# Solving Unit Commitment and Economic Dispatch Simultaneously Considering Generator Constraints by Using Nested PSO

Murtaza Farsadi[1], Hadi Hosseinnejad[2], and Tohid Sattarpour Dizaji[1]

[1] Urmia University, Urmia, Iran
m.farsadi@urmia.ac.ir, t.sattarpur@yahoo.com
[2] Department of Electrical Power Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
h.hosseinnejad@iaurmia.ac.ir

## Abstract

**Unit Commitment and Economic Dispatching are the most important part of every plant-planning project. Unit Commitment is the problem that searches the economical way for power generation, when the power consumption and altered constraints of the power plants are considered. In addition, economic dispatch as the process of power plants to produce energy at the lowest cost to satisfy consumers and considering any operating limits related to generation and transmission is complement of it and another important intention in power systems. In the past years, these kinds of problems are used to solve just by solving first one of them then another one. In this paper, it's decided to solve this two-parted problem simultaneously. For reaching this goal, it's used Particle Swarm Optimization twice but simultaneously as nested PSO. The other important part is considering some kind of limits that usually faced in these solutions, which is placed in this paper.**

## 1. Introduction

As the main part of every project for constructing a power plant, it is necessary to find out which kind of plants must be used. Immediate it's find out, the second action is programming this plants for working in a special schedule of a special region. Economic Dispatch (ED) is one of the fundamental issues in power system operation. In addition, it is an optimization problem and its objective is optimizing the total generation cost of plants, while satisfying limits. Earlier struggles on solving ED problems have hired various optimization methods. These methods contain the lambda iteration method, the base point and participation factors method, and the gradient method etc. [1, 2, and 3].

The used method in this paper for solving these two problems together is nested Particle Swarm Optimization (PSO). Nested PSO is a sub-group of Particle Swarm Optimization (PSO), which solve two or more problem together and simultaneously.

The first action which named Unit Commitment (UC) that showed in "Fig. 1". In this figure, the plants showed by rectangular and consumers illustrated by trapezius. The cross sign shows that each plant inject the power or not. The circles can be distributed generation or capacitors which are separated from network. The lines between plants and consumers show the transmission lines in power network. The second action and scheduling named Economic Dispatch that showed in "Fig. 2".

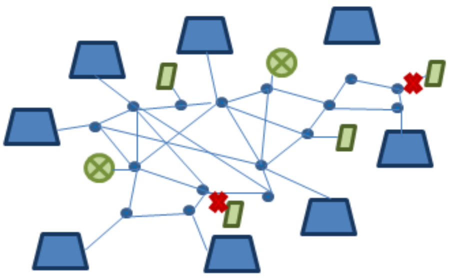

**Fig. 1.** Representation of unit commitment solution.

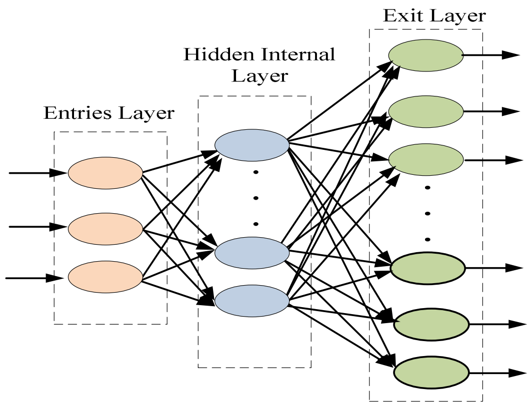


**Fig. 2.** Economic Dispatch for a typical network.

In this figure there are some steps to find out the amount of power that each plant injects. The first circles here are the priory answers in entries layer and middle one shows the hidden layer which the algorithm apply on the first layer and exit layer is final answers. In this typical example, there is units as initial answers.

The main goal in UC is finding out the type of plants that must use to most efficiency and find participating amount of power for every founded plant. In past efforts for solving this problem, some kind of methods has been used as it mentioned including lambda-iteration, gradient method and using evolutionary algorithms [4, 5, 6, and 7].

