

HYBRID PARTICLE SWARM OPTIMIZATION FOR SOLVING MULTI-AREA ECONOMIC DISPATCH PROBLEM

Huynh Thi Thanh Binh

School of Information and Communication Technology,
HaNoi University of Science and Technology, Ha Noi, Viet Nam
binhht@soict.hut.edu.vn

ABSTRACT

We consider the Multi-Area Economic Dispatch problem (MAEDP) in deregulated power system environment for practical multi-area cases with tie line constraints. Our objective is to generate allocation to the power generators in such a manner that the total fuel cost is minimized while all operating constraints are satisfied. This problem is NP-hard. In this paper, we propose Hybrid Particle Swarm Optimization (HGAPSO) to solve MAEDP. The experimental results are reported to show the efficiency of proposed algorithms compared to Particle Swarm Optimization with Time-Varying Acceleration Coefficients (PSO-TVAC) and RCGA.

KEYWORDS

Multi-Area Economic Dispatch, Particle Swarm Optimization.

1. INTRODUCTION

In recent years, energy deregulation of the electricity industry has been deployed in many countries to improve the economic efficiency of power system operation. It has been used widely in a more competitive market of power industry. However, even in a competitive environment, we have to guarantee the adequate level of reliability to supply for customers. Therefore, economic dispatch (ED) is one of the main problems in many energy systems. In this paper, we focus on solving ED problem in multi-area environment, so we call this problem MAEDP (Multi-Area Economic Dispatch problem).

Our MAEDP system is comprised of three components. The generators are the electricity suppliers. The areas are where to receive and consume electricity from generators in our system. The tie-lines are the ways to transport electricity between two areas. The objective of MAEDP is to determine the generation levels and the power interchange between two areas which would minimize total fuel costs in all areas while satisfying power balance, generating limit and transmission capacity constraints. If an area with excess power is not adjacent to a power deficient area, or the tie-line between the two areas is at the transmission limit, it is necessary to find an alternative path between these two areas in order to transmit additional power.

The most simple cost function of each generator can be represented as a quadratic function as follows.

$$F_u(P_u) = a_u P_u^2 + b_u P_u + c_u \quad (1)$$

Where a_u, b_u, c_u are fuel-cost coefficients of the generator u and P_u is the generated power of the generator u .

In fact, the cost function of the *MAED* problem has non-differential points depending on the valve-point effects or the change of fuels; therefore, the cost function is usually a non-smooth function. This paper will consider two cases of non-smooth cost functions. The first case considers the valve-point effect where the objective function is normally expressed as the superposition of a sinusoidal function and a quadratic function. Another addresses multiple fuels where the objective function is described as a set of piecewise quadratic functions.

1.1. ED Problem Considering Valve-Point Effects

The generator with multi-valve steam turbines has input-output curve which is very different from the smooth cost function. To calculate the accurate cost curve of each generator, the valve-point effects must be included in the cost model. Therefore, the sinusoidal function is incorporated into the quadratic function. The fuel cost of the generator u can be formulated as follows:

$$F_u(P_u) = a_u P_u^2 + b_u P_u + c_u + |e_u \times \sin(f_u \times (P_u^{\min} - P_u))| \quad (2)$$

Where e_u and f_u are the coefficients of the generator u representing valve-point effects.

1.2. ED Problem Considering Multiple Fuels

When the generators are supplied by multiple fuel sources, the cost of each generator is represented by several piecewise quadratic functions reflecting the effects of fuel changes. The fuel cost function for such a case should be practically expressed as:

$$F_u(P_u) = \begin{cases} a_u^{l1} P_u^2 + b_u^{l1} P_u + c_u^{l1} & \text{if } P_u^{\min} \leq P_u \leq P_u^{l1} \\ a_u^{l2} P_u^2 + b_u^{l2} P_u + c_u^{l2} & \text{if } P_u^{l1} \leq P_u \leq P_u^{l2} \\ \dots & \dots \\ \dots & \dots \\ a_u^{lk} P_u^2 + b_u^{lk} P_u + c_u^{lk} & \text{if } P_u^{l(k-1)} \leq P_u \leq P_u^{\max} \end{cases} \quad (3)$$

Where a_u^{lk}, b_u^{lk} and c_u^{lk} are the fuel-cost coefficients of the generator u and $k = 1, 2, \dots, k$: the number of available fuels.

