

OPTIMISASI PELETAKAN DAN *SIZING DISTRIBUTED GENERATION (DG)* MENGGUNAKAN *TWO LAYER PARTICLE SWARM OPTIMIZATION (TLPSO)*

Efrita Arfah Z

Jurusan Teknik Elektro, FTI-ITATS Surabaya
Jl. Arief Rahman Hakim 100 Tlp 031 5945043
Email : zuliary_efri@yahoo.co.id

ABSTRAK

Two Layer Particle Swarm Optimization (TLPSO) untuk optimisasi peletakan dan sizing DG. Distributed Generation (DG) merupakan pembangkit tersebar dengan tujuan menekan losses daya riil pada sistem. Data yang digunakan pada sistem ini adalah 30 bus IEEE. TLPSO yang merupakan perkembangan dari PSO terbukti lebih unggul di dibandingkan dengan PSO. TLPSO mampu menurunkan losses sebesar 57.3% sedangkan PSO sebesar 51.11%. dibandingkan dengan sebelum peletakan DG. Losses daya riil setelah peletakan DG dengan optimisasi TLPSO turun sebesar 12.6% dibandingkan PSO.

Kata kunci: Distributed generation (DG), peletakan dan sizing optimal, Two Layer Particle Swarm Optimization (TLPSO)

ABSTRACT

Two Layer Particle Swarm Optimization (TLPSO) for laying and sizing optimization DG. Distributed Generation (DG) is the power spread with the aim of suppressing the real power losses in the system. The data used in this system is the IEEE 30 bus. TLPSO which is the development of PSO is shown to be superior in comparison with PSO. TLPSO able to reduce losses by 57.3% and amounted to 51.11% of PSO. Compared with before laying DG. Real power losses after laying the optimization TLPSO DG decreased by 12.6% compared to PSO.

Keywords: Distributed generation (DG), placement and optimal sizing, Layer Two Particle Swarm Optimization (TLPSO)

PENDAHULUAN

Pembangkit kecil tersebar atau biasa disebut *Distributed Generation (DG)*, seiring berkembangnya ilmu pengetahuan dan teknologi dewasa ini menjadi salah satu penyelesaian dalam sistem untuk menekan *losses*. Alasan ekonomi menjadi salah satu alasan peletakan DG karena *losses* sistem dapat tereduksi, otomatis biaya dapat berkurang. Selain itu DG cukup efisien, handal dan cukup sederhana. (Mohammadi, 2011)

Pengertian DG itu sendiri yaitu pemasangan generator kecil didekat pusat beban pada seluruh jaringan distribusi untuk memenuhi kebutuhan beban lokal dan pemenuhan preventif untuk melonjaknya permintaan daya. *Losses* daya pada sistem distribusi yang tinggi disebabkan karena tegangan rendah dan oleh karena itu arus menjadi besar (Keane, 2005). Peletakan DG dengan ukuran dan bus yang tepat dapat mengoptimalkan kerja sistem. Berbagai metode optimisasi telah dikembangkan dalam penelitian peletakan dan ukuran yang optimal DG. *Two Layer Particle Swarm Optimization (TLPSO)* sebagai pengembangan dari PSO diusulkan dalam penelitian ini untuk memecahkan optimisasi peletakan dan jumlah optimal tiga tipe DG dengan menggunakan data sistem 30 bus IEEE.

TLPSO untuk meningkatkan keragaman dari partikel PSO biasa, kinerja TLPSO n kali lebih efisien dibanding dengan PSO biasa, dimana n adalah bilangan real sebagai representasi running program pengujian untuk mencari optimalitas. Desain dari TLPSO yaitu strukturnya terdiri dari dua layer (layer atas dan layer bawah) maka swarm M dari particle dan satu swarm dari partikel di bangkitkan masing-masing pada layer atas dan layer bawah. Setiap posisi global terbaik dari setiap swarm layer bawah akan di set pada posisi partikel dari swarm layer atas. Oleh karena itu, posisi global terbaik (Gbest) pada setiap layer atas secara langsung

dipengaruhi oleh setiap partikel swarm layer bawah sehingga keragaman partikel meningkat untuk menghindari terjebak local optimum. Disamping itu operasi mutasi ditambahkan pada setiap partikel masing-masing swarm pada layer bawah berdasarkan peluang mutasi, maka partikel baru akan muncul dari lokal optimum untuk mendapatkan global optimum (Chen, 2011).

