

## MODIFIED IMPROVED PARTICLE SWARM OPTIMIZATION FOR OPTIMAL GENERATOR SCHEDULING

Maickel Tuegeh<sup>1</sup>, Soeprijanto<sup>2</sup>, Mauridhi H Purnomo<sup>3</sup>

<sup>1</sup>Jurusan Teknik Elektro Universitas Sam Ratulangi Manado, Sulawesi Utara

<sup>2</sup>Jurusan Teknik Elektro ITS, Surabaya 60111

<sup>3</sup>Jurusan Teknik Elektro ITS, Surabaya 60111

E-mail: m\_tuegeh@elect-eng.ITS.ac.id, adisup@ee.its.ac.id, hery@ee.its.ac.id

### ABSTRAK

Generator scheduling merupakan bagian penting dalam pengoperasian sistem tenaga listrik. Tidak ada penjadwalan yang baik dapat menyebabkan biaya yang sangat besar dalam proses pengoperasian terutama pada sisi pembangkitan, juga dapat menyebabkan tidak ada koordinasi dalam menyalurkan tenaga listrik. Optimasi penjadwalan dari generator diperoleh dengan kecerdasan buatan diantaranya menggunakan teknik modified improved Particle Swarm Optimization (MIPSO). Hasil simulasi menunjukkan bahwa metode yang digunakan memberikan performa yang sangat baik dan hasil ini juga dibandingkan dengan metoda iterasi lamda untuk melihat keakuratan dari hasil yang didapat. Metode ini diaplikasikan pada dua contoh sistem dan diselesaikan dengan menggunakan Matlab.

**Kata Kunci:** Optimal Generator Scheduling, MIPSO.

### 1. PENGANTAR

Suatu sistem tenaga listrik memiliki tujuan untuk membangkitkan kemudian menyalurkan serta memanfaatkan energi listrik yang terbangkit tersebut. Suatu sistem tenaga listrik berdasarkan batasannya pada suatu sistem yang lengkap secara umum mengandung empat unsur, yaitu:

- a. Unsur pembangkit tenaga listrik
- b. Sistem transmisi
- c. Saluran distribusi
- d. Unsur pemakaian atas utilitas, yang terdiri atas instalasi pemakaian tenaga listrik.

Semakin berkembangnya jumlah penduduk menyebabkan penyebaran penduduk yang semakin luas. Hal ini juga berdampak pada kebutuhan akan energi listrik yang semakin hari makin meningkat dan juga tersebar luas. Kebutuhan akan listrik yang sangat besar tidak dapat dipenuhi oleh satu atau dua unit pembangkit tapi harus dipenuhi oleh beberapa unit pembangkit. Sistem tenaga listrik dengan banyak jenis pembangkit yang ada merupakan suatu kelebihan, tapi hal tersebut menyebabkan semakin kompleks permasalahan yang ada. Dengan karakteristik yang berbeda-beda maka pengoperasian suatu generator pada pembangkit harus optimal untuk melayani kebutuhan beban yang ada sehingga biaya operasi dapat ditekan. Pengoptimalan *generator scheduling* dalam sistem tenaga listrik sangat diperlukan, karena proses pembangkitan dan penyaluran dalam sistem tenaga listrik memerlukan biaya yang sangat besar. Koordinasi antar pembangkitan diperlukan dalam upaya melakukan *optimal generator scheduling* untuk memperoleh biaya yang minimum. Teknik solusi dalam masalah optimisasi dapat menggunakan kecerdasan buatan. Salah satu teknik solusi masalah

optimisasi dalam kecerdasan buatan menggunakan teknik *Particle Swarm Optimization (PSO)*. Kelebihan utama algoritma PSO adalah mempunyai konsep sederhana, mudah diimplementasikan, dan efisien dalam perhitungan jika dibandingkan dengan algoritma matematika dan teknik optimisasi heuristik lainnya.

Dalam penelitian ini, metoda PSO digunakan untuk menyelesaikan optimal generator scheduling dan memberikan hasil dengan performa yang baik.

