International Journal of Computational Intelligence and Applications
Vol. 15, No. 2 (2016) 1650006 (25 pages)
© World Scientific Publishing Europe Ltd.
DOI: 10.1142/S1469026816500061

Drug Delivery Based on Swarm Microrobots

Anan Banharnsakun

Computational Intelligence Research Laboratory (CIRLab)
Department of Computer Engineering
Faculty of Engineering at Sriracha
Kasetsart University Sriracha Campus
Chonburi 20230, Thailand
ananb@ieee.org

Tiranee Achalakul

Department of Computer Engineering
King Mongkut's University of Technology Thonburi
Bangkok 10140, Thailand

Romesh C Batra

Department of Engineering Science and Mechanics
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061, USA

Received 12 February 2016
Accepted 21 February 2016
Published 14 June 2016

Advances in the development of technology have led to microrobots applications in medical fields. Drug delivery is one of these applications in which microrobots deliver a pharmaceutical compound to targeted cells. Chemotherapy and its side effects can then be minimized by this method. Two major constraints, however, must be considered: the robot's onboard energy supply and the time needed for drug delivery. Furthermore, a microrobot must avoid biological restricted areas which we treat as obstacles in the path. The main objectives of this work were to find optimal paths to targeted cells and avoid collision with obstacles in the paths under a dynamic environment. In this study, we controlled motion of microrobots based on the concept of swarm intelligence. Artificial Bee Colony (ABC), the Best-so-far ABC, and the Particle Swarm Optimization (PSO) methods were employed to implement the collision detection and the boundary distance detection modules. Forces that drove or resisted blood flow as well as pressure in blood vessels were considered to approximate the effects of the environment on the microrobots. Numerical experiments were conducted using various obstacle environments. The results confirm that the proposed approaches were successful in avoiding obstacles and optimizing the energy consumption used to reach the target.

Keywords: Target finding and obstacle avoiding; drug delivery; microrobots; optimization; swarm intelligence; artificial bee colony (ABC); best-so-far ABC; particle swarm optimization (PSO).

1. Introduction

Advances in the field of nanotechnology and microtechnology have led to the adoption of microrobots in medical applications, specifically in the field of nanomedicine.[1–3] Microrobots are machines or devices whose components are at or close to the micrometer scale. The importance of a microrobot is increasing as today's biomedical technologies require innovative systems to replace difficult procedures.

Many diagnostic tools based on a microrobot have emerged in the past few decades. Dario *et al.*[4] proposed the concept and described the design and fabrication of a new system for colonoscopy, an important procedure for the diagnosis of various pathologies, based on a microrobot capable of propelling semi-autonomously along the colon. Since a low invasive treatment is very important, Ishiyama *et al.*[5] presented magnetic micromachines, a tiny magnetically driven spinning screw, to treat infected tissues or even to burrow into tumors and kill them with heat instead of having a surgery. An Electromagnetic-based Actuation (EMA) system for three-dimensional locomotion and drilling by a microrobot was introduced by Yu *et al.*[6] for a treatment of cardiovascular diseases.

In addition, there are many promising applications[7] for microrobots in the circulatory system including performing targeted drug delivery, removing plaque (rotational atherectomy), destroying blood clots (thrombolysis), acting as stents (a small mesh tube that is used to treat narrow or weak arteries), acting as occlusions to intentionally starve a region of nutrition, and administering therapy for aneurysms.

In a drug delivery process, microrobots generally travel inside a blood vessel with the goal of administering a pharmaceutical compound directly onto targeted cells, i.e., the cancerous or infected cells. The direct delivery process improves the medical efficacy and minimizes risks introduced by chemotherapy.[2] In optimizing the paths, microrobots have to consider the shortest distance between two points while avoiding obstacles in the paths. Obstacles are biological restricted areas.

Many researchers have addressed the above-mentioned issues. To plan trajectories for a mobile robot in partially known environments, Stentz[8] proposed the "D*" algorithm. "D*" resembles the "A*" algorithm, a heuristic search algorithm using a best-first search on a least-cost path, except that it is dynamic in the sense that the arc cosine parameter can be changed during the search process. In this work,[8] the problem space was formulated as a set of states denoting robot locations connected by directional arcs, each of which has an associated cost. The robot started at a particular state and moved across arcs (incurring the cost of traversal) to other states until it reached the goal state. Tsuzuki *et al.*[9] applied the Simulated Annealing (SA), a generic probabilistic method inspired by annealing in metallurgy, as the path-planning algorithm. The path can be represented as linear, Bezier, or interpolated spline trajectories. Each step of the SA algorithm attempted to find the new position of robot by a random solution. The new solution might then be accepted with a probability that depended both on the difference between the corresponding objective values and also on the function of time. The Genetic Algorithm (GA), a heuristic search that mimics the process of natural evolution, was proposed by Tao and Zhang[10] for online and offline path planning based on area coverage.

In recent years, algorithms based on swarm intelligence have been applied to solve the path finding problem. Brand *et al.*[11] worked with the Ant Colony Optimization

(ACO) algorithm, which is inspired by the ant foraging behavior. The ACO could find the shortest and collision free route in a grid network for robot path planning. Obstacles with various shapes and sizes were considered to simulate a dynamic environment in this work. Ganganath *et al.*[12] proposed an ACO-based off-line path planner algorithm that can be realized with most real-world robots, which are kinematically constrained. The Particle Swarm Optimization (PSO) algorithm, mimicking a bird flock or a fish school, was also proposed for path planning by Qin *et al.*,[13] Chen and Li,[14] and Zhang and Li.[15] Moreover, to avoid obstacles, self-organized trajectory planning based on PSO was introduced by Hla *et al.*[16] who simplified the problem by considering only circular shaped obstacles. Zhang *et al.*[17] introduced a constrained multi-objective PSO algorithm to solve the robot path planning problem, in which the robots must evade the uncertain danger sources. PSO-based motion planner was presented by Deepak *et al.*[18] for an autonomous mobile robot in avoiding obstacles and generating feasible trajectories within its unknown environments. To improve the efficiency of microrobots' path finding, the quorum sensing technique, which is the ability of bacteria swarms to communicate and coordinate via molecule signaling, was initiated by Chandrasekaran and Hougen.[19] Hossain and Ferdous[20] explored the bacterial foraging optimization algorithm to the problem of mobile robot navigation in order to determine the shortest feasible path to move from any current position to the target position in an unknown environment with moving obstacles. A cuckoo search-based approach was presented by Mohanty and Parhi[21] for mobile robot navigation in an unknown environment populated by a variety of obstacles.

However, consideration of the distance of the microrobot traveling alone may not be sufficient for the drug delivery process. Two major constraints should be considered. One is that the robot can become inactive and thus not deliver the drug to the target if its energy is not sufficient and the other is that the efficiency of the drug will be dropped if it cannot be delivered to the target cells within a certain time period.[1,3] In this study, we have extended our scope from the previous work[13,15,22] and found the optimal path with an obstacle avoidance mechanism that considers the energy consumption and time taken by the microrobots to travel to the target cell. The Artificial Bee Colony (ABC) algorithm,[23] the Best-so-far ABC algorithm,[24] and the PSO algorithm[25] have been employed for path planning. The techniques for collision detection and the boundary distance detection, as well as their avoidance on both static and dynamic obstacles will be described. In addition, the drag force experienced by a microrobot during its trajectory will be addressed. Forces that drive or resist blood flow as well as pressure in blood vessels will be considered to mimic a more realistic environment in this work.

The paper is organized as follows. Section 2 describes the circulation environment used in our drug delivery framework. Section 3 presents the problem statement. Section 4 provides the microrobots model including microrobots movement, obstacle detection and avoiding techniques. Section 5 proposes the adoption of swarm intelligence algorithms in drug delivery process. Section 6 describes the numerical experiments and discusses simulation results. Section 7 summarizes conclusions of the work.

2. Circulation Environment

In this section, we will provide the background knowledge that was used in our drug delivery framework including the fluid flow concept for blood circulation in arteries and the drag force acting on microrobots when they moved through the fluid to the target cell.