$P_u^{l(k-1)}, P_u^{lk}$: capacity consumption

P_u^{\min}, P_u^{\max} : minimum and maximum capacity of generator u

The objective of *MAEDP* is to determine the generated powers P_u of generators so that the total fuel cost for the N number of generators is minimal. Therefore, the objective function of *MAEDP* is as follows.

$$\text{Minimize } \sum_{u=1}^N F_u(P_u) \quad (4)$$

Where $F_u(P_u)$ is the fuel cost function of generator u .

Subject to:

i. Area Power Balance Constraints

$$\sum_{\forall u|u \in a} P_u + \sum_{\forall l|T_{o_l}=a} P_l = PD_a + \sum_{\forall l|From_l=a} P_l \quad (5)$$

ii. Generating Limit Constraints

$$\sum_{\forall u|u \in a} P_u + \sum_{\forall l|T_{o_l}=a} P_l = PD_a + \sum_{\forall l|From_l=a} P_l \quad (6)$$

iii. Tie-line Limit Constraints

$$-P_l^{\max} \leq P_l \leq P_l^{\max} \quad (7)$$

P_l^{\max} : maximum transmission capacity of line l ;

P_l : transmission capacity of line l .

PD_a : addition charge demand of generator a

In this paper, we propose new algorithm to solve the *MAEDP*, called Hybrid Particle Swarm Optimization (*HGAPSO*), which is the combination of Particle Swarm Optimization (*PSO*) algorithm and Genetic Algorithm (*GA*) model. This combination not only able to change search area by considering the value of $P_{best,i}$ and G_{best} , but it also improve the running time.

The rest of this paper is organized as follows. Section 2 describes the related works. Our new proposed algorithm is showed in section 3. Section 4 gives our experiments and computational and comparative results. The paper concludes with discussions and future works in section 5.

2. RELATED WORKS

Economic dispatch is a complex problem with many different models and has been set up for a long time. Through twenty years, along with the development of competitive electricity markets and smart grid technology, many new models of the problem have been introduced. At the same time, the heuristic algorithms also provide a new approach to solve the *ED* problem.

In 1994, Bakirtzis et al. proposed a Binary-Coded Genetic Algorithm for traditional *ED* problem with the generating limit constraints and power balance constraints [3, 4]. In 2005, Hamid Bouzeboudja et al. solve that problem by Real-Coded Genetic Algorithm [5] and Jong-Bae Park et al. proposed *PSO* algorithm [1, 2] for the *ED* problem which has the non-smooth cost function [6]. With smooth cost functions, they have provided the global solution satisfying the constraints. And their global solution has a high probability for 3-generator system and it is better than other heuristic approaches for 40-generator system. However, they have just solved *ED* problem with one or two area. The next year, *GA* and *PSO* approaches are modified to apply to the more complex model of *ED* problem such as: generators with valve point effect and multiple fuels, ramp rate limit, thermal generator forbidden zones...

MAEDP problem with transmission capacity limit constraints were also interested in solving in recent years. Zarei et al. introduced Direct Search Method (*DSM*) for the two-area problem in 2007 [7]. In 2008, Nasr Azadani et al. solve the two-area problem with spinning reserve constraints by *PSO* [8]. In 2009, Prasanna et al. proposed algorithms which are based on the combined application of Fuzzy logic strategy incorporated in both Evolutionary Programming

and Tabu-Search algorithms for the n-area problem. In 2010 and 2011, Manjaree Pandit et al. introduced and compare different versions of *PSO* for the *MAED* problem with n-area [8, 9]. The results show that the *PSO* algorithm achieves better solutions and faster computation time than previous methods.

In [14], Binh et al. proposed genetic algorithm, called *RCGA*, for solving *MAEDP*. The main idea of this algorithm is to make a modified *GAs* employing real valued vector for representation of the chromosomes. The use of real number representation in the *GAs* has a number of advantages in numerical function optimization compared to the binary representation as: no need to convert chromosomes to binary code; requires less memory capacity; no loss of accuracy incurred by discretion of the binary value and easy implementation of different genetic operators. This algorithm gives good results, but the running time is so great.

