Hindawi Publishing Corporation
Journal of Artificial Evolution and Applications
Volume 2008, Article ID 745694, 8 pages
doi:10.1155/2008/745694

Research Article

Generating Complete Bifurcation Diagrams Using a Dynamic Environment Particle Swarm Optimization Algorithm

Julio Barrera, Juan J. Flores, and Claudio Fuerte-Esquivel

Programa de Graduados e Investigación en Ingeniería Eléctrica, Universidad Michoacana de San Nicolas de Hidalgo, Ciudad Universitaria, 58030 Morelia Michoacan, Mexico

Correspondence should be addressed to Julio Barrera, julio.barrera@gmail.com

Received 20 July 2007; Accepted 28 November 2007

Recommended by Jim Kennedy

A dynamic system is represented as a set of equations that specify how variables change over time. The equations in the system specify how to compute the new values of the state variables as a function of their current values and the values of the control parameters. If those parameters change beyond certain values, the system exhibits qualitative changes in its behavior. Those qualitative changes are called bifurcations, and the values for the parameters where those changes occur are called bifurcation points. In this contribution, we present an application of particle swarm optimization methods for dynamic environments for plotting bifurcation diagrams used in the analysis of dynamical systems. The use of particle swarm optimization methods presents various advantages over traditional methods.

1. INTRODUCTION

A dynamic system is represented as a set of equations that specify how variables change over time. The minimum set of variables that uniquely determines the state of a system is called state variables. The equations in the system specify how to compute the new values of the state variables as a function of their current values and the values of the control parameters. If we allow the control parameters to change, the system changes with them. If those parameters change beyond certain values, the system exhibits qualitative changes in its behavior. Those qualitative changes are called bifurcations, and the values for the parameters where those changes occur are called bifurcation points.

If we allow one parameter to vary and plot the norm of the vector of state variables for which we find fixed points of the system, against the changing parameter, we obtain what we call a bifurcation diagram. In a bifurcation diagram, we see that fixed points may disappear, appear, or even change their stability nature as the changing parameter varies. Those changes may occur even for infinitesimal changes in the control parameter. Bifurcation diagrams are a tool used in stability analysis of dynamic systems.

Traditionally, bifurcation diagrams are plotted starting from a fixed point and using the so-called continuation methods to determine how the fixed point moves with changes in the control parameter.

In this paper, we propose an alternative by using an intelligent optimization method, namely, particle swarm optimization (PSO); we can determine all fixed points of the system for a given value of the control parameter. Iterating through the allowed range of the control parameter, we can build the whole bifurcation diagram in one pass, without the need for continuation methods.

PSO is one of the several bioinspired techniques found in artificial intelligence. PSO algorithms are inspired in the behavior of bird flocking and fish schooling [1]. When a bird finds a region with food, the others will follow it in its direction.

To illustrate our ideas and implementation work, we use a set of examples in this paper. The first examples show how the system is able to produce bifurcation diagrams for problems belonging to each one of the classes known as normal forms. Normal forms typify dynamic systems by the kind of bifurcations they may exhibit. The results match very accurately those shown in text books and the results produced with XPPAUT [2]. The last example shows a power system and its governing differential equations. Even though the system may look like a toy problem, it is in fact a benchmark in electrical engineering, for the richness of the set of qualitative features that its behavior exhibits.

The rest of the article is organized as follows. Section 2 provides the necessary background on dynamic systems and bifurcation diagrams. Section 3 mentions the particular ideas in the implementation of the PSO optimizer used in this work. (We assume the reader is already familiar with PSO, and just mention the fine details. The novice reader may consult [3].) Section 4 proposes a novel approach to the construction of bifurcation diagrams, using a PSO-based optimizer to determine all fixed points of a dynamic system. Section 5 illustrates the validity and applicability of our proposal through several examples. Finally, Section 6 concludes the work with a comparison between our proposal and traditional methods, and states several directions of research found in our agenda.

