Hybrid Bacterial Foraging and Tabu Search Optimization (BTSO) Algorithms for Lyapunov's Stability Analysis of Nonlinear Systems

Suphaphorn Panikhom, Nuapett Sarasiri and Sarawut Sujitjorn*

***Abstract*—**This article presents brief descriptions of the bacterial foraging optimization (BFO), the tabu search (TS) and the hybrid algorithms thereof namely bacterial foraging-tabu search optimization (BTSO) algorithms. The proposed hybrid BTSO algorithms perform search rapidly, and render a high-quality solution according to the operation of the adaptive tabu search (ATS). The BTSO algorithm is applied to stability analysis of linear and nonlinear systems based on the Lyapunov's methods. The stability analysis results are compared with the threshold accepting (TA) method. The article also covers the reviews of the TA and the Lyapunov's methods, respectively.

***Keywords*—**Bacterial foraging optimization, hybrid algorithms, Lyapunov's method, nonlinear systems, tabu search, threshold accepting, stability.

I. Introduction

In recent years, bio-inspired and metaheuristic algorithms have played very active roles in solving complex optimization problems. Some of the well-known bio-inspired algorithms include genetic algorithm, ant colony, bee colony, particle swarm optimization, bacterial foraging optimization, etc. Among these, the bacterial foraging optimization (BFO) originated in 2002 [1]-[2] has recently received much attention in academic research. Some published papers have reported a deficit of the original BFO in that under some situations it takes a very long time to render an elite solution. In other words, the BFO needs a very large number of iterative loops to track down the global solution. This problem has been resolved by introducing an adaptive jump in the chemotaxis step of the BFO. The modified version is known as the adaptive BFO, or ABFO [3]-[4].

Metaheuristics have emerged for a number of years, and successfully played major roles in economics and management fields. Among the existing algorithms, the tabu search (TS) [5]-[7] has demonstrated many successful applications in real-world problems as evidenced by a vast number of publications worldwide. The simplistic TS occasionally encounters an undefeatable local trap. In such case, the algorithm moves around a local solution endlessly without any improvement, otherwise hits the iteration limits. The drawback has been overcome by introducing the backtracking (BT) and the adaptive search radius (AR) mechanisms to the TS. This modified version of the TS has been referred to as the adaptive tabu search, or ATS [8]. The ATS has been successfully applied to identification [9], control [10]-[12], signal processing [13], etc.

Lyapunov's stability methods have been successfully applied for many years by engineers and scientists [14]-[15]. The direct method of Lyapunov's is regarded as clean and concise. Nonetheless, for some systems, finding Lyapunov function is not straightforward. Doing this manually in some cases is very time consuming, perhaps not possible. Once the Lyapunov function is obtained for the system of interest, the next practical issue becomes seeking for the region of attraction. In order to find this, some computational approaches, e.g. geometrical, numerical methods, etc., have to be applied. For instance, various previous works have proposed the construction of Lypaunov functions based on conventional methods [16]-[17], numerical methods [18]-[19] and artificial intelligent methods [20]-[21].

It is an interest to find a general tool to assist the construction of the Lyapunov function and the stability region for a nonlinear system. Among those existing algorithms the BFO, ATS, BTSO and the threshold accepting methods are good candidates. Since the article proposes the BTSO algorithms, the descriptions of the BFO, ATS and the BTSO appear in section 2,3 and 4 respectively. Section 5 presents the review of the Lyapunov's methods for stability analysis of dynamic systems. Section 6 presents the TA the BTSO methods applied to stability analysis. The results are compared with the previous work [22] for the same systems as appeared in [23]-[24].

II. Bacterial Foraging Optimization (BFO)

The BFO algorithm imitates the foraging behavior of bacteria, and consists of 4 main steps namely chemotaxis, swarming, reproduction and elimination-dispersal, respectively. The chemotaxis step imitates the swift movement of bacteria by a fixed distance or height. Under some complex circumstances, this conventional BFO algorithm may take very long time to reach a satisfactory solution. To resolve this, some modifications have been made to the BFO to have an adaptive chemotaxis step, i.e. the jumping height or distance is adjustable to cope with various environments. This modified

Manuscript received September 8, 2010. Research grants from Rajamangala University of Technology Isan and Royal Golden Jubilee Ph.D. Program under Grant PHD/0091/2551 are greatly acknowledged.

The authors are with the School of Electrical Engineering, Suranaree University of Technology, 111 University Avenue, Muang District, Nakhon Ratchasima, 3000, Thailand (correspondence : sarawut@sut.ac.th)

version has been known as the adaptive BFO or ABFO algorithm, which is more efficient than the conventional BFO. The procedural list below provides the ABFO algorithm.