In next section a new approach using *PSO* will be introduced. The new algorithm can achieve better result than previous methods in some known benchmark and the running time are improved.

3. HYBRID OF GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION FOR SOLVING MAEDP

3.1. Individual Representation

We propose to apply real-coded to encode solution. An individual is represented by a chromosome whose length is equal to $U + L$ (U : the number of generators; L : the number of tie-lines in the system).

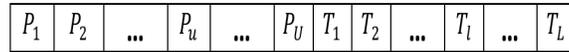


Figure 1. Individual representation

Figure 1 shows that $P_i (i=1 \rightarrow U)$ represents the generated power of the generator i , and $T_j (j=1 \rightarrow L)$ represents the transport power in tie-line j .

Each individual has two components which are position and velocity vector. With *MAEDP*, the position and velocity vector are represented bellow:

$$X_i = [(P_{i1}, P_{i2}, \dots, P_{iU}), (T_{i1}, T_{i2}, \dots, T_{iL})] \tag{8}$$

$$V_i = [(VP_{i1}, VP_{i2}, \dots, VP_{iU}), (VT_{i1}, VT_{i2}, \dots, VT_{iL})] \tag{9}$$

In the first population, we initialize randomly the velocity value as satisfying:

$$(P_{iu}^{\min} - \varepsilon) - P_{iu}^0 \leq VP_{iu}^0 \leq (P_{iu}^{\max} + \varepsilon) - P_{iu}^0 \tag{10}$$

$$or (T_{il}^{\min} - \varepsilon) - T_{il}^0 \leq VT_{il}^0 \leq (T_{il}^{\max} + \varepsilon) - T_{il}^0 \tag{11}$$

Where

ε : the very small positive real number,

P_{iu}^0 : initial power of generator u on individual i ,

T_{il}^0 : initial power of tie-line l on individual i .

VP_{iu}^0, VT_{il}^0 : initial velocity vector of power of generator u or power of tie-line l on individual i .

3.2. Fitness function

We use fitness function to estimate the optimality of solution. Fitness value of an individual is calculated by the total of the fuel cost of all generators and penal cost when the system does not guarantee the energy balance constraint. Therefore, this function is as follows:

$$Fitness = \sum_{u=1}^U F_u(P_u) + PenCoef * \sum_{a=1}^A AreaV_a \quad (12)$$

$$AreaV_a = \left| (PG_a + Pin_a) - (PD_a + Pout_a) \right| \quad (13)$$

Where P_u is the generated power of generator u .

PenCoef : the penal coefficient.

U, A : the number of generators and the number of areas respectively.

$AreaV_a$ is penal cost when the system does not guarantee the energy balance constraint and calculated as above.

3.3. Individuals updated

Each iteration, position and velocity are updated by

$$V_i^{k+1} = V_i^k * w + C_1 * R_1 * (P_{best,i} - X_i^k) + C_2 * R_2 * (G_{best} - X_i^k) \quad (14)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (15)$$

- V_i^k : velocity of individual i at iteration k ;
- X_i^k : position of individual i at iteration k ;
- $P_{best,i}$: the best position of individual i up to the iteration k ;
- G_{best} : the best position of population up to the iteration k ;
- w : inertia weight
- C_1 : cognitive acceleration coefficient;
- C_2 : social acceleration coefficient;
- R_1, R_2 : randomize number between 0 and 1.

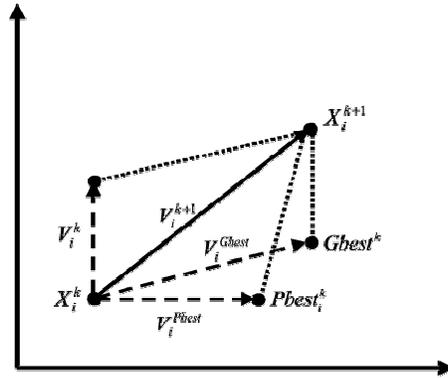


Figure 2. Individuals updated

Inertia weight: High inertia weight is good for global search while small one is good for local search [2]. So, it is better if we use high inertia weight in the first step and descending to the final step. We propose time-varying inertia weight as below.

$$w = w_{\min} + (w_{\max} - w_{\min}) \times \frac{\text{iter}_{\max} - \text{iter}}{\text{iter}_{\max}} \quad (16)$$

In which

- w_{\min}, w_{\max} : range on inertia weight;
- iter : the number of iteration;
- iter_{\max} : maximum iteration.

