International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016 [ICSSCCET 2016]

ISBN	978-81-929866-6-1
Website	icssccet.org
Received	25 – February – 2016
Article ID	ICSSCCET041

VOL	02
eMail	icssccet@asdf.res.in
Accepted	10 - March – 2016
eAID	ICSSCCET.2016.041

Harmony Search Algorithm for Optimal Reactive Power Dispatch

Deepak Kumar R[1], Muthusamy k[2]
Assistant Professor, [1,2]Department of EEE, Karpagam Institute of Technology, Coimbatore, Tamilnadu, India.

Abstract— *Optimal reactive power dispatch problem in power systems has thrown a growing influence on secure and economical operation of power systems. However, this issue is well known as a nonlinear, multimodal and mixed-variable problem. In the last decades, computation intelligence-based techniques, such as genetic algorithms (GAs), differential evolution (DE) algorithms and particle swarm optimization (PSO) algorithms, etc., have often been used for this aim. In this work, a harmony search-based reactive power dispatch method is proposed.*

1. INTRODUCTION

The reactive power optimization has played an important role in optimal operation of power system. It is a sub problem of the optimal power flow (OPF) calculation, which adjusts all kinds of controllable variables, such as generator voltages, transformer taps, shunt capacitors/inductors, etc., and handles a given set of physical and operating constraints to minimize transmission losses or other concerned objective functions. It is well-known that the reactive power optimization is a nonlinear and multimodal optimization problem with a mixture of discrete and continuous variables.

Harmony search is a salient meta-heuristics, which turned out to be capable of resolving a wide variety of highly nonlinear and complex engineering optimization problems with outstanding convergence performance. In this study, a harmony search algorithm is proposed to derive the optimal reactive power dispatch.

2. Problem Formulation

2.1 Objective Function

The objective of the reactive power optimization is to minimize the active power loss in the transmission network, which can be defined as follows:

$$\min p_{loss} = f(\vec{x}_1, \vec{x}_2) = \sum_{k \in N_l} g_k (V_i^2 + V_j^2 - 2V_iV_j \cos\theta_{ij})$$

Where $f(\vec{x}_1, \vec{x}_2)$ denotes the active power loss function of the transmission network $\vec{x}_1$, is the control variable vector $\vec{x}_1 = [\mathrm{V_G} \quad K_T \quad Q_C]^T$, $\vec{x}_2$ is the dependent variable vector $\vec{x}_2 = [\mathrm{V_L} \quad Q_G]^T$. is the generator voltage(continuous), T_K is the transformer tap, Q_C is the shunt capacitor/inductor (integer), is the load-bus voltage, Q_G is the generator reactive power,

Cite this article as: Deepak Kumar R, Muthusamy k. "Harmony Search Algorithm for Optimal Reactive Power Dispatch". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 218-222. Print.

$K = |(i, j)|$, $i \in N_B$ $j \in N_i$ g_k is the conductance of branch K, is the voltage angle difference between bus i and j, is the injected active power at bus i, P_{Di} is the demanded active power at bus i, V_i is the voltage at bus i, G_{ij} is the transfer conductance between bus i and j, Q_{Gi} is the injected reactive power at bus i, Q_{Di} is the demanded reactive power at bus i, N_E is the set of numbers of network branches, N_{PQ} is the set of numbers of PQ buses, N_B is the set of numbers of total buses, is the set of numbers of buses adjacent to bus I (including bus), N_0 is the set of numbers of total buses excluding slack bus, N_C is the set of numbers of possible reactive power source installation buses, N_G is the set of numbers of generator buses, N_T is the set of numbers of transformer branches, S_l is the power flow in branch l , the superscripts "min" and "max" in denote the corresponding lower and upper limits, respectively.

Control variables are self-constrained, and dependent variables are constrained using penalty terms to the objective function. So the objective function is generalized as follows:

$$f = P_{loss} + \lambda_V \sum_{N_V^{lim}} \Delta V_L^2 + \lambda_Q \sum_{N_Q^{lim}} \Delta Q_G^2$$

where λ_V and λ_Q are the penalty factors, $N_V^{\lim}$ is the set of numbers of load-buses on which voltage outside limits, $N_Q^{\lim}$ is the set of numbers of generator buses on which injected reactive power outside limits, ΔV_L and ΔV_G are defined as

$$\Delta V_L = \begin{cases} V_L^{min} - V_L, & \text{if } V_L < V_L^{min} \\ V_L - V_L^{max}, & \text{if } V_L < V_L^{max} \end{cases}$$

$$\Delta Q_G = \begin{cases} Q_G^{min} - Q_G, & \text{if } Q_G < Q_G^{min} \\ Q_G - Q_G^{max}, & \text{if } Q_G > Q_G^{max} \end{cases}$$

2.1 Problem Constraints

Equality constraints

Constraint 1: active power constraints

$$P_{Gi} - P_{Di} = V_i \sum_{j \in N_i} V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) \qquad i \in N_0$$

Constraint 2: reactive power constraints

$$Q_{Gi} - Q_{Di} = V_i \sum_{j \in N_i} V_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}) \qquad i \in N_{PQ}$$

Inequality constraints

Constraint 3: bus voltage capacity constraints

$$V_i^{min} \le V_i \le V_i^{max} \qquad i \in N_B$$

Constraint 4: transformer tap-setting constraint

$$T_K^{min} \le T_K \le T_K^{max} \qquad k \in N_T$$

Constraint 5: reactive power generation constraint

$$Q_{Gi}^{min} \le Q_{Gi} \le Q_{Gi}^{max} \qquad i \in N_G$$

Cite this article as: Deepak Kumar R, Muthusamy k. "Harmony Search Algorithm for Optimal Reactive Power Dispatch". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 218-222. Print.