Reduksi *losses* daya riil dalam sistem distribusi dibutuhkan dalam operasi sistem tenaga listrik yang efisien. *Losses* sistem tenaga dapat dirumuskan seperti pada persamaan (Jain, 2010).

$$P_L = \sum_{i=1}^n \sum_{j=1}^n A_{ij} (P_i P_j + Q_i Q_j) + B_{ij} (Q_i P_j - P_i Q_j)$$

Dimana

$$A_{ij} = \frac{R_{ij} \cos(\delta_i - \delta_j)}{V_i V_j}$$

$$B_{ij} = \frac{R_{ij} \sin(\delta_i - \delta_j)}{V_i V_j}$$

Dimana P_i dan Q_i injeksi bersih daya riil dan reaktif pada bus i ; R_{ij} resistansi saluran antara bus i dan j ; V_i dan δ_i tegangan dan sudut pada bus i .

Tujuan dari teknik peletakan adalah untuk meminimalkan total *losses* daya riil. Fungsi obyektif dapat ditulis sebagai:

$$P_L = \sum_{k=1}^{N_{SC}} loss_k$$

Dengan beberapa batasan yang digunakan yaitu:

1. Batasan keseimbangan daya, dapat dirumuskan sebagai:

$$\sum_{i=1}^N P_{DG_i} = \sum_{i=1}^N P_{D_i} + P_L$$

2. Batasan tegangan :

$$|V_i|^{min} \leq |V_i| \leq |V_i|^{max}$$

3. Batasan arus

$$|I_{ij}| \leq |I_{ij}|^{max}$$

Dimana : $Loss_k$ merupakan pada bagian k , N_{SC} jumlah total dari bagian, P_L *losses* daya riil sistem, P_{DG_i} daya riil pembangkitan DG pada bus i dan P_{D_i} permintaan beban pada bus i .

METODE PENELITIAN

Particle Swarm Optimization (PSO)

PSO merupakan salah satu algoritma optimisasi yang dirumuskan pertama kali oleh Kennedy dan Eberhart pada tahun 1995 yang terinspirasi kebiasaan dari sekumpulan burung atau ikan dalam populasi untuk mencari makan. Individu dalam populasi tersebut dinamakan partikel (*particle*), tiap-tiap partikel mempunyai posisi (*position*) dan *velocity* yang berubah berdasarkan pengalaman dari partikel terbaik (P_{best}). Partikel terbaik adalah partikel yang paling dekat dengan sumber makanan (Mohammadi, 2011).

Perubahan posisi dipengaruhi oleh perubahan *velocity* dari masing-masing partikel, sesuai dengan persamaan (Oumar, 2009). Velocity dihitung berdasarkan P_{best} dan G_{best} yang dirumuskan pada persamaan (Kennedy, 1995).

$$S_{id}^{k+1} = S_{id}^k + v_{id}^{k+1}, i = 1, 2, \dots, n$$

$$d = 1, 2, \dots, m$$

$$v_{id}^{k+1} = (\omega) v_{id}^k + c_1 rand \times (P_{best_{id}} - S_{id}^k) + c_2 rand \times (G_{best_d} - S_{id}^k) v_{id}^{k+1}$$

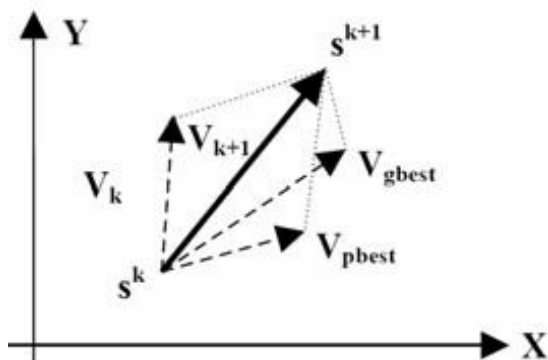
Dimana dimana S^k titik pencarian arus, S^{k+1} modifikasi titik pencarian, v^k velocity arus, v^{k+1} modifikasi velocity dari agen i , $v_{P_{best}}$ velocity yang berdasarkan P_{best} , $v_{G_{best}}$ velocity yang berdasarkan G_{best} , n jumlah anggota dalam grup, m jumlah anggota dalam partikel, P_{best} adalah P_{best} dari agen i , G_{best} merupakan G_{best} dari grup, ω fungsi weight untuk velocity dari agen i , c_1 koefisien weight.

