# Novel Metaheuristic Hybrid Spiral-Dynamic Bacteria-Chemotaxis Algorithms for Global Optimisation

A. N. K. Nasir[1] and M. O. Tokhi[2].

[1]Faculty of Electrical & Electronics Engineering, Universiti Malaysia Pahang, Pekan Pahang Malaysia.
[2]School of Engineering, London South Bank University, 103 Borough Road, London SE1 0AA, UK
[1]kasruddin@ump.edu.my, [2]tokhim@lsbu.ac.uk

**ABSTRACT**

This paper presents hybrid spiral-dynamic bacteria-chemotaxis algorithms for global optimisation and their application to control of a flexible manipulator system. Spiral dynamic algorithm (SDA) has faster convergence speed and good exploitation strategy. However, the incorporation of constant radius and angular displacement in its spiral model causes the exploration strategy to be less effective hence resulting in low accurate solution. Bacteria chemotaxis on the other hand, is the most prominent strategy in bacterial foraging algorithm. However, the incorporation of a constant step-size for the bacteria movement affects the algorithm performance. Defining a large step-size results in faster convergence speed but produces low accuracy while defining a small step-size gives high accuracy but produces slower convergence speed. The hybrid algorithms proposed in this paper synergise SDA and bacteria chemotaxis and thus introduce more effective exploration strategy leading to higher accuracy, faster convergence speed and low computation time. The proposed algorithms are tested with several benchmark functions and statistically analysed via nonparametric Friedman and Wilcoxon signed rank tests as well as parametric $t$-test in comparison to their predecessor algorithms. Moreover, they are used to optimise hybrid Proportional-Derivative-like fuzzy-logic controller for position tracking of a flexible manipulator system. The results show that the proposed algorithms significantly improve both convergence speed as well as fitness accuracy and result in better system response in controlling the flexible manipulator.

*Keywords- Spiral dynamics; bacteria chemotaxis; PD-like control; fuzzy logic control; flexible manipulator system.*

## 1. INTRODUCTION

Metaheuristic algorithms are currently gaining popularity among researchers worldwide due to their effectiveness in finding optimum solution and reliability in solving real world problems. They are widely employed to solve various complex problems in science and engineering [1], medicine [2], finance [3], management [4], etc. Metaheuristic algorithms are normally have the following four basic characteristics. First, they are inspired by natural phenomena in the universe such as biological-inspired or nature-inspired. Second, they include a stochastic approach where an optimum solution is not known and it is randomly searched in a feasible area. Third, they are free of gradient or do not use derivative term in finding solution and finally, they involve several parameters to be defined at the initial stage in order to function properly [5]. Examples of well known metaheuristic algorithms inspired from biological phenomena are bacterial foraging algorithm (BFA) [6], particle swarm optimisation (PSO) [7], genetic algorithm (GA) [8], ant colony (ACO) [9] and bee colony [10]. On the other hand, examples of metaheuristic algorithms inspired from natural phenomena other than living organisms are simulated annealing (SA) [11], tabu search (TS) [12], spiral dynamic algorithm (SDA) [13], harmony search [14] and central force optimisation [15]. Selection of user defined parameters at the intialisation is a challenging task and it affects the performance of the algorithm very much. A study on parameter variation on the performance of metaheuristic algorithm has been presented in [79]. Some researchers utilised mathematical formulation [16], [21] while some other researchers adopted intelligent approaches such as fuzzy logic [75], [76], [77] to adaptively vary the parameters based on certain variables or conditions. Another important factor that determines the successful application of metaheuristic algorithms in solving problems is a balance between exploration and exploitation strategies. The exploration is a technique of the algorithm to search an optimum solution thoroughly and diversely within a search area while exloitation is the ability to intensify the search operation within a more promising and feasible region. Both exploration and exploitaion can also be referred to as diversification and intensification strategies. A metaheuristic algorithm in general implements swarm searching behaviour in its operation. In order to improve the performance in terms of processing time for the whole search operation, in real-time applications, some researchers have utilised parallel computing architecture such as graphic processing unit. Unlike standard central processing unit, here the motion of each search agent to find an optimal solution is done in a parallel manner. Some of the metaheuristic algorithms that have successfully been implemented on parallel computing architectures include GA, PSO, differential evolution (DE) and SA [80], [81]. In this work, two metaheuristic algorithms namely

BFA and SDA, their development, modification and application to solve engineering problems are selected as the main subject in this paper.

SDA is a nature-inspired optimisation algorithm introduced by Tamura and Yasuda [13]. SDA appears to have good potential for further development due to its simplicity, fast convergence and low computation time. The strategy in this algorithm is generally based on spiral phenomena in nature such as spiral of galaxy, hurricane, tornado and nautilus shell. The most important feature of SDA is a spiral model that determines the shape and charateristic of the spiral. The advantage of having spiral strategy in metaheuristic algorithms is that it comprises both diversification and intensification phases. Diversification can be found at the initial stage or at the outermost layer of the spiral where the search point moves away from the original location towards the centre of the spiral with a large step size. Intensification occurs at the inner layer of the spiral where the search point moves with smaller step size towards the final point. It is interesting to point out that the spiral model has a dynamic step size in its motion starting with larger step size and getting smaller when it approaches the final point, which is located at the centre of the spiral. During the diversification phase, the search point has the opportunity to explore optimum solution with large distance while intensification takes place, it then exploits the solution optimally within a more confined, remote and promising area that could not be reached during the diversification phase. Combination of these two strategies will possibly allow an algorithm to search the area thoroughly and effectively. Although, SDA has covered both diversification and intensification phases, many publications in the literature have found that the exploration strategy of SDA still can be further improved to avoid the algorithm from getting trapped into local optima solutions.

SDA is a relative new algorithm and there are very few works found in the literature about the development and application of this algorithm to solve real world problems. [16] has conducted a study to vary the spiral model radius and angular displacement with respect to individual point fitness cost at every iteration. The authors have proposed adaptive mathematical formulations based on linear, quadratic and exponential funcions as well as adaptive intelligent method based on fuzzy logic inference. The proposed algorithms have been tested with several unimodal and multimodal standard benchmark functions. The authors have proposed other versions of adaptive SDA where instead of incorporating only fitness cost into the adaptive equation, the best fitness cost of the current iteration is also included [17]. The adaptive algorithms have been applied to optimisation of dynamic model of flexible manipulator system and the results show that the algorithms successfully avoid local optima. [18] has proposed hybrid spiral-dynamic BFA (HSDBF) to optimise PD control for position tracking control of flexible manipulator system. The authors placed bacteria-chemotaxis phase prior to hybrid spiral-dynamic bacteria-chemotaxis algorithm to enhance the exploration strategy and to avoid the original algorithm from being trapped into local optima. The tumble and swim actions for the bacteria during the chemotaxis phase were made in spiral motion rather than random. This algorithm however was employed to solve low dimension problems. [19] has proposed hybrid spiral–bacteria foraging algorithm (HSBFA) and hybrid bacteria-chemotaxis spiral-dynamic algorithm (HBCSD). In HSBFA, the authors incorporated the spiral model into chemotaxis phase of BFA to assist the exploration strategy of BFA, where the bacteria motion is guided towards global best position at every iteration. The algorithm has produced faster convergence speed and better fitness accuracy. However, incorporating the spiral model into BFA has increased the total computation cost needed to complete the whole search operation. As a result, the authors proposed another algorithm namely HCSA, which replaced the reproduction, elimination and dispersal phases of BFA with the presence of spiral model in the chemotaxis phase aiming to reduce the total computational time but maintaining the performances. In HBCSD, the authors placed bacteria-chemotaxis in the first part to cover the exploration strategy while the SDA was placed in the second to cover the exploitation strategy. In HBCSD the bacteria move in random motion as the original BFA for the tumble action. The proposed algorithms were employed to optimise high order dynamic model of flexible systems. HSDBF in [18] and HBCSD in [19] have collaborative type structure where the combination of the chemotaxis phase and SDA is in series and the combined features are executed one after another in a seqential order. A summary of related works on adaptive and hybrid types based on SDA is shown in Table I.

Table I: Summary of related works on adaptive and hybrid SDA.

| References | Strategies | Applications |
|---|---|---|
| [16], [17] | ASDA: Adaptive spiral model radius and angle based on fitness cost. | Benchmark functions test, linear dynamic modelling of flexible manipulator system. |
| [18], [19] | HSDBF, HBCSD: Bacteria-chemotaxis phase followed by SDA (Collaborative type algorithm). | PD control for position tracking of flexible manipulator system. |
| [19] | HSBFA: Spiral model incorporated into BFA. | Linear dynamic modelling of flexible system. |
| Proposed methods | HSDBC-S, HSDBC-R: Integrated chemotaxis phase into SDA (Integrative type algorithm). | Benchmark functions test with statistical analysis and PD-like FLC control for flexible manipulator system. |

BFA on the other hand, is a well known bio-inspired optimisation algorithm introduced by Passino [6]. It is developed based on unique strategy of Escherichia Coli (E-coli) bacteria in exploring food or nutrient during the whole of their life cycle within lower-intestine or warm-blooded organisms for their survival [20]. Three important phases of E-coli bacteria strategy which form a BFA are chemotaxis, reproduction, and elimination and dispersal. These three phases are placed sequentially after one another. The chemotaxis phase is the first phase of BFA and it is the most prominent stage in BFA. Here, bacteria move in tumble and swim actions continuously. Tumble is where the bacteria move randomly one step ahead aroud their original position, trying to find a better location than the current one. If the new location of bacteria has more food or nutrient, then they continuously swim a few more steps in the same direction as tumble. On the contrary, if the new location has less food source than the current one, then the bacteria perform another tumble action looking for a new and better position. Depending on the amount of food source at this new location, the bacteria again swim in the same direction as tumble or perform another tumble action. This process is continuously repeated during the chemotaxis phase. The reproduction phase comes after the chemotaxis where at this stage, the population of bacteria is classified into two groups based on their health condition. Here, the healthier group is duplicated to preserve the best and important features of their predecessor. In terms of the optimisation algorithm point of view, the bacteria population is sorted in ascending order based on their fitness cost. The fittest bacterium is the one that has the least fitness cost. The first half of the bacteria population is considered as healthier compared to the remaining half of the bacteria population and they are copied exactly the same as the original population. Duplicating healthier group of bacteria increases the total number of bacteria in the population. From the algorithmic point of view, this leads to unfavour situation where it can increase the total computation cost of the algorithm. In order to keep the size of population the same as the original population, the less healthy bacteria are eliminated. This ensures that only the best and fittest bacteria remain in the population. The final step is to relocate or redistribute the whole fittest bacteria population within the feasible search area randomly. This is to give the opportunity for the bacteria to be placed closer to the global optimum solution in a faster way.

Since the introduction of BFA, a lot of modifications have been proposed by researchers worldwide to improve its performance. Moreover, there are various publications about the successful application of BFA to solve real world problems. These developments and successful application of BFA to solve real world problems have increased the popularity of BFA within the optimisation algorithm community. The modification of BFA can be divided into two main groups. The first group is the introduction of adaptive formulation to vary bacteria step size with respect to fitness cost or index of iteration. The second group is hybrid approach where the BFA is synergised with other metaheuristic algorithms.

Introduction of a powerful adaptive formulation into chemotaxis phase to change the bacteria step size can significantly improve BFA performance without increasing the complexity of the original BFA structure. [21] has conducted a comprehensive study investigating the convergence, stability and oscillation behaviour of BFA near the global optimum point. They have established a linear relationship between step size and deviation of individual bacterium from global best position. The algorithm has been tested with ten benchmark functions and applied to parameter estimation of frequency-modulated sound wave. [22] employed adaptive BFA to optimise forecasting model for prediction of various stock market indices. They have formulated a linear adaptive equation to vary bacteria step size based on the individual fitness cost without taking into account the global fitness cost in that particular iteration. Similar type of formulation was adopted by [23] to solve automatic circle detection problem for a digital image. [24] used delta modulation to change the bacteria step size where the deviation between actual and modulated signal is integrated and fed into a voltage regulator. The algorithm was employed to optimise parameters of a linear antenna array problem. [25] has proposed nonlinear adaptive formulation to change the step size based on the index and total number of chemotaxis. The algorithm was applied to solve economic dispatch problem with consideration to power losses and valve-point effects. [26] formulated an adaptive strategy for changing the step size with respect to the best fitness cost and current iteration number. The algorithm was tested with several standard benchmark functions. Another adaptive strategy was formulated by [27] where fitness cost and sine function were incorporated into the equation. The algorithm was used to optimise fuzzy entropy for segmentation of gray images. [28] proposed nonlinear adaptive formula based on exponential function in terms of current and maximum number of iterations. A certain range within $[C_{min}, C_{max}]$ for bacteria step size was introduced as they approach to the final search location. The algorithm was validated with various sets of parameters on two standard benchmark functions. Another strategy to vary the bacteria step size can be made based on intelligent approach. [29] has adaptively varied the bacteria step length through Takagi-Sugeno fuzzy logic approach based on minimum value of fitness cost. The algorithm was applied to estimate harmonic component of power system waveforms. [30] has proposed fuzzy adaptive BFA to optimise congestion management for a power generator. The authors have defined a total number of 52 fuzzy rules where the input and output for the fuzzy system were the production cost and the change of step size respectively. [31] have utilised the Mamdani-type fuzzy rule to vary bacteria step size based on absolute value of cost function. The algorithm was tested with several benchmark

functions and showed better performance compared to the original BFA. A summary of related works on adaptive types based on BFA is shown in Table II.

Tabel II: Summary of related works on adaptive BFA approach.

| References | Adaptive strategies |
|---|---|
| [21], [22], [23], [27] | Adaptive equation to vary bacteria step size based on fitness cost. |
| [24] | Bacteria step size based on problem (deviation between actual and modulated signal). |
| [25], [28] | Adaptive formulation based on index and iteration number. |
| [26] | Bacteria step size based on combination of fitness cost and iteration number. |
| [29], [30], [31] | Adaptive formulation based on fuzzy logic approach. |

The synergy of BFA and PSO is the most commonly found hybrid BFA in the literature. [32], [33], [34], [35] and [36] utilised chemotaxis strategy in BFA to serve as local search method while the velocity and position update equations in PSO were employed to act as global search method. [37] introduced a simplified version of BFA employing bacterial chemotaxis strategy and PSO velocity update equation to solve parameter identification problem of a heavy oil thermal cracking model. Reproduction and elimination stages were omitted to reduce computational time. [38], [39] and [40] introduced hybrid GA-BF algorithm by applying varying bacterial chemotaxis step size in BFA while employing modified mutation and crossover operation in GA. [41] developed cooperative BF-TS by combining adaptive BFA (ABFA) and adaptive tabu search (ATS). With limited exploration capability of ATS in the search space and complexity of ABFA, to provide suitable exploration at the early stage, the chemotaxis strategy of ABFA is incorporated into ATS. On the other hand, [42] used hybrid ABFA and ATS called BTSO, to analyse Lyapunov's stability of linear and nonlinear systems. [43] proposed an intelligent biogeography-based optimisation (BBO) where a BFA is combined with BBO. In the algorithm, to determine a valid migration of an individual from one place to another, chemotaxis behaviour of bacteria is adopted into BBO migration process. This ensures the island that receives the migrated solution and preserves its fitness level by only accepting individuals that contribute to a better fitness value. [44] introduced a hybrid algorithm namely BPSO-DE by synergising BFA, PSO, and DE to solve dynamic economic dispatch problem with valve-points effect. PSO-DE features contain evolutionary operators and velocity update equation, and are used to perform exploration search over the entire search space while bacterial chemotaxis strategy with adaptive step-size in BFA is used to perform local search to enhance exploitation. [45] introduced a chemotaxis differential evolution where a BFA is hybridised with DE algorithm. In the algorithm, the mutation process in DE is combined with the chemotaxis strategy of bacteria. A summary of related works on hybrid type based on BFA is shown in Table III.