There are two types of fluid flow[26]: laminar and turbulent. While the direction of the laminar flow is parallel to the vessel wall, the direction of the turbulent flow is not parallel to the vessel wall and blood flows in different directions. There is a considerable turbulent flow at the branches of large arteries. However, in small vessels, blood flow is predominantly laminar.

Reynolds number (R_e) is a dimensionless number that gives a measure of the ratio of inertial forces to viscous forces and consequently quantifies the relative importance of these two forces for given flow conditions. Reynolds number can be calculated by

$$R_e = \frac{\text{inertial force}}{\text{visous force}} = \frac{\rho \nu d}{\mu}, \tag{1}$$

where

$\rho =$ the density of the fluid (kg/m^3),
$\nu =$ the average speed of the fluid over the cross section of the tube (m/s),
$d =$ the tube diameter (m),
$\mu =$ the dynamic fluid viscosity (Pa·s or kg/m·s).

For two-dimensional laminar flow, the velocity increases toward the center of a tube. The velocity profile as a function of radius illustrated in Fig. 1 can be described by

$$\nu = -\frac{1}{2\mu}\frac{\Delta P}{L}(h^2 - y^2), \tag{2}$$

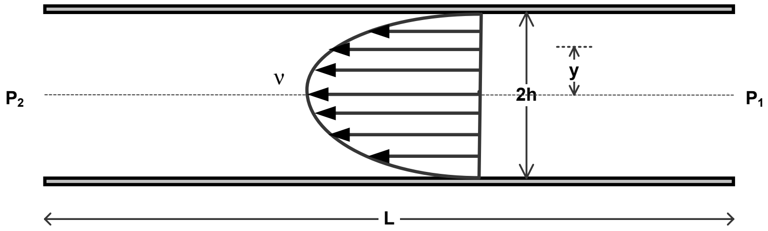

Fig. 1. Velocity profile for laminar flow in circular tube.[26]

where

$L =$ length of the vessel (m),
$\Delta P =$ pressure drop in a segment of blood vessel of length L(Pa),
$h =$ internal radius of the tube (m),
$y =$ radial coordinate of a point (m).

When a microrobot moves through a fluid, it will experience the "drag force" (F_D)[27] as shown in Fig. 2.

This is the force that resists the motion of an object through the fluid. The drag force on a submerged object has two components and can be calculated by using Eq. (3).

- A pressure drag (form drag) F_p:

  The integration of components in the direction of motion of all the pressure forces exerted on the surface of the body.

- A friction drag (surface drag) F_f:

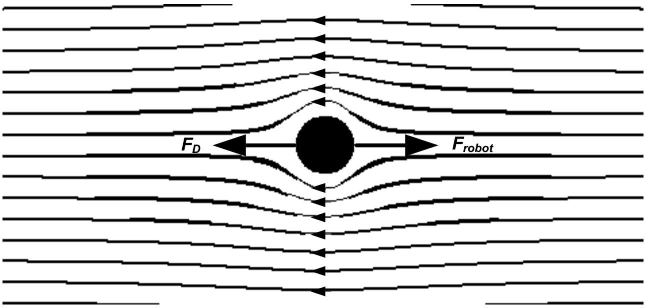

Fig. 2. Drag force on a microrobot.[27]

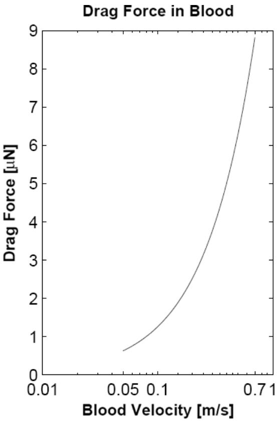

Fig. 3. Drag force in blood.[28]

The friction of the fluid against the surface of the object that is moving through it:

$$F_D = F_p + F_f = \frac{1}{2}\rho v_R^2 C_d A, \tag{3}$$

where

$F_D =$ drag force (N),
$v_R =$ velocity of the object relative to the fluid (m/s),
$C_d =$ drag coefficient (a dimensionless parameter),
$A =$ frontal area of the moving object (m^2).

A typical value of drag force taken from Ref. 28 is shown in Fig. 3.

3. Problem Statement

The following three constraints on the microrobot's ability for a drug delivery process were considered within the scope of this work:

- The robot's onboard energy supply is limited. Once the available energy has been consumed, the robot can become inactive and thus not deliver the drug to the target.
- The time used by robots to deliver the drug to the target.
- Avoidance of the biological obstacles in blood vessel is mandatory.

In the work described here, we studied a two-dimensional problem as the first step; its generalization to three-dimensional problem will be considered in a future work. The microrobots were assumed to move in this space under the following assumptions:

- The starting and target positions with respect to a given reference coordinate system were known.
- Microrobots were not allowed to move outside the blood vessel boundary.
- Microrobots might occupy the same location with respect to the reference coordinate system.
- Microrobots did not have power outage.
- There were two types of obstacles: static obstacles and dynamic obstacles. The static obstacles could be described as polygons with boundaries represented by linear equations while the dynamic obstacles could be described as circular. In the circulation environment, the static obstacles and the dynamic obstacles represented platelets adhered to the wall of a vessel and the red blood cells, respectively.
- Velocity of each dynamic obstacle as a function of time was known to microrobots.
- The speed of each dynamic obstacle was constant and randomly generated at the beginning of the analysis.
- The speed of all microrobots was greater than the speed of all dynamic obstacles in each time step.

Based on Eqs. (2) and (3), when a microrobot moved from one point to another inside the blood vessel as shown in Fig. 4, the objective function used in optimizing the energy consumption of microrobots for traveling to the target cell as a fitness value was given by Eq. (4).

From

$$\begin{aligned}
\Delta W_{R,t} &= \mathbf{F}_{D,R,t} \cdot \Delta \mathbf{d}_{R,t}, \\
\Delta W_{R,t} &= \frac{1}{2}\rho C_d A (v_{R,t} - v_B)^2 \frac{\mathbf{v}_{R,t} - \mathbf{v}_B}{\left|\mathbf{v}_{R,t} - \mathbf{v}_B\right|} \cdot \Delta \mathbf{d}_{R,t}, \\
\Delta W_{R,t} &= \frac{1}{2}\rho C_d A \left(v_{R,t} - \frac{1}{2\mu}\frac{\Delta P}{L}(h^2 - y_{R,t}^2)\right)^2 \frac{\mathbf{v}_{R,t} - \mathbf{v}_B}{\left|\mathbf{v}_{R,t} - \mathbf{v}_B\right|} \cdot \Delta \mathbf{d}_{R,t}.
\end{aligned}$$

Then
Minimize

$$W_R = \sum_{t=1}^{S} \left(\frac{1}{2}\rho C_d A \left(v_{R,t} - \frac{1}{2\mu}\frac{\Delta P}{L}(h^2 - y_{R,t}^2)\right)^2 \frac{\mathbf{v}_{R,t} - \mathbf{v}_B}{\left|\mathbf{v}_{R,t} - \mathbf{v}_B\right|} \cdot \Delta \mathbf{d}_{R,t} \right), \tag{4}$$

subject to

$$\begin{array}{lll}
\sqrt{(x_T - x_{R,t})^2 + (y_T - y_{R,t})^2} = 0 & \text{when} \quad t = S, & \\
x_{lb} < x_{R,t} < x_{ub} & \text{for} \quad t = 1, 2, 3, \ldots, S, & \\
y_{lb} < y_{R,t} < y_{ub} & \text{for} \quad t = 1, 2, 3, \ldots, S, & \\
d_{Ro,t} \geq \varepsilon & \text{for} \quad o = 1, 2, 3, \ldots, M & \text{for all} \quad t,
\end{array}$$

where

W_R = sum of energy consumption from starting point to target point for robot R (J),
S = number of robot's steps,
M = number of obstacles,
$\Delta \mathbf{d}_{R,t}$ = incremental displacement of robot R at time t (m),
$x_{R,t}$ = x-coordinate of robot R at step t,
$y_{R,t}$ = $y$coordinate of robot R at step t,
x_T = x-coordinate of target,

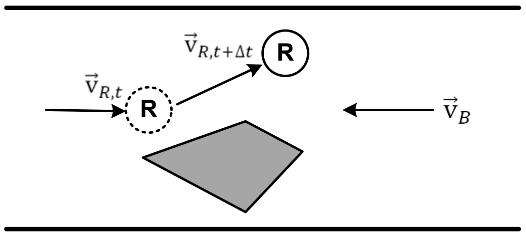

Fig. 4. Microrobot movement from one point to another inside the blood vessel.

y_T = y-coordinate of target,
x_{lb} = lower bound for x-coordinate in the search space,
x_{ub} = upper bound for x-coordinate in the search space,
y_{lb} = lower bound for y-coordinate in the search space,
y_{ub} = upper bound for y-coordinate in the search space,
$d_{Ro,t}$ = distance between robot R and obstacle o at step t (m),
ε = threshold distance value (m).