Algorithm 1. ABFO
Initialization of parameters: S, p, N_C, N_S, N_{re}, N_{ed}, p_{ed}, $C(i)$, $d_{attract}$, $w_{attract}$, $h_{repellant}$ and $w_{repellant}$.
Elimination-dispersal loop:
while $l \leq N_{ed}$ **do**
  $l = l + 1$
  Reproduction loop:
  while $k \leq N_{re}$ **do**
    $k = k + 1$
    Chemotaxis loop:
    while $j \leq N_c$ **do**
      $j = j + 1$
      for each bacterium $i=1,2,\ldots,S$ **do**
        Compute $J(i, j, k, l)$
        $J(i,j,k,l) = J(i,j,k,l) + J_{CC}\left(\theta^i(j,k,l), P(j,k,l)\right)$
        $J_{last} = J(i, j, k, l)$
        Tumble:
        Generate a p-dimensional random vector
        $\Delta_m(i) \in \Re^p$, $m=1,2,\ldots,p$ on [-1,1].
        Move:
        Compute the adaptive step size, $C(i$

$$C(i) = \frac{\left|J^i(\theta)\right|}{\left|J^i(\theta)\right| + \alpha} = \frac{1}{1 + \dfrac{\alpha}{\left|J^i(\theta)\right|}}$$

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^{\mathrm{T}}(i)\Delta(i)}}$$

        compute the objective function

$$J(i,j+1,k,l) = J(i,j+1,k,l) + J_{cc}\left(\theta^i(j+1,k,l), P(j+1,k,l)\right).$$

        Swim:
        Let $m=0$
        while $m < N_S$ **do**
          $m=m+1$
          if $J(i, j+1, k, l) < J_{last}$, define
          $J_{last} = J(i, j+1, k, l)$ and compute

$$\theta^i(j+1,k,l) = \theta^i(j+1,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^{\mathrm{T}}(i)\Delta(i)}}$$

          Use this θ^i to compute the new $J(i, j+1, k, l)$
          else
            $m = N_S$
          end if
        end while
      end for
    end while
    Compute for each bacterium $i=1,2,\ldots,S$, for given k and l

$$J^i_{health} = \sum_{j=1}^{N_c+1} J(i, j, k, l).$$

    Eliminate S_r of bacteria with high values of J_{health}, and split the other bacteria into two at their locations.
  end while
  For each bacterium, with probability P_{ed} eliminate the bacterium and disperse to new location.
end while

III. Adaptive Tabu Search (ATS)

Tabu search (TS) has been originated by Glover to solve combinatorial optimization problems. A vast amount of publications are available to support that the algorithm is worldwide acceptable as an efficient tool for optimization problems. Some research results have shown that the simplistic TS is suffered from local solution deadlocks [8]. Adaptive mechanisms have been added to resolve the deadlock problem. The mechanisms are referred to as backtracking (BT) and adaptive search radius (AR), respectively. Such mechanisms enhance the exploitation and the exploration capabilities of the TS. The modified version listed in the procedures below has been known as the adaptive tabu search (ATS).

Algorithm 2. ATS
Initialize search parameters: R, N, TL, $count_{max}$, BT and AR.
Randomly select initial solution S_0 from search space:
define S_0 = *best_neighbor*
  for $coun \leq count_{max}$
    - Generate a neighbourhood with an initial search radius R, set $S_1(r)$ = *neighbourhood*.
    - Evaluate J each member belonging to $S_1(r)$.
      define S_1 = *best_neighbor1*.
        if $J_1 < J_0$ store S_0 in the TL
  define $S_0 = S_1$
  else
      Store S_1 in the TL.
      Invoke the BT when a deadlock occurs.
        if $n \geq BT$
          $n=n+1$
          look back in the TL.
          best_error = RANK(TL)
        else
          $n=0$
          define S_0 = *best_neighbor*
            best_error = best_error
        end if
        Invoke the AR when the current solution S_0 is close to local minima
        if *best_error<best_error_expect1*
          $R=R_1$
        end
        if *best_error<best_error_expect2*
          $R=R_2$
        end
        …
      end if
  end for

IV. HYBRID BTSO ALGORITHMS

From observation, the search movement of the ATS is limited within the preset search radius resulting in limited exploration on the search space. Consequently, some parts of the search space containing high-quality solutions may not be explored. As a result, the algorithm consumes a considerably long time to struggle for an elite solution. In the contrary, the ABFO algorithm via its adaptive chemotaxis step can explore various parts of the entire search space. This provides more chances to grasp at least one elite solution at the very beginning of the search providing that an initial exploration is not repeated too many times. The ABFO algorithm yet has some drawbacks concerning its complexity, possibility to be locked up by a local solution, for instance. These deficits can be overcome by the BT and the AR mechanisms of the ATS. Therefore, the algorithms have been combined, and designated as the hybrid bacterial foraging and tabu search, or BTSO algorithms. The steps for executing the BTSO algorithms are as follows:

STEP 1: Call ABFO.
STEP2: Transport the solution obtained from the ABFO to the ATS as an initial solution.
STEP3: Call ATS.
STEP4: Report solution and exit.

Fig.1 BTSO algorithms.

To investigate the performance, the algorithms have been tested against 3D surface optimization problems. These include the Bohachevsky's, Rastrigin's and Shubert's functions, or BF, RF and ShuF, respectively. Representations of the problems appear in the following expressions:

$$f(x,y)=x^2+2y^2-0.3\cos(3\pi x)-0.4\cos(4\pi y)+0.7 \qquad (1)$$

$$f(x,y)=x^2+y^2-10\cos(2\pi x)-10\cos(2\pi y)+20 \qquad (2)$$

$$f(x_1,x_2)=\sum_{i=1}^{5} i\cos((i+1)x_1+i)*\sum_{i=1}^{5} i\cos((i+1)x_2+i) \qquad (3).$$

where = $-10\le x_1, x_2 \le 10$

In particular, the ShuF in (3) contains 18 global solutions. Table 1 summarizes the results obtained from running the MATLAB codes on a duo-core PC (Pentium IV 2.4GHz 640Mbytes of SD-RAM.). The data indicate that the hybrid BTSO algorithms are superior to the others. The proposed algorithms consume considerably short search time. This is supported by the average deadlocks encountered by the proposed algorithms are much less than those of the two algorithms. It is not unexpected because the search behaviour of the ABFO provides rather thorough exploration of the entire search space in a short duration, however, lacks the intensification characteristic.