Table III: Summary of related works on hybrid BFA approach.

| Hybrid types | References | Strategies |
|---|---|---|
| BFA-PSO | [32], [33], [34], [35], [36] | <ul><li>Exploitation: chemotaxis strategy of BFA.</li><li>Exploration: velocity and position update equation of PSO.</li><li>Incorporating velocity and position update equation of PSO into BFA.</li><li>Enhancing exploration capability of BFA.</li></ul> |
| Simplified BFA-PSO | [37] | <ul><li>Excluded reproduction, elimination and dispersal events.</li></ul> |
| BFA-GA | [38], [39], [40] | <ul><li>Exploitation: chemotaxis strategy of BFA.</li><li>Exploration: modified mutation and crossover operations of GA.</li><li>Synergising modified mutation and crossover operations of GA with chemotaxis strategy of BFA.</li><li>Enhancing the exploitation capability of GA.</li></ul> |
| BFA-TS | [41], [42] | <ul><li>Exploitation: ATS features.</li><li>Exploration: ABFA features.</li><li>Synergising ABFA and ATS features.</li><li>Enhancing exploitation capability of BFA.</li></ul> |
| BFA-BBO | [43] | <ul><li>Exploitation: chemotaxis strategy of BFA.</li><li>Exploration: BBO features.</li><li>Incorporating chemotaxis strategy of BFA into BBO.</li><li>Enhancing exploitation capability of BBO.</li></ul> |
| BFA-PSO-DE | [44] | <ul><li>Exploitation: chemotaxis strategy with adaptive step-size of BFA.</li><li>Exploration: PSO-DE features.</li><li>Incorporating PSO-DE operators into BFA.</li><li>Enhancing exploration capability of BFA.</li></ul> |
| BFA-DE | [45] | <ul><li>Exploitation: chemotaxis strategy of BFA.</li><li>Exploration: DE features.</li></ul> |

| | | • Incorporating chemotaxis strategy of BFA into DE. |
| | | • Enhancing exploitation capability of DE. |

Flexible robot manipulators are used in the manufacturing industry as tools in the production process. Unlike a rigid manipulator, a flexible manipulator is lighter in weight, has smaller actuator, faster response, consumes less power, is less expensive, operates cost-efficiently, has higher payload to robot weight ratio and offers more safety to the user [46],[47]. A single-link flexible manipulator is considered in this work. This is a single-input multi-output system comprising rigid and flexible dynamics. An electromechanical actuator at the hub of the system produces rotational motion of the manipulator, and fast motion-induced vibration propagates along the flexible beam to the payload at the end-point of the beam. The control strategy for the flexible manipulator system can be realised through conventional control [48], [49] or intelligent control [50], [51], [52], [53] approaches. Fuzzy logic control (FLC) is one type of intelligent control approaches that increasingly attracts interest of researchers. The nonlinear control laws of FLC are more robust compared to conventional feedback control when dealing with unprecision and uncertainty of the controlled system. A more promising intelligent control approach to handle higher degree of uncertainty and information deficiency is through utilisation of interval type-2 fuzzy logic [61]. It is an extended and improved version of a classical type fuzzy logic. Interval type-2 fuzzy logic has been extensively used to solve complex problems in various applications. [68] has reported that it has been applied in the area of pattern recognition [69], system identification [70], theory [71], prediction [72] and classification [73]. In the area of control design, it has been extensively applied in a noisy and changing environment [62], [63], [64] and [65]. The drawback of FLC is the need for human expert to formulate linguistic description of fuzzy rules before they can be converted into a control signal to control a system. Moreover, training of such system is relatively complex due to large number of tuning parameters [78]. However, this limitation can be solved by applying an optimisation algorithm as a tuning tool, and this can lead to better performance than classical fuzzy tuning approach. In [66], the authors revealed that GA, PSO and ACO were the commonly used algorithms to optimise interval type-2 fuzzy in the area of intelligent control. However, the application of GA has been significantly reduced. [67] used a chemical optimisation algorithm to optimise interval type-2 fuzzy for the application of autonomous mobile robots. The authors pointed out that the interval type-2 FLC performed better than classical FLC in the presence of higher degree of uncertainty. [74] applied PSO and GA to optimise membership functions of interval type-2 FLC for a real-time application to control speed of a DC motor based on FPGA. The authors concluded that PSO outperformed GA in terms of accuracy and resulted in faster runtime. Another promising control approach for controlling a flexible system is feedforward input command shaping introduced by [54]. It is a common approach to suppress vibration of a flexible system where it cancels resonance poles of a system by convolving a desired command with a sequence of impulses. However, through conventional design approach for the input command shaping, information about the system characteristics such as resonance frequencies and corresponding damping ratios must be known prior to controller design [55]. This limitation can be overcome efficiently if an optimisation algorithm is employed in the design where it can produce optimum solution even if no prior knowledge about the system is known. In the application part of this paper, the SDA, BFA and the proposed optimisation algorithms are employed to design and optimise a hybrid PD-like FLC for position tracking of a single-link flexible robot manipulator.

This paper is an extended study of previously published work at a conference, where the hybrid spiral-dynamics bacteria-chemotaxis algorithms were introduced [56], [57]. The proposed algorithms offer different strategies compared to the existing hybrid type presented in the literature as shown in Table I. First, it is an integrative type algorithm where the chemotaxis phase of BFA is integrated into SDA. Unlike the collaborative type structure as proposed by [18] and [19], the combined features of integrative type struture are executed in a parallel manner. Both chemotactic strategy and SDA are executed at the same time to balance the exploration and exploitation. In the strategy, SDA is the main and host algorithm where it provides platform for the chemotaxis phase of BFA. Second, the reproduction, elimination and dispersal phases of BFA are excluded so that the simple structure of SDA can be retained to achieve fast convergence speed and reduced computation time. It is different than the HSBFA in [19], which only adopted spiral equation into the chemotaxis phase of BFA and executed BFA as a whole algorithm, requiring more processing time than the original BFA. The proposed algorithms have simpler structure and shorter execution time than the original BFA. Third, the spiral radius and angle of the spiral model is determined based on fitness cost of each bacterium. It offers more dynamic step size compared to the original SDA.

The development of the proposed algorithms has been motivated by the following issues. BFA provides good exploration strategy due to the chemotaxis approach but suffers from oscillation towards the end of its search process if large step size is used. It offers fast convergence speed but low accurate solution. On the contrary, for a small step size, it offers good exploitation and better accuracy but with slow convergence. There is unbalanced combination between the exploration and exploitation, and this makes the algorithm unable to produce both accuracy and faster convergence speed at the same time. Moreover, the total computation time for

the algorithm to complete the whole search operation is relatively high due to its complex structure. This situation is not favoured especially when it is adopted for solving a complex real world problem, implementation on a real-time application such as on-line tuning of controller parameters or dynamic modelling of a real physical system. In such situations, it requires high speed computing machine, which is very costly. On the other hand, SDA provides better stability when approaching the optimum point due to dynamic spiral step in its trajectory motion, has faster convergence speed and reduced computation time due to a relatively simple structure. However, it may potentially cause the search agents to get trapped into local optima due to insufficient exploration strategy throughout the search space and deterministic approach of the spiral trajectory thus offers low accurate solution at the end. The objectives of the proposed algorithms are to improve the performance of the original BFA and SDA in terms of both convergence speed and fitness accuracy as well as to reduce the complexity of the original BFA structure for shorter computation time.

Statistical analyses of the proposed algorithms tested on various multi-modal, unimodal and different landscapes of benchmark functions are presented in the paper. The performance of the proposed algorithms on the benchmark functions is compared with the predecessor algorithms through non-parametric Friedman and Wilcoxon signed rank tests and parametric *t*-test analysis. Moreover, the paper presents an application of the hybrid schemes in optimising a hybrid intelligent control technique for a laboratory-scale flexible manipulator system. Classical type FLC is applied in this work rather than interval type-2 FLC since no payload is considered and the presence of relatively lower degree of uncertainty for controlling the hub angle position of the flexible manipulator system. It also provides a challenging and complex platform with fewer tuning parameters and faster computing time to assess the performance of the proposed algorithms in comparison to their predecessor algorithms.

The rest of the paper is organised as follows. Detailed description of the proposed algorithms is presented in section 2. Section 3 discusses the implementation and associated results of the proposed algorithms on a set of benchmark functions. Section 4 presents a description of the single-link flexible robot manipulator, the method used to design hybrid PD-like FLC for hub-angle position tracking and the corresponding results. The conclusions drawsn from the work presented in the paper are given in section 5.

## 2. HYBRID SPIRAL-DYNAMIC BACTERIA-CHEMOTAXIS ALGORITHMS

This section describes the proposed hybrid spiral-dynamic bacteria-chemotaxis (HSDBC) algorithms. The HSDBC is a combination of bacterial chemotaxis strategy used in BFA and SDA. It preserves the strengths and eliminates the weaknesses of both SDA and BFA algorithms. The HSDBC algorithm is divided into two types based on the swimming strategy of bacteria. In the first type namely HSDBC-S, bacteria swim through spiral approach while in the second type namely HSDBC-R the bacteria adopt a random approach.

The parameters and description used in *n*-dimensional HSDBC optimisation algorithm are presented in Table IV. Step-by-step description of the algorithms is presented in Figure 1. To get a clearer picture, the HSDBC-S and HSDBC-R flowcharts are shown in Figure 2(a) and Figure 2(b) respectively. In the proposed hybrid approach, bacterial chemotaxis strategy is employed in step 2 to balance and enhance exploration and exploitation of the search space.

Table IV: Parameters for HSDBC-S and HSDBC-R algorithms.

| Symbols | Descriptions |
|---------|--------------|
| $\theta_{tumble}$ | Bacteria angular displacement on $x_i - x_j$ plane around the origin for tumbling. |
| $\theta_{swim}$ | Bacteria angular displacement on $x_i - x_j$ plane around the origin for swimming. |
| $r_{tumble}$ | Spiral radius for bacteria tumble. |
| $r_{swim}$ | Spiral radius for bacteria swim. |
| $m$ | Number of search points. |
| $k_{max}$ | Maximum iteration number. |
| $N_{sw}$ | Maximum number of swim. |
| $x_i(k)$ | Bacteria position. |
| $C$ | Bacteria step size. |
| $R^n$ | *n x n* square matrix. |

<div style="border:1px solid">

**Step 0: Preparation**

Select the number of search points (bacteria) $m \geq 2$, parameters $0 < \theta_{tumble}, \theta_{swim} \leq 2\pi$, $0 < r_{tumble}, r_{swim} \leq 1$ of $S_n(r, \theta)$, maximum iteration number, $k_{\max}$ and maximum number of swims, $N_s$ for bacteria chemotaxis. Set $k = 0$, $s = 0$.

Step 1: Initialisation

Set initial points $x_i(0) \in R^n$, $i = 1, 2, ..., m$ in the feasible region at random and centre $x^*$ as

$x^* = x_{i_g}(0)$, $i_g = \arg\min_i f(x_i(0))$, $i = 1, 2, ..., m$.

**Step 2: Bacteria chemotaxis**

(i) Bacteria tumble

(a) Update $x_i$

$x_i(k+1) = S_n(r_{tumble}, \theta_{tumble})x_i(k) - (S_n(r_{tumble}, \theta_{tumble}) - I_n)x^*$, $i = 1, 2, ..., m$.

(ii) Bacteria swim

(a) Check number of swims for bacteria $i$.

If $s < N_s$, then check fitness,

Otherwise set $i = i + 1$, and return to step (i).

(b) Check fitness

If $f(x_i(k+1)) < f(x_i(k))$, then update $x_i$,

Otherwise set $s = N_s$, and return to step (i).

(c) Update $x_i$

$x_i(k+1) = S_n(r_{swim}, \theta_{swim})x_i(k) - (S_n(r_{swim}, \theta_{swim}) - I_n)x^*$, $i = 1, 2, ..., m$ for HSDBC-S

$x_i(k+1) = x_i(k+1) + step\ size \times random\ direction$, $i = 1, 2, ..., m$ for HSDBC-R

**Step 3: Updating $x^*$**

$x^* = x_{i_g}(k+1)$, $i_g = \arg\min_i f(x_i(k+1))$, $i = 1, 2, ..., m$.

**Step 4: Termination criterion**

If $k = k_{\max}$ then terminate. Otherwise set $k = k + 1$,

and return to step 2.

</div>

**Fig. 1. Description of the proposed hybrid spiral-dynamics bacteria-spiral-chemotaxis and bacteria-random-chemotaxis algorithms.**

*2.1.    HYBRID SPIRAL-DYNAMIC BACTERIA-SPIRAL-CHEMOTAXIS*

The bacteria move from low nutrient location towards higher nutrient location, placed at the centre of a spiral. The most important factor of HSDBC-S algorithm is the respective diversification and intensification at the early phase and later phase of the spiral motion. In the diversification phase, bacteria are located at low nutrient location and move with larger step size thus producing faster convergence. On the other hand, in the intensification phase, bacteria are approaching rich nutrient location and move with smaller step size hence eliminating oscillation around the optimum point. However, to avoid the bacteria from getting trapped into local optima points, swim action is adopted in step 2(ii), Figure 1. Bacteria continuously swim in spiral form towards optimum point if the next location has higher nutrient value compared to the previous location until the maximum number of swims is reached. In the swim action, the spiral radius value is not necessarily the same as in tumble. Giving larger or smaller radius enhances the exploration and exploitation strategy of the bacteria respectively. More importantly, the exploration and exploitation are kept alternated during the search operation. Instead of having constant radius or constantly varying step size, another spiral radius value or adaptive spiral radius is adopted in the swim action. It may introduce variation in the spiral radius itself and step size of the bacteria for the whole search operation. Moreover, through the adaptive approach, the determination of the spiral radius for swim is based on the fitness cost of each bacterium. This in turn gives more chances for the bacteria to move within the feasible space hence resulting in more accurate solution. Beside, the variations of the spiral radius through the adaptive approach can control the convergence speed of the algorithm. The convergence speed may also be improved if the average size of spiral radius is employed in the swim action.