### 2. PARTICLE SWARM OPTIMIZATION (PSO)

#### 2.1 Dasar PSO

Algoritma *Particle Swarm Optimization (PSO)* diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995, proses algoritmanya diinspirasi oleh perilaku sosial dari binatang, seperti sekumpulan burung dalam suatu swarm.

*Particle Swarm Optimization (PSO)* adalah salah satu dari teknik komputasi *evolusioner*, yang mana populasi pada PSO didasarkan pada penelusuran algoritma dan diawali dengan suatu populasi yang random yang disebut dengan *particle*.

Berbeda dengan teknik komputasi *evolusioner lainnya*, setiap *particle* di dalam PSO juga berhubungan dengan suatu *velocity*. *Particle-particle* tersebut bergerak melalui penelusuran ruang dengan *velocity* yang dinamis yang disesuaikan menurut perilaku historisnya. Oleh karena itu, *particle-particle* mempunyai kecenderungan untuk bergerak ke area penelusuran yang lebih baik setelah melewati proses penelusuran.

*Particle Swarm Optimization (PSO)* mempunyai kesamaan dengan genetic algorithm yang mana dimulai dengan suatu populasi yang random dalam bentuk matriks. Namun PSO tidak memiliki operator

evolusi yaitu crossover dan mutasi seperti yang ada pada *genetic algorithm*. Baris pada matriks disebut *particle* atau dalam *genetic algorithm* sebagai kromosom yang terdiri dari nilai suatu *variable*. Setiap *particle* berpindah dari posisinya semula ke posisi yang lebih baik dengan suatu *velocity*.

Pada algoritma PSO vektor *velocity* diupdate untuk masing-masing *particle* kemudian menjumlahkan vektor *velocity* tersebut ke posisi *particle*. *Update velocity* dipengaruhi oleh kedua solusi yaitu *global best* yang berhubungan dengan biaya yang paling rendah yang pernah diperoleh dari suatu *particle* dan solusi *local best* yang berhubungan dengan biaya yang paling rendah pada populasi awal. Jika solusi *local best* mempunyai suatu biaya yang kurang dari biaya solusi global yang ada, maka solusi *local best* menggantikan solusi *global best*.

Kesederhanaan algoritma dan performansinya yang baik, menjadikan PSO telah menarik banyak perhatian di kalangan para peneliti dan telah diaplikasikan dalam berbagai persoalan optimisasi sistem tenaga seperti economic dispatch, design kontrol PID pada sistem AVR, kontrol tegangan dan daya reaktif, *unit commitment* dan lain sebagainya.

PSO telah populer menjadi optimisasi global dengan sebagian besar permasalahan dapat diselesaikan dengan baik di mana variabel-variabelnya adalah bilangan riil.

Beberapa istilah umum yang biasa digunakan dalam Optimisasi *Particle Swarm* dapat didefinisikan sebagai berikut :

1. *Swarm* : populasi dari suatu algoritma.
2. *Particle*: anggota (individu) pada suatu *swarm*. Setiap *particle* merepresentasikan suatu solusi yang potensial pada permasalahan yang diselesaikan. Posisi dari suatu *particle* adalah ditentukan oleh representasi solusi saat itu.
3. *Pbest (Personal best)*: posisi *Pbest* suatu *particle* yang menunjukkan posisi *particle* yang dipersiapkan untuk mendapatkan suatu solusi yang terbaik.
4. *Gbest (Global best)* : posisi terbaik *particle* pada *swarm*.
5. *Velocity* (vektor): vektor yang menggerakkan proses optimisasi yang menentukan arah di mana suatu *particle* diperlukan untuk berpindah (*move*) untuk memperbaiki posisinya semula.
6. *Inertia weight* : *inertia weight* di simbolkan  $w$ , parameter ini digunakan untuk mengontrol dampak dari adanya *velocity* yang diberikan oleh suatu *particle*.