V. LYAPUNOV'S METHODS

A nonlinear system can be represented by

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},t) \qquad (4)$$

for a non-autonomous one, and

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x}) \qquad (5)$$

for an autonomous system. At the equilibrium $\mathbf{x}_e = 0$, the following conditions hold : $\mathbf{f}(\mathbf{x}_e)=0$ and $\dot{\mathbf{x}}_e = 0$.

A. The Lyapunov's indirect method

Consider an autonomous system, the Jacobian at the equiblibrium point can be defined as

$$\mathbf{A}=\left.\frac{\partial f(\mathbf{x})}{\partial x}\right|_{x_e=0} \qquad (6).$$

- If all eigenvalues of $\mathbf{A}$ are strictly in the left-half complex plane, then the asymptotic stability at the equilibrium point of the linearized system can be concluded.
- If at least one eigenvalue of $\mathbf{A}$ is strictly in the right-half complex plane, then the instability of the linearized system is concluded.
- If all eigenvalues of $\mathbf{A}$ are in the left-half complex plane but at least one of them is on the $j\omega$ axis, then the linearized system is said to be marginally stable but one cannot conclude anything from the linear approximation.

TABLE I
SUMMARY OF THE RESULTS (AVERAGED OVER 50 TRIALS).

Test functions	Average search times (seconds)			Average search rounds			Average deadlocks	
	ABFO	ATS	BF-TS	ABFO	ATS	BF-TS	ATS	BF-TS
BF	5.754	12.89	6.83	12.06	661.98	151.20	56.28	14.28
RF	48.63	14.79	5.81	60.20	811.02	323.30	343.18	32.20
ShuF	182.83	3.28	2.80	715.12	68.06	55.28	7.60	4.44

B. The Lyapunov's direct method

According to the method, the Lyapunov function $V(\mathbf{x})$ must be found, and the stability can be concluded without knowing the solutions of the equation governing the systems. $V(\mathbf{x})$ must be scalar, positive definite and differentiable.

A nonlinear system can be said to have a globally asymptotically stable equilibrium, iff

- $V(\mathbf{x}) > 0$
- $\dot{V}(\mathbf{x}) < 0$
- $V(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$.

A thorough treatment of the theorems can be found in [14]-[15].

Consider a nonlinear mass-spring-damper system described by

$$m\ddot{x} + b\dot{x}|\dot{x}| + k_0 x + k_1 x^3 = 0 \qquad (7).$$

The energy function is used as the Lyapunov function candidate expressed by

$$V(\mathbf{x}) = \frac{1}{2}m\dot{x}^2 + \int_0^x (k_0 y + k_1 y^3)dy$$
$$= \frac{1}{2}m\dot{x}^2 + \frac{1}{2}k_0 x^2 + \frac{1}{4}k_1 x^4 \qquad (8).$$

It can be clearly seen that $V(\mathbf{x})$ is scalar, differentiable, positive definite and unbounded. The function $V(\mathbf{x})$ is, hence, the Lyapunov function. Next,

$$\dot{V}(\mathbf{x}) = \dot{x}(-b\dot{x}|\dot{x}|) = -b|\dot{x}^3| \qquad (9)$$

is negative definite. Therefore, the globally asymptotic stability of the system can be concluded.

According to the indirect method, the quadratic Lyapunov function can be generally applied. It can be expressed by

$$V(\mathbf{x}) = \mathbf{x}^T\mathbf{P}\mathbf{x} > 0 \qquad (10)$$

,whereas $\mathbf{x}$ is the state vector, and $\mathbf{P}$ is a symmetrically scalar matrix. The following equations must be satisfied

$$\dot{V}(\mathbf{x}) = \mathbf{x}^\mathbf{T}\mathbf{P}\mathbf{x} + \mathbf{x}^\mathbf{T}\mathbf{P}\mathbf{x} \qquad (11)$$

$$\dot{V}(\mathbf{x}) = -\mathbf{x}^\mathbf{T}\mathbf{Q}\mathbf{x} < 0 \qquad (12)$$

$$\mathbf{Q} = \mathbf{P}\mathbf{A} + \mathbf{A}^\mathbf{T}\mathbf{P} \qquad (13)$$

,and

$$\mathbf{Q} = \mathbf{Q}^\mathbf{T} \qquad (14).$$

VI. Nonlinear Stability Analysis

A. The indirect method

As reviewed earlier, the stability of any linearized systems can be concluded via the quadratic Lyapunov function. Regarding this, the matrices $\mathbf{P}$ and $\mathbf{Q}$ must exist and satisfy the Lyapunov equation. Generic threshold accepting (TA) algorithms [25]-[26] have been modified to search for these matrices. The procedural list below accommodates the stability analysis of the second-order systems.

STEP 0: Initialize solutions $\mathbf{P}$ and $\mathbf{Q}$ by zeroing all elements.
STEP 1: Generate randomly within the ranges [-10,+10] the elements of $\mathbf{P}$. Determine the positive definiteness of $\mathbf{P}$. If $\mathbf{P}$ is not positive definite, go to **STEP 1**.
STEP 2: Compute $\mathbf{Q}$ based on $\mathbf{P}$. Determine the negative definiteness of $\mathbf{Q}$. If $\mathbf{Q}$ is not negative definite, go to ***STEP 1***.
STEP 3: Accept solutions $\mathbf{P}$ and $\mathbf{Q}$. Exit.