Furthermore, incorporating only chemotaxis part of bacteria and adopting adaptive radius, the simple structure of SDA can be retained.
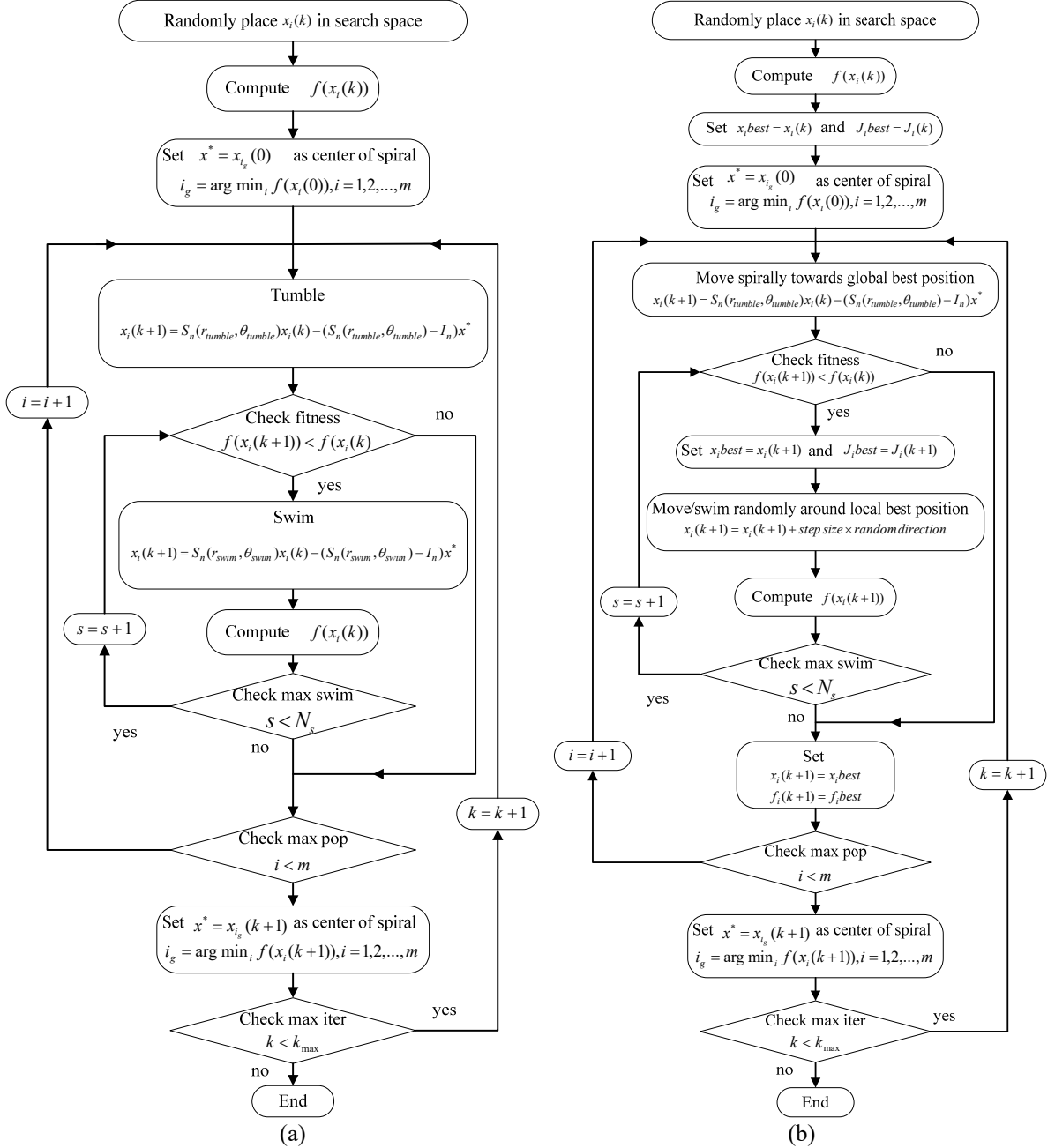


Fig. 2. Flowchart of the proposed hybrid algorithms, (a) Hybrid spiral-dynamic bacteria-spiral-chemotaxis algorithm and (b) Hybrid spiral-dynamic bacteria-random-chemotaxis algorithm.

## 2.2. HYBRID SPIRAL-DYNAMIC BACTERIA-RANDOM-CHEMOTAXIS

Instead of swimming in spiral manner where a deterministic approach is adopted, in HSDBC-R the bacteria swim in random direction, within 360° around their current position giving more flexibility to the bacteria in finding better path. Here, a combination of deterministic and random approaches is adopted during the tumble and swim respectively. Combination of these two approaches will complement each other where in the spiral approach, the trajectory of the bacteria is predetermined by the radius and the angle while in the random approach, the bacteria move freely in any direction. In terms of step size, the deterministic approach provides dynamic step size, which may balance the constant step size in the random approach. Moreover, in the tumble phase, the bacteria are guided by global best position, and this can speed up the convergence while in the swim phase, the bacteria move individually and independent to any other positions. A constant step size is adopted

8

for the swim action as shown in step 2(ii)(c) of Figure 1. Since the SDA has problem with local optima due to insufficient exploration, defining a large step size in the swim action can help the algorithm to explore more efficiently. Defining small step size helps the algorithm to improve the exploitation part. Instead of having the exploitation at the end of search operation only, here, additional exploitation in the swim with random trajectory is included, and it alternates repetitively with the exploration part in the tumble action. The HSDBC-R flowchart is depicted in Figure 2(b). Notice that, after the bacteria move one-step ahead using the spiral model, they perform random tumble and swim actions without referring to the global best position. Moreover, an additional step is incorporated into the chemotaxis phase through the introduction of *fbest* and *xbest*. This is to ensure that the best solution and best fitness are always preserved when the bacteria move from one location to another when performing tumble and swim actions. In addition, it eliminates the oscillation problem in the algorithm convergence.

## 3. EXPERIMENTS AND RESULTS WITH BENCHMARK FUNCTIONS

This section presents an extensive study on the performance of the proposed hybrid algorithms in comparison to the original SDA and BFA with several well-known benchmark functions.

### 3.1. BENCHMARK FUNCTIONS

Eight well-known benchmark functions with different fitness landscapes, dimensions and complexities are used in this work. The mathematical representations and descriptions of the benchmark functions are shown in Table V [21]. The functions $f_1$-$f_6$ have the global optimum value of zero while the functions $f_7$ and $f_8$ have the global optimum values of 3 and -1.0316 respectively. All the functions used in the test are continuous and differentiable.

Table V: Mathetical representation and description for the benchmark functions.

| Functions | Mathematical representations and descriptions | Ranges | Theoretical global optima |
|---|---|---|---|
| Sphere ($f_1$) | $f_1(x) = \sum_{i=1}^{p} x_i^2$ <br><br> continuous, differentiable, separable, scalable, unimodal | $(-5.12, 5.12)^p$ | $f_1(0) = 0$ |
| Rosenbrock ($f_2$) | $f_2(x) = \sum_{i=1}^{p-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ <br><br> continuous, differentiable, non-separable, scalable, unimodal | $(-5, 10)^p$ | $f_2(1) = 0$ |
| Ackley ($f_3$) | $f_3(x) = -20\exp\left(-20\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right) + 20 +$ <br><br> continuous, differentiable, non-separable, scalable, multimodal | $(-15, 30)^p$ | $f_3(0) = 0$ |
| Rastrigin ($f_4$) | $f_4(x) = \sum_{i=1}^{p} [x_i^2 - 10\cos(2\pi x_i) + 10]$ <br><br> continuous, differentiable, separable, scalable, multimodal | $(-5, 5)^p$ | $f_4(0) = 0$ |
| Griewank ($f_5$) | $f_5(x) = \frac{1}{4000}\sum_{i=1}^{p} x_i^2 - \prod_{i=1}^{p}\cos\frac{x_i}{\sqrt{i}} + 1$ <br><br> continuous, differentiable, non-separable, scalable, multimodal | $(-600, 600)^p$ | $f_5(0) = 0$ |
| Dixon & Price ($f_6$) | $f(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$ <br><br> continuous, differentiable, non-separable, scalable, unimodal | $(-10, 10)^p$ | $f_6(2^{-\frac{2^i - 2}{2^i}}) = 0$ |
| Goldstein & Price ($f_7$) | $f_7(x) = \{1 + (x_0 + x_1 + 1)^2(19 - 14x_0 + 3x_0^2 - 14x_1 - 6x_0x_1 + 3x_1^2)\}$ <br> $\{30 + (2x_0 - 3x_1)^2 \times (18 - 32x_0 + 12x_0^2 + 48x_1 - 36x_0x_1 + 27x_1^2)\}$ <br> continuous, differentiable, non-separable, non-scalable, multimodal | $(-2, 2)^2$ | $f_7(0, -1) = 3$ |
| Hump ($f_8$) | $f_8(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^6$ <br><br> continuous, differentiable, non-separable, non-scalable, multimodal | $(-5, 5)^2$ | $f_8(0.0898, -0.71)$ <br> $= -1.0316$ |

*3.2.* COMPARATIVE TEST STRATEGY

The main aim of the benchmark functions test is to acquire statistical data which can be used to compare the performances of the proposed algorithms with those of the original algorithms. The tests focus mainly on these performance metrics: 1) The fitness accuracy achieved by the algorithm at the last fitness evaluation (FE) which may also be considered as the closeness of the solution to the global optimum point. 2) The speed of the algorithm to converge to the global optimum point in terms of the number of FEs. 3) The frequency of the final solution to reach the global optimum point. In this work, the dimensions for both $f_7$ and $f_8$ are defined as 2 while dimensions 15, 30, 45 and 60 are tested for function $f_1$-$f_6$.

In these tests, 30 independant runs were executed for each algorithm under study and, the mean and standard deviation values were generated. Moreover, the best and the worst values achieved by each algorithm are also presented as these can show the closeness of the final solution to the theoretical global optima. The number of total iterations for each algorithm is defined based on the total number of FEs which is set the same for all algorithms. Here, the maximum number of FEs is set to 80,000. The performances of the proposed algorithms and the original algorithms in terms of fitness accuracy or significant improvement are compared by conductiong Friedman and Wilcoxon tests and *t*-test on the best mean values achieved by the proposed algorithms and the BFA or SDA. The alpha value of 5% degree of significance is defined for all tests giving 95% confidence interval for the generated results. Following the work of [58], symmetrical initilisation method is used for all the problems. The initial parameter used in the test for the BFA, SDA, HSDBC-R and HSDBC-S are presented in Table VI. The initial parameter values for the proposed hybrid algorithms were selected heuristically while the parameters for the BFA were adopted from [6], [32] and the SDA parameters were adopted from [13].

Table VI: Initial parameter values for BFA, SDA, HSDBC-R and HSDBC-S.

|  | BFA | SDA | HSDBC-R | HSDBC-S |
|---|---|---|---|---|
| Parameter | $S=50$<br>$N_s=4$<br>$N_c=100$<br>$N_{re}=4$<br>$N_{ed}=2$<br>$C=0.1$ | $m=50$<br>$\theta=0.95$<br>$r=\pi/4$ | $S=50$<br>$N_s=4$<br>$\theta=0.95$<br>$r=\pi/4$<br>$C=1$ | $S=50$<br>$N_s=4$<br>$\theta_{tu}=0.95$<br>$r_{tu}=\pi/4$<br>$\theta_{sw}=0.99$<br>$r_{sw}=2$ |

*3.3.* EMPIRICAL RESULTS

The analytical results for the benchmark functions tests representing the mean, standard deviation (SD), best and worst accuracy are presented in Table VII. The best mean among the algorithms is highlighted in bold font, while the best mean between the SDA and BFA is italicised. Notice that, both the proposed hybrid algorithms achieved better performance compared to the SDA and BFA for all benchmark functions. Except only for the 15 dimensions Rosenbrock function, the BFA achieved the best mean among the algorithms. Comparing the proposed hybrid algorithms, it is noted that HSDBC-S dominated the best mean result. The HSDBC-R had better mean than HSDBC-S for lower order 15 and 30 dimensions Rastrigin function. Comparing the original algorithms, it can be seen that SDA dominated the best mean result especially for functions $f_1$, $f_3$, $f_4$, $f_5$, and $f_6$. BFA achieved better mean for functions $f_2$, $f_7$ and $f_8$ compared to SDA. On the other hand, for the SD, the proposed HSDBC-S dominated the results especially for functions $f_2$, $f_3$, $f_5$, $f_7$ and $f_8$. BFA dominated the SD result for functions $f_1$ and $f_6$ while the HSDBC-R had better SD for function $f_4$. The best accuracy is the best value achieved by the algorithms to the targeted global optima location within 80,000 FEs. It is noted that HSDBC-S dominated the best accuracy for functions $f_2$, $f_3$, $f_6$, $f_7$ and $f_8$. Both the proposed hybrid algorithms shared the domination for the function $f_1$ while for functions $f_4$ and $f_5$, HSDBC-R, HSDBC-S and SDA shared the best performance. Overall, for the results presented in Table VII, HSDBC-S showed the best performance compared to all other algorithms.

Table VII: Analytical results for BFA, SDA, HSDBC-R, HSDBC-S tested on benchmark functions.