2. BACKGROUND

Electric power systems are the large physical systems interconnecting various devices to perform generation, delivery, and consumption of electricity. From an engineering point of view, the task in the power system operation is to maintain proper frequency and voltage magnitude within appropriate tolerance so that the system may operate at a stable equilibrium point in the steady state. Power system equilibrium equations which determine the system's operation state typically depend on a very large number of parameters. Hence, numerous parameters such as generation, load, network conditions, and so forth, that can change with time and circumstances, affect the system behavior. Under normal variations of those parameters, the operating point varies smoothly so the variation can be tracked by local linear analysis. However, this behavior changes qualitatively at certain critical parameter values such that the equilibrium point becomes unstable causing operational problems. One such operation problem occurs in the area of system magnitude voltages, which gradually decrease as the system load increases. This voltage decrement could be spread uncontrollably throughout the power system causing a total (blackout) or partial disruption on the power system operation. The phenomenon of this catastrophic event is referred to as a voltage collapse. Nowadays, voltage collapse problems are considered to be the principal threat to power system stability, security, and reliability in many utilities around the world [4, 5].

Several methods have been proposed to find the critical parameters that make an equilibrium point become unstable or disappear, producing a voltage collapse [5]. However, in the last decade, bifurcation theory received considerable attention from researches to increase the understanding of the complex behavior associated to the voltage collapse phenomenon [6–13]. Bifurcation theory refers to characterizing sudden changes in the qualitative response of the system as its parameters are varied smoothly and continuously over a specified range [13, 14]. When these changes relate to qualitative changes occurring in the neighborhood of an equilibrium point or limit cycle such that an equilibrium point or limit cycle appears, disappears, or losses stability, they are referred to as local bifurcations. In bifurcation problems, it is very useful to consider a space formed by system state variables and parameters, called state-control space. In this space, locations at which bifurcations occur are called bifurcation points.

The numerical analysis to locate these points is based on the principle of continuation [14, 15]. A continuation method generates a chain of solutions from an established solution of the equations representing the system under analysis. The solution branch thus established can then be examined for bifurcation points, at which a qualitative change of the preceding solution type can be observed. The representation of this branch of solutions in the state-control space is referred to as bifurcation diagram. The continuation methods applied to obtain bifurcation diagrams are based on prediction-correction schemes [15]. Generally, most prediction-correction schemes have similar prediction steps, prediction along the tangent direction, but different correction steps [15].

Opposite to the continuation methods commonly used to compute bifurcation diagrams, this paper proposes a heuristic-based method to assess these diagrams. The proposed approach solves the set of nonlinear equations representing the system under analysis by applying particle swarm optimization.

This new method is first tested to obtain bifurcation diagrams of normal forms related to generic bifurcations. In order to validate the new proposal, it is applied to a benchmark power system proposed in [8] to analyze the bifurcation diagram involved in a voltage collapse process. The model developed in [8] has been thoroughly tested by several authors [9–11, 13] in order to show that several kinds of bifurcations coexist in the voltage collapse phenomenon at different levels of system loadings. The results obtained compared well with those obtained with the popular continuation package.

3. PARTICLE SWARM OPTIMIZATION METHODS FOR DYNAMIC ENVIRONMENTS

The problem of plotting a bifurcation diagram can be stated as the problem to find all solutions of a nonlinear equation system in a given region and trace the solutions when a change in one or more parameters is introduced. This is the case of environments which present changes in the position or number of the optima of the given fitness function in the presence of one or more parameters that change. The particle swarm optimization method, developed by Li et al. [16], presents good results in this type of environments.

3.1. *Particle swarm optimization*

Particle swarm optimization is a nature inspired algorithm that has been developed for function optimization; particle swarm algorithms are inspired in the behavior of bird flocking and fish schooling [1]. When a bird finds a region with food, other birds in the flock will follow it in its direction. In the particle swarm algorithm, a point in the search space is viewed as a particle; the fitness value of a particle is equivalent to the fitness value in genetic algorithms.

In a first state, all particles have no velocity, and the particle with optimal value of fitness is selected and all the other particles are directed towards it. A velocity is computed for each of the other particles and the position of each particle is updated. This procedure is repeated for a given number of times until an optimum or a limit is reached. Variations of those algorithms have been developed for multimodal problems [3, 16].

3.2. *Genetic algorithms and species*

The particle swarm optimization method, developed by Xiaodong Li, takes two concepts from genetic algorithms for multimodal optimization: species and clearing. Dividing a population in species is a method used in genetic algorithms to preserve diversity; it consists of selecting the best available individual; all individuals with a distance less than a parameter value, called radius, with respect to the selected individual are considered to be in the same species. The selected individual and the individuals in the species are marked as used and the procedure is repeated until all individuals belong to a species. A good example is the species conservation algorithm of J.-P. Li [17]. In this method, the best individual in a species is preserved to be reinserted in the next generation of the population.