4. Microrobot Model

In a microrobot model designed for drug delivery, factors to be considered include the following: how the microrobot is powered, how the microrobot finds the target cells, what the communication methods are, and how microrobots can be removed when the job is finished. This section briefly reviews the technology that can potentially address these questions.

Research[3,29–31] has suggested some possible sources of energy for microrobots. The Adenosine Triphosphate (ATP) synthase enzyme, which can be applied in rotary bio-molecular motor-powered nanodevices, is an example of the energy source. Other examples include the remote inductive power presented by Takeuchi and Shimoyama,[30] and the use of CMOS for active telemetry and power supply for implanted devices introduced by Sauer *et al.*[31]

Chemical and biochemical sensors[1,3,10] can be used to sense the environment and detect the target cells and obstacles. The sensors detect changes in volume, concentration, displacement and velocity, pressure, or temperature of cells to identify the target.

Radio frequency identification device (RFID) technology[32] can be used to locate and track objects or even to remotely control human biological functions. RFID is thus often mentioned in microrobots communication methods.

In order to excrete the microrobots when their task has been completed, the microrobots must be created with disposable materials. An alternative is to create a microrobot that can anchor itself to a blood vessel for easier surgical removals.[3,10,33]

However, in order to design the microrobot for the simulation, the mathematical model used to simulate the movement of a microrobot and the mechanism used to recognize the obstacles must be described. The following subsections provide the model and the methods to support these requirements.

4.1. *Microrobot movement*

The movement of microrobots from one point to another inside the blood vessels can be modeled as shown in Fig. 5.

Let $(x_{i,s}, y_{i,s})$ be the position of robot i at time s, θ_i be the angle describing the direction of motion of robot i, $\upsilon_{i,s}$ be the velocity of robot i at time s, and Δt be the

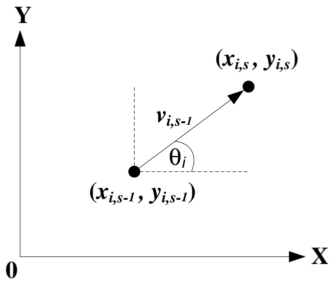

Fig. 5. Schematic sketch of microrobot positions.

time step. Then

$$x_{i,s} = x_{i,s-1} + \upsilon_{i,s-1} \cos\theta_i \, \Delta t, \tag{5}$$

$$y_{i,s} = y_{i,s-1} + \upsilon_{i,s-1} \sin\theta_i \, \Delta t. \tag{6}$$

4.2. *Obstacle detection and avoiding techniques*

In real applications, microrobots should be able to detect obstacles in their paths using their chemical and/or biochemical sensors. However, under the simulated environment, there exists the need to design a computational algorithm to recognize obstacles. In this paper, three computational methods are proposed: the collision detection, the boundary distance detection, and the obstacle avoiding.

4.2.1. *Collision detection*

To detect a collision between a microrobot and an obstacle, the point of intersection of a robot trajectory with an object boundary is identified. In Fig. 6, the intersected coordinate point $(x_{\text{cross}}, y_{\text{cross}})$ from a microrobot trajectory line and a boundary of the static obstacle is calculated. If this point is located between the lower bound and

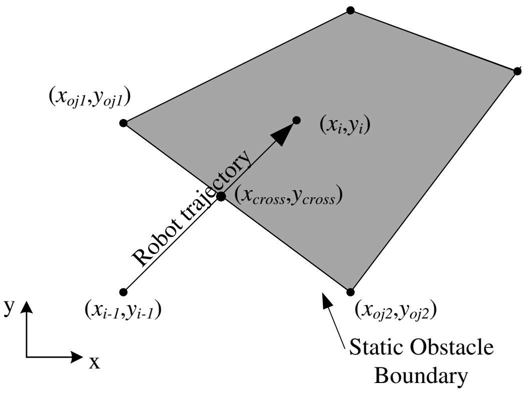

Fig. 6. Collision detection on a static obstacle.

the upper bound on both of these two lines, the collision is detected. The point of intersection $(x_{\text{cross}}, y_{\text{cross}})$ can be calculated with the following algorithm.

Let (x_{i-1}, y_{i-1}) be a robot position at time $i-1$, (x_i, y_i) be a robot position at time i, $y_o = m_o x_o + c_o$ be a linear equation of a static obstacle boundary, and $y_r = m_r x_r + c_r$ be a linear equation of the robot's trajectory. Then

$$x_{\text{cross}} = \frac{-(c_o - c_r)}{m_o - m_r}, \tag{7}$$

$$y_{\text{cross}} = m_o x_{\text{cross}} + c_o. \tag{8}$$

The robot and the static obstacle will collide if the following criteria are satisfied:

$$(x_{oj1} \le x_{\text{cross}} \le x_{oj2} \wedge x_{i-1} \le x_{\text{cross}} \le x_i),$$

and

$$(y_{oj2} \le y_{\text{cross}} \le y_{oj1} \wedge y_{i-1} \le y_{\text{cross}} \le y_i).$$

In Fig. 7, the point of intersection $(x_{\text{cross}}, y_{\text{cross}})$ of a microrobot trajectory and a boundary of the dynamic obstacle is shown; the algorithm to find $(x_{\text{cross}}, y_{\text{cross}})$ is given below.

Let $r^2 = (y_o - y_{oc})^2 + (x_o - x_{oc})^2$ be the boundary of a circular dynamic obstacle. Then

$$\begin{aligned} x_{\text{cross1}} = {} & (-2(m_r c_r - m_r y_{oc} - x_{oc}) - ((2(m_r c_r - m_r y_{oc} - x_{oc}))^2 \\ & - 4(m_r^2 + 1)((c_r - y_{oc})^2 + x_{oc}^2 - r^2))^{1/2})/2(m_r^2 + 1), \end{aligned} \tag{9}$$

$$y_{\text{cross1}} = m_r x_{\text{cross1}} + c_r. \tag{10}$$

and

$$\begin{aligned} x_{\text{cross2}} = {} & (-2(m_r c_r - m_r y_{oc} - x_{oc}) + ((2(m_r c_r - m_r y_{oc} - x_{oc}))^2 \\ & - 4(m_r^2 + 1)((c_r - y_{oc})^2 + x_{oc}^2 - r^2))^{1/2})/2(m_r^2 + 1), \end{aligned} \tag{11}$$

$$y_{\text{cross2}} = m_r x_{\text{cross2}} + c_r. \tag{12}$$

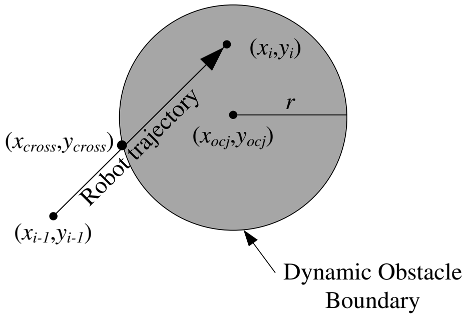

Fig. 7. Collision detection on a dynamic obstacle.