Acceleration factors

$$C_1 = C_{1i} + (C_{1f} - C_{1i}) \times \frac{\text{iter}}{\text{iter}_{\max}} \quad (17)$$

$$C_2 = C_{2i} + (C_{2f} - C_{2i}) \times \frac{\text{iter}}{\text{iter}_{\max}} \quad (18)$$

$C_{1i}, C_{1f}, C_{2i}, C_{2f}$: initial and final value of cognitive acceleration and social acceleration.

3.4. Crossover Operator

In population i , we choose two randomly individuals called parent_1 and parent_2 and recombine them to create two childs. The position and velocity vectors of offspring are formulated respectively as followings:

$$\text{Child}_1(x_i) = p_i * \text{parent}_1(x_i) + (1.0 - p_i) * \text{parent}_2(x_i) \quad (19)$$

$$\text{Child}_2(x_i) = p_i * \text{parent}_2(x_i) + (1.0 - p_i) * \text{parent}_1(x_i) \quad (20)$$

The velocity of the child is calculated as below:

$$Child_1(\vec{v}) = \frac{parent_1(\vec{v}) + parent_2(\vec{v})}{|parent_1(\vec{v}) + parent_2(\vec{v})|} parent_1(\vec{v}) \quad (21)$$

$$Child_2(\vec{v}) = \frac{parent_1(\vec{v}) + parent_2(\vec{v})}{|parent_1(\vec{v}) + parent_2(\vec{v})|} parent_2(\vec{v}) \quad (22)$$

p_i : randomized number between 0 and 1.

3.5. Proposed algorithm structure

1. Procedure GeneticAlgorithm
2. Begin
3. Initialize population
4. Update G_{Best}
5. While !(Maximum_Number Iterations)
6. Individual Updated
7. For $i = 1$ to number_crossover do
 - Begin
 8. Random $parent_1, parent_2$
 9. Random p_i
 10. Create $Child_1, Child_2$
 11. Calculate $Child_1(\vec{v}), Child_2(\vec{v})$
 12. Endfor
 13. Update $P_{best, i}, G_{Best}$
 14. Endwhile
 15. Return G_{Best}

4. EXPERIMENTAL RESULTS

4.1. Problem Instances

In our experiments, we used five test systems that have different sizes and nonlinearities. They are: test systems I taken from [10] (one area, 3 generators, quadratic cost function), test system II taken from [11] (2 areas, 4 generators, quadratic cost function), test system III taken from [12] (3 areas, 10 generators, with three fuel options), test system IV taken from [13] (2 areas, 40 generators, with valve-point effects), and test system V taken from [13] (2 areas, 120 generators, quadratic cost function). With each test system, we create four different values of total power demand (PD). Therefore, we have 20 test cases.

4.2. Experiment Setup

We experiment our proposed algorithm independently and compare its performance with *PSO-TVAC* [9, 10] and *RCGA* [14].

4.3. System Setting

The parameters are used in our algorithm:

Population size: 400

Max iteration: 100

Minimum inertia weight: 0.4

Maximum inertia weight: 0.9

Initial value of cognitive acceleration factors: 1.8

Final value of cognitive acceleration factors: 0.2

Initial value of social acceleration factors: 0.2

Final value of social acceleration factors: 1.9

Crossover probability: 0.7

Our system is run 20 times for each test set. All the programs are run on a machine with Intel Core 2 Duo P7450 2.13Ghz, RAM 3GB, Windows 7 Professional, and are installed in C# language.