The fitness value of the individuals that are in the same species can be modified by a function to improve the probability of the best individual in the species being selected in the matting process. In some cases, only a given number of individuals are maintained with their fitness value without change, and the value of fitness of all other individuals is set to zero or they are placed in a random position within the search space; this procedure is called clearing. An example of this method is examined in the work of Pétrowski [18] where the fitness value of all individuals in a species, except for the best individual, is set to zero. A complete review of some of the methods of genetic algorithms can be found in [19].

3.3. *Hybrid particle swarm optimization method*

In addition to the basic particle swarm optimization method, two steps are added; after the initial swarm is created, the particle with the best fitness value is selected and all particles with distance less than the value of a parameter r are selected as individuals of a species; the global optimum of the particles in the species is set to the particle initially selected. The particles in the species are marked as used, and the next particle with the best fitness value available is selected to form a new species. The procedure is repeated until all particles belong to a species (the fitness value of all particles is preserved). After all species are formed, only a given number of particles are permitted in the species. Only the n particles closest to the particle with the best fitness value are preserved; the remaining particles are reinitialized at random positions.

Once the previous steps have been made, the velocity and position of all the particles are updated according to the rules given as follows:

$$\begin{aligned} v_{\text{new}} &= \chi[v + C_1R_1(P_{\text{best}} - P) + C_2R_2(P_{\text{global}} - P)], \\ P_{\text{new}} &= P + V_{\text{new}}, \end{aligned} \tag{1}$$

where v, P are the current velocity and position, respectively, R_1 and R_2 are random numbers generated in the range of $[0, 1]$, C_1 and C_2 are the learning factors, P_{best} is the position of the best fitness of the particle at the current iteration, P_{global} is the position of the particle with the best fitness in the swarm (in this case, in the species), and v_{new}, P_{new} represent the new position and velocity of the particle. The constant χ is calculated according to

$$\begin{aligned} \phi &= C_1 + C_2, \\ \chi &= \frac{2}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|}. \end{aligned} \tag{2}$$

The constant χ is called constriction coefficient [3, 16, 20], and it prevents that the particles explore too far away from the search space. Thus, we do not need a $v_{\max}$ limit for the velocity of the particles.

4. GENERATING BIFURCATION DIAGRAMS

Given the set of differential equations (see (3)) that models a dynamic system with one parameter:

$$\dot{x} = f(x, \alpha), \quad x \in R^n,\ \alpha \in R^1, \tag{3}$$

the generation of the bifurcation diagram consists of finding one stable fixed point and then using a continuation method to find the next fixed point based on the condition

$$f(x, \alpha) = 0 \tag{4}$$

which defines a smooth one-dimensional curve in R^{n+1}. Our approach to this problem is to determine all points $x \in X$, for a given region $X \subset R^n$, that satisfy (4) for a given value of the parameter α.

To find all points x using particle swarm optimization, we first transform the problem of finding all solutions for (4) into a problem of finding maxima using the transformation

$$g(x) = \frac{1}{1 + |f(x)|}. \tag{5}$$

This nontrivial transformation sends all points x such that $f(x) = 0$ to 1, and maps the domain of the function $f(x)$ to $(0, 1]$, thus zeroes of f map to maxima of function g.

Starting with a value $\alpha = \alpha_0$, the hybrid particle swarm optimization method is applied to a given number of iterations to find all maxima of function g with the fixed value α_0. Once all the points x for a given region and fixed parameter value α_0 are found, they are recorded; then the parameter α is changed to $\alpha_1 = \alpha_0 + \Delta\alpha$ and a new set of fixed points x is determined. The search is based on the information accumulated in the previous swarm. A bifurcation diagram is generated by changing the parameter α and recording all solutions found for each value α_k of the parameter.

5. RESULTS

We present two examples. The normal forms for systems with one parameter that exhibit bifurcations are examined; the