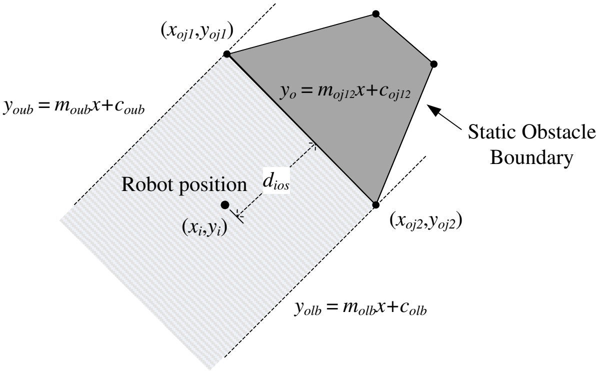

Fig. 8. Boundary distance detection on the static obstacle.

The robot and the dynamic obstacle will collide if the following criteria are satisfied:

$$\begin{aligned}(x_{i-1} &\leq x_{\text{cross1}} \leq x_i \wedge y_{i-1} \leq y_{\text{cross1}} \leq y_i)\\ &\wedge (r^2 = (y_{\text{cross1}} - y_{oc})^2 + (x_{\text{cross1}} - x_{oc})^2),\end{aligned}$$

or

$$\begin{aligned}(x_{i-1} &\leq x_{\text{cross2}} \leq x_i \wedge y_{i-1} \leq y_{\text{cross2}} \leq y_i)\\ &\wedge (r^2 = (y_{\text{cross2}} - y_{oc})^2 + (x_{\text{cross2}} - x_{oc})^2).\end{aligned}$$

4.2.2. *Boundary distance detection*

A microrobot is required to have a minimum distance from boundaries of an obstacle. The distance, d_{ios}, between a microrobot position and boundary line of a static obstacle, shown in Fig. 8, is given by Eq. (13).

Let

$$\begin{aligned}m_{oj12} &= \frac{y_{oj2} - y_{oj1}}{x_{oj2} - x_{oj1}},\\ c_{oj12} &= y_{oj1} - m_{oj12}x_{oj1},\\ m_{oub} &= m_{olb} = \frac{-1}{m_{oj12}},\\ c_{oub} &= y_{oj1} - m_{oub}x_{oj1},\\ c_{olb} &= y_{oj2} - m_{olb}x_{oj2}.\end{aligned}$$

Then

$$d_{ios} = \frac{|m_{oj12}x_i - y_i + c_{oj12}|}{\sqrt{m_{oj12}^2 + 1}}. \tag{13}$$

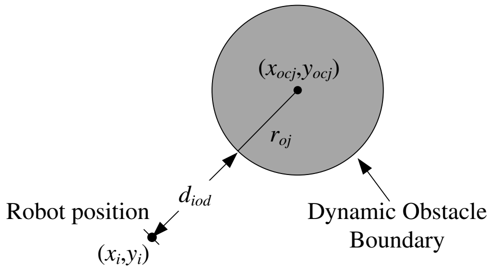

Fig. 9. Boundary distance detection on a dynamic obstacle.

The robot's current position must satisfy the following criteria:

$$y_i \geq m_{olb}x_i + c_{olb} \wedge y_i \leq m_{oub}x_i + c_{oub}.$$

For a circular obstacle, the distance, d_{iod}, between a microrobot position and its boundary, shown in Fig. 9, is calculated by

$$d_{iod} = \sqrt{(x_i - x_{ocj})^2 + (y_i - y_{ocj})^2} - r_{oj}. \tag{14}$$

4.2.3. *Avoiding obstacles*

This section describes a method used to avoid obstacles. The next-step position of a microrobot is analyzed with all positions of static obstacles and all next-step positions of dynamic obstacles in order to avoid any collision using the following pseudo-code:

```
for all obstacles
while (collision(robot's next-step position, static obstacle's position)
$\wedge$ collision(robot's next-step position, dynamic obstacle's next-step position)
$\wedge$ distance(robot's next-step position, static obstacle's position) $< d_{iod}$
```

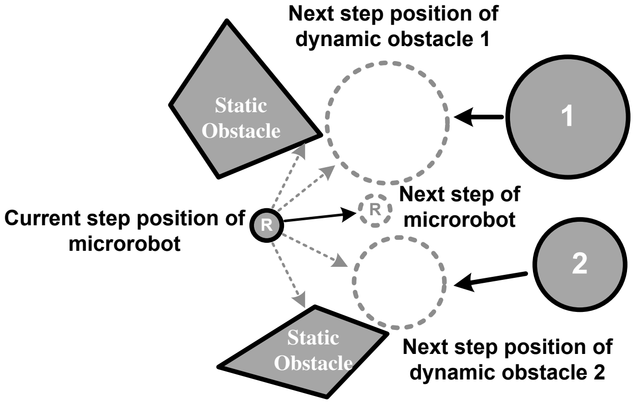

Fig. 10. Schematic sketch of avoiding obstacles.

```
$\wedge$ distance(robot's next-step position, dynamic obstacle's next-step position)
$< d_{ios}$)
    {
        Find robot's next step()
    }
```

Figure 10 illustrates an example of avoiding method by the microrobot when it encounters obstacles. The microrobot will try to find the next-step position by avoiding any collision and maintaining the distance between its next-step position and the positions of all obstacles. The details of finding the microrobot's next-step position will be provided in Sec. 5.

5. Adoption of Swarm Intelligence Algorithms in Drug Delivery

This section proposes three algorithms: (i) the ABC algorithm, (ii) the Best-so-far ABC algorithm, and (iii) the PSO algorithm to control mechanism of microrobots in drug delivery process. The following subsections provide a brief background on each algorithm.

5.1. *Artificial bee colony algorithm*

The ABC algorithm introduced by Karaboga[23] is one of the popular approaches used to find an optimal solution. This algorithm is inspired by the behavior of honey bees when seeking a quality food source.[34] The performance of ABC algorithm has been compared with that of other optimization methods such as the GA, the Differential Evolution (DE) algorithm, Evolution Strategies (ES), PSO, and Particle Swarm Inspired Evolutionary Algorithm (PS-EA). The comparison of results[35] for several optimization problems has shown that the ABC algorithm can produce better optimal solutions and thus is more effective than other methods. The ABC algorithm uses a set of computational agents called honey bees to find the optimal solution. The honey bees can be categorized into three groups: employed bees, onlooker bees and scout bees. Each solution in the search space consists of a set of optimization parameters which represent a food source position. The number of employed bees is equal to the number of food sources. In other words, there is only one employed bee investigating each food source. The quality of food source is called its "fitness value" and is associated with its position.

In the algorithm, the employed bees will be responsible for investigating their food sources (using fitness values) and sharing the information to recruit the onlooker bees. The onlooker bees will make a decision to choose a food source based on this information. A food source with a higher quality will have a larger probability of being selected by onlooker bees. An employed bee whose food source is rejected by employed and onlooker bees will change to a scout bee to search for new food sources randomly.

This process of a bee swarm seeking, advertising, and eventually selecting the best-known food source is used to find the optimal solution. Notice that the food sources are selected based on group decision making by the swarm. Independence and interdependence in collective decision making are important factors in this mechanism. Figure 11 illustrates the flow chart of the algorithm developed for finding the paths.

First, initial solutions consisting of a speed (v) and the direction of motion (θ) for each microrobot are generated. These parameters set are treated as the food sources. Each food source is used to move the microrobot from the current position to the next position in a search space.

The food sources will be updated by the employed bees. The choices are based on the neighborhood of the previously selected food sources. The position of the new food source can be calculated from

$$x'_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}). \tag{15}$$

In Eq. (15), x'_{ij} is the new feasible food source, which is selected by comparing the previous food source (x_{ij}) and the randomly selected food source from the neighboring food source (x_{kj}). ϕ_{ij} is a random number between $[-1, 1]$ which is used to adjust the old food source to become the new food source in the next iteration. $k \in \{1, 2, 3, \ldots, SN\} \wedge k \neq i$ and $j \in \{1, 2, 3, \ldots, D\}$ and are randomly chosen indices.