Berikut pers fungsi koefisien dari weight:

$$\frac{\omega_{max} - i \omega_{min}}{k_{max}} \cdot k$$

$$\omega_i = \omega_{max} - i$$

dimana $\omega_{min}, \omega_{max}$ minimum dan maksimum dari weight...., k, k_{max} arus dan maksimum iterasi.



Gambar 1 konsep pencarian PSO

Berikut ini prosedur algoritma PSO untuk optimisasi peletakan dan *sizing* dari DG dengan tujuan meminimalkan losses:

1. Input data bus dan line bus, batas tegangan dan parameter PSO
2. Hitung *losses* sebelum ada peletakan DG
3. Bangkitkan secara random posisi dan velocity dari partikel.
4. Untuk masing- masing partikel, jika tegangan memenuhi syarat hitung *losses* daya, jika tidak memenuhi partikel tersebut akan di abaikan.
5. Untuk masing-masing partikel bandingkan nilai fitness dengan nilai fitness P_{best} , jika nilai fitness tersebut lebih rendah dari nilai P_{best} set nilai tersebut sebagai P_{best} yang baru
6. Bandingkan P_{best} dengan G_{best} , jika P_{best} lebih kecil G_{best} jadikan P_{best} sebagai G_{best} yang baru; jika tidak maka menggunakan G_{best} yang lama
7. Update velocity dan posisi masing-masing partikel menggunakan persamaan 6 dan 7
8. Cek iterasi, jika iterasi mencapai maksimum maka berhenti atau jikan belum maka kembali ke point 4.
9. Cetak hasil.

Two Layer Particle Swarm Optimization (TLPSO)

Two Layer Particle Swarm Optimization (TLPSO) mempunyai dua layer yaitu layer atas dan layer bawah. Sejumlah M *swarm particle* dibangkitkan pada layer bawah. Masing-masing global *best* posisi pada masing-masing *swarm* pada layer bawah ditetapkan sebagai posisi partikel di layer atas, sehingga global *best* posisi pada layer atas secara tidak langsung dipengaruhi partikel pada masing masing *swarm* layer bawah. Selanjutnya operator mutasi ditambahkan pada partikel dimasing-masing *swarm* dilayer bawah, penambahan operator ini ditujukan agar terjadi varian baru pada partikel sehingga tidak mudah terjebak pada kondisi lokal optimum. (Chen, 2011)

Berikut ini prosedur algoritma TLPSO:

1. Tentukan layer atas dan bawah dan parameter TLPSO.
2. Bangkitkan secara random inialisasi posisi dan velocity dari partikel. Hitung fitness P_{best} dan G_{best} pada masing-masing layer

- a. Bangkitkan secara random posisi awal dari partikel, $(x^{j1} x^{j2}, \dots x^{jN})$,

$j \in \{1, 2, \dots, M\}$, pada *swarm* layer bawah. Posisi dari partikel ditunjukkan sebagai:

$$x^{jk} = [x_1^{jk}, x_2^{jk} \dots x_D^{jk}],$$

$$j \in \{1, 2, \dots, M\}$$

$$k \in \{1, 2, \dots, N\}$$

Dimana $x_i^{jk}, i \in \{1, 2, \dots, D\}$ berikut hasil yang di bangkitkan secara random:

$$x_i^{jk} = x_i^{min} + (x_i^{max} - x_i^{min}) \cdot$$

$rand()$

Rentang dari parameter x_i^{jk} didefinisikan sebagai $[x_i^{min}, x_i^{max}]$

b. Pembangkitan secara random velocity vector awal dari partikel, $\{v^{j1}, v^{j2}, \dots, v^{jN}\}, j \in \{1, 2, \dots, M\}$, pada swarm layer bawah. Vektor velocity ditunjukkan pada pers.