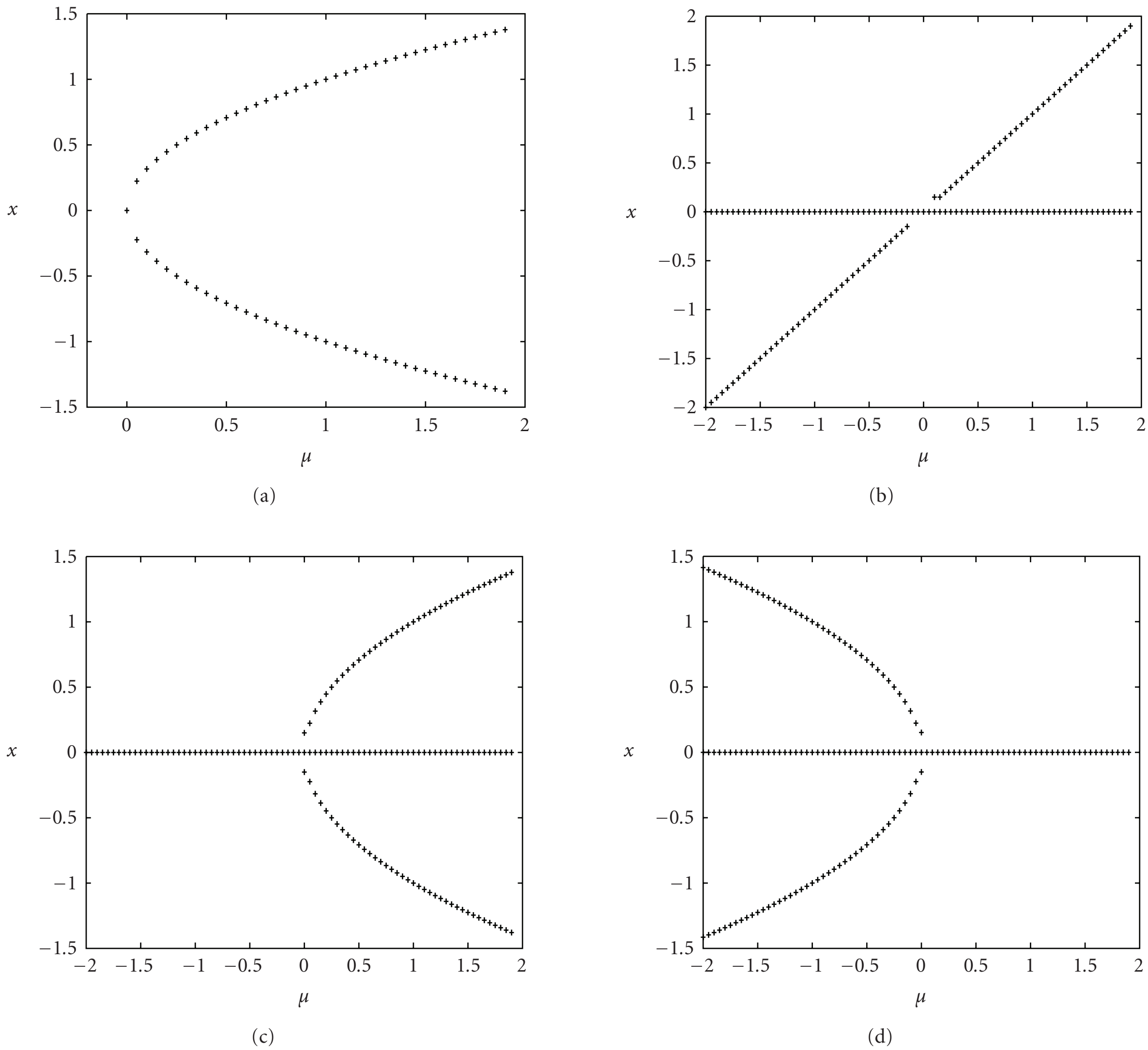

Figure 1: Bifurcation diagrams generated with the PSO for the normal forms: (a) saddle-node bifurcation (see (6)), (b) transcritical bifurcation (see (7)), (c) pitchfork supercritical bifurcation (see (8)), and (d) pitchfork subcritical bifurcation (see (9)).

bifurcation types that the systems exhibit are saddle-node, transcritical, pitchfork supercritical, and pitchfork subcritical [21, 22] (see Figure 1), and a comparative of bifurcation diagrams generated with the XPPAUT software and the particle swarm optimization method for an electrical network that illustrates the voltage collapse phenomenon arising from the load variation is presented.