The microrobots use the food sources generated by the employed bees to update their positions. Collision detection will be performed and the objective score will be calculated using Eq. (14). If the new candidate position does not collide with any obstacle and gives a better objective value than the old position, the microrobot will move to the new position.

The onlooker bees will then select food sources from the employed bees. Food sources of better objective values have higher chances of being selected. The probability that a food source will be selected is given by

$$P_i = \frac{\text{fit}_i}{\sum_{n=1}^{N} \text{fit}_i}, \tag{16}$$

where N is the number of food sources and fit_i is the fitness value of the food source i. A smaller objective score indicates a better fitness value.

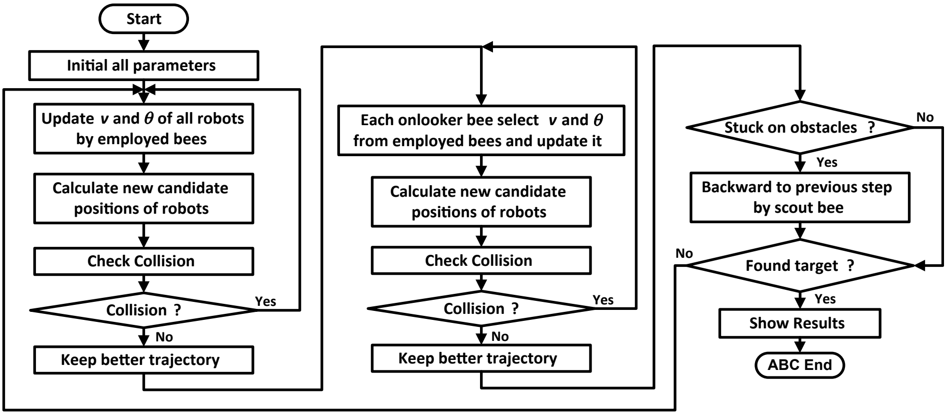
Start
Initial all parameters
Update *v* and *θ* of all robots by employed bees
Calculate new candidate positions of robots
Check Collision
Collision ?
Yes
No
Keep better trajectory
Each onlooker bee select *v* and *θ* from employed bees and update it
Calculate new candidate positions of robots
Check Collision
Collision ?
Yes
No
Keep better trajectory
Stuck on obstacles ?
No
Yes
Backward to previous step by scout bee
No
Found target ?
Yes
Show Results
ABC End

Fig. 11. Flow chart of microrobots control mechanism using ABC.

The onlooker bees will then select the food sources that produce better fitness values and update those food sources using the same algorithm as the employed bees. Microrobots will use these food sources to update their positions. The collision detection and the objective scoring processes will be performed repeatedly so that the microrobots will keep moving to new positions.

The microrobot that cannot avoid an obstacle within a certain period of time will backtrack to its previous position. The new food sources will then be randomly generated by the scout bees.

The process will be repeated until the microrobots reach the target or the number of iterations equals the Maximum Cycle Number (MCN).

5.2. *Best-so-far ABC algorithm*

The Best-so-far ABC has been proposed by Banharnsakun *et al.*[24] to enhance the exploitation and exploration processes in the ordinary ABC algorithm. While the exploration process is related to the independent search for an optimal solution, the exploitation uses existing knowledge to bias the search. Experimental results have demonstrated that the Best-so-far ABC is able to produce higher quality solutions with faster convergence than the original ABC and other state-of-the-art heuristic-based algorithms.[24,36–38]

In the Best-so-far ABC, there are three modification modules including the best-so-far ABC module, the adjustable search radius module, and the objective-value-based comparison module. Note that in this work we only employ the best-so-far ABC module to control motions of microrobots in the drug delivery process. Solutions consisting of a speed (υ) and the direction of motion (θ) for each microrobot are treated as the food sources and will be updated by the bee agents similarly to the steps in the ordinary ABC algorithm.

In the ordinary ABC algorithm, each onlooker bee selects a food source based on a probability that varies according to the fitness function explored by a single employed bee. Then the new candidate solutions are generated by updating the onlooker solutions as shown in Eq. (15). However, changing only one dimension of the solution x_i in the original ABC results in a slow convergence rate.

In the best-so-far ABC module, all onlooker bees use existing information from all employed bees to make a decision on a new candidate food source. Thus, the onlookers can compare information from all candidate sources and are able to select the best-so-far position. The new method used to calculate a candidate food source is

$$\nu_{id} = x_{ij} + \Phi f_b(x_{ij} - x_{bj}), \tag{17}$$

where

ν_{id} = the new candidate food source for onlooker bee position i in dimension $d, d = 1, 2, 3, \ldots, D$,
x_{ij} = the selected food source position i in a selected dimension j,
Φ = a random number between -1 and 1,

f_b = the fitness value of the best food source found so far,
x_{bj} = the best so far food source in selected dimension j.

5.3. *Particle swarm optimization algorithm*

PSO[25] algorithm is inspired by the mechanism of bird flocks and fish schools that is used to synchronize their movement to avoid collisions during their foraging. In the PSO method, the member in this swarm is called a "particle". The position of each particle stands for a feasible solution that will be tracked and updated. The "pbest" value is used to call its own best position of each particle and the "gbest" value is used to call the best position, which is obtained from comparison of pbest on all particles in the swarm. In each iteration, the position of each particle will be updated by its velocity that is calculated from the cognitive value, which is influenced by the best position found by a particle itself (different value between its pbest and its current position) and a social value which is modeled by the influence of the best position found by other particles (different value between gbest and its current position). Consequently, the position of each particle will converge to the optimal solution.

The position of each particle in the PSO algorithm is adjusted by its velocity as

$$\varphi_{id} = \omega \upsilon_{id} + c_1 R_1 (p_{id} - x_{id}) + c_2 R_2 (p_{gd} - x_{id}), \tag{18}$$

$$x'_{id} = x_{id} + \varphi_{id}, \tag{19}$$

where φ_{id} is the velocity of particle, υ_{id} is the previous velocity of particle, x_{id} is the previous position of particle, x'_{id} is the new position of particle, ω is the inertial constant factor, c_1, c_2 are values of cognition and social component, respectively, R_1, R_2 are two random numbers independently generated in [0, 1], p_{id} is the pbest of particle i in dimension d, p_{gd} is the gbest of particle i in dimension d.

In the PSO method, solutions consisting of speed (ν) and the direction of motion (θ) are treated as the positions of the particles and will be tracked and updated by using Eqs. (18) and (19).

The microrobots use these solutions to update their positions. The mechanisms to detect and to avoid the obstacles are also employed on the PSO method the same as applied on the ABC method and the Best-so-far ABC method.

6. Experiments and Results

We studied the performance of the three proposed techniques including the Best-so-far ABC, the ABC, and the PSO methods in two sets of experiments: static obstacle environment and dynamic obstacle environment. For both experiments, we aimed at comparing and evaluating the solution quality obtained from these three methods in the perspectives of the target hit rate, the energy consumption, and the time step used by microrobot to deliver the drug to the target. To make a clearer analysis of the performance on the proposed algorithms, the A* search algorithm, a well-known path finding method,[39,40] is also included and can be used for comparison with the proposed method in these experiments.

Each of the experiment sets was repeated 30 times with different random seeds. In the static environment experiment, the complex search spaces are presented in terms of the random position of the obstacles whereas in the dynamic environment experiment, the complex search spaces are conducted in terms of the random movement of the obstacles that is different in each run of 30 times.

The proposed methods were programmed in C++ and all experiments were run on a PC with Intel Core i7 CPU, 2.0 GHz. The size of the search space is set as 150*1000, the starting point at (0,75) and the target point at (1000,75). For the ABC and the Best-so-far ABC methods, the size of population was 10, and the number of iterations (MCN) was 1000. There were 20 obstacles located at random positions in the search space.