(10).

$$v^{j1} = [v_1^{jk}, v_2^{jk}, \dots, v_D^{jk}], j \in \{1, 2, \dots, M\}$$

$$k \in \{1, 2, \dots, N\}$$

Dimana $v_i^{jk}, i \in \{1, 2, \dots, D\}$ yang dibangkitkan secara random dengan:

$$v_i^{jk} = \frac{x_i^{max} - x_i^{min}}{20} \cdot rand()$$

c. Tentukan posisi terbaik individu awal dari $x^{jk}, x_{Pbest}^{jk}, j \in \{1, 2, \dots, M\}, k \in \{1, 2, \dots, N\}$, pada swarm layer bawah. Individu dengan posisi terbaik ditentukan dengan:

$$x_{Pbest}^{jk} = x^{jk}, j \in \{1, 2, \dots, M\},$$

$$k \in \{1, 2, \dots, N\}$$

d. Tentukan posisi terbaik secara global dari swarm jth, $y^j, j \in \{1, 2, \dots, M\}$ pada layer bawah.

Hitung nilai fitness dari $x_{Pbest}^{jk}, f(x_{Pbest}^{jk}), k \in \{1, 2, \dots, N\}$ dan kemudian tentukan indeks K dari

x_{Pbest}^{jk} dengan nilai fitness terbaik oleh

$$f(x_{Pbest}^{jk})$$

$$\max_{k=1}^N \text{Kemudian atur}$$

$$K = \arg \min$$

$$y^j = x^{jk}, j \in \{1, 2, \dots, M\}$$

e. Tentukan posisi awal dari partikel dalam swarm layer atas. Posisi global terbaik $\{y^1, y^2, \dots, y^M\}$ di atur untuk posisi partikel dalam swarm layer atas.

f. Atur vector velocity awal dari $w^j, w^j, j \in \{1, 2, \dots, M\}$. Vector velocity ditunjukkan pada pers (12).

$$w^j = [w_1^j, w_1^j, \dots, w_D^j],$$

$$j \in \{1, 2, \dots, M\}$$

Dimana, $w_i^j, i \in \{1, 2, \dots, D\}$ yang dibangkitkan secara random dengan rumus

$$w_i^j = \frac{x_i^{max} - x_i^{min}}{20} \cdot rand()$$

g. Tentukan individu awal dengan posisi terbaik dari partikel $\mathbf{p}_{Pbest}^j, j \in \{1, 2, \dots, M\}$ pada swarm layer atas. Individu dengan posisi terbaik ditentukan dengan:

$$\mathbf{p}_{Pbest}^j = \mathbf{p}^j, j \in \{1, 2, \dots, M\}$$

h. Tentukan posisi global terbaik dari swarm, \mathbf{p}_{Gbest} pada layer atas. Hitung nilai fitness dari $\mathbf{p}_{Pbest}^j, f(\mathbf{p}_{Pbest}^j) j \in \{1, 2, \dots, M\}$ dan selanjutnya cari indeks J dari \mathbf{p}_{Pbest}^j dengan menggunakan nilai fitness terbaik dari

$$J = \arg \max_{j=1} f(\mathbf{p}_{Pbest}^j).$$

atur $\mathbf{p}_{Gbest} = \mathbf{p}_{Pbest}^J$

3. Tentukan jumlah iterasi

4. Update vector velocity $\mathbf{v}^{jk}, j \in \{1, 2, \dots, M\}, k \in \{1, 2, \dots, M\}$ dan vector posisi \mathbf{x}^{jk} pada swarm layer bawah

$$\begin{aligned} \mathbf{v}^{jk} &= \mathbf{v}^{jk} + c_1 \cdot \text{rand}() \cdot (\mathbf{x}_{Pbest}^{jk} - \mathbf{x}^{jk}) \\ &+ \mathbf{v}^{jk} c_2 \cdot \text{rand}() \cdot (\mathbf{p}^j - \mathbf{x}^{jk}) \\ \mathbf{x}^{jk} &= \mathbf{x}^{jk} + \mathbf{v}^{jk} \end{aligned}$$

5. Mutasi patikel secara random

$\mathbf{x}^{jk} = [x_1^{jk} \ x_2^{jk} \ \dots \ x_D^{jk}], j \in \{1, 2, \dots, M\}, k \in \{1, 2, \dots, M\}$. Jika $\text{rand}() < P_m$ maka $\mathbf{x}_{i^k}^{jk} = i$

$$\mathbf{x}_{i^k}^{jk} = \mathbf{x}_{i^k}^{jk} - (\mathbf{x}_{i^k}^{jk} - \mathbf{x}_{i^k}^{min}), \text{ dimana } i^k = \text{rand}() \cdot i$$

$$\text{rand}() + 0.5 \cdot i$$

6. Update individu dengan posisi terbaik $\mathbf{x}_{Pbest}^{jk}, j \in \{1, 2, \dots, M\}, k \in \{1, 2, \dots, M\}$

posisi terbaik secara global $\mathbf{p}^j, j = 1, 2, \dots, M$.