Genetic Algorithm (GA), PSO and dynamic programming are most-used methods in past decades [8, 9, 10, 11 and 12]. An optimization method known as GA that is a kind of probabilistic heuristic algorithms is using methods inspired by natural manner, such as inheritance, mutation, selection, and crossover [13]. Particle swarm optimization (PSO) is one of other methods for optimization that is a computational algorithm to optimize by iteratively trying to improve a candidate solution. PSO optimizes a problem by having a population of first accidental solutions [14]. Dynamic programming is other method for solving a complex problem by breaking it down into a collection of simpler sub problems. It is applicable for problems to exhibiting the properties of overlapping sub problems and optimal substructure [15]. The most important thing, which must be noticed, is that each of the previous methods has some problems, i.e. the Dynamic Programming method may finally effects on the sizes of the ED problem, thus requiring giant computations. About the GA method which in the past have been hired successfully to solve many complicate optimization problems, it's important to know that there are lacks in GA. i.e., where the parameters being optimized are so depend on crossover or mutation or both together, most of the time the offsprings are the same as old generation [16]. Moreover, the GA by finding a local optimum led no improvement sometimes and stuck in a special place [16 and 17]. PSO developed by 'Kennedy' and 'Eberhard', is one of the optimization algorithms. It is used in solving continuous nonlinear optimization problems [18, 19]. The PSO technique can produce first-rate solutions with littler calculation and stable answers than other methods [18, 19]. In this paper, it's hired PSO method for optimizing the ED problem and UC problem together.

The proposed method studies the features of a generator such as ramp rate limits and prohibited operating zone which it's seen in an actual power system operation. The feasibility of the proposed method examined on a typical network consists on some plants.

## 2. Methodology

Nested PSO is a kind of PSO that it's used. In this algorithm, two PSO combined together or nested to solve separate but related problems simultaneously. Here, there is an inner and an outer PSO that work together. The benefit is that it's easy to reach two goals by considering other goal when achieving other one. Pseudo code of PSO is discribed below:

```
For each particle
        Initialize particle
End
        Do
        For each particle
        Calculate fitness value
        If the fitness value is better than the best fitness
value (pBest) in history
        set current value as the new pBest
        End
Choose the particle with the best fitness value of all the
particles as the gBest
     For each particle
        Calculate particle velocity according equation
        Update particle position according equation
     End
While maximum iterations or minimum error criteria is not
attained
```

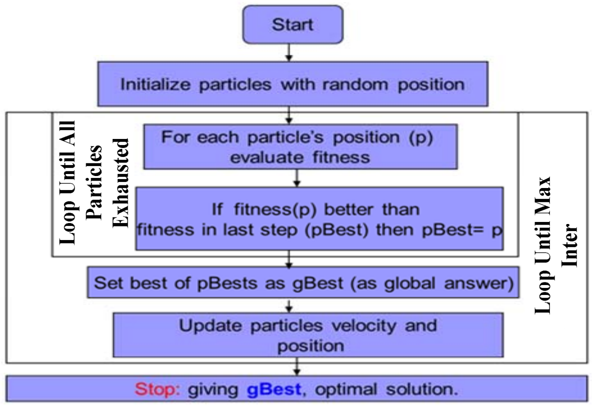


**Fig. 3.** PSO's chart of action.

The main method as it described can be illustrated as chart in "Fig. 3". In this chart and it's related pseudo, 'p' is the position or first generation of PSO as a priory answer, the best solution (fitness) particle has achieved so far is showed by "pBest", and the best value obtained by any particle as global best is modeled by "*gBest*". It's considered $P=X_k$ and updated position (answer) illustrated by $X_{k+1}$. [20]

Basic algorithm as proposed by 'Kennedy' and 'Eberhart' can present by this factors:

**Table 1.** Parameters definitions.

| | |
|---|---|
| $X_k^i$ | Particle position ( current particle or solution) |
| $V_k^i$ | velocity of agent *i* at iteration *k*. |
| $P_k^i$ | Best "remembered" individual particle position *i*. |
| $X_k^g$ | Best "remembered" swarm position |
| $C_1$,$C_2$ | Cognitive and social parameters (learning factors) |
| $r_1$,$r_2$ | Random numbers between 0 and 1 |