5.1. Normal forms for systems with one parameter

A fixed point associated to a dynamic system where the Jacobian matrix presents all eigenvalues with nonzero real parts is called hyperbolic fixed point. If we examine the behavior of this fixed point under the variation of the parameter, the eigenvalues of the Jacobian matrix condition can only be violated in two ways: the real part of an eigenvalue approaches zero, or two eigenvalues reach the imaginary axis. When the condition is violated, the bifurcations that the dynamic system presents can be one of the bifurcations arising in the systems represented by

$$\dot{x} = \mu - x^2, \tag{6}$$

$$\dot{x} = (\mu - x)x, \tag{7}$$

$$\dot{x} = (\mu - x^2)x, \tag{8}$$

$$\dot{x} = (\mu + x^2)x. \tag{9}$$

These equations are called normal forms or topological normal forms, and they are employed to reduce a given nonlinear system to the simplest possible form preserving the dynamics in a neighborhood of the fixed point where the condition is violated. In this context, by means of a variable change, a dynamic system with one parameter, close to the hyperbolic fixed point, can be rewritten as one of the normal forms, having the same bifurcation diagram. The variable change is often nontrivial.

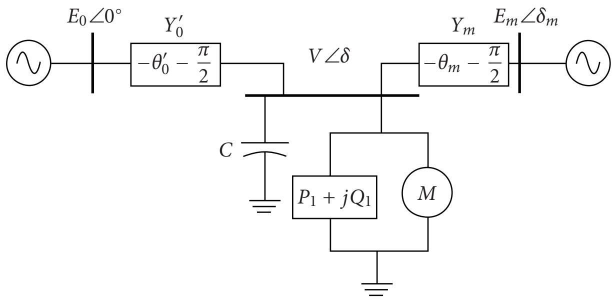

Figure 2: Diagram for the power system analyzed.

For the normal forms, the particle swarm optimization algorithm was set up with a swarm of 500 particles, a maximum of 10 particles per species, and $r = 0.15$. The search space for all normal forms was the interval $[-2, 2]$. The initial value $\mu = -2$ was incremented in $\Delta\mu = 0.05$ until $\mu = 2$. For each μ_k, the PSO was run for 100 cycles. The bifurcation diagrams that were found for each one of the normal forms are shown in Figure 1; these results match those in the bibliography (see [14, 21, 22]).

5.2. Bifurcation diagram for an electrical power system

The electrical network shown in Figure 2 has been used in [8, 23] and other papers to illustrate the voltage collapse phenomenon arising from the load variation. The three-node equivalent circuit is to be viewed as an equivalent circuit of a local area of interest connected to a large network. The network is modeled as an infinite bus represented by a voltage source providing constant voltage magnitude and phase, $E_0\theta_0$, regardless of power flow. The generator terminal bus is $E_m\delta_m$. The complex admittance of the transmission lines connected to the generator and infinite bus, a load, and a capacitor are also shown in Figure 2. The measured voltage at the load terminal is $V\delta$.

The system's dynamics are governed by the following four ordinary differential equations:

$$\begin{aligned}
\dot{\delta}_m &= \omega,\\
\dot{\omega} &= \frac{1}{M}\left(P_m - D_m\omega + VE_mY_m\sin(\delta-\delta_m-\theta_m) + E_m^2Y_m\sin\theta_m\right),\\
\dot{\delta} &= \frac{1}{K_{qw}}\left(-K_{qv2}V^2 - K_{qv}V + Q - Q_0 - Q_1\right),\\
\dot{V} &= \frac{1}{TK_{qw}K_{pv}}\left[-K_{qw}(P_0+P_1-P) + (K_{pw}K_{qv} - K_{qw}K_{pv})V\right.\\
&\qquad \left. + K_{pw}(Q_0 + Q_1 + Q) + K_{pw}K_{qv2}V^2\right].
\end{aligned} \tag{10}$$

In these equations, the state variables are defined as follows. δ_m is the generator voltage angle phase, ω is the rotor speed, δ is the load voltage phase angle, and V is the load voltage magnitude. The functions P and Q appearing in these equations represent, respectively, the active and reactive powers supplied to the load by the network. They are given by

$$\begin{aligned}
P &= -VE_0'Y_0'\sin(\delta+\theta_0') - VE_mY_m\sin(\delta-\delta_m+\theta_m)\\
&\quad + V^2(Y_0'\sin\theta_0' + Y_m\sin\theta_m),\\
Q &= VE_0'Y_0'\cos(\delta+\theta_0') + VE_mY_m\cos(\delta-\delta_m+\theta_m)\\
&\quad - V^2(Y_0'\cos\theta_0' + Y_m\cos\theta_m).
\end{aligned} \tag{11}$$

In the above equations, instead of including the capacitor in the circuit, a Thevenin equivalent circuit with the capacitor has been derived with values given by

$$\begin{aligned}
E_0' &= \frac{E_0}{\sqrt{1 + C^2Y_0^{-2} - 2CY_0^{-1}\cos\theta_0}},\\
Y_0' &= Y_0\sqrt{1 + C^2Y_0^{-2} - 2CY_0^{-1}\cos\theta_0},\\
\theta_0' &= -\frac{\pi}{2} + \tan^{-1}\frac{C - Y_0\cos\theta_0}{-Y_0\sin\theta_0}.
\end{aligned} \tag{12}$$

Other quantities appearing in (10)–(12) are constant parameters, relating to either the load or the network and generator. All of these parameters are fixed during the analysis, except for Q_1, the reactive power demand of the load. The load, network, and generator parameters are given in Table 1.