| No | Function | Dim | Accuracy | SDA | BFA | HSDBC-S | HSDB-R |
|---|---|---|---|---|---|---|---|
| 1 | Rastrigin, $f_1$ | 15 | Mean | *56.2282* | 71.7664 | 49.8473 | **44.5291** |
|  |  |  | SD | 21.4073 | 10.0497 | 29.8361 | 13.9978 |
|  |  |  | Best | 22.0649 | 42.4882 | 14.9244 | 22.0333 |
|  |  |  | Worst | 102.8827 | 87.0167 | 100.4904 | 70.7086 |
|  |  | 30 | Mean | *177.3411* | 227.8996 | 145.0974 | **141.4004** |
|  |  |  | SD | 42.0284 | 15.2076 | 40.7532 | 36.2115 |
|  |  |  | Best | 109.7006 | 191.0726 | 77.6066 | 71.3614 |
|  |  |  | Worst | 256.3584 | 255.9851 | 221.8472 | 215.0208 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 45 | Mean | *326.5830* | 399.5583 | **261.9378** | 281.4467 |
| | | | SD | 51.6715 | 21.3011 | 58.4445 | 49.6190 |
| | | | Best | 184.5732 | 365.2128 | 167.1521 | 194.3177 |
| | | | Worst | 391.1996 | 442.6647 | 415.8885 | 376.3560 |
| | | 60 | Mean | *465.1590* | 583.9721 | **395.4618** | 420.6546 |
| | | | SD | 58.8316 | 27.2864 | 54.4317 | 69.4549 |
| | | | Best | 373.3858 | 507.5568 | 307.4419 | 277.7842 |
| | | | Worst | 601.6010 | 640.2427 | 548.2187 | 549.5155 |
| 2 | Sphere, $f_2$ | 15 | Mean | *0.0460* | 0.1453 | **$1.02 \times 10^{-16}$** | 0.0016 |
| | | | SD | 0.0985 | 0.0398 | $2.10 \times 10^{-16}$ | 0.0025 |
| | | | Best | $1.57 \times 10^{-10}$ | 0.0618 | $1.56 \times 10^{-18}$ | $1.10 \times 10^{-10}$ |
| | | | Worst | 0.4460 | 0.2139 | $9.29 \times 10^{-16}$ | 0.0105 |
| | | 30 | Mean | 11.3489 | *0.6566* | **$2.40 \times 10^{-9}$** | 0.1037 |
| | | | SD | 8.1754 | 0.1011 | $7.19 \times 10^{-9}$ | 0.0641 |
| | | | Best | 1.2464 | 0.4697 | $1.29 \times 10^{-11}$ | 0.0084 |
| | | | Worst | 33.5903 | 0.8894 | $3.32 \times 10^{-8}$ | 0.2625 |
| | | 45 | Mean | 44.5252 | *4.1923* | **$5.51 \times 10^{-7}$** | 0.4320 |
| | | | SD | 23.2000 | 0.9020 | $1.07 \times 10^{-6}$ | 0.1219 |
| | | | Best | 11.5869 | 2.8172 | $2.51 \times 10^{-8}$ | 0.2481 |
| | | | Worst | 109.8495 | 6.3640 | $4.49 \times 10^{-6}$ | 0.7359 |
| | | 60 | Mean | 59.6307 | *25.9662* | **$1.99 \times 10^{-4}$** | 0.6456 |
| | | | SD | 22.9318 | 4.4920 | $1.78 \times 10^{-4}$ | 0.1476 |
| | | | Best | 18.6768 | 15.1910 | $1.68 \times 10^{-5}$ | 0.4123 |
| | | | Worst | 121.5477 | 33.9146 | $9.38 \times 10^{-4}$ | 0.9314 |
| 3 | Griewank, $f_3$ | 15 | Mean | *0.2285* | 207.2228 | **0.0970** | 0.1621 |
| | | | SD | 0.2652 | 40.6498 | 0.0459 | 0.1051 |
| | | | Best | 0.0137 | 116.9222 | 0.0123 | 0.0082 |
| | | | Worst | 1.1990 | 290.1331 | 0.2213 | 0.3938 |
| | | 30 | Mean | *44.9101* | 572.9713 | **0.0136** | 8.6370 |
| | | | SD | 33.1576 | 42.4306 | 0.0113 | 16.1938 |
| | | | Best | 5.6035 | 475.3584 | $1.43 \times 10^{-7}$ | 0.0221 |
| | | | Worst | 128.4010 | 664.2639 | 0.0394 | 71.5929 |
| | | 45 | Mean | *157.2306* | 936.8041 | **0.0067** | 69.6809 |
| | | | SD | 101.6197 | 74.9055 | 0.0079 | 41.3909 |
| | | | Best | 25.4580 | 741.5544 | $3.65 \times 10^{-4}$ | 5.4645 |
| | | | Worst | 449.1162 | 1057.4 | 0.0348 | 140.6752 |
| | | 60 | Mean | *234.5367* | 1320.5 | **0.1123** | 45.6662 |
| | | | SD | 92.3632 | 96.7467 | 0.1049 | 33.4395 |
| | | | Best | 99.7027 | 1112.7 | 0.0194 | 2.5428 |
| | | | Worst | 463.9472 | 1484.7 | 0.5675 | 128.1832 |
| 4 | Goldstein & Price, $f_4$ | 2 | Mean | *3.0000* | 3.0790 | 3.0000 | **3.0000** |
| | | | SD | $2.80 \times 10^{-15}$ | 0.0765 | $2.23 \times 10^{-15}$ | **$1.34 \times 10^{-15}$** |
| | | | Best | 3.0000 | 3.0043 | 3.0000 | 3.0000 |
| | | | Worst | 3.0000 | 3.2865 | 3.0000 | 3.0000 |
| 5 | Hump, $f_5$ | 2 | Mean | *$4.65 \times 10^{-8}$* | 0.0013 | **$4.65 \times 10^{-8}$** | $4.65 \times 10^{-8}$ |
| | | | SD | $8.41 \times 10^{-17}$ | 0.0013 | **$5.63 \times 10^{-17}$** | $5.63 \times 10^{-17}$ |
| | | | Best | $4.65 \times 10^{-8}$ | $1.09 \times 10^{-6}$ | $4.65 \times 10^{-8}$ | $4.65 \times 10^{-8}$ |
| | | | Worst | $4.65 \times 10^{-8}$ | 0.0063 | $4.65 \times 10^{-8}$ | $4.65 \times 10^{-8}$ |
| 6 | Ackley, $f_6$ | 15 | Mean | *2.7560* | 17.2150 | **0.2733** | 0.4668 |
| | | | SD | 1.5520 | 0.8505 | 0.6686 | 0.5895 |
| | | | Best | 0.1578 | 14.2413 | $3.29 \times 10^{-9}$ | 0.0024 |
| | | | Worst | 7.6937 | 18.2857 | 2.3168 | 1.6462 |
| | | 30 | Mean | *10.3492* | 18.5364 | **1.0815** | 2.3763 |
| | | | SD | 2.9361 | 0.3884 | 0.9423 | 2.5354 |
| | | | Best | 5.6647 | 17.4426 | $2.67 \times 10^{-5}$ | 0.6873 |
| | | | Worst | 15.7385 | 19.1476 | 2.6605 | 15.4803 |
| | | 45 | Mean | *15.5079* | 18.8840 | **2.5513** | 5.3842 |
| | | | SD | 1.9447 | 0.2426 | 3.3276 | 5.3402 |
| | | | Best | 9.0870 | 18.3453 | 0.0033 | 1.7744 |
| | | | Worst | 17.9486 | 19.3364 | 18.650 | 18.1202 |
| | | 60 | Mean | *16.3385* | 19.1621 | **3.3505** | 9.7372 |
| | | | SD | 1.5981 | 0.1771 | 4.2232 | 6.5962 |
| | | | Best | 12.3176 | 18.7347 | 0.9355 | 2.5041 |
| | | | Worst | 18.5273 | 19.4170 | 18.7603 | 17.6647 |
| 7 | Dixon & Price, $f_7$ | 15 | Mean | 21.2076 | *2.4073* | **0.6713** | 1.0066 |
| | | | SD | 37.1136 | 0.7815 | 0.0176 | 0.5801 |
| | | | Best | 0.6690 | 1.5245 | 0.6667 | 0.6685 |
| | | | Worst | 138.1960 | 5.7020 | 0.7448 | 2.9900 |
| | | 30 | Mean | 30142 | *1644.7* | **0.8720** | 28.4950 |
| | | | SD | 63151 | 1163.2 | 0.3714 | 26.5510 |
| | | | Best | 368.8575 | 174.3992 | 0.6667 | 3.0957 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | Worst | 268500 | 5183.1 | 2.5058 | 102.0375 |
| | | 45 | Mean | 141120 | *115390* | **7.0515** | 74.5114 |
| | | | SD | 129630 | 33979 | 22.9881 | 75.2632 |
| | | | Best | 4865.4 | 36225 | 0.6686 | 17.8003 |
| | | | Worst | 607590 | 180080 | 125.8619 | 311.1949 |
| | | 60 | Mean | *285880* | 778280 | 19.6118 | 134.6677 |
| | | | SD | 228680 | 115660 | 36.0167 | 108.5157 |
| | | | Best | 17710 | 450890 | 0.9558 | 19.0963 |
| | | | Worst | 1091300 | 969210 | 190.8919 | 388.0243 |
| 8 | Rosenbrock, $f_8$ | 15 | Mean | 293.2540 | **28.3812** | *49.0480* | 49.9760 |
| | | | SD | 716.7687 | 3.6979 | 83.3906 | 62.0846 |
| | | | Best | 6.9860 | 21.5729 | 6.3186 | 7.9985 |
| | | | Worst | 3921 | 34.3244 | 338.6489 | 268.7396 |
| | | 30 | Mean | 15610 | *699.4071* | **54.0812** | 196.6493 |
| | | | SD | 18883 | 314.8351 | 66.0543 | 218.4484 |
| | | | Best | 618.4347 | 172.2762 | 23.5363 | 33.1709 |
| | | | Worst | 67199 | 1756.6 | 346.2318 | 857.5338 |
| | | 45 | Mean | 97738 | *30564* | **77.6202** | 398.2330 |
| | | | SD | 87041 | 13109 | 55.8222 | 319.0449 |
| | | | Best | 1581 | 8229.6 | 35.3515 | 156.3455 |
| | | | Worst | 326710 | 61866 | 242.9060 | 1691.7 |
| | | 60 | Mean | *133400* | 199390 | **135.3810** | 522.2300 |
| | | | SD | 75235 | 62914 | 197.8978 | 407.3238 |
| | | | Best | 27401 | 60315 | 52.0677 | 134.9322 |
| | | | Worst | 396650 | 321830 | 1122.0 | 1661.4 |

The performances of the four algorithms in terms of fitness accuracy were tested through nonparametric Frieman test and Wilcoxon signed rank test and the result is presented in Table VIII. Friedman test is used to rank the performance of three or more algorithms while the Wilcoxon signed rank test is used to compare the performance between two algorithms. The algorithm with lower rank in the Friedman test is considered as the algorithm with the highest accuracy achievement. In both tests, the algorithms are considered to have an equal performance if the two-tailed, $p$ value is equal to or more than 0.05 with 5% degree-of-significance. Notice that, based on Friedman test, HSDBC-S achieved the lowest ranking followed by HSDBC-R, SDA and BFA. In other words, HSDBC-S was the best algorithm in terms of accuracy achievement. Based on Wilcoxon signed rank test, it is noted that the $p$ value of SDA-BFA was bigger than 0.05, which means that the difference between BFA and SDA in terms of accuracy achievement was not significant and thus had equal performance. Wilcoxon test result for other than SDA-BFA showed that the $p$ value was less than 0.05. This, with reference to the Friedman test and $p$ value of Wilcoxon test, means that both HSDBC-R and HSDBC-S ouperformed BFA and SDA with significant improvement. The results also show that HSDBC-S had better performance than HSDBC-R with significant difference.

Table VIII: Result of Friedman test and Wilcoxon signed rank test.

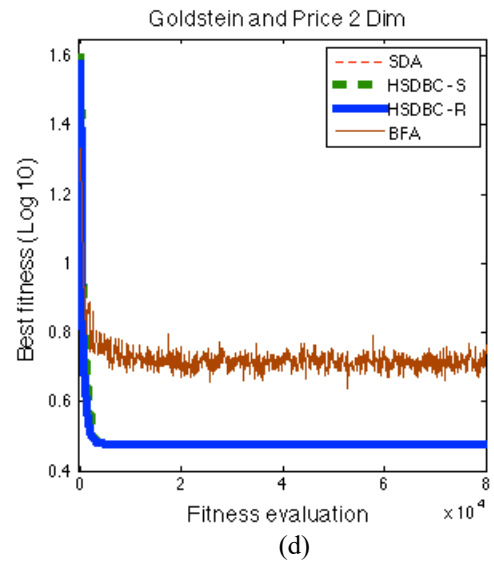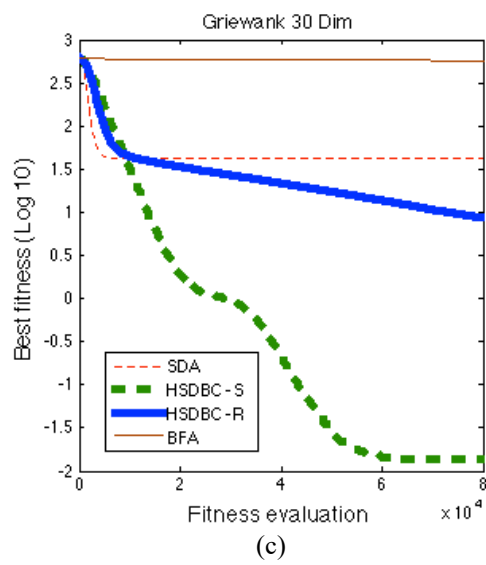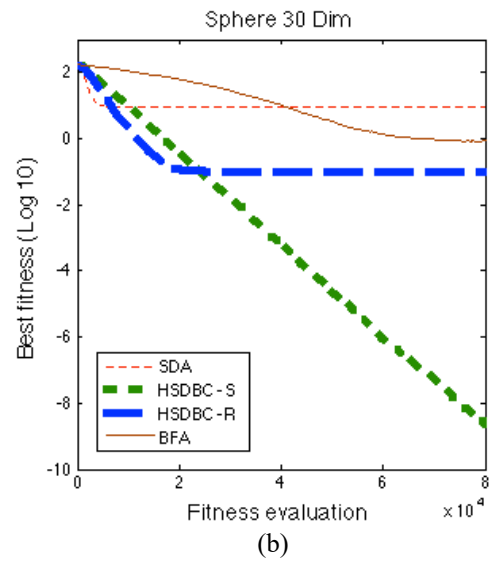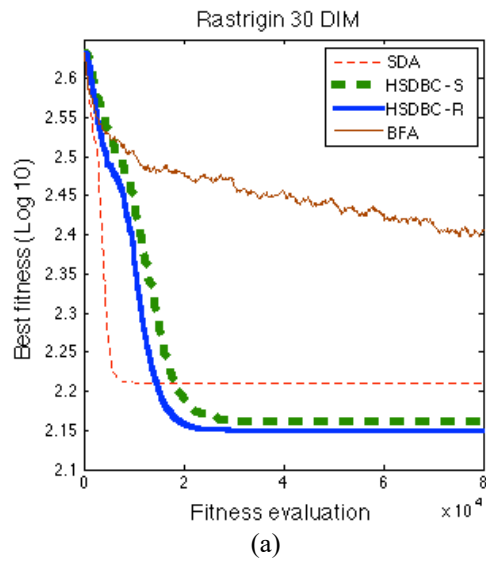| Algorithm | Friedman test | | | Wilcoxon signed rank test | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean rank | $p$ | $\chi^2$ | SDA-BFA | | | SDA-HSDBCS | | | SDA-HSDBCR | | |
| | | | | $R^+$ | $R^-$ | $p$ | $R^+$ | $R^-$ | $p$ | $R^+$ | $R^-$ | $p$ |
| SDA | 3.27 | | | 203 | 148 | 0.48 | 0 | 300 | <0.05 | 0 | 300 | <0.05 |
| BFA | 3.58 | <0.05 | 60.38 | BFA-HSDBCS | | | BFA-HSDBCR | | | HSDBCS-HSDBCR | | |
| | | | | $R^+$ | $R^-$ | $p$ | $R^+$ | $R^-$ | $p$ | $R^+$ | $R^-$ | $p$ |
| HSDBCS | **1.19** | | | 11 | 340 | <0.05 | 11 | 340 | <0.05 | 277 | 23 | <0.05 |
| HSDBCR | 1.96 | | | | | | | | | | | |

The significant difference between the best mean of the proposed algorithms (bold font) and the best mean of the original algorithms (italicised) from Table VII were evaluated through $t$-test approach. The overall result for the $t$-test representing the standard error, $t$-value, 95% confidence interval and two-tailed, $p$ value are shown in Table IX. As the degree of significace is 5% and degrees-of-freedom (DOF) is 58, if the two-tailed value is equal to or less than 0.05, then the mean difference is said to be significant. In other words, the result of the proposed algorithm was improved significantly compared to the standard algorithm. With reference to the result in Table VII, the BFA has achieved the best mean for 15 dimensions function $f_8$. However, the result of the $t$-test has shown that it has two-tailed value 0.1803, greater than 0.05, which may be interpreted as equal performance to HSDBC-S. Besides, for functions $f_4$ and $f_5$, the $t$-test has shown the two-tailed values above 0.05; these were 0.1232 and 0.2347 respectively. This result is aligned with the result

shown in Table VII. The rest of the results show that the two-tailed value was less than 5% degree of significance, which concludes that the performance difference of the proposed hybrid algorithms to the original algorithms is statistically significant and better. Notice that both functions $f_4$ and $f_5$ have low dimension of 2 and SDA performed very well giving equal performance to the hybrid schemes. However, for higher order dimensions on other functions, the performance of SDA remarkably reduced and might not be suitable to use as an efficient tool especially for solving real world problems.