Constraint 6: reactive power source installation constraint

$$Q_{Ci}^{\min} \leq Q_{Ci} \leq Q_{Ci}^{\max} \qquad i \in N_C$$

Constraint 7: power flow of each branch constraint

$$S_l \leq S_l^{\max} \qquad l \in N_l$$

3. Harmony Search Algorithm

3.1 Introduction

Calculus has been used in solving many scientific and engineering problems. For optimization problems, however, the drawbacks of differential calculus technique are

- When the objective function is step-wise, discontinuous, or multi-model
- When decision variables are discrete rather than continuous.

Harmony search (HS) algorithm was recently developed in an analogy with music improvisation process where music players improvise the pitches of their instruments to obtain better harmony.

3.2 Analogy between Music and Optimization

Musician generates a note for finding best harmony. Similarly decision variable generates a value for finding global optimum. For example, when a musician is improvising, he or she has three possible choices:

- Playing any famous tune exactly from his or her memory
- Playing something similar to the aforementioned tune (thus adjusting the pitch slightly)
- Composing new or random notes.

Geem et al. formalized these three options into quantitative optimization process in 2001, and the three corresponding components become:

- Usage of harmony memory
- Pitch adjusting
- Randomization

3.3 Algorithm

Step 1: Initialize the optimization problem and HS algorithm parameters.
Step 2: Initialize the harmony memory.
Step 3: Evaluation of Harmony Memory Matrix
Step 4: Improvise a new harmony.
Step 5: Update the harmony memory.
Step 6: Check the stopping criterion.

Step 1: Initialize the optimization problem and HS algorithm parameters

The optimization problem is specified as follows

$$\min\{ f(x) \mid x \in X \}$$

$$\text{Subject to } g(x) \geq 0 \quad \text{and } h(x) = 0$$

F(x) is the objective function, g(x) is the inequality constraint, h(x) is the equality constraint, xi is the set of each decision variable, X is the set of the possible range of values for each decision variable i.e. ${}_L xi \leq X_i \leq {}_U xi$, ${}_L xi$ is the lower bound of decision variable and ${}_U xi$ is the upper bound of decision variable. The HS algorithm parameters are Harmony memory size (HMS), or the number of

Cite this article as: Deepak Kumar R, Muthusamy k. "Harmony Search Algorithm for Optimal Reactive Power Dispatch". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 218-222. Print.

solution vectors in the harmony memory; Harmony memory considering rate (HMCR); Pitch adjusting rate (PAR); Number of decision variables (NVAR) and Number of improvisations (ITER) or stopping criterion.

Step 2: Initialize the harmony memory

The harmony memory (HM) is a memory location where all the solution vectors (sets of decision variables) are stored. The HM matrix is filled with as many randomly generated solution vectors as the HMS

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{nvar-1}^1 & x_{nvar}^1 \\ x_2^1 & x_2^2 & \cdots & x_{nvar-1}^2 & x_{nvar}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{nvar-1}^{HMS-1} & x_{nvar}^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{nvar-1}^{HMS} & x_{nvar}^{HMS} \end{bmatrix}$$

Step 3: Evaluation of harmony memory matrix

The objective function values are calculated. The solutions evaluated are sorted in the matrix in the increasing order of objective function values.

Step 4: Improvise a new harmony

Generating a new harmony is called 'improvisation'. A New Harmony vector is generated based on three rules: (a) memory consideration, (b) pitch adjustment and (c) random selection.

a). Memory Consideration

A new harmony is improvised (generated) by selecting each design variable from either harmony memory or from the entire discrete set. The probability that a design variable is selected from the harmony memory is controlled by a parameter called harmony memory considering rate (HMCR). To execute this probability, a random number r_i is generated between 0 and 1 for each variable. If this random number r_i is smaller than or equal to HMCR, the variable is chosen from harmony memory in which case it is assigned any value randomly from the corresponding column of the harmony memory. Otherwise if this random number is greater than HMCR a random value is assigned to the variable from the entire discrete set. HMCR is chosen between 0.7 and 0.95.

$$x_i' \leftarrow \begin{cases} x_i' \in \{x_i^1, x_i^2 \ldots\ldots x_i^{HMS}\} & \textbf{if } \mathbf{r_i \le HMCR} \\ x_i' \in X_i & if \quad ri > HMCR \end{cases}$$

b). Pitch Adjustment

If a design variable attains its value from harmony memory, it is checked whether this value should be pitch-adjusted or not. Similar to HMCR parameter, it is operated with a probability known as pitch adjustment rate (PAR).