The bifurcation diagram for the system, relating the voltage magnitude V to the bifurcation parameter Q_1 (reactive power load), is obtained by the proposed approach. In order to show the validity of this result, the same analysis is carried out employing the continuation/bifurcation software package XPPAUT [24].

For the electrical network, the particle swarm optimization algorithm was set up with a swarm of 500 particles, 20-particle admittance for species, and $r = 0.1$. The intervals for the four variables for the electrical network are shown in Table 2. The initial value $Q_1 = 10$ was incremented in $\Delta Q_1 = 0.01$ until a Q_1 value of 11.5 is reached. For each Q_{1k}, the PSO was run for 100 cycles. The fixed points that were found for electrical network with the PSO and the bifurcation diagram generated by the XPPAUT software package are shown in Figure 3.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we report our research work on the application of intelligent optimization methods, namely, particle swarm optimization, to the production of bifurcation diagrams. Bifurcation diagrams are a tool for stability analysis of dynamic systems, with applications to electrical power systems, biology, chemistry, economics, and so forth.

The obtained results correspond to those produced by XPPAUT [24], which is the standard tool for producing bifurcation diagrams. This fact validates our results in the sense that the bifurcation diagrams produced with our method are as good approximations to the ideal ones as those produced by XPPAUT.

Table 1: Constant values for the symbols in the differential equation system.

Symbol	Value	Symbol	Value	Symbol	Value
K_{pw}	0.4	K_{pv}	0.3	K_{qw}	−0.03
K_{qv2}	2.1	T	8.5	K_{qv}	−2.8
P_0	0.6	P_1	0	Q_0	1.3
E'_0	2.5	P_m	1	E_m	1
M	0.3	Y_m	5	Y'_0	8
θ'_0	−0.2094	Q_1	10	θ_m	−0.08726
D_m	0.05	—	—	—	—

Table 2: Values of the ranges of search for variables.

Variable	Range	Variable	Range	Variable	Range	Variable	Range
δ_m	$[0, 1]$	ω	$[-1, 1]$	δ	$[0, 1]$	V	$[0, 2]$

Comparing our results with those produced by XPPAUT, the proposed method presents several advantages.

(1) XPPAUT requires 14 parameters to produce a bifurcation diagram, while ours depends only on four $(r, \Delta r, N_p, n)$.

(2) The sensitivity of XPPAUT to those parameters is very high, while our method is sensitive only to r, the radius. This dependence, though, affects only the accuracy of the results. If those parameters are not set right, XPPAUT may not produce any diagram at all.

(3) Besides those parameters, XPPAUT needs a stable fixed point to start a bifurcation diagram. If the engineer/scientist does not count with one, the system cannot produce any results. Our approach does not need initial conditions at all.

(4) In terms of time, XPPAUT is faster than our method. For a given dynamic system, XPPAUT may produce results in, let us say, 15 seconds, while ours may take 15 minutes. Nevertheless, taking into account the setting of all parameters that XPPAUT needs for operation, it may take a student/engineer about 3 days to complete an experiment. Using our system, the total time to produce results is still 15 minutes.

(5) XPPAUT uses a continuation method (that is why it needs initial conditions to start with) and produces only a segment of the total bifurcation diagram, stopping where it finds a bifurcation. Our approach produces the whole bifurcation diagram for the specified region.

(6) An additional effect of using our approach is to further the automation of the production of bifurcation diagrams, since they can be produced without any human intervention, other than the specification of the system (through a set of differential equations). Using XPPAUT, on the other hand, the user needs to specify new initial conditions to produce more segments of the diagram, or to select bifurcation points to continue plotting a given segment.

(7) Finally, XPPAUT produces only bifurcation diagrams of systems with one parameter (two-dimensional diagrams). Using our approach, we have already produced bifurcation diagrams for systems with more than one parameter, being able to plot them for 2 parameters varying at the same time (three-dimensional diagrams).