For the parameter setting of the PSO method, the number of particles was 10, the parameters used in PSO were defined as: $c_1 = c_2 = 2, \omega = 0.7$, and the number of iterations was 1000. Note that the number of function evaluations per iteration in the ABC and that in the Best-so-far ABC were not equal to that in the PSO algorithm. The number of function evaluations in the ABC and the Best-so-far ABC is equal to twice per iteration (for employed bee and onlooker bee phases) but there is only once per iteration in the ordinary PSO algorithm. For a fair comparison, the PSO algorithm in this experiment was modified to perform the function evaluation twice in each iteration.

In order to mimic a somewhat realistic environment for flow in a blood vessel, the properties of blood flow reported in Refs. 26–28, 41–43 used in our experiment are listed in Table 1.

6.1. *Static environment experiment*

In this experiment, the three methods were used to analyze the same problem. The objective was to find the optimal path that microrobots used to minimize energy consumption for traveling from the starting point to the target point.

The example of search space initially created is shown in Fig. 12. The microrobots will move from the starting point on the left of the search space to the target on the far right of the search space.

Table 1. The properties of blood flow in arteries.

Parameter	Value
Differential pressure, ΔP	20 Pa
Vessel length, L	10 cm
Vessel radius, R	2 mm
Blood density, ρ	1060 kg/m^3
Blood viscosity, μ	4 mPa·s
Drag coefficient, C_d	0.47
Frontal area of microrobot, A	0.179 mm^2

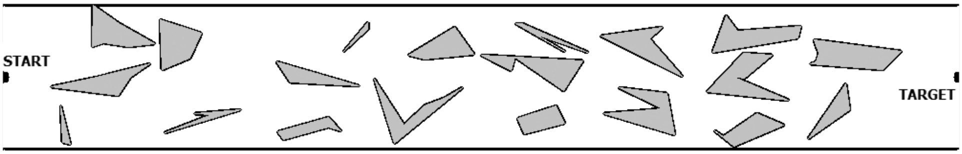

Fig. 12. The sample of initial search space.

Figure 13 presents the trajectory of the microrobot when the simulation has been completed with optimization method. The result showed that the control mechanism by our designed methods would adjust the new position in each time step of the microrobot to avoid obstacles and to move ahead to its target, while the minimization of robot's energy consumption was still maintained. The microrobot would follow the path for which the position where the resistance from the drag force was low in order to minimize its energy.

The results obtained with the Best-so-far ABC, the ABC, and the PSO methods on the static obstacle environment are listed in Table 2. The "Target hit rate" column shows the percentage of the average number of microrobots that reach the target. The "Energy consumed" column and the "Time step used" column show the energy consumed and the number of iterations used by microrobots to reach the target point, respectively. The "Target hit rate" can be calculated by

$$\text{Taget hit rate} = \frac{\begin{array}{c}\text{total number of robots that reach}\\ \text{the target from 30 experiments}\end{array}}{\text{total number of robots used in 30 experiments}} * 100. \tag{20}$$

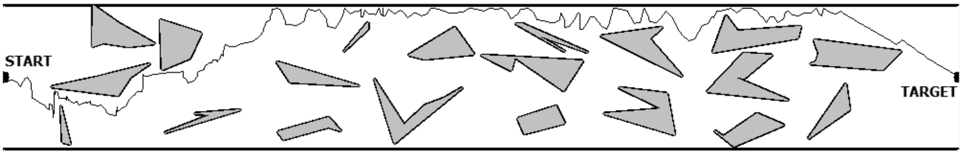

Fig. 13. The completed path of the microrobot.

Table 2. Results for the static obstacle environment.

		Energy Consumed (J)			Time Step Used		
Method	Target Hit Rate	Best	Average	S.D.	Best	Average	S.D.
A* search	**87%**	6.61E−08	7.81E−08	7.25E−09	358	423	62
Best-so-far ABC	**100%**	**6.05E−08**	**7.36E−08**	**6.94E−09**	**236**	**283**	**27**
ABC	**100%**	6.52E−08	7.57E−08	7.13E−09	312	354	51
PSO	73%	6.76E−08	7.92E−08	1.22E−08	394	562	92

From Table 2, it can be seen that the Best-so-far ABC method generates either better quality results in terms of the target hit rates, the energy consumed, and the time step used than the A* search, the ABC, and the PSO methods or at least equal. Microrobots control mechanism using both the Best-so-far ABC and the ABC algorithms were able to find the target with 100% success rate whereas the A* search and the PSO algorithms were able to find only 87% and 73% of the time, respectively. The results also showed that the energy consumed by the microrobot based on the Best-so-far ABC and the ABC approaches was less than that of the microrobot based on the A* search and PSO approaches. The best and the average of the energy consumed for the Best-so-far ABC approach are 6.05E−08 J and 7.36E−08 J whereas the ABC approach obtained 6.52E−08 J and 7.57E−08 J, the A* search approach obtained 6.61E−08 J and 7.81E−08, and the PSO approach obtained 6.76E−08 J and 7.92E−08 J, respectively. Moreover, in terms of the time steps used, the results showed that the microrobots based on the Best-so-far ABC technique could reach to the target point faster than did the microrobots based on the A* search, the ABC, and the PSO techniques. The best and the average of the time step used for the Best-so-far ABC technique were 236 and 283 whereas the A* search technique used 358 and 423, the ABC technique used 312 and 354, and the PSO technique used 394 and 562, respectively. It is also apparent that the Best-so-far ABC method was more robust than the ABC and the PSO methods as shown by the lower standard deviations in the S.D. column on both the energy consumed and the time step used terms.

Figure 14 shows the comparison of the convergence speed in terms of time steps (number of iterations) for the four proposed methods on the static obstacle environment. It shows that the Best-so-far ABC method converges to the target point substantially faster than the A* search, the ABC, and the PSO methods with a much smaller number of iterations needed.

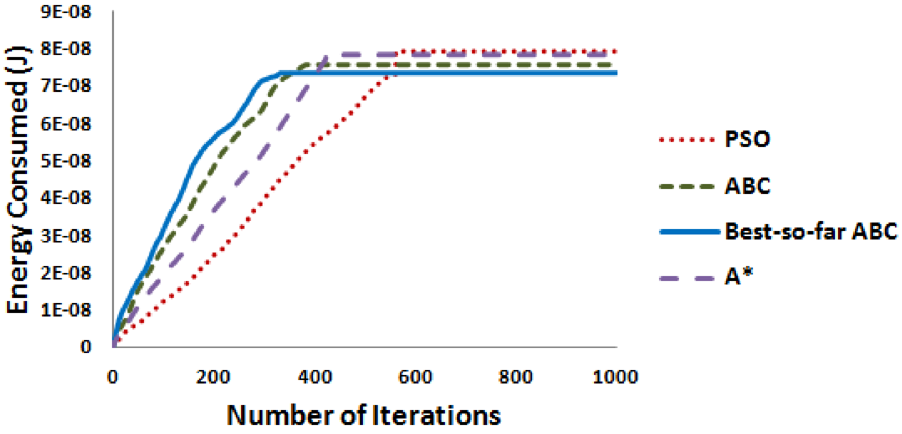

Fig. 14. The convergence speed in terms of time steps under the static obstacle environment.

6.2. *Dynamic environment experiment*

For this experiment, to validate the effectiveness of our proposed methods on dynamic environment, the dynamic obstacles were introduced. There were 10 static obstacles (polygon shape) and 10 dynamic obstacles (circular shape) located at random positions in the search space. All parameter settings were set the same as those in the previous experiment. Figure 15 illustrates the snapshot of microrobot trajectory from the starting point to the target point by the Best-so-far ABC methods.

From Fig. 15, it is obvious that the microrobot controlled by our proposed methods could find the optimal path without any collision with the obstacles. Especially for the time step from $T = 25$ to $T = 50$ and $T = 75$ to $T = 150$, the proposed methods based on the collision detection and the boundary distance detection techniques could help the microrobot to avoid collision with the dynamic obstacles during traveling to the target point. While the proposed methods tried to control the microrobot to avoid the dynamic obstacle, the results exhibited in Fig. 15 also show that the proposed methods still adjusted the direction of the path for the microrobot to travel to the position where the resistance from the drag force was low.

The results obtained with our proposed methods on the dynamic obstacle environment are listed in Table 3.