Table IX: Results of *t*-test between the best mean of original algorithms and best mean of HSDBC.

| No | Function | Dim | Standard error | t - value | 95% Confidence interval | Two tailed, P | Significant improve. |
|----|----------|-----|----------------|-----------|-------------------------|---------------|----------------------|
| 1 | Rastrigin | 15 | 4.6698 | 2.5053 | 2.3516 - 21.0468 | 0.0151 | Yes |
| | | 30 | 10.1286 | 3.5484 | 15.6662 - 56.2154 | 7.75e-4 | Yes |
| | | 45 | 14.2428 | 4.5388 | 36.1351 - 93.1552 | $2.90 \times 10^{-5}$ | Yes |
| | | 60 | 14.6333 | 4.7629 | 40.4055 - 98.9889 | $1.31 \times 10^{-5}$ | Yes |
| 2 | Sphere | 15 | 0.0180 | 2.5570 | 0.0100 - 0.0820 | 0.0132 | Yes |
| | | 30 | 0.0185 | 35.5554 | 0.6196 - 0.6935 | $4.46 \times 10^{-41}$ | Yes |
| | | 45 | 0.1647 | 25.4570 | 3.8626 - 4.5219 | $3.62 \times 10^{-33}$ | Yes |
| | | 60 | 0.8201 | 31.6611 | 24.3244 - 27.6077 | $2.69 \times 10^{-38}$ | Yes |
| 3 | Griewank | 15 | 0.0491 | 2.6758 | 0.0331 - 0.2299 | 0.0097 | Yes |
| | | 30 | 6.0537 | 7.4163 | 32.7786 - 57.0143 | $5.85 \times 10^{-10}$ | Yes |
| | | 45 | 18.5531 | 8.4743 | 120.0858 -194.3620 | $9.87 \times 10^{-12}$ | Yes |
| | | 60 | 16.8631 | 13.9016 | 200.6692 - 268.1797 | $4.06 \times 10^{-20}$ | Yes |
| 4 | Goldstein & Price | 2 | 5.67e-16 | 1.5644 | -0.02e-14 - 0.20e-14 | 0.1232 | No |
| 5 | Hump | 2 | 1.84e-17 | -1.2008 | -0.59e-16 - 0.14e-16 | 0.2347 | No |
| 6 | Ackley | 15 | 0.3085 | 8.0464 | 1.8650 - 3.1003 | 5.12e-11 | Yes |
| | | 30 | 0.5630 | 16.4619 | 8.1408 - 10.3946 | $1.58 \times 10^{-23}$ | Yes |
| | | 45 | 0.7037 | 18.4127 | 11.5480 - 14.3651 | $6.68 \times 10^{-26}$ | Yes |
| | | 60 | 0.8244 | 15.7546 | 11.3379 - 14.6383 | $1.27 \times 10^{-22}$ | Yes |
| 7 | Dixon & Price | 15 | 0.1427 | 12.1647 | 1.4504 - 2.0217 | $1.33 \times 10^{-17}$ | Yes |
| | | 30 | 212.3715 | 7.7402 | 1218.7 - 2068.9 | $1.67 \times 10^{-10}$ | Yes |
| | | 45 | 6203.7 | 18.5986 | 102960 -127800 | $4.05 \times 10^{-26}$ | Yes |
| | | 60 | 41751 | 6.8467 | 202280 - 369430 | $5.30 \times 10^{-9}$ | Yes |
| 8 | Rosenbrock | 15 | 15.2399 | -1.3561 | -51.1729 - 9.8392 | 0.1803 | No |
| | | 30 | 58.7322 | 10.9876 | 527.7605 - 762.8912 | $8.43 \times 10^{-16}$ | Yes |
| | | 45 | 2393.4 | 12.7376 | 25695 - 35277 | $1.89 \times 10^{-18}$ | Yes |
| | | 60 | 1373.6 | 9.7017 | 105770 - 160760 | $9.42 \times 10^{-14}$ | Yes |

The performance of the algorithms on the benchmark functions test in terms of convergence speed might be observed via graphical approach. The convergence plots comparing all algorithms for 30 dimension functions $f_1$-$f_3$, $f_6$-$f_8$ and 2 dimension functions $f_4$ and $f_5$ are depicted in Figure 3. It is noted that SDA had the fastest speed but for 30 dimension problems it got trapped into the local optima position. The BFA on the other hand, had the slowest speed among other algorithms; hence for 80,000 fitness iterations it was unable to reach the global optimum point satisfactorily. On the contrary, using the proposed hybrid schemes, the search points settled down to a location nearer to the global optima solution thus giving better fitness accuracy, and the convergence speed was relatively slower compared to SDA but much better than BFA.

(a) Rastrigin 30 DIM

(b) Sphere 30 Dim

(c) Griewank 30 Dim

(d) Goldstein and Price 2 Dim

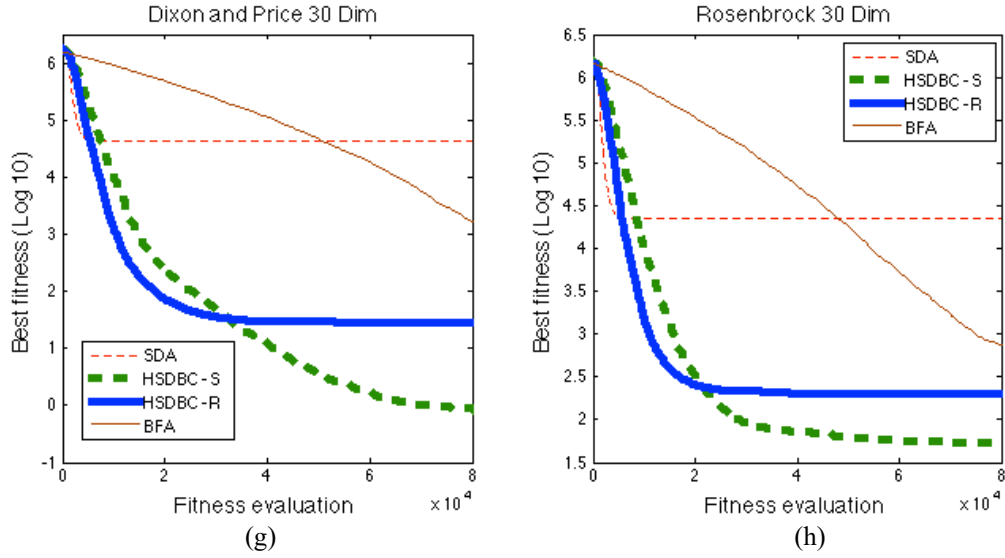(e) Hump 2 Dim

(f) Ackley 30 Dim
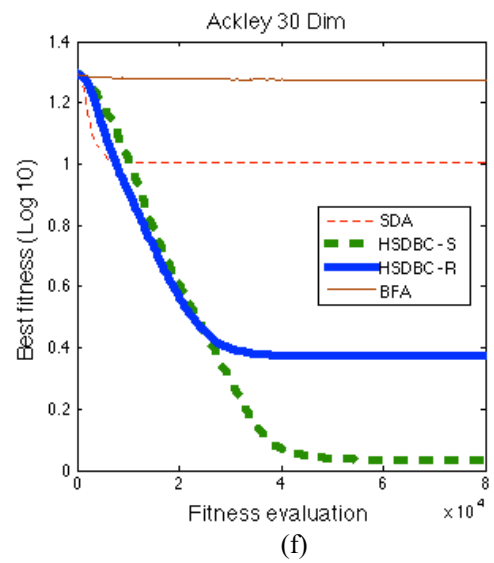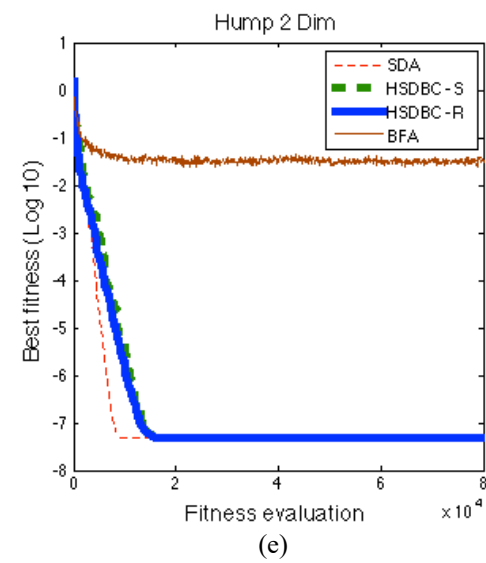
**Fig. 3. Convergence plot for BFA, SDA, HSDBC-S and HSDBC-R tested on benchmark functions, (a) Rastrigin function, (b) Sphere function, (c) Griewank function, (d) Goldstein and Price function, (e) Hump function, (f) Ackley function, (g) Dixon and Price function and (h) Rosenbrock function.**

## 4. DESIGN OF HYBRID INTELLIGENT CONTROLLER FOR FLEXIBLE ROBOT MANIPULATOR

This is a comparative test between the proposed hybrid algorithms and original algorithms to solve a real world problem in the engineering field.

### 4.1. EXPERIMENTAL RIG OF FLEXIBLE ROBOT MANPULATOR

A schematic representation of the single-link flexible robot manipulator is shown in Figure 4. The physical characteristics of the beam such as length, $l$ width, $w$ and thickness, $h$ are defined as 960 mm, 19.008 mm and 3.2004 mm respectively. The Young modulus, $E = 71 \times 10^9$, the second moment of inertia $I_b = 0.04862$ kg m$^2$, mass density/volume, $\rho = 2710$ kg m$^{-3}$ and hub inertia $I_h = 5.86 \times 10^{-4}$ kg m$^2$ are further specifications of the system.
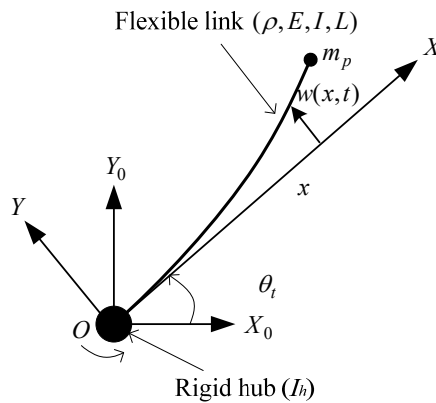


**Fig. 4. Schematic representation of single link flexible robot manipulator.**

The single-link flexible robot manipulator arm developed by [59] and [60] is used in this work and its schematic diagram is shown in Figure 5. The system consists of a single flexible link made of aluminum beam and attached to an electromechanical motor. A U9M4AT type printed circuit board with bi-directional drive amplifier is used to rotate the motor shaft in both counterclockwise and clockwise directions. As there are three outputs of interest to be gauged from the system, three different sensor units are attached to the body of system. An accelerometer is placed at the tip of the beam and used to measure the end-point acceleration while an encoder with a resolution of 2,048 pulses/revolution and a tachometer are attached to motor shaft and used to measure hub-angle and hub-velocity respectively. Moreover, a personal computer (PC) embedded with Pentium Celeron 500 MHz processor is connected with PCL818G interfacing unit to the flexible robot

manipulator arm. Matlab/Simulink software installed in the PC is used as a tool for controlling and manipulation of the system.
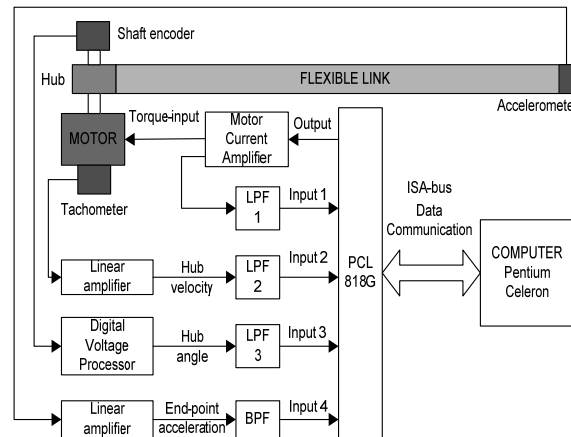


**Fig. 5.** Schematic diagram of single-link flexible robot manipulator rig [59], [60].

In general, there are one input and three output responses to be observed. Here, a random signal input is applied to the system and represented as electrical motor torque. The measured output responses of the system are hub-angle, hub-velocity and end-point acceleration. However, the hub-angle and end-point acceleration are the output responses of interest especially when dealing with controller design. In a real world application, the capability of the controller in dealing with position tracking problem of hub-angle and suppression of vibration at the end-point are mainly of concern. The flexible robot manipulator used in this study has three dominant resonance modes within the range [0-100] Hz as shown in Figure 6. Motion-induced vibration in the manipulator response appears and is dominated by the three resonance frequencies noted in Figure 6, while the system is in operation without vibration control.
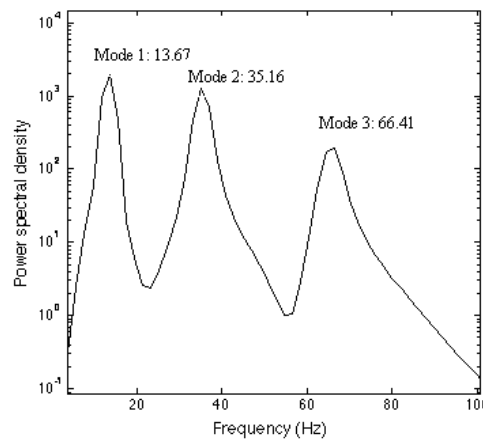


**Fig. 6.** Power spectral density for flexible robot manipulator.

### 4.2. PD-LIKE FUZZY-LOGIC CONTROL

In this work, PD-like FLC is implemented to track the hub angle position for a desired bang-bang input. The block diagram of PD-like FLC for a flexible system is depicted in Figure 7 [50]. Hub angle response is fed back and compared with the desired input. The difference between the actual and desired hub angle response is multiplied by a constant $k_1$ representing proportional value and scaling factor for fuzzy input 1. The hub velocity representing derivative of the hub angle is taken directly from the system and multiplied by a constant $k_2$ which is a derivative value or scaling factor for fuzzy input 2. Both of the outputs of the $k_1$ and $k_2$ are then injected into the input of the FLC. The output of the controller is then multiplied by another constant $k_3$ or scaling factor for fuzzy output before it passes as an input torque into the flexible robot manipulator. Considering processing time and the performance of the FLC, five triangular type membership functions or fuzzy sets are defined for both fuzzy inputs and fuzzy output giving 25 fuzzy rules in total. The membership functions for both inputs and output are uniformly distributed on the universe of discourse within a range of [-1, 1] as shown in Figure 8. The Mamdani-type with centre of area defuzzification method is used due to its

intuitiveness, widespread acceptance and suitability in dealing with human reasoning [31]. Besides, min type operator for implication and max type operator for aggregation are used in the scheme. Another important feature of fuzzy logic scheme is linguistic rule that comprises IF-THEN statements to establish relationship between antecedent and consequent. The general form of fuzzy logic linguistic rule for FLC is defined as:

$$\text{IF } \varepsilon \text{ is A AND } \upsilon \text{ is B THEN } \tau \text{ is C} \tag{1}$$

where A, B and C are linguistic values defined by fuzzy sets {negative big, negative small, zero, positive small, positive big}, represented by {NB, NS, Z, PS, PB}.The $\varepsilon$, $\upsilon$ and $\tau$ are the hub angle error, hub velocity and input torque to the flexible system respectively. The 'IF $\varepsilon$ is A AND $\upsilon$ is B' is known as antecedent while '$\tau$ is C' is known as consequent.
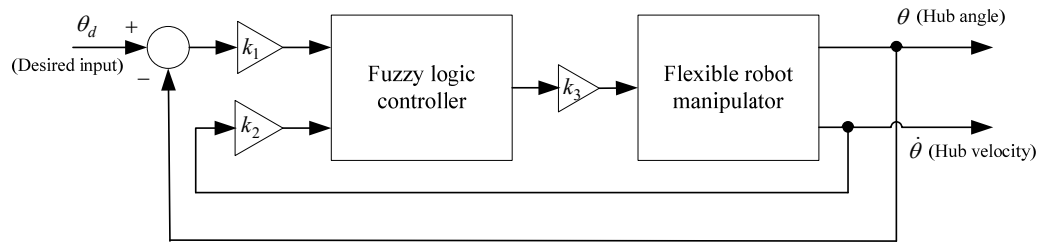