Usually, $C_1$ and $C_2$ considered as $C_1= C_2=2$ in PSO. Position of individual particles updated can described as equation (1):

$$X_{k+1}^{i}=X_{k+!}^{i}+V_{k+1}^{i} \tag{1}$$

The velocity can calculate as equation (2):

$$V_{k+1}^{i}=V_k^i+C_1r_1(P_k^i-X_k^i)+C_2r_2(P_k^g-X_k^i) \tag{2}$$

By considering equation (2) that have three parts consists on inertia ($V_k^i$), personal influence ($C_1r_1(P_k^i-X_k^i)$) and social influence ($C_2r_2(P_k^g-X_k^i)$), it can show as the concept in "Fig. 4" [21, 22, and 23].

The end term as it described in pseduo is while maximum iterations or minimum error criteria is not attained, which one arrive first.

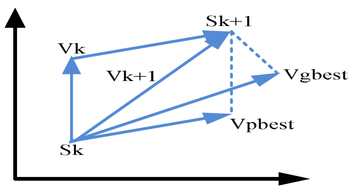


**Fig. 4.** Concept of modification of a searching point by PSO.

## 3. Problem Definition

As first step, the problem should model. The Economic Dispatch is a sub problem of the Unit Commitment, so it will define after defining UC.

As it is mentioned in UC, it should find out which kind of plants is better to use in a specific kind of plan. For this aim the following kind of plant is used as a priory choice. The plants that detailed in "Table 1" are six typical units:

**Table 2.** Unit's properties.

| Unit | $P_{i\,(min)}$ {MW} | $P_{i\,(max)}$ {MW} | $\alpha_i$ {\$} | $\beta_i$ {\$/MW} | $\gamma_i$ {\$/MW$^2$} |
|---|---|---|---|---|---|
| 1 | 100 | 500 | 240 | 7.0 | 0.0070 |
| 2 | 50 | 200 | 200 | 10.0 | 0.0095 |
| 3 | 80 | 300 | 220 | 8.5 | 0.0090 |
| 4 | 50 | 150 | 200 | 11.5 | 0.0090 |
| 5 | 50 | 200 | 220 | 10.5 | 0.0080 |
| 6 | 50 | 120 | 190 | 12.0 | 0.0075 |

Where $\alpha_i$, $\beta_i$ and $\gamma_i$ are the cost equation's coefficients for the *i*-th generator in the equation (3):

$$\min F_t = \sum_{i=1}^{m} F_i(P_i) = \sum_{i=1}^{m} \alpha_i + \beta_i P_i + \lambda_i P_i^2 \tag{3}$$

Where $P_{i\,(min)}$ and $P_{i\,(max)}$ are the constraints of plant *i* for producing power in Watt. The main Goal is optimization of equation $F_t$ as mentioned above.

After denoting this plant because it's important to be cautious about limits of each plant, it should define some prohibited zones that cannot be an answer, and our plants cannot produce this part of power. As it is showed in "Table 3" below, for example it is 80 to 90 and 110 to 120 for unit number 4.

**Table 3.** Unit's limits.

| Unit | $P_{i0}$ {MW} | $UR_i$ {MW/h} | $DR_i$ {MW/h} | Prohibited Zones {MW} | Cost {\$ Mil.} |
|---|---|---|---|---|---|
| 1 | 440 | 80 | 120 | [210-240], [350-380] | 50 |
| 2 | 170 | 50 | 90 | [90-110], [140-160] | 35 |
| 3 | 200 | 65 | 100 | [150-170], [210-240] | 40 |
| 4 | 150 | 50 | 90 | [80-90], [110-120] | 28 |
| 5 | 190 | 50 | 90 | [90-110], [140-150] | 32 |
| 6 | 110 | 50 | 90 | [75-85], [100-105] | 22 |