4.4. Computational Results

Table 1. Comparison of the best results found by *HGAPSO*, *PSO-TVAC* and *RCGA*

No	Test system	PD	Min		
			PSO-TVAC	RCGA	HGAPSO
1	I	1	100,00%	100,00%	100,00%
2		2	100,00%	100,00%	100,00%
3		3	100,00%	100,02%	100,00%
4		4	100,00%	100,01%	100,00%
5	II	1	100,00%	100,52%	100,02%
6		2	100,00%	100,41%	99,98%
7		3	100,00%	100,24%	100,07%
8		4	100,00%	100,06%	100,00%
9	III	1	100,00%	97,17%	98,02%
10		2	100,00%	100,16%	99,89%
11		3	100,00%	105,23%	100,01%
12		4	100,00%	105,96%	100,00%
13	IV	1	100,00%	99,09%	100,62%
14		2	100,00%	100,35%	101,77%
15		3	100,00%	101,44%	104,19%
16		4	100,00%	99,30%	100,13%
17	V	1	100,00%	104,25%	101,12%
18		2	100,00%	100,55%	99,19%
19		3	100,00%	103,32%	103,00%
20		4	100,00%	102,59%	103,01%

Table 2. Comparison of the average results found *HGAPSO*, *PSO-TVAC* and *RCGA*

No	Test system	PD	Average		
			PSO-TVAC	RCGA	HGAPSO
1	I	1	100,00%	99,91%	100,10%
2		2	100,00%	99,94%	100,04%
3		3	100,00%	100,03%	100,03%
4		4	100,00%	100,02%	100,00%
5	II	1	100,00%	99,27%	99,25%
6		2	100,00%	99,87%	99,75%
7		3	100,00%	100,03%	99,96%
8		4	100,00%	116,41%	99,96%
9	III	1	100,00%	93,57%	95,93%
10		2	100,00%	101,40%	99,84%
11		3	100,00%	123,64%	100,00%
12		4	100,00%	111,87%	99,99%
13	IV	1	100,00%	95,81%	99,51%
14		2	100,00%	98,12%	102,86%
15		3	100,00%	98,02%	103,39%
16		4	100,00%	96,79%	101,91%
17	V	1	100,00%	101,54%	101,79%
18		2	100,00%	100,16%	101,76%
19		3	100,00%	100,09%	100,97%
20		4	100,00%	99,29%	103,62%

Table 3. Comparison of the running time found by *HGAPSO*, *PSO-TVAC* and *RCGA*

No	Test system	PD	Time		
			PSO-TVAC	RCGA	HGAPSO
1	I	1	100,00%	241,65%	89,49%
2		2	100,00%	255,36%	91,39%
3		3	100,00%	254,70%	91,72%
4		4	100,00%	254,47%	94,88%
5	II	1	100,00%	280,11%	86,04%
6		2	100,00%	278,21%	87,92%
7		3	100,00%	286,46%	89,19%
8		4	100,00%	281,13%	88,65%
9	III	1	100,00%	450,02%	81,35%
10		2	100,00%	455,81%	82,32%
11		3	100,00%	457,01%	82,75%
12		4	100,00%	462,80%	83,10%
13	IV	1	100,00%	614,82%	79,29%
14		2	100,00%	611,01%	79,49%
15		3	100,00%	622,53%	79,58%
16		4	100,00%	617,50%	78,76%
17	V	1	100,00%	975,79%	87,59%
18		2	100,00%	1198,74%	89,69%
19		3	100,00%	1257,07%	90,97%
20		4	100,00%	1258,36%	91,32%

- Table 1 shows that the best results found by *HGAPSO* are better than or equal to *PSO-TVAC* and *RCGA* in 9/20 test cases.
- Table 2 shows that the average results found by *HGAPSO* are better than or equal to *PSO-TVAC* and *RCGA* in 8/20 test cases.
- Table 3 shows that the running time of *HGAPSO* is better than *PSO-TVAC* and *RCGA* in all of the test cases.
- The experiment results show that the combination between *PSO* and *GA* can find better results in the fastest running time.

5. CONCLUSION

In this paper, we proposed new hybrid particle swarm optimization algorithm for solving *MAEDP*. We experimented on five test systems. With each test system, we create 4 test cases which are different from the value of total power demand. The results show that our proposed approaches are stable and quite effective with *MAEDP*. The running time of *HGAPSO* is fastest compare to *PSO-TVAC* and *RCGA*.

In the future work, we are planning to improve the algorithm for solving *MAEDP* with more constraints. Moreover, we hope that we can find the other approach with better results for *MAEDP*.