**Fig. 7. Block diagram of the PD-like fuzzy logic control.**
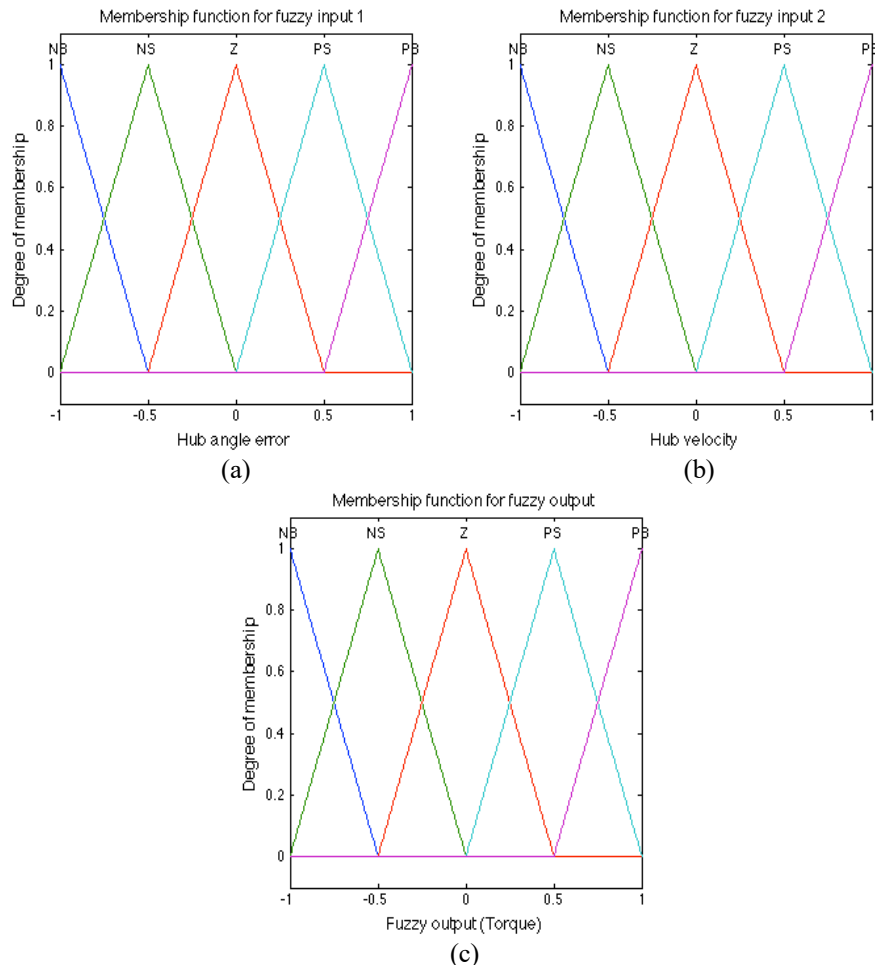


**Fig. 8. Membership functions, (a) Hub angle, (b) hub velocity and (c) fuzzy output (torque).**

17

*4.3. HSDBC-BASED PD-LIKE FUZZY-LOGIC CONTROL*

The application of an optimisation algorithm as a tuning tool makes the design of PD-like FLC more efficient. Unlike conventional tuning methods, here the parameters $k_1$, $k_2$ and $k_3$, the generation of rule base and the nonlinear mapping between the inputs and output are automatically determined and optimised by the optimisation algorithms.

In the controller optimisation, a total of 103 variables are defined. The first 75 variables represent the fuzzy sets for hub angle error, hub velocity and fuzzy output. The next 25 variables represent the fuzzy weight for each fuzzy rule. The last three variables in the first part are for scaling factors $k_1$, $k_2$ and $k_3$. In order to optimise the rule base, the linguistic variables of fuzzy set {NB NS Z PS PB} need to be transformed into numerical values as {1 2 3 4 5} repectively. Therefore, a range of [1, 5] is defined for the first 75 variables while a range [0, 1] is defined for the fuzzy weight. All the fractional numbers representing the fuzzy set are rounded to the nearest integer number. The '0' value for the fuzzy weight means that the rule is ignored while the fuzzy weight with value '1' means that the rule is fully used in generating the fuzzy output. This is a method of weighting a rule relative to other rules. The error function for tracking the hub angle used here is a mean of absolute error which is formulated as:

$$J = \frac{1}{N} \sum_{i=1}^{N} \left( \left| \theta_{act}(i) - \theta_{des}(i) \right| \right) \tag{2}$$

where N is the number of data points, $\theta_{act}(i) - \theta_{des}(i)$ is the error between the actual and desired signal. The initial parameter values for BFA, SDA, HSDBC-R and HSDBC-S used to optimise the controller for tracking hub angle position are shown in Table X. The parameter values for the algorithms were heuristically tuned for optimal performance.

Table X: Initial parameter values for BFA, SDA, HSDBC-R and HSDBC-S.

|  | BFA | SDA | HSDBC-R | HSDBC-S |
|---|---|---|---|---|
| Parameter | $S=50$<br>$N_s=4$<br>$N_c=50$<br>$N_{re}=4$<br>$N_{ed}=5$<br>$C=0.5$ | $m=50$<br>$\theta=\pi/4$<br>$r=0.95$ | $S=50$<br>$N_s=4$<br>$\theta=\pi/4$<br>$r=0.95$<br>$C=0.008$ | $S=50$<br>$N_s=3$<br>$\theta_{tu}=\pi/4$<br>$r_{tu}=0.95$<br>$\theta_{sw}=\pi/4$<br>$r_{sw}=0.65$ |

Table XI and Figure 9 show the gain value and fuzzy surface of the optimised PD-like FLC respectively. It is noted that the output torque obtained with the proposed schemes was higher compared to the original algorithms. This may affect the rise time and percentage overshoot of the hub angle response. The fuzzy rule of the optimised PD-like FLC is shown in Table XII. Notice that for the SDA-based rules, there is redundancy between rules 3 and 8 with the same weight. Rules 14, 18 and 19 also resulted the same linguistic scheme, but they had different weights. In this case, the rule with the highest weight (rule 18) is considered while all other rules with lower weights (rules 14 and 19) are ignored. Therefore, there are only 22 effective rules for the SDA-based FLC. For BFA-based FLC, rules 15 and 24 shared the same linguistic scheme but with different weights hence the effective number of rules were 24. For HSDBC-R-based FLC, rules 3 and 10, rules 14 and 24, and rules 17 and 25 shared the same linguistic scheme with different weights, thus giving 22 effective rules in total. The HSDBC-S-based FLC had 21 effective rules and it had the most redundancy with different weights compared to others. As shown in Table XII(d), rules 1 and 20, rules 3 and 12, rules 11 and 15, rules 24 and 25 shared the same linguistic schemes but with different weights.

Table XI: Optimised gain values for PD-like FLC inputs and output.

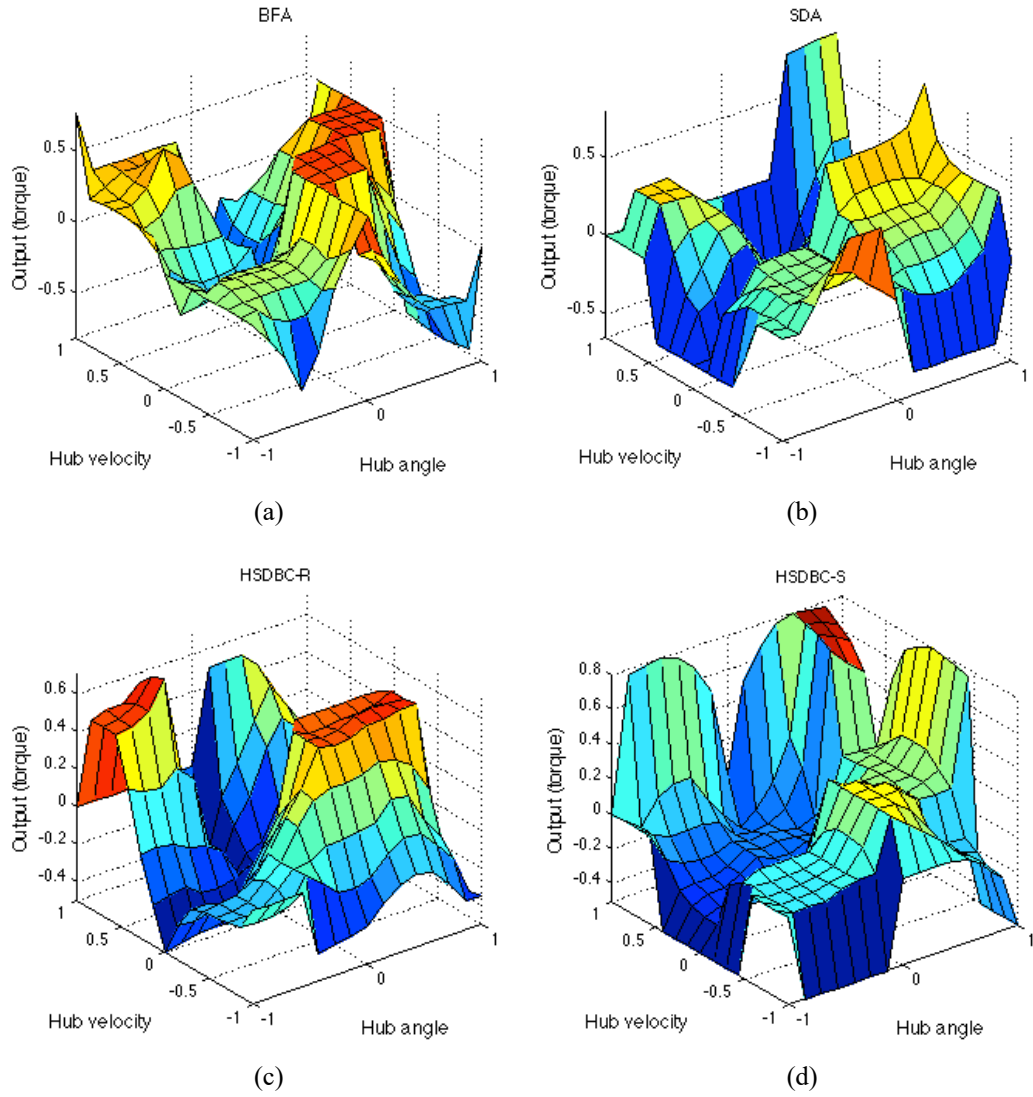|  | BFA | SDA | HSDBC-R | HSDBC-S |
|---|---|---|---|---|
| Parameter | $k_1=0.0164$<br>$k_2=0.0023$<br>$k_3=0.3247$ | $k_1=0.0297$<br>$k_2=0.0025$<br>$k_3=0.2800$ | $k_1=0.0230$<br>$k_2=0.0018$<br>$k_3=0.4452$ | $k_1=0.0299$<br>$k_2=0.0025$<br>$k_3=0.4947$ |

(a)

(b)

(c)

(d)

**Fig. 9. Fuzzy surface of the PD-like fuzzy-logic control, (a) BFA-based, (b) SDA-based, (c) HSDBC-R-based and (d) HSDBC-S-based.**

Table XII: Fuzzy rules for PD-like fuzzy logic control.

(a) SDA-based

| No. | Hub angle | Hub velocity | Torque | Weight |
|-----|-----------|--------------|--------|--------|
| 1 | PB | Z | PB | 0.4 |
| 2 | NB | NB | Z | 0.6 |
| 3 | NS | PS | PS | 0.6 |
| 4 | PB | NS | Z | 0.2 |
| 5 | PB | PB | PB | 0.4 |
| 6 | Z | PS | NS | 0.8 |
| 7 | PS | NS | PB | 0.4 |
| 8 | NS | PS | PS | 0.6 |
| 9 | NS | NS | PS | 1 |
| 10 | PB | NS | PS | 0.6 |
| 11 | PS | NB | NS | 0.4 |
| 12 | PB | PS | NB | 0.1 |
| 13 | NB | Z | NS | 1 |
| 14 | NS | NS | Z | 0.4 |
| 15 | NS | NB | PS | 1 |
| 16 | NS | NS | NB | 0.6 |
| 17 | PS | Z | PS | 0.8 |
| 18 | NS | NS | Z | 0.8 |
| 19 | NS | NS | Z | 0.6 |
| 20 | NB | NB | PS | 0.6 |
| 21 | PS | NS | Z | 1 |

(b) BFA-based

| No. | Hub angle | Hub velocity | Torque | Weight |
|-----|-----------|--------------|--------|--------|
| 1 | NS | Z | NS | 1 |
| 2 | NS | NB | NB | 0.6 |
| 3 | NB | PS | Z | 0.8 |
| 4 | PS | PB | NS | 0.4 |
| 5 | NB | PS | PB | 0.6 |
| 6 | Z | NS | PS | 0.6 |
| 7 | NS | PB | Z | 0.4 |
| 8 | NB | NB | Z | 0.6 |
| 9 | NS | PB | PB | 0.6 |
| 10 | PS | Z | PS | 0.6 |
| 11 | NB | PB | PB | 0.2 |
| 12 | NS | PS | PB | 0.8 |
| 13 | Z | NB | PS | 0.8 |
| 14 | PS | PS | PS | 0.4 |
| 15 | Z | PS | NS | 0.2 |
| 16 | NS | NS | PS | 0.8 |
| 17 | NS | NS | NS | 0.6 |
| 18 | PS | NB | NS | 0.8 |
| 19 | PB | NS | NB | 0.6 |
| 20 | PS | Z | PB | 0.4 |
| 21 | NS | Z | NB | 0.4 |

| No. | Hub angle | Hub velocity | Torque | Weight |
|---|---|---|---|---|
| 22 | NB | NB | NS | 0.6 |
| 23 | NB | NS | NS | 0.4 |
| 24 | PS | PS | NB | 0.4 |
| 25 | NS | PS | Z | 0.4 |

| No. | Hub angle | Hub velocity | Torque | Weight |
|---|---|---|---|---|
| 22 | PS | NB | Z | 0.4 |
| 23 | NS | Z | PS | 0.2 |
| 24 | Z | PS | NS | 0.6 |
| 25 | PB | PS | PS | 0.4 |

(c) HSDBC-R-based

| No. | Hub angle | Hub velocity | Torque | Weight |
|---|---|---|---|---|
| 1 | PB | NB | Z | 0.29 |
| 2 | Z | NS | PS | 0.89 |
| 3 | Z | PS | NS | 0.32 |
| 4 | PS | NB | NS | 0.25 |
| 5 | PS | Z | PS | 0.23 |
| 6 | PS | NS | Z | 0.17 |
| 7 | NB | NS | NB | 0.31 |
| 8 | PS | PB | PS | 0.42 |
| 9 | PS | PS | NS | 0.11 |
| 10 | Z | PS | NS | 0.86 |
| 11 | NS | NS | Z | 0.53 |
| 12 | NB | PS | PS | 0.33 |
| 13 | PS | NS | PB | 0.88 |
| 14 | PS | NB | Z | 0.53 |
| 15 | PB | NB | NS | 0.89 |
| 16 | PB | PB | Z | 0.52 |
| 17 | NS | Z | NS | 0.65 |
| 18 | NS | PS | PB | 0.35 |
| 19 | Z | NB | NS | 0.76 |
| 20 | NS | Z | Z | 0.05 |
| 21 | Z | NB | PB | 0.19 |
| 22 | NB | NS | Z | 0.85 |
| 23 | NB | Z | NS | 0.59 |
| 24 | PS | NB | Z | 0.89 |
| 25 | NS | Z | NS | 0.38 |