In the "Table 3" the $UR_i$ and $DR_i$ are Up-Ramp limit and Down-Ramp limit of generator *i,* and $P_{i0}$ the current output power of it, thus:

$$P_i - P_{i0} \le UR_i \tag{4}$$

$$P_{i0} - P_i \le DR_i \tag{5}$$

The cost is the other part that will consider in \$ million. Now for considering prohibited operating zones, it's important to define equation (6):

$$Pi \in \begin{cases} P_i^{\min} \le P_i \le P_{i,1}^{l} \\ P_{i,j-1}^{u} \le P_i \le P_{i,j-1}^{l}, j = 2,3,\ldots,nj; i = l,\ldots,m \\ P_{i,n}^{u} \le P_i \le P_i^{\max} \end{cases} \tag{6}$$

As PSO is a continues algorithm, it's needed to use PSO as discontinues so equation (7) will use:

$$0 \le X \le 1 \rightarrow k = \min\{\lfloor (M+1)X \rfloor, M\} \tag{7}$$

Where $\lfloor \quad \rfloor$ is sign for integer part of equation between it, and $X$ is produced coincidental by PSO. It's supposed in this paper that there is 6 units, So *k* belong to $\{0,1,2,\ldots,6\}$ , and the "Unit 0" means that there is no need any unit there, with all factors that are zero or not a number in real world, this is for that the model can use the algorithm for calculation by computer.

After finding answers for UC, PSO tried again for gaining ED producing answers in each iteration. The ED planning must run the optimal generation dispatch between the operating units to satisfy the system demand and practical operation constraints of generators that include the ramp rate limits and the prohibited operating zones as mentioned above.

For ED, the "Table 1" and "Table 2" used again of course without any need to cost column. It is obvious only the types are chosen that came from last step of UC's PSO. For example if the answers are {1, 2, 3, 5}; algorithm will use just this plants information for external PSO and other information are not need. The best solution in every step will save to compare with incoming answers.

## 4. Used algorithm structures

As it is mentioned PSO is used in this paper as nested PSO and contain two interrelate part of a whole as it is showed in "Fig. 5".

In "Fig. 5", the internal PSO is for optimization of UC and external PSO is used for optimization of ED, that this two PSO algorithm used together at same time and external PSO use internal PSO's results online and real time for its optimization and improvement of results.

## 5. Algorithm's main parts

After describing algorithm as a whole, the next step is recognizing how to apply it to especial problem that here is ED an UC. Our algorithm's main parts are most like other evolutionary algorithms such as genetic, ant colony (a sub group of PSO), etc. For more contact between parts of algorithm it is important to first define a priory population or exactly position as mentioned it by answer in final conclusion.

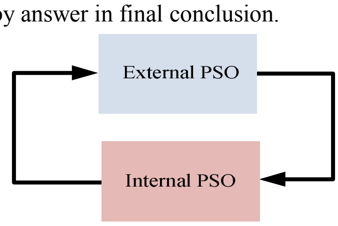


**Fig. 5.** The nested PSOs connection as a whole.

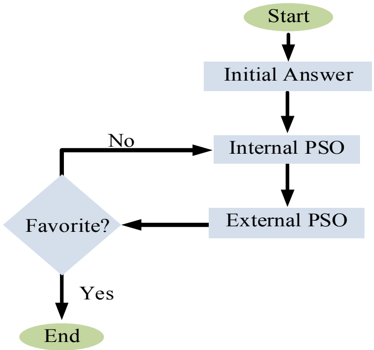


**Fig. 6.** Contact between internal and external PSOs.

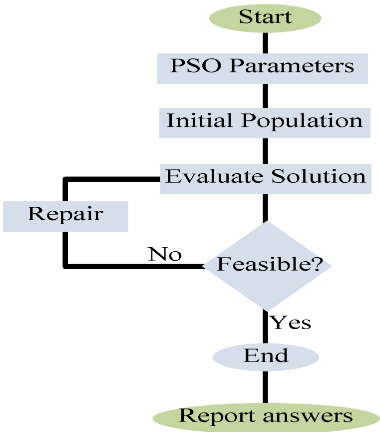


**Fig. 7.** Every separate PSO's structure.

The contact between internal and external PSO by applying in algorithm, is illustrated in "Fig. 6". Each PSO is however with contact with other one but it's a separate PSO algorithm that can describe with "Fig. 7":

The parameters of the used PSO for both internal and external PSO are:

- Maximum number of iterations for external PSO=10;
- Maximum number of iterations for internal PSO=100;
- Population size (Swarm Size) =10;
- Inertia weight =1;
- Inertia weight damping Ratio =0.99;
- And velocity limits are:
- $Vel_{Max}$= 0.1* ($Var_{Max}$-$Var_{Min}$);
- $Vel_{Min}$= -$Vel_{Max}$;

The end term as it referred in last part, is while maximum iterations or minimum error criteria is not attained , that it is choosed the maximum iteration which is 10, but it must be considered that every iteration of this 10 iteration fo external PSO have 100 iteration inside it for internal PSO, so there is 1000 iterations at last step.

## 6. Equations

The typical system covers six thermal units, 26-buses, and 46 transmission lines [24]. The load demand is equal 1263 MW. The characteristics of the six thermal units are the same in Tables 1 and 2 in last part. As it's mentioned before, the goal is to optimize equation (8) by minimizing it:

$$\min F_t = \sum_{i=1}^{m} F_i(P_i) = \sum_{i=1}^{m} \alpha_i + \beta_i P_i + \lambda_i P_i^2 \tag{8}$$

But, the main limit is considering in equation (9):

$$\sum_{i=1}^{m} P_i = P_D + P_L \tag{9}$$

Where, $P_i$ is the power produced by plant *i-th,* $P_D$ is demand power, *m* is number of units and $P_L$ is power loss.

By considering conditions in last parts it is easy to formalize the limits as equation (10):

$$\max(P_i^{\min}, P_i^0 - DR_i) \le P_i \le \min(P_i^{\max}, P_i^0 + UR_i) \tag{10}$$

Where, $P_i$ and other factors mentioned in last section. The $UR_i$ and $DR_i$ are Up-Ramp limit and Down-Ramp limit of generator *i,* and $P_{i0}$ is the current output power of it and $P_i$ is the power produced by plant *i.*

As it's mentioned $P_L$ is power loss, $P_L$ is equal to:

$$P_L = \sum_{i=1}^{m}\sum_{j=1}^{m} P_i B_{ij} P_j + \sum_{i=1}^{m} B_{0i} P_i + B_{00} \tag{11}$$

Where, *m* is the number of generators committed to the operating system and *Pi* is the power output of the *i-th* generator. In normal operation of the system, the loss coefficients can describe with matrix *B* as follows:

$$B_{ij} = 10^{-3} * \begin{bmatrix} 0.0425 & 0.0300 & 0.0175 & -0.0025 & -0.0125 & -0.0050 \\ 0.0300 & 0.0350 & 0.0225 & 0.0025 & -0.0150 & -0.0025 \\ 0.0175 & 0.0225 & 0.0775 & 0.0000 & -0.0250 & -0.0150 \\ -0.0025 & 0.0025 & 0.0000 & 0.0600 & -0.0150 & -0.0200 \\ -0.0125 & -0.0150 & -0.0250 & -0.0150 & -0.3225 & -0.0050 \\ -0.0050 & -0.0025 & -0.0150 & -0.0050 & -0.0050 & 0.3750 \end{bmatrix}$$

$$B_{0i} = 10^{-3} * [-0.3908 \quad -0.1279 \quad 0.7047 \quad 0.0591 \quad 0.2161 \quad -0.6635]$$
$$B_{00} = 0.056$$

$B_O$ and $B_{OO}$ are constant, but *Bij* is a matrix which depends on network buses.

## 7. Programming

In first step, it is necessary to model UC and ED, which one the models should consist of "cost" and "Zero Plant" as it's described in previous section.

For modeling ED and UC, data should inter from "Table 1" and "Table 2". The prohibited zones and UR and DR are the most important part must recognized by algorithm. Therefore, it is important to inter this part as main part of limits to model part.

As second step the *B* matrix and plants properties is needed for algorithm. It should considered that the cost of each plant is separate from cost of power for that plants, and one is used to UC and internal PSO and other one for ED and external PSO.