Fig.2 TA algorithms to search for $\mathbf{P}$ and $\mathbf{Q}$.

Example 1 Consider the system [15]

$$\dot{\mathbf{x}}_1 = -2x_1 - x_1x_2$$
$$\dot{\mathbf{x}}_2 = x_1x_2 - x_2 \qquad (15)$$

having $\mathbf{A} = \left.\frac{\partial f}{\partial x}\right|_{x_e=0} = \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}$.

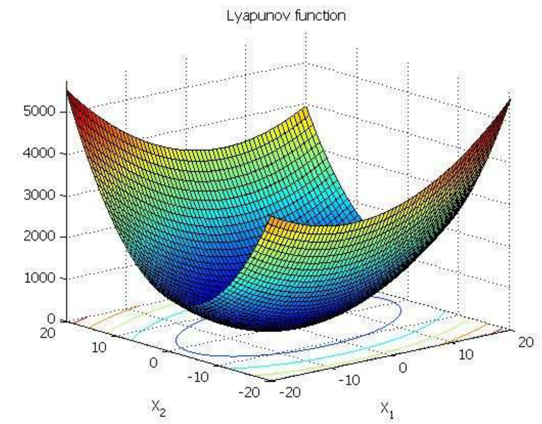
The search spent 4 iterations to reach for $\mathbf{P} = \begin{bmatrix} 5.9006 & -2.0793 \\ -2.0793 & 5.0789 \end{bmatrix}$ and $\mathbf{Q} = \begin{bmatrix} -4.1585 & 1.2575 \\ 1.2575 & -5.9992 \end{bmatrix}$.

Thus, $V(\mathbf{x})$ and $\dot{V}(\mathbf{x})$ are of the forms

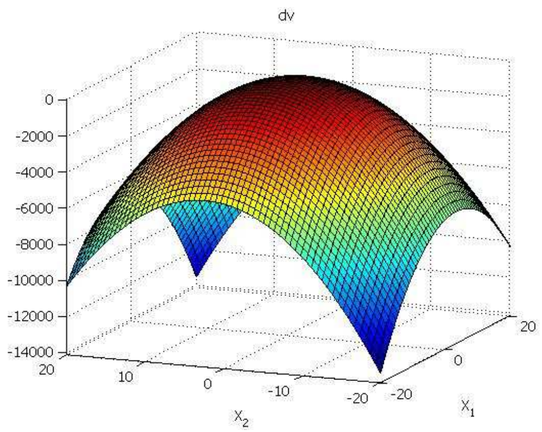
$$V(\mathbf{x}) = 5.9006x_1^2 - 5.0789x_2^2 - 2.0793x_2x_1 - 2.0793x_1x_2 \qquad (16)$$

$$\dot{V}(\mathbf{x}) = -14.3162x_2^2 + 14.3162x_1x_2 - 11.8012x_1x_2 \qquad (17),$$

whose surface plots are illustrated in Figs. 3(a) and (b), respectively, and the asymptotic stability at the origin can be concluded.

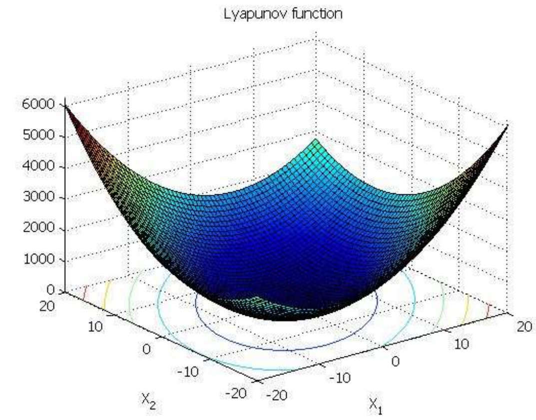


(a)

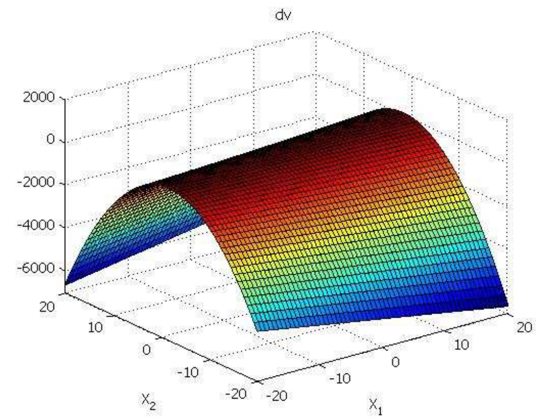


(b)

Fig.3 (a) Lyapunov function $V(\mathbf{x})$, *(b)* $\dot{V}(\mathbf{x})$ of example 1.



(a)



(b)

Fig. 4 *(a)* Lyapunov function $V(\mathbf{x})$, *(b)* $\dot{V}(\mathbf{x})$ of example 2.

Example 2 Consider the *Van der pol* system

$$\dot{\mathbf{x}}_1 = -x_2$$
$$\dot{\mathbf{x}}_2 = x_1 + (x_1{}^2 - 1)x_2 \qquad (18)$$

having $\mathbf{A} = \left.\frac{\partial f}{\partial x}\right|_{x_e=0} = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}$.