Jika $f(\mathbf{x}^{jk}) > f(\mathbf{p}_{Pbest}^j), j \in \{1, 2, \dots, M\}, k \in \{1, 2, \dots, M\}$

maka $\mathbf{x}_{Pbest}^{jk} = \mathbf{x}^{jk}$

Jika $f(\mathbf{x}^{jk}) > f(\mathbf{p}^j), j \in \{1, 2, \dots, M\}, k \in \{1, 2, \dots, M\}$

maka $\mathbf{p}^j = \mathbf{x}_{Pbest}^{jk}$

7. update vector velocity $\mathbf{w}^j, j = 1, 2, \dots, M$ dan posisi $\mathbf{p}^j, j = 1, 2, \dots, M$ pada swarm layer atas.

$$\mathbf{w}^j = \mathbf{w}^j + c_1 \cdot \text{rand}() \cdot (\mathbf{p}_{Pbest}^j - \mathbf{p}^j)$$

$$+c_2 \cdot rand() \cdot (p_{Gbest}^j - p^j)$$

$$p^j = p^j + w^j$$

8. update posisi terbaik dari individu $p_{Pbest}^j, j=1,2,\dots, M$

Jika $f(p^j) > f(p_{Pbest}^j), j=1,2,\dots, M$ maka $p_{Pbest}^j = p^j$

Jika $f(p_{Pbest}^j) > f(p_{Gbest}^j) j=1,2,\dots, M$ Maka $p_{Gbest}^j = p_{Pbest}^j$

9. Menurunkan v^{jk} dan w^j dengan konstanta $r \in \forall 0.1 \vee \zeta$

$$v^{jk} = v^{jk} \cdot r.$$

$$w^j = w^j \cdot r.$$

10. Menurunkan r dengan konstanta $w \in |0.1|$, jika $r > 0.1$ maka atur $r = r$, atau sebaliknya $r = 0.1$

11. Iterasi. Iterasi+1, jika iterasi ζG maka kembali ke tahap 12; atau sebaliknya jika iterasi ζG kembali ke tahap 4

12. Fitness terbaik

HASIL DAN PEMBAHASAN

Metode yang digunakan dalam penelitian peletakan dan *sizing* DG diterapkan pada \square elate IEEE 30 bus yang merupakan \square elate distribusi dengan konfigurasi dan *sizing* yang bervariasi. Sistem 30 bus mempunyai total beban 283.4 Mw dan 126.2 Mvar, dan total daya riil 283.4 MW dan daya reaktif sebesar 0.3644Mvar.

Perhitungan optimisasi didapatkan peletakan dan *sizing* DG pada bus yang bervariasi dan perbaikan aliran daya yang terbaik dengan menggunakan algoritma TLPSO sehingga *losses* daya minimum.

Parameter untuk algoritma TLPSO dan PSO ditunjukkan pada \square elay 1. Hasil simulasi didapatkan grafik konvergensi untuk TLPSO dan PSO dengan waktu komputasi 57.497201detik dan 56.256369 detik. Perbedaan waktu komputasi karena PSO listing program lebih pendek, hal ini karena TLPSO memiliki proses yang lebih panjang dibandingkan dengan PSO biasa yang hanya memiliki satu layer.