After modeling, there is two parse solutions for external and internal to use in cost function for each one separately. For example, the parse solution for UC as it's described in previous section is modeled as following codes:

```
Function k=ParseSolutionExt(x,M)
    k=min (floor ((M+1)*X), M);
End
```

Where $X$ is is produced by PSO coincidentally and $M$ is the the number of plants in model, that here is equal to 6.

The main idea is using a function that can handle discontinuous model for PSO. Therefore, the main cost function for this part will use mentioned parse solution. For other PSO in ED the parse solution which must used is as follows:

```
Pmin=model. Plants. Pmin;
  Pmax=model.Plants.Pmax;
  P=Pmin+(Pmax-Pmin).*X;
```

By considering prohibited zones, for this aid the used pseudo is:

```
for i=1:nPlant
    for j=1:numel(PZ{i})
      if P(i)>PZ{i}{j}(1) && P(i)<PZ{i}{j}(2)
        % Correction
        if P(i)<(PZ{i}{j}(1)+PZ{i}{j}(2))/2
          P(i)=PZ{i}{j}(1);
        else
          P(i)=PZ{i}{j}(2);
        end
      end
    end
  end
```

The pseudo can illustrate as "Fig. 8". In this figure, the method which used is a modle to run away from prohibited zone as it's coming. Where "nPlant" is the number of founded plants for special plan, and "numel(PZ{i})" is depend on the number of prohibited zones in a special plant. When it's founded that the *P(i)* as answer or position is in prohibited zone , if it's near to second part of porhibited zone which cut down two equal part, *P(i)* will be equal to *B*, otherwise it will be *A*.

After using this method, there is two costs function for each seprate PSO that use together for solve problem. Cost function in algorithm is just like the equation that mentioned in equation (8), but here the cost consist of all the plants cost together by put each unit's $P_i$ in equation (12). The last answers must minimize this function.

$$F_t = \sum_{i=1}^{m} F_i(P_i) = \sum_{i=1}^{m} \alpha_i + \beta_i P_i + \lambda_i P_i^2 \qquad (12)$$

## 8. Results

After running program and 10 iteration for main PSO and 100 for inner PSO, the best solution after 1000 iterations, showed in "Fig. 9".

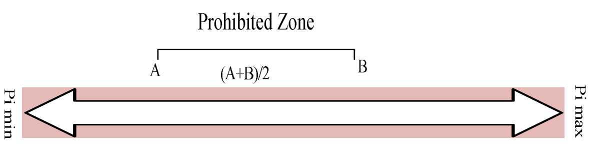


**Fig. 8.** Prohibited zones applying on algorithm.

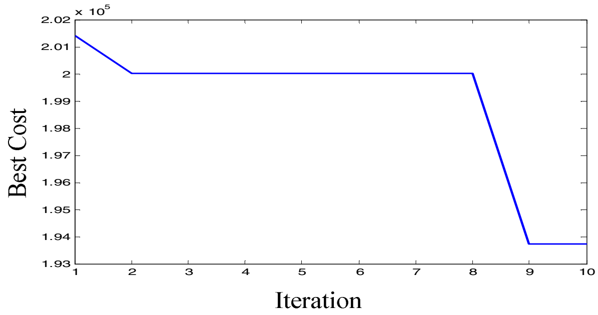


**Fig. 9.** External iteration for optimizing cost.

The important notice is that the "Fig. 9" shows just the best solutions answer and the total decrease in cost will appear in "Fig. 12". Here, it should consider that the 10 iteration as external iteration are only visible iterations but every iteration consist on 100 other internal iteration.

In addition, other results are:

- In iteration 10; best cost = 193721.2063$ Million.

The best cost after 10 iteraion, optimazed to 193721.20 $ Million and the type of plant are:

- k: [0,4,4,1,6,1]
- F: [0,28000,28000,50000,22000,50000]
- FTotal: 178000
- z: 193720

Where, $k$ is type of best plants for use in this plan, that are two of 4, two of 1 and a 6 or brifly {1,1,4,4,6} which illustrated in "Fig. 10 ".

Here, $F$ shows the cost of each plant that the whole cost will be 178000 $ Million, and the rest of it is ED's cost. So ED's cost will be 193721.20-178000=15721.2 $ Million.