By 7 iterations, the search returned $\mathbf{P} = \begin{bmatrix} 3.149 & \text{-}1.001 \\ \text{-}1.001 & 8.726 \end{bmatrix}$ and $\mathbf{Q} = \begin{bmatrix} \text{-}12.597 & 3.004 \\ 3.004 & \text{-}17.453 \end{bmatrix}$ Hence,

$$V(\mathbf{x}) = 2.1479x_1{}^2 \text{ -}8.7265x_2{}^2 \text{ -}1.0013x_1x_2 \qquad (19)$$

$$\dot{V}(\mathbf{x}) = 14.3162\, x_2x_1 \text{ -}14.3162\, x_2{}^2 \text{ -}11.8012\, x_1x_2 \qquad (20).$$

Fig. 4 depicts the surface plots of $V(\mathbf{x})$ and $\dot{V}(\mathbf{x})$ to confirm the asymptotic stability at the origin of the system.

B. The direct method

This section pays an interest in the method proposed by *Xiao-Lin* and *You-Lin* [22]. The method considers the polynomial system of the form

$$\begin{cases} \dfrac{dx}{dt} = y, \\ \dfrac{dy}{dt} = f(x, y) \end{cases} \qquad (21),$$

where

$$f(x, y) = P(x) + Q(x)y + R(x)y^2 \qquad (22)$$

$$P(x)=\sum_{i=1}^{N}p_i x^i,\quad Q(x)=\sum_{i=0}^{N}q_i x^i,\quad R(x)=\sum_{i=0}^{N}r_i x^i \quad (23).$$

This section demonstrates the usefulness of the TA and the BTSO algorithms to seek for the suitable coefficients of the Lyapunov function candidate. Moreover, the search approaches provide larger stable regions than those provided by the previous method.

Example 3 Consider the system

$$\begin{cases}\dfrac{dx}{dt}=y,\\ \dfrac{dy}{dt}=-x+x^3-y\end{cases} \quad (24)$$

and the Lyapunov function candidate

$$V(x,y)=Q_0(x)+Q_1(x)y+Q_2(x)y^2 \quad (25).$$

Since

$$\begin{aligned}\frac{dV}{dt}=&-Q_1(x)[x-x^3]+[\dot{Q}_0(x)-Q_1(x)-2(x-x^3)Q_2(x)]y\\&+[\dot{Q}_1(x)-2Q_2(x)]y^2+\dot{Q}_2(x)y^3\end{aligned} \quad (26),$$

let $\dot{Q}_2(x)=0$ and $\dot{Q}_1(x)-2Q_2(x)=-\varepsilon$, one can obtain

$$\begin{cases}Q_2(x)=c\\ Q_1(x)=(2c-\varepsilon)x\end{cases} \quad (27).$$

Hence, the Lyapunov function candidate can be rewritten as

$$V(x,y)=Q_0(x)+(2c-\varepsilon)xy+cy^2 \quad (28)$$

STEP 0: Initialize $\alpha_5,\alpha_7,\alpha_9...,\alpha_{2N-1}=0$.
STEP 1: Generate possible coefficient randomly: $\varepsilon,\alpha_3\in[0\ 1]$ and $c,\alpha_1\in[0\ 1\times10^{-8}]$. If $0\le\varepsilon\le c\le\alpha_3$ is false, then go to ***STEP 1***.
STEP 2: If $\dot{V}$ is not negative definite, then go to ***STEP 1***.
STEP 3: Accept solutions $\varepsilon,\alpha_3,c,\alpha_1,c$. Exit.

Fig. 5 TA algorithms for the direct method of example 3.

having

$$\frac{dV}{dt}=-(2c-\varepsilon)x(x-x^3)+(\dot{Q}_0(x)-(4c-\varepsilon)x+2cx^3)y-\varepsilon y^2 \quad (29)$$

Let $\dot{Q}_0(x)-(4c-\varepsilon)x+2cx^3)=\sum_{i=1}^{N}\frac{\alpha_{2i-1}}{2i}x^{2i}$, then

$Q_0(x)=(2c-\frac{\varepsilon}{2})x^2-\frac{c}{2}x^4+\sum_{i=1}^{N}\frac{\alpha_{2i-1}}{2i}x^{2i}$, and

$$\frac{dV}{dt}=-(2c-\varepsilon)x(x-x^3)+(\sum_{i=1}^{N}\frac{\alpha_{2i-1}}{2i}x^{2i})y-\varepsilon y^2 \quad (30)$$

The function $V(x,y)$ will be positive definite iff $0\le\varepsilon\le c\le\alpha_3$ and $\alpha_1,\alpha_3,\alpha_5,...,\alpha_{2N-1}\ge 0$. For the direct method problems, the TA procedures are declared in Fig. 5.

Fig. 6 illustrates the stability regions around the equilibrium at origin. The dotted line represents the result from the direct calculation [27]; the dashed and dark solid lines represent those from the TA and the BTSO method, respectively. Fig.7 depicts the cost function, which is the inverse of the area of stable region, during the search by BTSO. MATLAB returns the area of 6.2824, 7.0999 and 7.2097 units for the stable regions found by the direct calculation, the TA and the BTSO, respectively, with the corresponding $V(x,y)$ function in (31)-(33).

$$V(x,y)=x^2+xy+0.5y^2 \quad (31)$$

, where $\varepsilon=0,c=0.5,\alpha_1=0,\alpha_3=1$

$$V(x,y)=0.9550x^2-0.0357x^4+0.9550xy+0.4775y^2 \quad (32)$$

,where
$\varepsilon=2.1663\times10^{-7},c=0.4775,\alpha_1=1.544\times10^{-7},\alpha_3=0.8124$

$$V(x,y)=0.9803x^2-0.2498x^4+0.9803xy+0.4902y^2 \quad (33)$$

, where
$\varepsilon=7.1026\times10^{-8},c=0.4902,\alpha_1=3.4153\times10^{-8},\alpha_3=0.7811$.