There are several directions to work on. You may have noticed from Figure 3 that the parabole in the bifurcation diagram presents a discontinuity; this is caused by the radius value needed to determine to which species a particle belongs. Since the method is based on the radius of species, it can only detect one solution per species, and since there is one species per region of radius r, their representative individuals must be at least at a distance r from each other. So, the method cannot detect solutions appearing closer than the radius r, even though regions may intersect. Therefore, we are missing solutions in areas where they become very close together. From a practical viewpoint, the closeness of the points does not have a practical implication given that the bifurcation region has been determined, as well as solution trajectories emanating from the bifurcation point. Hence, further calculations, such as stability analysis and computation of doubling period branches, can be done from those fixed points composing the solution trajectories. The mentioned problem can be fixed by using a variable-radius strategy, similar to the variable-step integration methods [25]. Another approach to solve this problem is to use the convex hull method presented by Barrera and Flores in [26].

One feature found in XPPAUT and not in our implementation is the determination of period-doubling fixed points. Those points correspond to complex roots of the characteristic equation of the differential equation. Those solutions represent limit cyclic or strange attractor solutions. The extension to determine those is, in principle, straightforward. By extending the search space to include imaginary components in the roots, the same search procedure will be able to find real and imaginary roots. We do expect to find detail that will need refinement, once this extension is implemented.

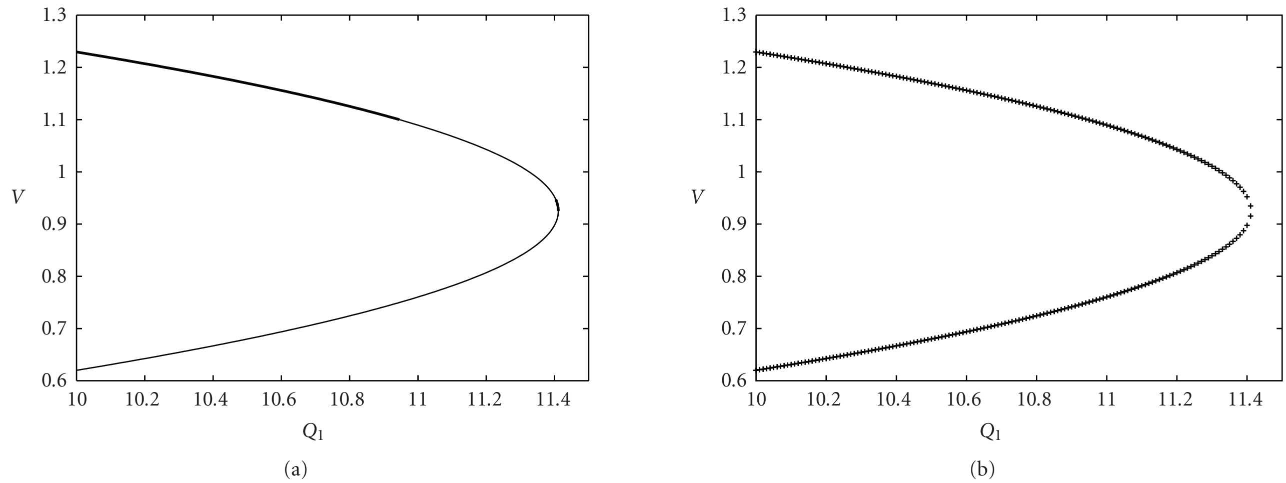

Figure 3: Bifurcation diagrams obtained (a) with the XPPAUT software package, and (b) hybrid particle swarm optimization.

Another feature not included in our implementation is discerning between stable and unstable fixed points. The determination of the nature of the fixed points has been in the literature (see [22]) for a while; so it will not present an obstacle.

In summary, we will continue this area of research, with the goals of producing alternative solutions to mathematical and engineering problems in the area of dynamical systems, and furthering the automation of the overall process, leaving more work to the computer and less to the scientist. As in all applications of computer science, the idea is not to replace the human being behind this process, but to free him/her from tedious work providing more time to do more productive work.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks (ICNN '95)*, pp. 1942–1948, Perth, Australia, December 1995.

[2] B. Ermentrout, Xppaut5.96—the differential equations tool, 2006.

[3] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, San Francisco, Calif, USA, 2001.

