

# A Novel Particle Swarm Algorithm Based Multi-agent for Coordinated Volt/Var Optimization

Wang Qiang<sup>1, a</sup> and Chen Jun<sup>2, b</sup>

<sup>1</sup> College of Electrical Engineering & New Energy, China Three Gorges University, Yichang, China

<sup>2</sup>Power Supply of Chongren County, Jiangxi Province, China

<sup>a</sup>wq00518@sohu.com, <sup>b</sup>chenjun@126.com

**Keywords:** Power system; Particle swarm algorithm; Multi-agent system; Coordinated volt/var optimization

**Abstract.** Coordinated volt/var optimization is fundamental to power dispatch and stability of power system, a novel particle swarm algorithm based multi-agent(MAPSO) is presented. The algorithm combines particle swarm optimization algorithm(PSO) with multi-agent system(MAS). The PSO-MAS utilize an agent as a particle to solve the optimization problem. Every agent lives in a fixed lattice with competition and cooperation by using its knowledge among neighbors. MAPSO integrates these agent-agent interactions and evolution mechanism of PSO to minimize the value of objective function. To test the performance of MAPSO for coordinated Volt/var optimization, simulation experiments were done on an IEEE 30-bus power system. The results show that MAPSO can converge to optimal solutions faster than many other approaches, and be applied to control volt/var to decrease the power loss and improve power quality.

## Introduction

Reactive power balance is a necessary condition to ensure voltage quality in power system. Reactive power optimization can make full use of reactive power sources, improve voltage quality, reduce grid loss and enhance voltage stability. Commonly used control means include ULTC, adjustable capacitor and generator. Essentially, reactive optimization is a discrete and nonlinear combination optimizing problem with constraints, solved by linear programming, nonlinear programming, quadratic programming, sensitivity analysis, mixed integer method[1] etc. All these approaches have their own advantages and fitness, but even if they assume that control variables are continuous, and that object functions are differential, sometimes only local optimal solution may be obtained consuming long time, and trapped into “dimension disaster” easily[2-3].

Particle swarm optimization algorithm (PSO) is a kind of technology evolution, from the study of birds of prey behavior, it essentially is belong to iterative random search algorithm, has the parallel processing characteristic, good robustness, easy to implement, principle can find the global optimal solutions of optimization problems with larger probability and high computational efficiency, has been successfully applied to solve all kinds of complex optimization problems [4]. Recently, particle swarm optimization (PSO) algorithm is introduced for power system, literature [5-7] were used for solving the unit commitment, transmission network expansion planning, and capacitor compensation for optimal allocation of power system optimization problems, have achieved good results.

Multi-agent system(MAS) is a network composed of agents which have the abilities of sensing, solving problems and communicating with each other. All these agents are distributed physically and logically, and behavior automatically, and complete complex tasks by negotiating, coordinating and cooperating.

A novel approach combining PSO with MAS(MAPSO) is constructed to solve Volt/Var optimization problem. Firstly, we create a lattice environment for agents to live. Each agent, namely a particle in PSO, is fixed in a lattice where it may compete and cooperate with neighbors, and evolve by PSO and self-learning continually to update its position to global optimal in the whole solution space fast and accurately. Finally, some simulation tests on IEEE 30 bus system are implemented, and the approach shows its outstanding merits of high convergence speed and computational accuracy.

## Reactive Power Optimization Model

Reactive power optimization is usually to modify control such variables as transformation ratio of adjustable transformer, compensation capacitor and the generator terminal voltage so on, in order to reduce active power loss and ensure the voltage within the provision range[8]. In this paper, the goal of minimizing the net loss can be expressed by the following equations:

$$\min f_Q = \sum_{k \in N_E} P_{kloss} = \sum_{k \in N_E} g_k (V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij}) \quad (1)$$

$$\text{s.t. } 0 = P_{Gi} - P_{Di} - V_i \sum_{j \in N_i} V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \quad i \in N_i, i \neq s \quad (2)$$

$$0 = Q_{Gi} - Q_{Di} - V_i \sum_{j \in N_i} V_j (G_{ij} \sin \theta_{ij} + B_{ij} \cos \theta_{ij}) \quad i \in N_{PQ} \quad (3)$$

$$V_{i,\min} \leq V_i \leq V_{i,\max} \quad i \in N_B \quad (4)$$

$$Q_{Gi,\min} \leq Q_{Gi} \leq V_{i,\max} \quad i \in N_G \quad (5)$$

$$T_{k,\min} \leq T_k \leq T_{k,\max} \quad k \in N_T \quad (6)$$

$$Q_{Ci,\min} \leq Q_{Ci} \leq Q_{Ci,\max} \quad i \in N_C \quad (7)$$

Where  $NE$ 、 $NPQ$ 、 $NG$ 、 $NB$ 、 $NT$  and  $NC$  are sets for branches,  $P-Q$  nodes, generator nodes, whole nodes, transformer branches and compensator nodes respectively.  $N_i$  is a set for all the nodes connected with node  $i$  and itself.  $s$  are balance nodes.  $P_{kloss}$  is active power loss of branch  $k$ .  $g_k$  is conductance of branch  $k$ .  $G_i$  and  $B_{ij}$  are coefficients of node admittance.  $P_i$  and  $Q_i$  are active power and reactive power injection respectively.  $V_i$  is voltage value of node  $i$ .  $\theta_{ij}$  is voltage phase difference between node  $i$  and  $j$ .  $Q_{Gi}$  is reactive generation power of node  $i$ .