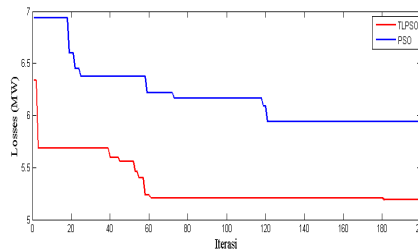
Tabel 1. Parameter TLPSO dan PSO

	TLPSO	PSO
Iterasi max	200	200
C ₁	0.7	0.7
C ₂	0.9	0.9
W	0.4	0.4


W_{max}	3	3
W_{min}	1	1
Swarm		20
Swarm bawah	10	
Swarm atas	5	

Sumber: data diolah

Grafik tersebut menunjukkan optimisasi DG yang ditunjukkan total *losses* daya aktif dari iterasi satusampai iterasi maximum yaitu 200. Penentuan iterasi maksimum dan parameter lain TLPSO di dapatkan dari proses trial and error dengan nilai acuan dari referensi penelitian yang pernah ada. Algoritma TLPSO dari iterasi pertama sampai terakhir mendapatkan nilai *losses* jauh lebih kecil dibandingkan dengan PSO, Hal ini disebabkan TLPSO swarm pada layer atas terdiri dari partikel-partikel terbaik yang ada pada layer bawah, dengan kata lain layer atas adalah sekumpulan Pbest pada masing-masing kelompok \square elayer bawah. Gbest pada TLPSO didapatkan dari lima kelompok partikel pada layer bawah yang merupakan representasi dari jumlah running PSO biasa sebanyak lima kali. Gbest didapatkan terbaik dari kelima tersebut. TLPSO lebih lambat konvergen yaitu pada iterasi ke 181 dibandingkan PSO yaitu pada 120. TLPSO terjadi local optimum pada iterasi 60 sampai 180, pada iterasi 180 terjadi keragaman partikel sehingga operator mutasi bekerja dan pada iterasi 181 terjadi partikel varian baru sehingga *losses* dapat turun lagi dan kemudian konvergen.



Gambar 1. Grafik konvergensi algoritma TLPSO dan PSO

(Sumber : Hasil simulasi )

Hasil simulasi dengan perbedaan variasi tipe DG dapat dilihat pada tabel 2. *Losses* sebelum di tambahkan DG sebesar 12.1496 MW, Optimisasi menggunakan PSO *losses* turun menjadi 5.938739 MW, optimisasi dengan TLPSO *losses* turun lagi menjadi 5.188660 MW. Jumlah daya aktif DG optimisasi antara TLPSO dan PSO sama yaitu 179 MW yang membedakan adalah jumlah peletakan bus dan Mvar yang di injeksikan. Jumlah bus pada TLPSO yang ditempatkan DG sebanyak delapan sedangkan PSO tujuh bus. semakin banyak bus yang di tempati DG semakin lebah baik untuk menurunkan *losses*

Tabel 2. Hasil simulasi dengan perbedaan variasi tipe DG

TLPSO		PSO	
Peletakan Optimal	<i>Losses</i>	Peletakan Optimal	<i>Losses</i> (MW)

No Bus	<i>Sizing</i> MW+jMVar	(MW)	No Bus	<i>Sizing</i> MW+jMVar	
12	60 + j0		15	40 + j0	
14	10 + j0		19	8 + j2	
16	3 + j2		21	5 + j0	
20	10 + j0	5.18866	22	6 + j4	5.93874
21	60 + j0		24	18 + j12	
25	12 + j8		28	99 + j0	
27	10 + j0		30	3 – j2	
30	14 + j1				
8 bus	179 + j10		7 bus	179 + j16	

KESIMPULAN

Algoritma TLPSO yang digunakan dalam paper ini untuk optimalisasi peletakan DG mampu menurunkan *Losses* sebesar 57,3% sedangkan PSO sebesar 51.11% dibandingkan dengan sebelum dilakukan peletakan DG. *Losses* daya riil setelah peletakan DG dengan menggunakan kedua metode TLPSO dan PSO juga menunjukkan penurunan sebesar 12.6% dibandingkan PSO

DAFTAR PUSTAKA

- Chen, C.C., 2011, "Two Layer Particle Swarm Optimization for unconstrained Optimization Problem", *Applied Soft Computing*, 11(1), pp. 295-304.
- Jain. N, Singh. S.N, and Srivastava. S.C., 2010, Particle Swarm Optimization Based method Optimal Siting and Sizing of Mutliple Distributed Generators. 16th National Power Sistem Conference. 15th-17thDecember .pp. 669-674
- Keane, A and M. O'Melley, 2005. Optimal allocation of embedded generation on distribution networks. *IEEE Trans. Power Sys.*, 20(3): 1640-1646.
- Kennedy J and Eberhart R, 1995, "Particle Swarm Optimizer," IEEE international Conference on Neural Networks (Pert, Australia), IEEE Service Center Piscataway, NJ, IV, pp1942-1948.
- Mohammadi M and Nasab, M.A. 2011. PSO Based multiobjective Approach for Optimal Sizing and placement of Distributed Generation. 2(8): pp.832-837.
- Oumarou.I, Jiang.D, and Yijia. C. 2009. Particle Swarm Optimization Applied to optimal Power Flow Solution. International Coference on Natural Computation.Fifth.Pp284-288.