The ED costs and BestSol will be:

- P: [0,149.3663,150,499.6838,120,418.9521]
- PTotal: 1338
- CTotoal: 15721
- PL: 74.9602
- PowerBalanceViolation: 0
- z: 15721.20
- TotalCost: 193720

Where, $P$ is amount of produced power by each plant, "Ptotal" is total $P$ that is needed, $PL$ is power loss, and $z$ is amount of ED's cost that added to UC's cost and TotalCost is 193721.20 $ Million.

PowerBalanceViolation is an consept to show how our answers are near to favarate ones.

$$P_{Total} \geq P_D + P_L \rightarrow Violation = 0 \qquad (13)$$

$$P_{Total} < P_D + P_L \rightarrow Violation > 0 \qquad (14)$$

By considering equation (13), it should always check that the function get the point for equation (9) or not.

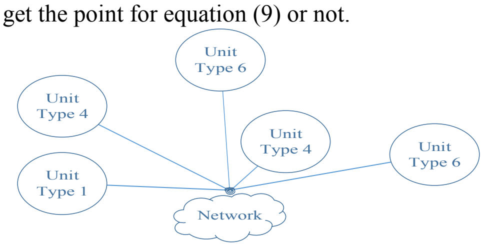


**Fig. 10.** Final results for UC.

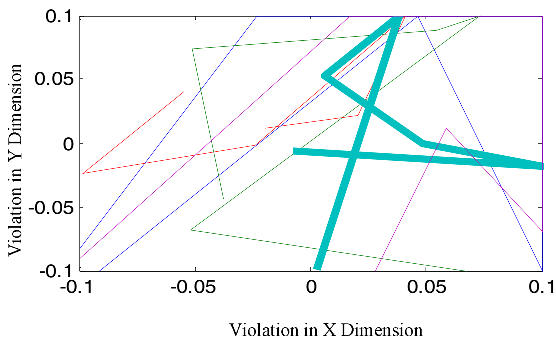


**Fig. 11.** External iteration for optimizing cost.

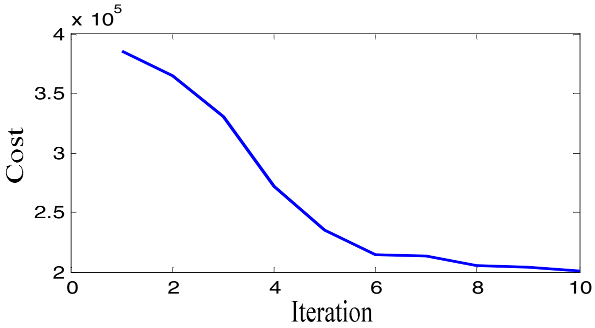


**Fig. 12.** Iteration's effect on cost.

For having this in programming, it's necessary to use equation (15) in below:

$$Violation = \max(1 - \frac{P_{Total} - P_L}{P_D}, 0) \quad (15)$$

The aim is to reduce violation to zero, or near to it, that in this paper it neared to zero. "Fig. 11" shows the violation of 10 iterations that it's tried to change it to zero in best solution. The best solution's line is thicker than other's line.

The "Fig.12" shows the cost reduction as all of the iterations where each iteration shows 100 iterations in it.

## 8. Conclusion

After finding all the results that mentioned in last parts, now it can say that by using two PSO as nested and together it is possible to get the answers more quickly and more feasible. In this method, which it's used both of ED and UC in a nested algorithm, when it is decided to find an answer to one, the other one examine it. In other aspect, it can reduce the cost, as it showed, to get the best answer at last by most efficiency. Having all kind of cost such as cost for ED and UC separately and even cost for each plant's power produce and construction is other subject that was important as it is showed.

## 7. References

[1] Bakirtzis, Anastasios, Vassilios Petridis, and Spyros Kazarlis. "Genetic algorithm solution to the economic dispatch problem." *IEE Proceedings-Generation, Transmission and Distribution*, 141.4 (1994): 377-382.

[2] Lee, Fred N., and Arthur M. Breipohl. "Reserve constrained economic dispatch with prohibited operating zones." *IEEE Transactions on Power Systems,* 8.1 (1993): 246-254.