Power balance equations would be used as equality constraints, node voltage, reactive generation power, transformer ratio and capacitance of compensation capacitor as inequality constraints. Generator terminal voltage, transformer ratio and capacitance of compensation capacitor of every node are control variables, so their constraints are satisfied by themselves. Voltage and reactive generation power of  $PQ$  nodes are status variables, therefore, they could formulate in punishment function as Eq.8:

$$F_Q = f_Q + \sum_{i \in N_{PQ}} \lambda_{Vi} (V_i - V_{i,\lim})^2 + \sum_{i \in (N_G + N_C)} \lambda_{Gi} (Q_{Gi} - Q_{Gi,\lim})^2 \quad (8)$$

Where  $\lambda_{Vi}$  and  $\lambda_{Gi}$  are punishment factors;  $V_{i,\lim}$  and  $Q_{Gi,\lim}$  can be expressed as:

$$V_{i,\lim} = \begin{cases} V_{i,\max}; & V_i > V_{i,\max} \\ V_{i,\min}; & V_i < V_{i,\min} \end{cases} \quad (9)$$

$$Q_{Gi,\lim} = \begin{cases} Q_{Gi,\max}; & Q_{Gi} > Q_{Gi,\max} \\ Q_{Gi,\min}; & Q_{Gi} < Q_{Gi,\min} \end{cases} \quad (10)$$

## MAPSO Algorithm

**Standard PSO Algorithm.** PSO algorithm[9] can obtain the solutions of optimization problems through information sharing and individual experience summary to modify individual action strategy. PSO algorithm initializes a particle swarm randomly, and then search the optimal solution through iteration. In each iteration, the particle updates itself by tracking the two extremes. One of them is the individual optimal solution  $p_{Best}$  found by the article itself, the other one is the global optimal solution  $g_{Best}$  obtained by the population. Each particle updates its velocity and position in solution space according the following equations:

$$v_{d+1} = w \cdot v_d + \varphi_1 \cdot rand() \cdot (p_{Best} - x_d) + \varphi_2 \cdot rand() \cdot (g_{Best} - x_d) \quad (11)$$

$$x_{d+1} = x_d + v_{d+1} \quad (12)$$

Where, subscript  $d$  denotes iteration times, and  $xd$  and  $vd$  are particle position and velocity in  $d$  iteration respectively.  $w$  is inertia constant, and  $\phi_1$  and  $\phi_2$  are learning factors, and  $rand()$  is a random ranged from 0 to 1.

**Agent and MAS.** Agent is a software entity which has the abilities of perception, problem solving and communication with other agent in the system so as to complete one or more functional goals. Agent has the following typical characteristics usually[10]:

- 1) In general, Agent lives in a specific environment, and Agent can only work in the environment.
- 2) Agent can perceive the environment located in, and make corresponding actions.
- 3) Agent can control its behavior or action autonomously without external intervention to perform its specific tasks.

MAS is a network composed of a number of loosely coupled, coarse grained Agent. These Agents are distributed in physical or logical, and their behaviors are autonomous, and they can accomplish complex control tasks and solve complicated problems through consultation, coordination and cooperation.

In general, MAS in solution of a problem needs to define the following four elements for every Agent: the intention and purpose, living environment, local environment and action strategies to achieve Agent's own intention and purpose.

MAPSO algorithm presented in this paper will explained in detail based on these four elements in the following.

**MAPSO Algorithm.** MAPSO algorithm is constructed with the main feature of PSO algorithm and MAS. In the survival environment built for Agent, every Agent is able to not only compete and cooperate with its neighbors and self-learn, but also adopt the evolution mechanism of PSO algorithm so that it can share information with global optimal Agent and correct the action strategies according to its own experience to converge global optimum rapidly and exactly.

#### (1) Definition of Agent intention

In MAPSO, suppose that any Agent  $\alpha$  equivalent to a particle in PSO has a fitness value determined by optimal problem. In solving reactive power optimization, the fitness value of Agent  $\alpha$  can be calculated as the Eq.8, that is,  $f(\alpha) = FQ$ .

The purpose of Agent  $\alpha$  is to reduce its fitness as much as possible under the operating conditions. In order to achieve its purpose, Agent will take corresponding actions to respond to its environment.

