cooperative Learning for Eradicating Cancer Cells Using Nanorobots,” in Proceedings of the IEEE International Conference on Robotics and Biomimetics, Qingdao, China, December 2016.

[5] Y. Chen, S. Shi, X. Yao, and T. Nakano, “Touchable Computing: Computing-inspired Bio-detection,” in IEEE TRANSACTIONS ON NANOBIOSCIENCE, 2017.

[6] A. Ummat, G. Sharma, C. Mavroidis, and A. Dubey, Biomedical engineering handbook, chapter Bio-nanorobotics: state of the art and future challenges. CRC Press, London, 2005.

[7] A. Cavalcanti, B. Shirinzadeh, T. Fukuda, and S. Ikeda, “Nanorobot for brain aneurysm,” Int J Robot Res, pp. 558–570, 2009.

[8] A. Cavalcanti, B. Shirinzadeh, and L. Kretly, “Medical nanorobotics for diabetes control,” Nanomedicine 4, pp. 127–138, 2008.

[9] A. Cavalcanti, B. Shirinzadeh, R. Freitas, and T. Hogg, “Nanorobot architecture for medical target identification,” Nanotechnol 19, pp. 1–15, 2008.

[10] A. Cavalcanti and R. Freitas, “Nanorobotics control design: a collective behavior approach for medicine,” IEEE Trans Nanobiosci 4, pp. 133–140, 2005.

[11] A. Cavalcanti, "Assembly automation with evolutionary nanorobots and sensor-based control applied to nanomedicine," IEEE Trans Nanotechnol 2, pp. 82–87, 2003.

[12] R. Freitas, "What is nanomedicine?," Nanomed 1, pp. 2–9, 2005.

[13] A. Cavalcanti and R. Freitas, "Autonomous multi-robot sensor-based cooperation for nanomedicine," Int J Nonlinear Sci Numer Simul 3, pp. 743–746, 2002.

[14] M. Lewis and G. Bekey, "The behavioral self-organization of nanorobots using local rules," in Proceedings of IEEE international conference on intelligent robots and systems, pp. 1333– 1338, 1992.

[15] S. Chandrasekaran and D. Hougen, "Swarm intelligence for cooperation of bio-nano robots using quorum sensing," in Bio micro and nanosystems conference, San Francisco, p. 104, June 2006.

[16] S. Zhang, S. Li, and Y. Guo (2013). Cooperative Control Design for Nano-robots in Drug Delivery. In Yi Guo (Ed.), Selected Topics in Micro/Nano-Robotics for Biomedical Applications (pp. 101– 123). New York, Springer.

[17] S. Ahmed, S. Amin, and T. Alarif, "Simulation for the Motion of Nanorobots in Human Blood Stream Environment," in Proceedings of Scientific Cooperation International Workshops on Electrical and Computer Engineering Subfields, Koc University, ISTANBUL/TURKEY, pp. 70-75, 2014.

[18] S. Ahmed, S. Amin, and T. Alarif, "Efficient Cooperative Control System for pH Sensitive Nanorobots in Drug Delivery," International Journal of Computer Applications, Volume 103 – No.1, 2014.

[19] S. Ahmed, S. Amin, and T. Alarif, "Assessment of Applying Path Planning Technique to Nanorobots in a Human Blood Environment," in Proceedings of the UKSim-AMSS 8th European Modeling Symposium on Mathematical Modeling and Computer Simulation, 2014.

[20] A. Lambe, "Reinforcement Learning for Optimal Path Length of Nanobots Using Dynamic Programming," in Proceedings of IEEE International Conference on Industrial and Information Systems, Peradeniya, Sri Lanka, December 2017.