If a design variable attains its value from harmony memory and if the random number is less than or equal to PAR, the new value is calculated as x_new=x_(old@)+b_(range)*∈

Where 〖x〗_(old @)the existing pitch is stored in the harmony memory and x_new is the new pitch after the pitch adjusting action. ∈ is a random number from uniform distribution with the range of [-1, 1].Otherwise the new value is randomly generated within the boundary value of the design variable. Usually PAR is chosen between 0.1 and 0.5.

c). Random Selection

The third component is the randomization, which is to increase the diversity of the solutions. Although the pitch adjustment has a similar role, it is limited to certain area and thus corresponds to a local search. The use of randomization can drive the system further to explore various diverse solutions so as to attain the global optimality.

Step 5: Update the harmony memory

After generating the New Harmony vector, its objective function value is calculated. If this value is better (lower) than that of the worst harmony vector in the harmony memory, it is then included in the matrix while the worst one is discarded out of the matrix. The updated harmony memory matrix is then sorted in ascending order of the objective function value.

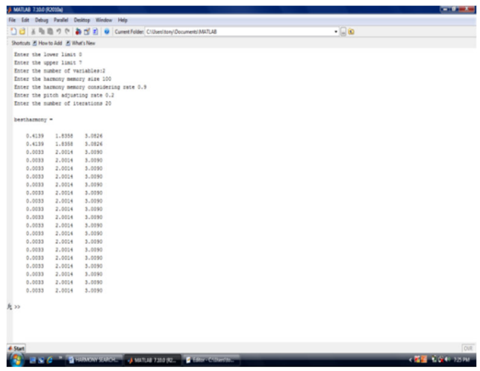
Step 6: Check the stopping criterion

If the stopping criterion (maximum number of improvisations) is satisfied, computation is terminated. Otherwise, Steps 4 and 5 are repeated.

4. Simulation and Evaluation of the Proposed Approach

The Harmony search Algorithm was developed using MATLAB R2010a.Then the developed MATLAB coding was tested using the test functions such as Himmelblau or Banana function, Sum of different power function, Easom's function, Michalewicz's function and their global optimum were found to be best.

4.1 Output for Himmelblau function



5. Conclusion

Harmony search (HS) algorithm was developed in an analogy with music improvisation process where music players improvise the pitches of their instruments to obtain better harmony. The algorithm has the additional advantage of being easy to understand, simple to implement so that it can be used for a wide variety of design and optimization tasks. Harmony search is a salient meta-heuristics, which turned out to be capable of resolving a wide variety of highly nonlinear and complex engineering optimization problems with outstanding convergence performance.In this paper, Harmony search algorithm is developed and tested using some test functions such as Himmelblau or Banana function, Sum of different power function, Easom's function, Michalewicz's function and their global optimum were found to be best. The developed Harmony search algorithm will be used to solve the optimal reactive power dispatch problem.

6. References

1. J. A. Momoh, M. E. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993 part I & II," IEEE Trans. Power Syst., vol. 14, no. 1, pp. 96–111, Feb. 1999.
2. B. Zhao, C. X. Guo, and Y. J. Cao, "A multi-agent based particle swarm optimization approach for reactive power dispatch," IEEE Trans. Power Syst., vol. 20, no. 2, pp. 1070–1078, May 2005.
3. G. A. Bakare, G. K. Venayagamoorthy, and U. O. Aliyu, "Reactive power and voltage control of the Nigerian grid system using microgenetic algorithm," in Proc. IEEE Power Eng. Soc. General Meeting, San Francisco, CA, 2005, vol. 2, pp. 1916–1922.
4. Q. H. Wu, Y. J. Cao, and J. Y. Wen, "Optimal reactive power dispatch using an adaptive genetic algorithm," Int. J. Elect. Power Energy Syst., vol. 20, pp. 563–569, Aug. 1998.
5. V. Miranda, D. Srinivasan, and L. Proenca, "Evolutionary computation in power systems," Elect. Power Energy Syst., vol. 20, no. 2, pp. 89–98, 1998.
6. N. Rajkumar, T. Vekara, and J. T. Alander, "A review of applications of genetic algorithms to power systems," Preprint Submitted to Elsevier Science, 2003.
7. M. Fozdar, C. M. Arora, and V. R. Gottipati, "Recent trends in intelligent techniques to power systems," in Proc. 42nd Int. Universities Power Engineering Conf., 2007, pp. 580–591.
8. Y. del Valle et al., "Particle swarm optimization: Basic concepts, variants and applications in power systems," IEEE Trans. Evol. Comput., vol. 12, no. 2, pp. 171–195, 2008.
9. J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, Nov. 1995, vol. 4, pp. 1942–1948.
10. H. Yoshida et al., "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," IEEE Trans. Power Syst., vol. 15, no. 4, pp. 1232–1239, Nov. 2001.

Cite this article as: Deepak Kumar R, Muthusamy k. "Harmony Search Algorithm for Optimal Reactive Power Dispatch". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 218-222. Print.