The results from Table 3 showed that the time step used by microrobots based on all proposed methods on the dynamic obstacle environment increased when compared with that under the static environment. The increase in the number of time steps used was caused by the microrobots waiting for a dynamic obstacle to move far away from their directions. However, the Best-so-far ABC method still generated better than the A* search, the ABC, and the PSO methods, or equal quality results, in terms of the target hit rate, the energy consumed, and the time step used. The

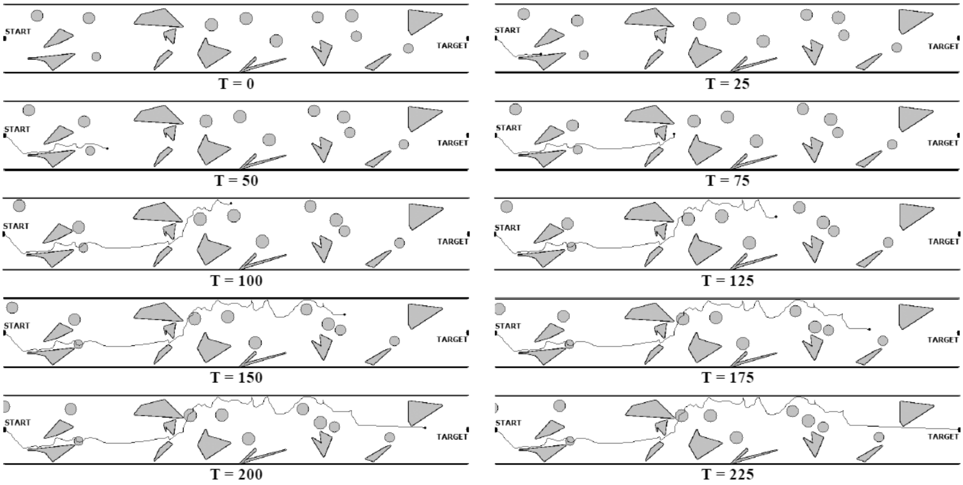

Fig. 15. The snapshot of microrobot trajectory under dynamic environment.

Table 3. The results on the dynamic obstacle environment.

		Energy Consumed (J)			Time Step Used		
Method	Target Hit Rate	Best	Average	S.D.	Best	Average	S.D.
A* search	**81%**	6.31E−08	7.33E−08	6.84E−08	386	513	73
Best-so-far ABC	**100%**	**5.52E−08**	**6.67E−08**	**5.92E−09**	**267**	**323**	**35**
ABC	**100%**	6.23E−08	7.11E−08	6.27E−09	354	437	57
PSO	70%	6.36E−08	7.63E−08	7.13E−09	462	624	94

target hit rate was still 100% for the Best-so-far ABC and the ABC methods whereas they dropped to 81% and 70% for the A* search and the PSO methods, respectively. The best and the average of the energy consumed for the Best-so-far ABC approach were 5.52E−08 J and 6.67E−08 J whereas the A* search obtained 6.31E−08 J and 7.33E−08 J, the ABC approach obtained 6.23E−08 J and 7.11E−08 J, and the PSO approach obtained 6.36E−08 J and 7.63E−08 J, respectively. Moreover, the Best-so-far ABC technique continued to give good results in terms of the time steps used compared with the A* search, the ABC, and the PSO techniques. The best and the average of the time step used for the Best-so-far ABC technique were 267 and 323 whereas the A* search technique used 386 and 513, the ABC technique used 354 and 437, and the PSO technique used 462 and 624, respectively.

The comparison of the convergence speeds in terms of number of iterations for the four proposed methods on the dynamic obstacle environment is shown in Fig. 16.

From Fig. 16, it is seen that the microrobot controlled by the Best-so-far ABC could converge to the target point faster than the A* search, the ABC, and the PSO methods. It also consumed less energy than both the ABC and the PSO in this case.

In summary, even though all proposed algorithms have a similar search process based on heuristic method, the A* search, the Best-so-far ABC and the ABC methods could provide better results than the PSO method. However, the A* search algorithm has characteristic of slow search speed and can easily fall into the failed

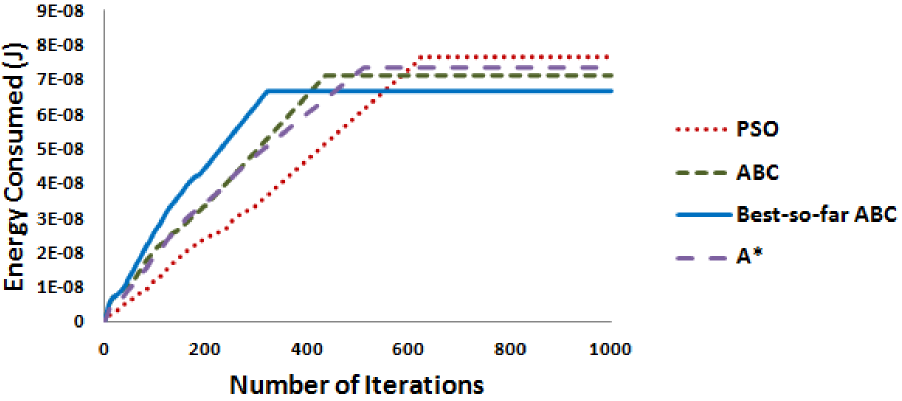

Fig. 16. The convergence speed in terms of time steps under the dynamic obstacle environment.

search state when trap obstacles are met in the unknown environments. The Best-so-far ABC and the ABC methods have both the exploitation and the exploration in their search process, while the PSO method only has the exploitation. The exploitation is handled by employed bees and onlooker bees, while the exploration is maintained by scout bees in the Best-so-far ABC and the ABC methods. If some directions of microrobot become trapped at any obstacle, the scout bees will try to randomly search the new direction again. Microrobots based on both the Best-so-far ABC and the ABC methods can achieve 100% target hit rate and provide low energy consumption as compared with the microrobots based on the A* search and the PSO methods.

7. Conclusions

Target finding and obstacle avoiding mechanisms based on swarm intelligence algorithms were employed in this work to design a microrobot for optimally delivering drugs to the target cell. Collision detection based on intersection lines concept and boundary distance detection based on Point-Line distance concept were used to avoid obstacles in microrobot's paths.

The swarm intelligence algorithms used in this work are the ABC algorithm, the Best-so-far ABC algorithm, and the PSO algorithm. The performance of the three methods was then compared together under the same parameter setting under various obstacle environments. In addition, forces that drive or resist blood flow, as well as pressure and flow in blood vessels, were considered to somewhat mimic the realistic environment. The target hit rate, the energy consumption, and the number of time steps used were set as the objectives in this drug delivery process. The results showed that the Best-so-far ABC and the ABC algorithms can be used to find the optimal path for the microrobots to deliver drug to the target better than the PSO algorithm. The microrobots can follow this optimal path to minimize their energy consumption and the time step used to reach the target point, as well as to avoid the biological obstacles.

In addition, the results on both static and dynamic environments also showed the evidence that the Best-so-far ABC method can help microrobots avoid any collision with obstacles while still continuing on the optimal path, leading toward convergence on the target faster than the ordinary ABC method. Microrobot's control mechanism using the Best-so-far ABC method is thus effective.

All in all, the major contribution of this work is the successful development of an effective and robust method with faster convergence for drug delivery problem, in which the concept of swarm intelligence is utilized to optimize the energy consumption used by microrobots to reach the target in an unknown environment with moving obstacles. However, there is still room for improvement in the capability of the proposed methods. In the future work, the drug delivery in the three-dimensional environments will be studied and the sensitivity of the parameter settings on the proposed algorithms will be addressed.

Acknowledgments

This work is partially supported by Faculty of Engineering at Sriracha, Kasetsart University Sriracha Campus (2558/1).