#### (2) Definition of environment

As for environment, a simple lattice structure is designed in this work, shown in Fig.1. Every Agent lives in the environment, and it is fixed in one of lattices. In Fig.1, each circle represents an Agent, and the data in the circle denotes the location of the Agent in the environment, which contains two elements, namely velocity and position of particle in PSO.  $Lsize$  is a positive integer, therefore, the total number of lattices is  $Lsize \times Lsize$ , equivalent to the size of population in PSO.

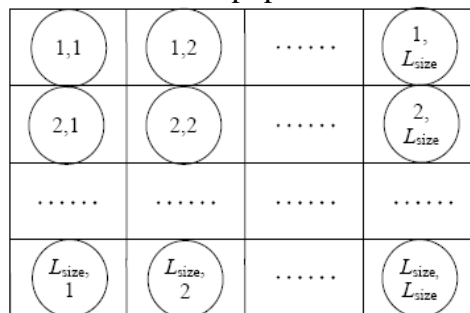


Fig.1 Environment for Agent fixed in the lattice

## Case Analysis

In order to verify the optimization effect of MAPSO, IEEE 30- bus power system is used as a case to compute reactive power optimization. The case system has 41 branches, 6 generator nodes and 22 load

nodes. The 6 generator nodes are indexed as 1,2,5,8,11 and 13, among which node 1 is balance node, and the reminding ones are  $P-V$  nodes. All the other nodes except generator ones in the system are  $P-Q$  nodes. Adjustable transformer branches are 6~9,6~10,4~12 and 27~28, and parallel capacitor nodes are 3,10 and 24. The total active power load  $P_{load}$  is 2.834, and reactive power load  $Q_{load}$  is 1.262. In MAPSO,  $Lsize$  usually ranges from 3 to 10, here  $Lsize=6$ , maximal iteration number  $Tmax=300$ , inertia constant  $w=0.7298$ , learning factor  $\phi_1=\phi_2=1.49618$ ,  $sLsize=3$  in self-learning operation, and local search radius  $sR=0.2$ .

In initialization, set ratios of generator terminal voltage and transformer to 1.0. We can obtain through flow calculation:  $\sum P_G=2.893857$ ,  $0.9801 \sum Q_G=99$ ,  $P_{loss}=0.059879$ . Voltage over-limits for three  $P-Q$  nodes:  $V_{26}=0.932$ ,  $V_{29}=0.940$ ,  $V_{30}=0.928$ ; Reactive power generation over-limit:  $Q_G=0.569$ . To test the efficiency of MAPSO, comparisons are implemented with PSO and SGA. Optimal results for 3 algorithms in 30 iterations are given in Table 1. From Table 1, grid power loss  $P_{loss}=0.048747$ , and grid power loss reduction ratio  $\eta_{SAVE}=18.5900\%$  gotten by MAPSO, which are better than SGA and PSO.

Table 1 Comparison of optimal results for different methods

	$\sum P_G$ [pu]	$\sum Q_G$ [pu]	$P_{loss}$ [pu]	$Q_{loss}$ [pu]	$P_{SAVE}$ [pu]	$\eta_{SAVE}$ [%]
SGA	2.88380	1.02774	0.049800	-0.23426	0.01008	16.8400
PSO	2.88330	0.82500	0.049262	-0.22920	0.01062	17.6200
MAPSO	2.88270	0.81950	0.048747	-0.22836	0.01113	18.5900

## Conclusion

A novel MAPSO algorithm is presented in this work. The algorithm constructs a MAS environment, where every agent, namely a particle in PSO, can compete and cooperate with neighbor and learn by itself through integration with PSO evolution mechanism. Simulation test results based IEEE 30 bus system show that The MAPSO algorithm can obtain the global optimal solution rapidly and exactly. Therefore, as a new attempt, this approach is very significant to solve the combination optimization problem with highly complex constraints in power system.

## References

- [1] Mamandur K R C, Chenoweth R D, in: IEEE Trans. on PAS, vol. 100 (2007), p. 3185
- [2] Yan Wei, and Luo Chunlei, in: Proceedings of the CSEE, vol. 23(2013), p. 76
- [3] Lee K Y, in: IEEE Transactions on Power System, vol. 13(2008), p. 101
- [4] Kennedy J, Eberhart R, in: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, vol. 4(2009), p.1942
- [5] Hu Jia sheng, and Cao Yijia, in: Proceedings of the CSEE, vol. 24(2014), p.24
- [6] Yuan Xiaohui, and Zhang Yongchuan in: Power System Technology, vol. 28(2015), p.14
- [7] Yu Xinmei, Li Yan, and Xiong Xinyin, in: Proceedings of the CSEE, vol. 23 (2013), p.26
- [8] Ma Jintao, Lai L L, and Yang Yihan, in: Proceedings of the CSEE, vol. 15(2011), p.347
- [9] Eberhart R C, and Shi Y, in: Proceedings of the Congress on Evolutionary Computing, IEEE Service Center, California, USA, (2014), p. 84
- [10] Liu Jiming, Jing Han, and Tang Y Y, in: Artificial Intelligence, vol. 136(2012), p.101