(d) HSDBC-S-based

| No. | Hub angle | Hub velocity | Torque | Weight |
|---|---|---|---|---|
| 1 | NS | PS | Z | 0.42 |
| 2 | NS | NS | PB | 0.78 |
| 3 | PS | NB | Z | 0.34 |
| 4 | Z | PS | PS | 0.25 |
| 5 | NS | Z | Z | 0.41 |
| 6 | Z | NS | Z | 0.56 |
| 7 | PS | Z | Z | 0.88 |
| 8 | NB | Z | NS | 0.44 |
| 9 | PS | PB | PB | 0.40 |
| 10 | NS | PS | NS | 0.59 |
| 11 | PS | Z | PS | 0.25 |
| 12 | PS | NB | Z | 0.92 |
| 13 | NS | Z | NS | 0.89 |
| 14 | PB | PS | NS | 0.009 |
| 15 | PS | Z | PS | 0.36 |
| 16 | NS | NB | NS | 0.24 |
| 17 | NS | PB | PB | 0.43 |
| 18 | PB | NB | NS | 0.24 |
| 19 | Z | NS | PB | 0.74 |
| 20 | NS | PS | Z | 0.37 |
| 21 | PB | Z | PB | 0.40 |
| 22 | NS | NS | NS | 0.71 |
| 23 | Z | PB | Z | 0.04 |
| 24 | Z | PS | NS | 0.91 |
| 25 | Z | PS | NS | 0.93 |

The results for the optimisation of the PD-like FLC for tracking hub angle position are shown in Figure 10. Figure 10(a) shows that the hybrid algorithms achieved better accuracy compared to the original algorithms. Notice that, the SDA and HSDBC-S had the fastest speed at the very early iteration. However, SDA got trapped at local optimum with fitness accuracy of 2.049 when it reached iteration 50. The BFA showed the worst performance among all other algorithms in terms of convergence speed. The graph shows that, at the early iterations, it was hardly converging to a better fitness location. It converged to fitness value of 2.315 and intercepted SDA at iterations 632 and 633 respectively. It achieved the best fitness value of 1.727 at iteration 700 and was unable to converge further until the end of the search operation. The HSDBC-R achieved the best accuracy of 1.560 followed by HSDBC-S which achieved the second best fitness accuracy of 1.609. HSDBC-R showed slower convergence at the early stage of the search, but it successfully intercepted SDA, BFA and HSDBC-S at iterations 81, 77 and 145 respectively. For a clearer visual representation, the best fitness cost achieved by the BFA, SDA, HSDBCS and HSDBCR is summarised in the bar chart shown in Figure 11.

The optimised hub angle position is shown in Figures 10(b) and 10(c) while its numerical result is shown in Table XIII with the best value highlighted in bold font. The result shows that HSDBC-R had the shortest rise time followed by BFA, HSDBC-S and SDA. The position optimised by BFA and HSDBC-R presented the highest and lowest overshoot respectively. On the contrary, the result optimised by HSDBC-R and BFA presented the highest and lowest undershoot respectively. In terms of settling time, HSDBC-S performed the best, followed by BFA, HSDBC-R and SDA. For the steady-state error, both hybrid schemes resulted in smaller offset from the desired position where HSDBC-R had the smallest error of 0.01 while HSDBC-S had the second smallest error with the offset value of 0.04. A comparison of overall performance for the time-domain response of the hub-angle position is presented in bar chart form in Figure 12. Overall, for optimisation of PD-like FLC for tracking hub-angle position, the result has shown that the proposed hybrid algorithms outperformed both the predecessor algorithms.
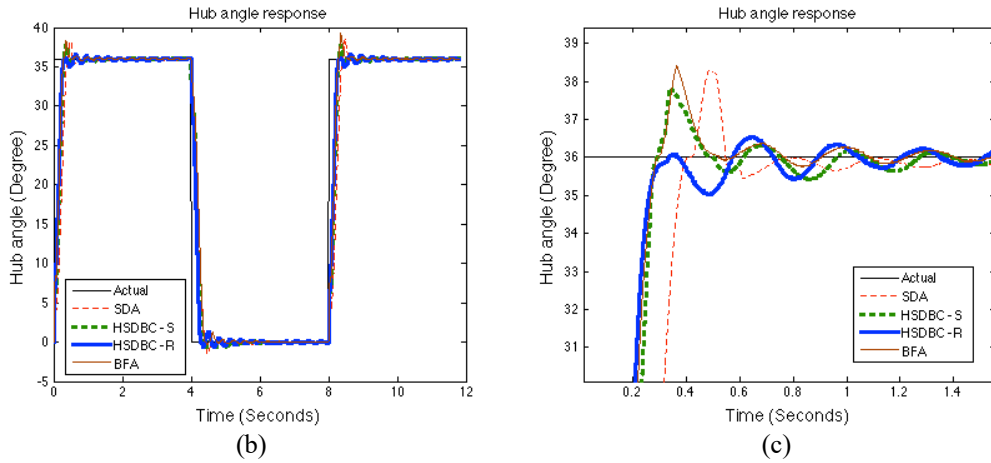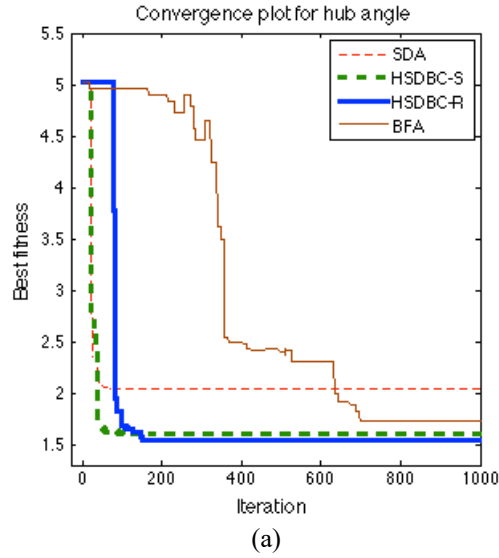
(a)



(b)                  (c)

**Fig. 10. Results of optimisation of the PD-like fuzzy logic control for tracking hub angle position, (a) Convergence plot of best fitness vs iteration, (b) Hub angle position in the time-domain and (c) Zoom-in of the hub angle position.**
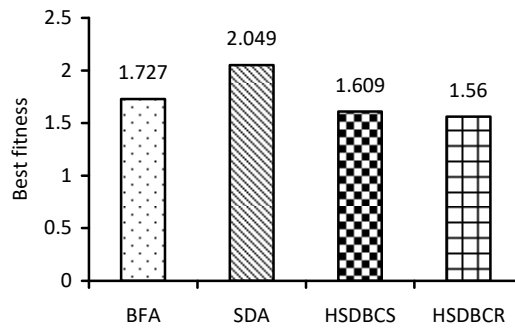


**Fig. 11. Best fitness cost of the BFA, SDA, HSDBCS and HSDBCR.**

Table XIII: Numerical result of the optimised hub angle position.

|  | BFA | SDA | HSDBC-S | HSDBC-R |
|---|---|---|---|---|
| *Settling time, $t_s$* | 0.4381 | 0.5396 | **0.4262** | 0.5188 |
| *Steady state error, $e_{ss}$* | 0.0500 | 0.1300 | 0.0400 | **0.0100** |
| *Rise time, $t_r$* | 0.2195 | 0.2725 | 0.2281 | **0.2074** |
| *Percentage overshoot, %os* | 6.6900 | 6.3333 | 5.1111 | **1.5000** |
| *Percentage undershoot, %os* | **0.6700** | 1.5555 | 1.6111 | 2.6944 |
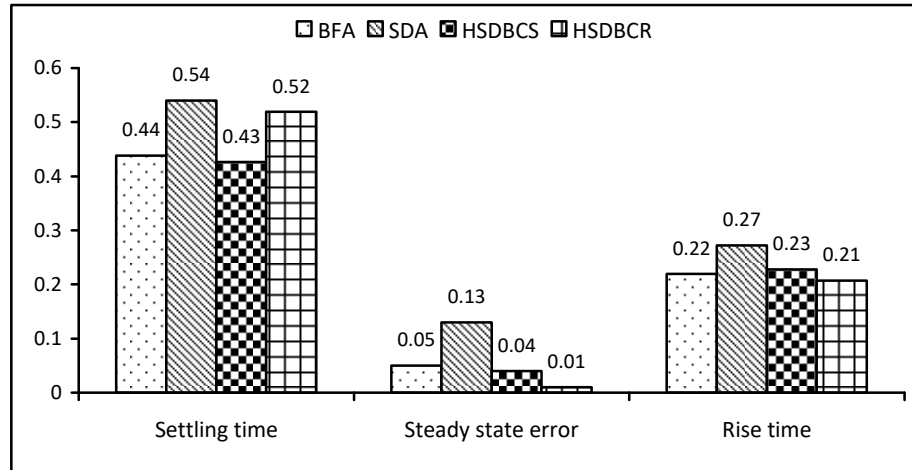
(a)
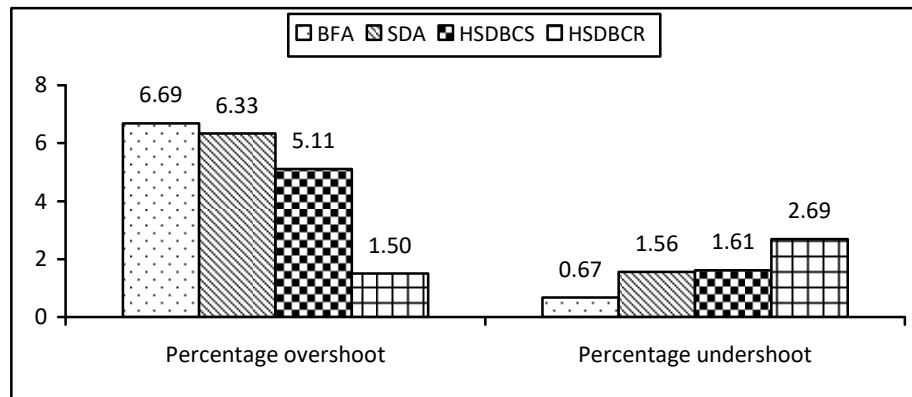


(b)

**Fig. 12. Comparison of the time-domain response for hub angle position in bar chart form, (a) settling time, steady-state error and rise time and (b) percentage overshoot and percentage undershoot.**

## 5. CONCLUSION

Two types of hybrid schemes synergising SDA and chemotactic strategy of BFA have been proposed in this paper. SDA with dynamic step size solves the oscillation problem around global optima for the BFA. Unlike the original SDA algorithm, in the proposed schemes, a swim strategy has been introduced to enhance the search strategy of SDA. Moreover, the reproduction, elimination and dispersal phases of the BFA have been excluded, retaining the simplicity of original SDA algorithm, hence avoiding long computational time for the algorithm to complete the whole process. The first type known as HSDBC-S uses spiral approach of swimming while the second type named HSDBC-R employs random approach of swimming. Incorporation of the swim action into the original SDA has balanced the exploitation and exploration aspects of the algorithm. The analytical results of tests based on nonparametric Friedman and Wilcoxon signed rank on eight well known benchmark functions have shown that the proposed algorithms statistically outperform the original SDA and BFA in terms of fitness accuracy as well as convergence speed with number of fitness evaluations.

The application of the algorithms to optimise hybrid conventional-intelligent controller for hub-angle position tracking of a flexible robot manipulator has shown that the hybrid schemes achieve better results especially on the fitness accuracy. The study has shown that the SDA has the fastest convergence speed at the very early iterations but it tends to get trapped at local optima. On the contrary, the BFA has the longest computation time and it hardly converges to a better location. Moreover, it is evidently shown that the hybrid schemes show better result for position tracking of the flexible manipulator. Overall, the hybrid schemes have shown the best performance where they have achieved the best accuracy with relatively faster convergence.

Future research will expand this algorithm to a multiobjective type for finding pareto front to give multiple options rather than a single solution where two or more conflicting objectives will be compromised for best performance. Also, adaptive formulation of bacteria step size based on intelligent fuzzy logic approach will be further considered to increase the algorithm performance. Moreover, comparison and testing of the

proposed algorithms with other state-of-the-art algorithms and benchmark functions will be further explored. In this paper, in terms of real world problems, a classical type FLC has been applied for position control of hub angle of a flexible manipulator. In future work, this will be extended to include hub angle position as well as vibration reduction at the end-point of the flexible manipulator system with payload variation where a more dynamic environment will be considered to provide a more challenging platform for testing and comparing interval type-2 FLC with classical type FLC using various optimisation algorithms. Finally, an online tuning approach through parallel computing architecture such as graphic processing unit for faster computation and a real time application for control design will be considered to show the effectiveness of the proposed algorithms in optimising the control scheme with hardware implementation.

## 6. REFERENCES

[1] Roberge, V., Tarbouchi, M. and Labonte, G. (2013). Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning, IEEE Transactions On Industrial Informaticsm, 9 (1), pp. 132-141.

[2] Firoozy, N. and Tavakoli, A. (2013). Breast tumour identification based on inverse scattering approach, IET Microwaves, Antennas & Propagation, 7 (7), pp. 535–542.

[3] Li-Ning, Z., Qi, Z., Da-chao, L. and Jing, A. (2010). Research on the Pre-warning Model of Enterprise Financial Crisis Based on the Information Entropy and PSO-ANN. Proceeding of 2010 International Conference on E-Business and E-Government, Guangzhou, China, 7-9 May 2010, pp. 1567 – 1570.

[4] Griffis, S. E., Bell, J. E., and Closset, D. J. (2012), Metaheuristics in Logistics and Supply Chain Management. Journal of Business Logistics, 2012, 33 (2), pp. 90 – 106.

[5] Boussaid, I., Lepagnot, J., and Siarry, P. (2013). A survey on optimization metaheuristics. Information Sciences (237). pp. 82-177.

[6] Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control System Magazine, June 2002, pp. 52 - 67.

[7] Kennedy, J. and Eberhart, R. C. (1995) "Particle swarm optimization", Proc. IEEE Intl. Conf. on Neural Networks, vol. IV, pp.1942 -1948.

[8] Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimisation and Machine Learning. Addison Wesley Longman, Publishing Co. Inc., New York.

[9] Dorigo, M. and Di caro, G. (1999). Ant colony optimization: a new meta-heuristic. Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, Washington, DC 06-09 Jul 1999, pp. 1470-1477.

[10] Karaboga D. and Basturk B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global optimization, 39 (3), pp. 459-471.

[11] Kirkpatrick. S. (1984). Optimization by simulated annealing: Quantitative studies. Journal of Statistical Physics, 34(5-6), pp 975-986.

[12] Glover, F. (1989). Tabu Search - Part 1". ORSA Journal on Computing, 1 (2), pp. 190-206.

[13] Tamura K., Yasuda K. (2011). Spiral multipoint search for global optimization. Proceeding of 10th International Conference on Machine Learning and Applications, pp. 470 – 475.

[14] Geem Z. W., Kim J. H.and Loganathan G.V. (2001). A New Heuristic Optimization Algorithm: Harmony Search, SAGE Publication, 76 (2), pp. 62-68.