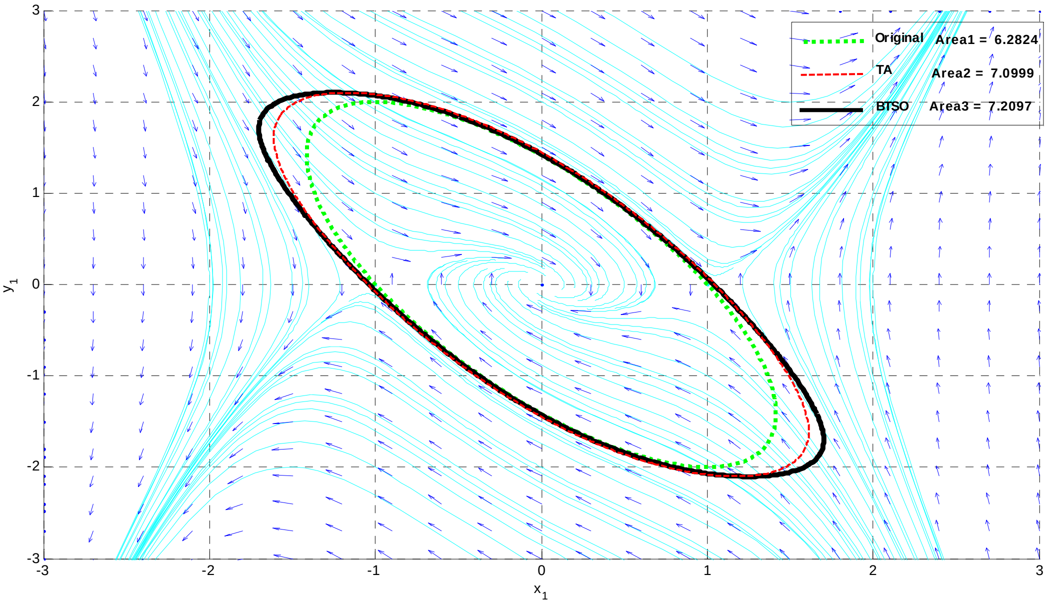


Fig.6 Stability regions of example 3.

The BTSO spent 620 iterations to reach for the $V(x, y)$ function in (33). Its area, although larger than the others, is rather limited by the coefficient constraints derived by the previous method [22]. Fig.7 depicts the convergence curve indicating that the BTSO stopped when the termination criterion (TC) was satisfied, i.e. $J = 0.1388 < 0.0139$. The recommendations for search parameter settings for the ABFO and the ATS can be found in [4] and [8], respectively.

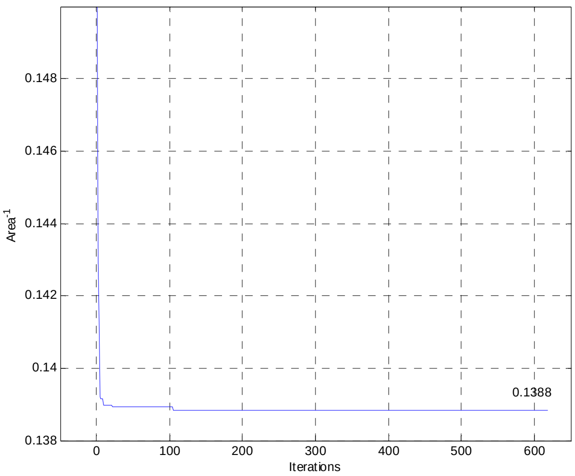


Fig. 7 Cost functions of example 3.

Example 4 Consider a nonlinear system of

$$\begin{aligned} \dot{\mathbf{x}}_1 &= -x_1 + 2x_1^2 x_2 \\ \dot{\mathbf{x}}_2 &= -x_2 \end{aligned} \tag{34}$$

having an equilibrium at the origin. This time the construction of the Lyapunov function follows the trajectories reversing technique [27]. The structure of the Lyapunov function is in the form of

$$V(x) = V(x, p) = \sum_{i=1}^{m} v_i(x)^T p_i \tag{35}$$

, in which p_i represents coefficients, and $v_i(x)$ is constructed from a homogeneous polynomial of order i. Using this technique, the function $V(x, p)$ based on (34)-(35) can be obtained as

$$\begin{aligned} V(x, p) = {} & p_{21}x_1^2 + p_{22}x_1x_2 + p_{23}x_2^2 + p_{41}x_1^4 + p_{42}x_1^3x_2 \\ & + p_{43}x_1^2x_2^2 + p_{44}x_1x_2^3 + p_{45}x_2^4 \end{aligned} \tag{36}$$

The coefficients p_{41} , p_{44} and p_{45} are set to zero [27]. Hence, the TA returns the $V(x, p)$ function in (37), whilst the BTSO returns (38). Both functions are positive definite, and

$$\begin{aligned} V(x_1, x_2) = {} & 0.273\, x_1^{\,2} + 0.546\, x_1 x_2 + 0.546\, x_2^{\,2} \\ & + 0.018\, x_1^{\,3} x_2 + 0.018\, x_1^{\,2} x_2^{\,2} \end{aligned} \tag{37}$$