References

1. A. Cavalcanti and R. A. Freitas, Nanorobotics control design: A collective behavior approach for medicine, *IEEE Trans. Nanobiosci.* **4** (2005) 133–140.
2. G. M. Patel, G. C. Patel, R. B. Patel, J. K. Patel and M. Patel, Nanorobot: A versatile tool in nanomedicine, *J. Drug Target.* **14** (2006) 63–67.
3. A. Cavalcanti, B. Shirinzadeh, R. A. Freitas and T. Hogg, Nanorobot architecture for medical target identification, *Nanotechnology* **19** (2008) 1–12.
4. P. Dario, M. C. Carrozza, L. Lencioni, B. Magnani and S. D'Attanasio, Micro robotic system for colonoscopy, in *Proc. IEEE Int. Conf. Robotics and Automation* (1997), pp. 1567–1572.
5. K. Ishiyama, M. Sendoh and K. I. Arai, Magnetic micromachines for medical applications, *J. Magn. Magn. Mater.* **242–245** (2002) 41–46.
6. C. Yu, J. Kim, H. Choi, J. Choi, S. Jeong, K. Cha, J.-O. Park and S. Park, Novel electromagnetic actuation system for three-dimensional locomotion and drilling of intravascular microrobot, *Sens. Actuators A: Phys.* **161** (2010) 296–304.
7. B. J. Nelson, I. K. Kaliakatsos and J. J. Abbott, Microrobots for minimally invasive medicine, *Annu. Rev. Biomed. Eng.* **12** (2010) 55–85.
8. A. Stentz, Optimal and efficient path planning for partially-known environments, in *Proc. IEEE Int. Conf. Robotics and Automation* (1994), pp. 3310–3317.
9. M. S. G. Tsuzuki, T. C. Martins and F. K. Takase, Robot path planning using simulated annealing, in *12th IFAC Symp. Information Control Problems in Manufacturing*, (2006), pp. 173–178.
10. W. M. Tao and M. Zhang, A genetic algorithm-based area coverage approach for controlled drug delivery using microrobots, *Nanomedicine* **1** (2005) 91–100.
11. M. Brand, M. Masuda, N. Wehner and X.-H. Yu, Ant colony optimization algorithm for robot path planning, in *Proc. 2010 Int. Conf. Computer Design and Applications* (2010), pp. 436–440.
12. N. Ganganath, C.-T. Cheng, C. K. Tse, An ACO-based off-line path planner for nonholonomic mobile robots, in *Proc. IEEE Int. Symp. Circuits and Systems* (2014), pp. 1038–1041.
13. Y. Q. Qin, D. B. Sun, M. Li and Y. G. Cen, Path planning for mobile robot using the particle swarm optimization with mutation operator, in *Proc. Int. Conf. Machine Learning and Cybernetics*, Vol. 4 (2004), pp. 2473–2478.
14. X. Chen and Y. Li, Smooth path planning of a mobile robot using stochastic particle swarm optimization, in *Proc. IEEE on Mechatronics and Automation* (2006), pp. 1722–1727.
15. Q. Zhang and S. Li, A global path planning approach based on particle swarm optimization for a mobile robot, in *Proc. 7th WSEAS Int. Conf. Robotics, Control, and Manufacturing Technology* (2007), pp. 263–267.
16. K. H. S. Hla, Y. Choi and J. S. Park, Obstacle avoidance algorithm for collective movement in nanorobots, *Int. J. Comput. Sci. Netw. Secur.* **8** (2008) 302–309.
17. Y. Zhang, D.-W. Gong and J.-H. Zhang, Robot path planning in uncertain environment using multi-objective particle swarm optimization, *Neurocomputing* **103** (2013) 172–185.

18. B. B. V. L. Deepak, D. R. Parhi and B. M. V. A. Raju, Advance particle swarm optimization-based navigational controller for mobile robot, *Arab. J. Sci. Eng.* **39** (2014) 6477–6487.
19. S. Chandrasekarn and D. F. Hougen, Swarm intelligence for cooperation of bio-nano robots using quorum sensing, in *Proc. BioMicro and Nanosystems Conf.* (2006), pp. 15–18.
20. Md. A. Hossain and I. Ferdous, Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique, *Robot. Auton. Syst.* **64** (2015) 137–141.
21. P. K. Mohanty and D. R. Parhi, Cuckoo search algorithm for the mobile robot navigation, in *Proc. 4th Int. Conf. Swarm, Evolutionary and Memetic Computing* (2013), pp. 527–536.
22. A. Banharnsakun, T. Achalakul and R. C. Batra, Target finding and obstacle avoidance algorithm for microrobot swarms, in *Proc. 2012 IEEE Int. Conf. Systems, Man, and Cybernetics* (2012), pp. 1610–1615.
23. D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.
24. A. Banharnsakun, T. Achalakul and B. Sirinaovakul, The best-so-far selection in artificial bee colony algorithm, *Appl. Soft Comput.* **11** (2011) 2888–2901.
25. J. Kennedy and R. C. Eberhart, Particle swarm optimization, in *Proc. 1995 IEEE Int. Conf. Neural Networks*, Vol. 4 (1995), pp. 1942–1948.
26. Y. C. Fung, *Biomechanics: Circulation*, 2nd edn. (Springer, New York, 1996).
27. J. B. Franzini and E. J. Finnemore, *Fluid Mechanics*, 9th edn. (McGraw-Hill, New York, 1997).
28. K. B. Yesin, K. Vollmers and B. J. Nelson, Modeling and control of untethered biomicrorobots in a fluidic environment using electromagnetic fields, *Int. J. Robot. Res.* **25** (2006) 527–536.
29. R. K. Soong, G. D. Bachand, H. P. Neves, A. G. Olkhovets, H. G. Craighead and C. D. Montemagno, Powering an inorganic nanodevice with a biomolecular motor, *Science* **290** (2000) 1555–1558.
30. S. Takeuchi and I. Shimoyama, Selective drive of electrostatic actuators using remote inductive powering, *Sens. Actuators A* **95** (2002) 269–273.
31. C. Sauer, M. Stanacevic, G. Cauwenberghs and N. Thakor, Power harvesting and telemetry in CMOS for implanted devices, *IEEE Trans. Circuits Syst.* **52** (2005) 2605–2613.
32. H. Aubert, RFID technology for human implant devices, *C. R. Phys.* **12** (2011) 675–683.
33. L. Rubinstein, A practical nanorobot for treatment of various medical problems, in *Proc. 8th Foresight Conf. Molecular Nanotechnology*, Vol. 2 (2002), pp. 208–214.
34. J. C. Biesmeijer and T. D. Seeley, The use of waggle dance information by honey bees throughout their foraging careers, *Behav. Ecol. Sociobiol.* **59** (2005) 133–142.
35. D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *J. Glob. Optim.* **39** (2007) 459–471.
36. A. Banharnsakun, T. Achalakul and B. Sirinaovakul, Job shop scheduling with the best-so-far ABC, *Eng. Appl. Artif. Intell.* **25** (2012) 583–593.
37. A. Banharnsakun, B. Sirinaovakul and T. Achalakul, The best-so-far ABC with multiple patrilines for clustering problems, *Neurocomputing* **116** (2013) 355–366.
38. A. Banharnsakun, B. Sirinaovakul and T. Achalakul, The performance and sensitivity of the parameters setting on the best-so-far ABC, in *The Ninth Int. Conf. Simulated Evolution and Learning*, eds. L. T. Bui *et al.*, (Springer, Berlin, 2012), pp. 248–257.

39. P. E. Hart, N. J. Nilsson and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. Syst. Sci. Cybern.* **4** (1968) 100–107.
40. J. Yao, C. Lin, X. Xie, A. J. Wang and C. C. Hung, Path planning for virtual human motion using improved A* algorithm, in *Proc. Seventh Int. Conf. Information Technology: New Generations* (2010), pp. 1154–1158.
41. J. Cutnell and K. Johnson, *Physics*, 4th edn. (Wiley, California, 1998).
42. R. Glaser, *Biophysics*, 4th edn. (Springer, New York, 2001).
43. K. B. Chandran, A. P. Yoganathan and S. E. Rittgers, *Biofluid Mechanics: The Human Circulation* (CRC/Taylor and Francis, Boca Raton, 2007).