[3] Kiani, M., and Seid H. Pourtakdoust. "State estimation of nonlinear dynamic systems using weighted variance-based adaptive particle swarm optimization." *Applied Soft Computing* 34 (2015): 1-17.

[4] Elattar, Ehab E. "A hybrid genetic algorithm and bacterial foraging approach for dynamic economic dispatch problem." *International Journal of Electrical Power & Energy Systems* 69 (2015): 18-26.

[5] Naka, Shigenori, et al. "Practical distribution state estimation using hybrid particle swarm optimization." *Power Engineering Society Winter Meeting, IEEE*. Vol. 2. IEEE, 2001.

[6] Jadoun, Vinay Kumar, et al. "Multi-area economic dispatch with reserve sharing using dynamically controlled particle swarm optimization." *International Journal of Electrical Power & Energy Systems* 73 (2015): 743-756.

[7] Singhal, Prateek K., et al. "Solution of unit commitment problem using enhanced genetic algorithm." *Power Systems Conference (NPSC), Eighteenth National*. IEEE, 2014.

[8] Banerjee, Sumit, Deblina Maity, and Chandan Kumar Chanda. "Teaching learning based optimization for economic load dispatch problem considering valve point loading effect." International Journal of Electrical Power & Energy Systems 73 (2015): 456-464.

[9] Alsaffar, Mohammad A., and Mohamed A. El-Sayed. "Emission constrained unit commitment of Kuwait power generation system using genetic algorithm." Journal of Engineering Research 2.2 (2014).

[10] Gaing, Zwe-Lee. "Particle swarm optimization to solving the economic dispatch considering the generator constraints." *IEEE Transactions on Power Systems,* 18.3 (2003): 1187-1195.

[11] Park, Jong-Bae, et al. "A particle swarm optimization for economic dispatch with nonsmooth cost functions." *IEEE Transactions on Power Systems,* 20.1 (2005): 34-42.

[12] Xu, Bin, et al. "Comparison between dynamic programming and genetic algorithm for hydro unit economic load dispatch." Water Science and Engineering 7.4 (2014): 420-432.

[13] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation,* 6.2 (2002): 182-197.

[14] Kennedy, James. "Particle swarm optimization." *Encyclopedia of Machine Learning*, Springer US, 2010. 760-766.

[15] S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, "Algorithms", p 173.

[16] Fogel, David B. "Evolutionary computation: toward a new philosophy of machine intelligence" ,Vol. 1. John Wiley & Sons, 2006.

[17] Eberhart, Russell C., and Yuhui Shi. "Comparison between genetic algorithms and particle swarm optimization." *Evolutionary Programming VII*, Springer Berlin Heidelberg, 1998.

[18] Kennedy, James. "Particle swarm optimization." *Encyclopedia of Machine Learning*. Springer US, 2010. 760-766.

[19] Shi, Yuhui, and Russell C. Eberhart. "Empirical study of particle swarm optimization." *Proceedings of the 1999 Congress on Evolutionary Computation,*Vol. 3. IEEE, 1999.

[20] Lisa L. Smith, Ganesh K. Venayagamoorthy, Phillip G. Holloway ,“Obstacle Avoidance in Collective Robotic Search Using Particle Swarm Optimization”, University of Missouri-Rolla, 2004.

[21] S. Doctor, G. K. Venayagamoorthy and V. Gudise, Optimal PSO for Collective Robotic Search Applications, *IEEE Congress on Evolutionary Computations*, June 19-23, 2004, Portland OR, USA, pp. 1390 –1395.

[22] Kurt Derr, Milos Manic, "Multi-Robot, Multi-Target Particle Swarm Optimization Search in Noisy Wireless Environments", , 2009 IEEE.

[23] Mahamed G. Omran and Andries P. Engelbrecht, "A Color Image Quantization Algorithm Based on Particle Swarm Optimization", Informatica 29 (2005) 261–269.

[24] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, “A particle swarm optimization for reactive power and voltage control considering voltage security assessment*,” IEEE Trans. Power Syst.,* vol.15, pp. 1232–1239, Nov. 2000.