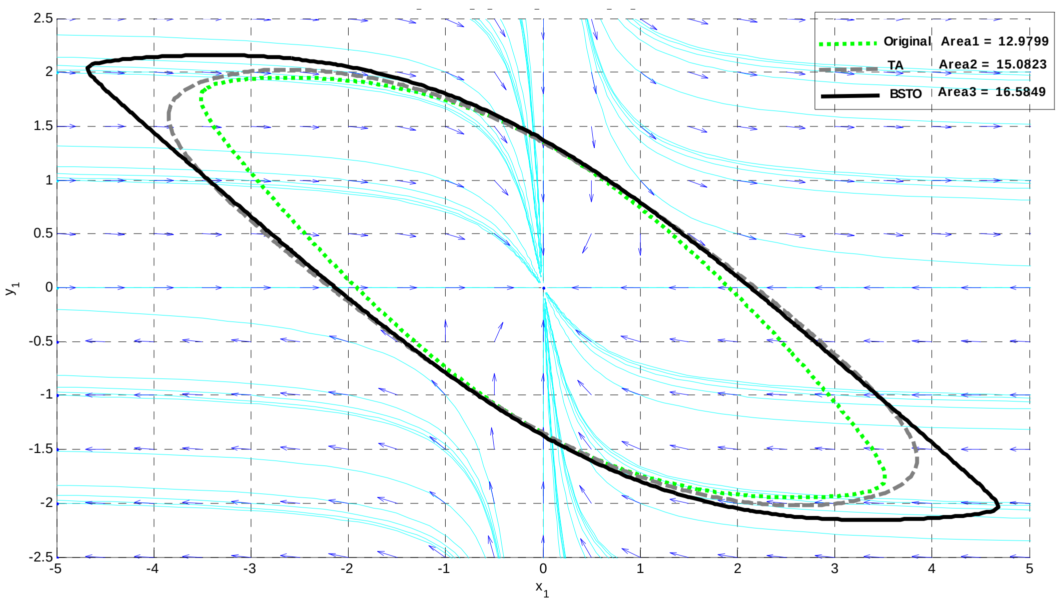


Fig. 8 Stability regions of example 4.

$$V(x_1,x_2)=0.222x_1^{\,2}+0.541x_1x_2+0.541x_2^{\,2} +0.0115x_1^{\,3}x_2+0.0183x_1^{\,2}x_2^{\,2} \qquad (38).$$

their corresponding $\dot{V}s$ are negative definite.

Fig. 8 illustrates the stability regions in the similar manner to those in Fig.6. The BTSO provides the largest region with the corresponding cost functions shown in Fig. 9. Its spent 256 iterations to reach for the solution by the TC of J=0.0603 (<0.0605).

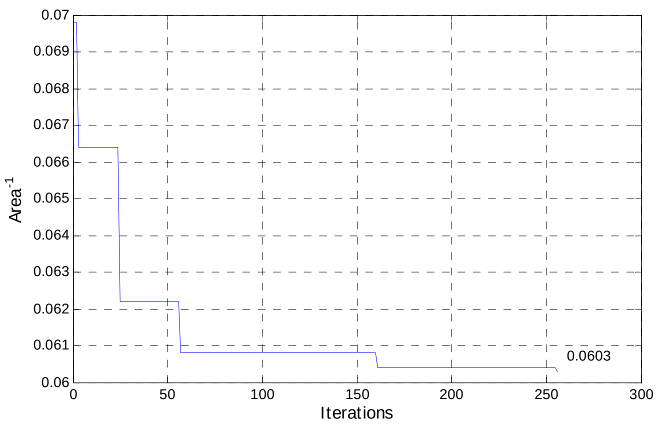


Fig.9 Cost functions of example 4.

VII. Conclusion

This paper has proposed hybrid algorithms namely bacterial foraging and tabu search optimization (BTSO). The paper has explained the adaptive bacterial foraging optimization (ABFO), the adaptive tabu search (ATS) and threshold accepting (TA) algorithm, respectively. The BTSO has been formed from the ABFO and the ATS. Using the 3D surface optimization problems, our computational results have shown the superiority of the BTSO. The proposed BTSO has been applied to stability analysis problems of nonlinear systems based on the Lyapunov's methods. The useful Lyapunov's functions have been found by searches. The results obtained by from the BTSO, the TA and the conventional methods [22], have been compared. The results show that faster convergence to the solutions of Lyapunov's functions is achieved with larger stability regions by using the proposed BTSO algorithms.

References

[1] K.M. Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control," *IEEE Control System Magazine*, vol. 22, pp. 52-67, 2002.

[2] Y. Liu and K.M. Passino, "Biomimicry of Social Foraging Bacteria for Distributed Optimization: Model, Principle, and Emergent Behaviors," *J. Optimization Theory and Applications*, vol. 115, no. 3, pp. 603-628, 2002.

[3] T. Datta, I.S. Misra, B.B. Mangaraj and S. Imtiaj, "Improved Adaptive Bacteria Foraging Algorithm in Optimization of Antenna Array for Faster Convergence," *Progress In Electromagnetics Research C.*, vol. 1, pp. 143-157, 2008.

[4] R. Majhi, G.Panda, B. Majhi and G. Sahoo "Efficient Prediction of Stock Market Indices Using Adaptive Bacterial Foraging Optimization (ABFO) and BFO Based Techniques," *Expert Systems with Applications*, vol.36, no.6, pp. 10097-10104, 2009.

[5] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers and Operations Research*, vol. 13, no.5, pp. 533-549, 1986.