REFERENCES

- [1] Kennedy J. & Eberhart R., (1995)“Particle swarm optimisation”, Proceedings of the IEEE international conference of neural network (ICNN’95), Vol. IV, pp1942-1948.
- [2] Shi Y. & Eberhart R., (1998) “A modified particle swarm optimizer”, IEEE, pp69-73.
- [3] Bakirtzis A., Petridis V. & Kazarlis S., (1994) “Genetic algorithm solution to the economic dispatch problem”, Generation, Transmission and Distribution, IEEE Proceedings, Vol. 141, No. 4, pp377-382.
- [4] H. Bouzeboudja, A. Chaker, A. Allali & B. Naama., (2005)“Economic dispatch solution using a real-coded genetic algorithm”, Acta Electrotechnica et Informatica, Vol. 5, No. 4.
- [5] Jong-Bae Park, Ki-Song Lee, Joong-Rin Shin & Lee K.Y., (2005) “A particle swarm optimization for economic dispatch with nonsmooth cost functions”, IEEE Transactions on Power Systems, Vol. 20, No. 1, pp34- 42.
- [6] M. Zarei, A. Roozegar, R. Kazemzadeh & J.M. Kauffmann., (2007) “Two Area Power Systems Economic Dispatch Problem Solving Considering Transmission Capacity Constraints”, World Academy of Science, Engineering and Technology.
- [7] Azadani E.N., Hosseinian S.H., Moradzadeh B. & Hasanpor P., (2008) “Economic dispatch in multi-area using particle swarm optimization in electricity market”, Power System Conference, 2008. MEPCON 2008. 12th International Middle-East, pp. 559-564.
- [8] M Sharma, M Pandit & L Srivastava., (2010)“ Multi-area economic dispatch with tie-line constraints employing evolutionary approach”, International Journal of Engineering, Science and Technology, Vol. 2, No. 3, pp132-149.
- [9] Rameshwar Singh, Kalpana Jain, Manjaree Pandit, (2011) “Comparison of PSO variants with traditional solvers for large scale multi-area economic dispatch”, International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2011), pp. 304-309.
- [10] F. Herrera, M. Lozano & J. L. Verdegay, (1995) “Tuning fuzzy controllers by genetic algorithms”, Int. J. Approx. Reas, Vol. 12, pp. 299-315.
- [11] Lin C.E. & Viviani G.L, (1984) “Hierarchical economic load dispatch for piecewise quadratic cost functions”, IEEE Transactions Power Apparatus and systems, Vol. PAS-103, pp.1170-1175.
- [12] Sinha N., Chakraborty R. & Chattopadhyay P.K., (2003) “Evolutionary programming techniques for economic load dispatch”, IEEE Transactions on Evolutionary Computation, Vol. 7, No. 1, pp83-93.
- [13] P.H. Chen and H.C. Chang, (1995) “Large scale economic dispatch approach by genetic algorithm”, IEEE Transactions on Power Systems, Vol. 10, No. 4, pp. 1919-1926.
- [14] Huynh Thi Thanh Binh, Tran Kim Toan, (2013) “Real-Coded Genetic Algorithm for Solving Multi-Area Economic Dispatch Problem”, 2013 IEEE Symposium Series On Computational Intelligence, Singapore, pp. 97-101.\

Author

Huynh Thi Thanh Binh has received Master and PhD degree in Computer Science field at Ha Noi University of Science and Technology on 1999 and 2011. Her research is computational Intelligence: genetic algorithms, heuristic algorithms. Dr. Huynh Thi Thanh Binh is now lecturer at Ha Noi University of Science and Technology, School of Information and Communication Technology. Dr. Binh is member of IEEE (from 2006 to now). Now, she is treasure of IEEE Viet Nam Section. Dr. Binh also is member of ACM. Dr. Binh has served as the organizing chair of the SoICT2011, SoICT2012, a program committee of the international conference SoCPaR 2013, KSE 2013, ACIIDS 2013, SEAL 2012, ICCASA 2012, ICCCI 2012, KSE 2012, reviewer of International Journal Intelligent Information and Database Systems, INFORMS Journal on Computing, Journal of Theoretical and Applied Computer Science.