[15] Formato, R. A. (2007). Central force optimization: A new metaheuristic with applications in applied electromagnetics. Progress In Electromagnetics Research, 77, pp. 425-491.

[16] Nasir, A. N. K., Tokhi, M. O., Abd. Ghani, N. M. and Raja Ismail, R. M. T. (2012). Novel Adaptive Spiral Dynamics Algorithms for Global Optimization. Proceeding of the 11th IEEE International Conference on Cybernatic Intelligent Systems 2012, August, 23-24, 2012, Limerick, Ireland, pp. 99-104].

[17] Nasir, A. N. K., Tokhi, M. O., Sayidmarie, O. and Raja Ismail, R. M. T. (2013). A Novel Adaptive Spiral Dynamic Algorithm for Global Optimization. Proceeding of the 13th UK Workshop on Computational Intelligence (UKCI 2013), Surrey, UK, 9-11 Sept 2013.

[18] Nasir, A. N. K., Tokhi, M. O., Abd. Ghani, N. M. and Ahmad, M. A. (2012). A novel hybrid spiral-dynamics bacterial-foraging algorithm for global optimization with application to control design. Proceeding of the 12th UK Workshop on Computational Intelligence (UKCI 2012), Edinburgh, UK, 5-7 Sept 2012, pp. 1-7.

[19] Nasir, A. N. K., Tokhi, M. O. and Abd. Ghani, N. M. (2013). Novel Hybrid Bacterial Foraging and Spiral Dynamics Algorithms. Proceeding of the 13th UK Workshop on Computational Intelligence (UKCI 2013), Surrey, UK, 9-11 Sept 2013.

[20] Li, T. et al. 2012, Draft genome sequence of Escherichia coli LCT-EC106, Journal of Bacteriology 194 (16), pp. 4443-4444.

[21] Dasgupta, S., Das, S., Abraham, A. and Biswas, A. (2009). Adaptive computational chemotaxis in bacterial foraging optimization: an analysis. IEEE Transaction on Evolutionary Computation, 13(4), pp. 919-941.

[22] Majhi, R., Panda, G., Majhi, B. and Sahoo, G. (2009). Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques. Expert System with Applications, 36 (6), pp. 10097-10104.

[23] Dasgupta, S., Das, S., Biswas, A. and Abraham, A. (2010). Automatic circle detection on digital images with an adaptive bacterial foraging algorithm. Soft Computing, 14(11), pp. 1151-1164.

[24] Datta, T., Misra, I. S., Mangaraj, B. B. and Imtiaj, S. (2008). Improved adaptive bacteria foraging algorithm in optimization of antenna array for faster convergence. Progress in Electromagnetics Research C, Vol. 1, pp.143-157.

[25] Farhat, I. A. and El-Hawary, M. E. (2010). Dynamic adaptive bacterial foraging algorithm for optimum economic dispatch with valve-point effects and wind power. IET Generation, Transmission and Distribution. 4(9). pp. 989-999.

[26] Chen, H., Zhu, Y. and Hu, K. (2011). Adaptive bacterial foraging optimization. Abstract and Applied Analysis, Vol. 2011.

[27] Sanyal N., Chatterjee, A. and Munshi, S. (2011). An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation. Expert System with Applications. 38(12), pp. 15489-15498.

[28] Niu, B., Wang, H., Tan, L. and Li, L. (2011). Improved BFO with adaptive chemotaxis step for global optimization. Proceeding of 2011 Seventh International Conference on Computational Intelligence and Security (CIS), Hainan, China, 3-4 Dec. 2011, pp. 76-80.

[29] Mishra S., (2005). Hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. IEEE Transaction on Evolutionary Computation, 9(1), pp. 61 – 73.

[30] Venkaiah, C. and Vinod Kumar D. M. (2011). Fuzzy adaptive bacterial foraging congestion management using sensitivity based optimal active power re-scheduling of generators. Applied Soft Computing, 11(8), 4921-4930.

[31] Supriyono H., Tokhi M. O. (2011). Adaptation schemes of chemotactic step size of bacterial foraging algorithm for faster convergence. Journal of Artificial Intelligence, 4(4), pp. 207 – 209.

[32] Biswas, A., Dasgupta, S., Das, S. and Abraham, A. (2007). Synergy of pso and bacterial foraging optimization – A comparative study on numerical benchmarks. Springer-Verlag Berlin Heidelberg, pp. 255 – 263.

[33] Chu, Y., Mi, H., Liao, H., Ji, Z. and Wu, Q. H. (2008). A fast bacterial swarming algorithm for high-dimensional function optimization, Proceeding of 2008 IEEE Congress on Evolutionary Computation, 1-6 June 2008, Hong Kong, pp. 3135 – 3140.

[34] Korani, W. M. (2008). Bacterial foraging oriented by particle swarm optimization strategy for PID tuning. Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation, Atlanta, GA, USA, July 12-16, 2008, pp. 1823 – 1826.

[35] Shen, H., Zhu, Y., Zhou, X., Guo, H. and Chang, C. (2009). Bacterial foraging optimization algorithm with particle swarm optimization strategy for global numerical optimization. Proceedings of 2009 World Summit on Genetic and Evolutionary Computation, Shanghai, China, June 12 – 14, 2009, pp 497 – 504.

[36] Hooshmand, R. A. and Pour, M. E. (2010). Corrective action planning considering FACTS allocation and optimal load shedding using bacterial foraging oriented by particle swarm optimization algorithm. Turkish Journal of Electrical Engineering and Computer Sciences, 18(4), pp. 597 – 612.

[37] Fang, Y., Liu, Y. and Liu, J. (2007). A novel simplified foraging optimization algorithm for parameter identification of nonlinear system model. Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, August 18 - 21, 2007, pp. 798 – 802.

[38] Kim, D. H., Abraham, A. and Cho, J. H. (2007). A hybrid genetic algorithm and bacterial foraging approach for global optimization. Journal of Information Sciences, 177, pp. 3918 – 3937.

[39] Chen, T. C., Tsai, P. W., Chu, S. C. and Pan, J. S. (2007). A novel optimization approach: Bacterial-GA foraging. Proceedings of the Second International Conference on Innovative Computing, Information and Control ICICIC '07, Kumamoto, Japan, September 5 - 7, 2007, pp. 391 – 394.

[40] Jadhav, D. G., Pattnaik, S. S., Devi, S., Lohokare, M. R. and Bakwad, K. M. (2010). Approximate memetic algorithm for consistent convergence. Proceeding of National Conference on Computational Instrumentation (NCCI 2010), Chandigarh, India, 19-20 March 2010, pp. 118 – 122.

[41] Sarasiri, N. and Sujitjorn, S. (2010). Bacterial foraging optimization and tabu search: performance issues and cooperative algorithms. Proceedings of the 10th WSEAS International Conference on Systems Theory and Scientific Computation, Taipei Taiwan, August 20-22, pp. 186 – 191.

[42] Panikhom, S., Sarasiri, N., and Sujitjorn, S. (2010). Hybrid bacterial foraging and tabu search optimization (BTSO) algorithms for Lyapunov's stability analysis of nonlinear systems. International Journal of Mathematics and Computers in Simulation, 3(4), pp. 81 – 89.

[43] Lohokare, M. R., Pattnaik, S. S., Devi, S., Panigrahi, B. K., Das, S. and Bakwad, K. M. (2009). Intelligent biogeography-based optimization for discrete variables. Proceeding of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), Coimbatore, India, 9 – 11 December, 2009, pp. 1088 – 1093.

[44] Praveena, P., Vaisakh, K. and Rao, S.R.M. (2010). A Bacterial foraging and PSO-DE algorithm for solving dynamic economic dispatch problem with valve-point effects. First International Conference on Integrated Intelligent Computing (ICIIC), Bangalore, India, 5-7 Aug. 2010, pp. 227 – 232.

[45] Biswas, A., Dasgupta, S., Das, S. and Abraham, A. (2007). A synergy of differential evolution and bacterial foraging optimization for global optimization. Journal of Neural Network World, 17(6), pp. 607 – 626.

[46] Tokhi, M.O., Mohamed, Z., Amin, S.H.M., Mamat, R. (2000). Dynamic characterization of a flexible manipulator system : theory and experiments. Proceeding of TENCON 2000, Kuala lumpur Malaysia, 24 – 27 Sept 2000, pp. 167-172.

[47] Ostergaard, E.H., (2012). Lightweight robot for everybody. IEEE Robotics & Automation Magazine, 19 (4), pp. 17–18.

[48] Alvarez-Ramirez, J., Kelly, R. and Cervantes, I. (2003). Semiglobal stability of saturated linear PID control for robot manipulators. Automatica, vol. 39, pp. 989-995.

[49] Cheong, J. and Lee, S. (2008). Linear PID Composite Controller and its Tuning for Flexible Link Robots. Journal of Vibration and Control, vol. 14, pp. 291-318.

[50] Alam, M. S. and Tokhi, M. O. (2008). Hybrid fuzzy logic control with genetic optimisation for a single-link flexible manipulator. Engineering Applications of Artificial Intelligence, vol. 21, pp. 858-873.

[51] Zain, M. Z. M., Tokhi, M. O. and Mohamed, Z. (2006). Hybrid learning control schemes with input shaping of a flexible manipulator system. Mechatronics, vol. 16, pp. 209-219, 2006.

[52] Subudhi, B. and Morris, A. S. (2009). Soft computing methods applied to the control of a flexible robot manipulator. Applied Soft Computing, vol. 9, pp. 149-158.

[53] Loudini, M. (2013). Modelling and Intelligent Control of an Elastic Link Robot Manipulator. International Journal of Advanced Robotic Systems, 10:81, pp. 1-18.

[54] Singer, N.C. and Seering, W.P., 1990. Preshaping command inputs to reduce system vibration. Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control 112 (1), 76–82.

[55] Singh, T., and Singhose, W.E., 2002. Tutorial on input shaping/time delay control of maneuvering flexible structures. Proceedings of the 2002 American Control Conference, Omnipress, Madison 1, 1717–1731.

[56] Nasir, A.N.K., Tokhi, M.O., Ghani, N.M.A. and Ahmad, M.A. (2012). A novel hybrid spiral dynamics bacterial chemotaxis algorithm for global optimization with application to controller design. Proceeding of the International Conference on Control, UKACC 2012, Cardiff, UK, 3-5 Sept 2012, pp. 753-758.

[57] Nasir, A. N. K. and Tokhi, M. O. (2013). A novel hybrid spiral-dynamics random- chemotaxis optimization algorithm with application to modelling of a flexible robot manipulator. Proceeding of 16th International Conference on Climbing and Walking Robots (CLAWAR2013), 14-17 July 2013, Sydney, Australia, pp. 667-674.

[58] Angeline, P. J. (1998). Evolutionary optimization versus particle swarm optimization: Philosophy and the performance difference. Proceeding of 7th Int. Conf. Evol. Programming Evol. Programming, LNCS vol. 1447, pp. 84–89.

[59] Azad, A.K.M., 1994. Analysis and design of control mechanisms for flexible manipulator systems. PhD Thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, UK.

[60] Poerwanto, H., 1998. Dynamic simulation and control of flexible manipulator systems. PhD thesis, Department of Automatic Control and Systems Engineer- ing, The University of Sheffield, UK.

[61] Castillo, O. and Melin, P. (2008). Type-2 Fuzzy Logic: Theory and Applications, Springer-Verlag, Heidelberg, Germany, 2008.

[62] Wagner, C. and Hagras, H. (2007). A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots, in: Proceedings of the IEEE Conference on Fuzzy Systems, London.

[63] Wagner, C. and Hagras, H. (2007). Evolving type-2 fuzzy logic controllers for autonomous mobile robots, Adv. Soft Comput. 41 (2007) pp. 16–25.

[64] Bingül, Z. and Karahan, O. (2011). A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control, Expert Syst. Appl. 38 (2011), pp. 1017–1031.

[65] Cao, J., Li, P., Liu, H. and Brown, D. (2008). Adaptive fuzzy controller for vehicle active suspensions with particle swarm optimization, in: Proceedings of SPIE- The International Society of Optical Engineering, vol. 7129, 2008.

[66] Castillo, O. and Melin, P. (2014). A review on interval type-2 fuzzy logic applications in intelligent control. Information Sciences, 279 (2014), pp. 615–631.

[67] Melin, P., Astudillo, L., Castillo, O., Valdez, F. and Garcia, M. (2013). Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm. Expert Systems with Applications, 40 (2013), pp. 3185–3195.

[68] Castillo, O. and Melin, P. (2012). Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review. Information Sciences, 205 (2012) pp. 1–19.

[69] Lopez, M., Melin, P. and Castillo, O. (2008). Optimization of response integration with fuzzy logic in ensemble neural networks using genetic algorithms, Studies in Computational Intelligence 154 (2008), pp. 129–150.

[70] Aliev, R.A., Pedrycz, W., Guirimov, B.G., Aliev, R.R., Ilhan, U., Babagil, M., and Mammadli, S. (2011). Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization, Information Sciences, 181 (9), pp. 1591–1608.

[71] Park, K.-J., Oh, S.-K. and Pedrycz, W. (2009). Design of interval type-2 fuzzy neural networks and their optimization using real-coded genetic algorithms, in: Proceedings of the IEEE Conference on Fuzzy Systems, Jeju, Korea, 2009, pp. 2013–2018.

[72] Qiu, Y., Zhang, Y.-Q. and Zhao, Y. (2007). Statistical genetic interval-valued fuzzy systems with prediction in clinical trials, in: Proceedings of the IEEE International Conference on Granular Computing, San Jose, 2007, pp. 129–132.

[73] Chua, T.W. and Tan, W.W. (2008). Genetically evolved fuzzy rule-based classifiers and application to automotive classification, Lecture Notes in Computer Science 5361 (2008), pp. 101–110.

[74] Maldonado, Y., Castillo, O. and Melin, P. (2013). Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications. Applied Soft Computing, 13 (2013), pp. 496–508.

[75] Valdez, F., Melin, P. and Castillo, O. (2009). Evolutionary method combining particle swarm optimization and genetic algorithms using fuzzy logic for decision making, in: Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 2114–2119.

[76] Sombra, A., Valdez, F., Melin, P., and Castillo, O. (2013). A new gravitational search algorithm using fuzzy logic to parameter adaptation. IEEE Congress on Evolutionary Computation, pp. 1068–1074.

[77] Valdez, F., Melin, P. and Castillo, O. (2014). A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation. Expert Systems with Applications, **41** (14), pp. 6459–6466.

[78] Liand, Q. and Mendel, J. M. (2000). Interval type-2 fuzzy logic systems: Theory and design. IEEE Transaction on Fuzzy Systems, **8** (5), pp. 535-550.

[79] Vázquez, J. C., Valdez, F. and Melin, P. (2013). Comparative study of particle swarm optimization variants in complex mathematics functions. Recent Advances on Hybrid Intelligent Systems, Springer Berlin Heidelberg, pp. 223-235.

[80] Valdez, F., Melin, P., and Castillo, O. (2013). Bio-inspired optimization methods on graphic processing unit for minimization of complex mathematical functions. Recent Advances on Hybrid Intelligent Systems, Springer Berlin Heidelberg, pp. 313-322.

[81] Kromer, P., Platos, J. and Snasel, V. (2014). Nature-inspired meta-heuristics on modern gpus: state of the art and brief survey of selected algorithms. Internatinal Journal Parallel Prog., 42, pp. 681–709.