[6] F. Glover, "Tabu Search – Part I," *ORSA J. Computing*, vol. 1, no. 3, pp. 190-206, 1989.

[7] F. Glover, "Tabu Search – Part II," *ORSA J. Computing*, vol. 2, no. 1, pp. 4-32, 1990.

[8] S. Sujitjorn, T. Kulworawanichpong, D. Puangdownreong and K-N. Areerak, "Adaptive Tabu Search and Applications In Engineering Design," *Book Chapters in Integrated Intelligent Systems for Engineering Design, ed. X.F. Zha and R.J. Howlett, IOS Press*, The Netherlands, pp. 233-257, 2006.

[9] D. Puangdownreong and S.Sujitjorn, "*Image approach to System Identification,*" *WSEAS Trans. Circuits and Systems*, vol.5, no.5, pp.930-938, 2006.

[10] S.Sujitjorn and S.Khawn-on, " Learning Control via Neuro-Tabu-Fuzzy Controller," *LNAI*, vol.4251, pp.833-840, 2006.

[11] D. Puangdownreong, C. U. Thaiwasin, and S. Sujitjorn, "Optimized Performance of a 2-Mass. Rotary System Using Adaptive Tabu Search," *WSEAS Trans. Circuits and Systems*, vol.5, no.3, pp.339-345, 2006.

[12] W. Ngernbaht, K-L. Areerak and S. Sujitjorn, "Resonance and friction compensations in a. Micro-hard Drive", *WSEAS Trans. Information Science and Applications*, vol.6, no.5, pp.736-745,2009.

[13] N. Sriyingyong, and K. Attakitmongcol "Wavelet-Based. Audio Watermarking Using Adaptive Tabu Search," *in Proc.1st Int.Symp. Wireless Pervasive Computing*, Phuket, Thailand, 2006, pp.1-5.

[14] J.J.E. Slotine, and W. Li, *Applied Nonlinear Control System*. Prentice Hall Inc., Englewood Cliffs,1991.

[15] H. K. Khalil, *Nonlinear Systems*, New York: MacMillan,1992.

[16] G.H. Golub, S. Nash, and C. Van Loan, "A Hessenberg-Schur Method for Problem AX+XB=C," *IEEE Trans. Automatic Control*, vol. 24, no.6, pp. 909-913, December 1979.

[17] X. A. Wang, and W. Dayawansa., "On Global Lyapunov Functions of Nonlinear Autonomous Systems," *in Proc. 38th Conf. Decision & Control*, pp.1635-1639, December 1999.

[18] T. Zhaolu, and G. Chuanqing, "A Numerical Algorithm for Lyapunov Equations," *Applied Mathematics and Computation 202*, pp.44–53, 2008.

[19] D. C. Sorensen, and Y. Zhou, "Direct Method for Matrix Sylvester And Lyapunov Equations," *J. Applied Mathematics*, vol.6, pp.277–303, January 2003.

[20] B. Grosman, and D. R. Lewin, "Lyapunov-based Stability Analysis Automated by Genetic Programming," *Automatica*, vol. 45, pp.252-256, December 2008.

[21] B. Carl, "Searching for Lyapunov Functions Using Genetic Programming,", http://www.aerojockey.com/files/lyapunovgp.pdf.

[22] L. Xiao-Lin, and J. Yao-Lin, "Numerical Algorithm for Constructing Lyapunov Functions of Polynomial Differential System," *J. Appl. Math. Comput.*, vol. 29, pp.247–262, 2009.

[23] J.R. Hewit and C. Storey, "Numerical application of Szego's method for constructing Lyapunov functions", *IEEE. Trans. Automatic Control*, vol. 14, pp.106–108,1969.

[24] C.C. Hang and J.A. Chang, "An algorithm for constructing Lyapunov functions based on the variable gradient method," *IEEE Trans. Automatic Control* , vol. 15, pp. 510–512, August 1970.

[25] G. Duech and T. Scheuer, "Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *J. Computational Physics*, vol.90, no.1, pp.161-175, 1990.

[26] G. Duech, "New optimization heuristics: Te great deluge algorithm and the record-to-record travel," *J. Computational Physics*, vol.104, no.1, pp.86-92, 1993.

[27] R .Genesio and A .Vicino, "New techniques for constructing asymptotic stability regions for nonlinear systems," *IEEE Trans. Circuits and Systems*, vol. 31, pp. 574–581, June 1984.

Suphaphorn Panikhom was born in Khon Kaen, Thailand, in 1978. She received her BEng and MEng degrees in control engineering from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand in 2003 and 2006, respectively. From June 2006 to May 2009, she has been with the Department of Electronics Engineering, Rajamangala University of Technology Isan, Khon Kaen Campus. She is currently working towards the PhD degree with the School of Electrical Engineering, Suranaree University of Technology (SUT), Thailand. Her present research interests are control systems, stability analysis of nonlinear control systems.

Nuapett Sarasiri was born in Chachoengsao, Thailand, in 1984. She received her BEng (EE) and MEng (EE) from Suranaree University of Technology (SUT), Thailand, in 2007 and 2009, respectively. She is currently working towards her PhD (EE) at SUT. Her research interests include power systems, power electronics, control, and search algorithms.

Sarawut Sujitjorn received his PhD (EE) in 1987 from the University of Birmingham, UK. He is a professor of electrical engineering at the School of Electrical Engineering, Suranaree University of Technology, Thailand. His research interests include control and intelligent systems, metaheuristics, power converters and drives.