[4] IEEE Working Group, "Voltage stability of power systems: concepts, analytical tools, and industry experience," Tech. Rep. IEEE 90 TH 0358-2-PWR, IEEE Power Systems Engineering Committee, New York, NY, USA, 1990.

[5] T. V. Cutsem and C. Vournas, *Voltage Stability of Electric Power Systems*, Power Electronics and Power System Series, Springer, New York, NY, USA, 1998.

[6] E. H. Abed and P. P. Varaiya, "Nonlinear oscillations in power systems," *International Journal of Electrical Power and Energy System*, vol. 6, no. 1, pp. 37–43, 1984.

[7] H. Kwatny, A. Pasrija, and L. Bahar, "Static bifurcations in electric power networks: loss of steady-state stability and voltage collapse," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 10, pp. 981–991, 1986.

[8] J. S. T. Thorp, I. Dobson, and H. D. Chiang, "A model of voltage collapse in electric power systems," in *Proceedings of the 27th IEEE Conference on Decision and Control (CDC '88)*, vol. 3, pp. 2104–2109, Austin, Tex, USA, December 1988.

[9] V. Ajjarapu and B. Lee, "Bifurcation theory and its application to nonlinear dynamical phenomena in an electrical power system," *IEEE Transactions on Power Systems*, vol. 7, no. 1, pp. 424–431, 1992.

[10] E. H. Abed, J. C. Alexander, H. Wang, A. M. A. Hamdan, and H. C. Lee, "Dynamic bifurcations in a power system model exhibiting voltage collapse," Tech. Research Report, Systems Research Center, Maryland University, College Park, Md, USA, February 1992.

[11] H. D. Chiang, T. P. Connen, and A. J. lueck, "Bifurcation and chaos in electric power systems," *Journal of the Franklin Institute*, vol. 33, no. 6, pp. 1001–1036, 1994.

[12] C. A. Canizares, "On bifurcations, voltage collapse and load modeling," *IEEE Transactions on Power Systems*, vol. 10, no. 1, pp. 512–522, 1995.

[13] D. J. Hill, "Special issue on nonlinear phenomena in power systems: theory and practical impli," *Proceedings of the IEEE*, vol. 83, no. 11, p. 1439, 1995.

[14] R. Seydel, *Practical Bifurcation and Stability Analysis From Equilibrium to Chaos*, Springer, New York, NY, USA, 2nd edition, 1994.

[15] S. Richter and R. DeCarlo, "Continuationmethods: theory and applications," *IEEE Transactions on Circuits and Systems*, vol. 30, no. 6, pp. 347–352, 1983.

[16] X. Li, J. Branke, and T. Blackwell, "Particle swarm with speciation and adaptation in a dynamic environment," in *Proceedings of the 8th Annual Conference Genetic and Evolutionary Computation (GECCO '06)*, vol. 1, pp. 51–58, New York, NY, USA, July 2006.

[17] J. -P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 207–234, 2002.

[18] A. Petrowski, "A new selection operator dedicated to speciation," in *Proceedings of the 7th International Conference on Genetic Algorithms (ICGA '97)*, T. Bäck, Ed., pp. 144–151, Morgan Kaufmann, San Francisco, Calif, USA, July 1997.

[19] S. W. Mahfoud, *Niching methods for genetic algorithms*, Ph.D. thesis, Genetic Algorithms Laboratory, University of Illinois at Uraban-Champaign, Urbana, Ill, USA, 1995.

[20] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[21] S. H. Strogatz and S. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*, Perseus Books, New York, NY, USA, 2000.

[22] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory*, Springer, New York, NY, USA, 1998.

[23] K. Walve, "Modeling of power system components at severe disturbances," in *Proceedings of the 15th International Conference on Large High Voltage Electric Systems*, pp. 38–18, Ljubljana, Slowenia, August 1986.

[24] E. Doedel, A. Champneys, T. Fairgrieve, Y. Kuznetsov, B. Sandstede, and X. Wang, "Auto97: continuation and bifurcation software for ordinary differential equations," User's Guide, Concordia University, Montreal, Canada, 1997.

[25] W. J. F. Govaerts, *Numerical Methods for Bifurcations of Dynamic Equilibria*, SIAM, Philadelphia, Pa, USA, 2000.

[26] J. Barrera and J. J. Flores, "Multimodal optimization by decomposition of the search space in regions," in *Proceedings of the 7th International Conference on Intelligent System Design and Applications (ISDA '07)*, pp. 863–868, Rio de Janeiro, Brazil, October 2007.