Prosedur standar untuk menerapkan algoritma PSO adalah sebagai berikut :

1. Inisialisasi populasi dari *particle-particle* dengan posisi dan *velocity* secara random dalam suatu ruang dimensi penelusuran.

2. Evaluasi fungsi *fitness* optimisasi yang diinginkan di dalam variabel  $d$  pada setiap *particle*.
3. Membandingkan evaluasi *fitness particle* dengan *Pbest*nya. Jika nilai yang ada lebih baik dibandingkan dengan nilai *Pbest*nya, maka *Pbest* diset sama dengan nilai tersebut dan  $P_i$  sama dengan lokasi *particle* yang ada  $X_i$  dalam ruang dimensional  $d$ .
4. Identifikasi *particle* dalam lingkungan dengan hasil terbaik sejauh ini.
5. *Update velocity* dan posisi *particle*.
6. Kembali ke step 2 sampai kriteria terpenuhi, biasanya berhenti pada nilai *fitness* yang cukup baik atau sampai pada jumlah maksimum iterasi.

Seperti halnya dengan algoritma evolusioner yang lain, algoritma PSO adalah sebuah populasi yang didasarkan penelusuran inisialisasi *particle* secara random dan adanya interaksi diantara *particle* dalam populasi. Di dalam PSO setiap *particle* bergerak melalui ruang solusi dan mempunyai kemampuan untuk mengingat posisi terbaik sebelumnya dan dapat bertahan dari generasi ke generasi.

## 2.2 Ukuran Swarm

Ukuran *swarm* atau populasi yang dipilih adalah tergantung pada persoalan yang dihadapi. Ukuran *swarm* yang umum digunakan berkisar antara 20 sampai 50. Hal tersebut telah dipelajari sejak dahulu bahwa PSO hanya perlu ukuran *swarm* atau populasi yang lebih kecil dibanding algoritma-algoritma evolusioner yang lain untuk mendapatkan solusi-solusi terbaik.

## 2.3 Koefisien Akselerasi

Pada umumnya nilai-nilai untuk koefisien akselerasi  $c_1$  dan  $c_2 = 2.0$ . Namun demikian, nilai koefisien akselerasi tersebut dapat ditentukan sendiri yang digunakan di dalam penelitian yang berbeda, biasanya nilai  $c_1$  dan  $c_2$  adalah sama dan berada pada rentang antara 0 sampai 4.

## 2.4 Inertia Weight

Perubahan *velocity* pada algoritma PSO terdiri atas tiga bagian yaitu *sosial part*, *cognitive part* dan *momentum part*. Ketiga bagian tersebut menentukan keseimbangan antara kemampuan penelusuran *global* dan *local*, oleh karena itu dapat memberikan performansi yang baik pada PSO.

Parameter *inertia weight* digabungkan dengan *sosial part* di dalam algoritma PSO standar. Persamaan dinamis dari PSO dengan *inertia weight* ( $w$ ) dimodifikasi menjadi :

$$V_{id} = wV_{id} + c_1 \text{rand}_1(P_{id} - X_{id}) + c_2 \text{rand}_2(P_{gd} - X_{id}) \quad (1)$$

dan

$$X_{id} = X_{id} + V_{id} \quad (2)$$

Persamaan (1) sama dengan persamaan (2) kecuali ada penambahan parameter baru yaitu *inertia weight* ( $w$ ). *Inertia weight* diperkenalkan untuk keseimbangan antara kemampuan penelusuran *global* dan *local*. Secara umum parameter *inertia weight* ( $w$ ) diperoleh dengan menggunakan persamaan berikut :

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{Iter_{\max}} Iter \quad (3)$$

Dengan :

$$w_{\max} - w_{\min} = \text{Weight awal dan akhir}$$

$$Iter_{\max} = \text{jumlah iterasi maksimum}$$

$$Iter = \text{jumlah iterasi yang ada}$$

*Inertia weight* yang besar dapat memudahkan dalam penelusuran *global* sedangkan *inertia weight* yang kecil memudahkan untuk penelusuran *local*. Penggunaan *inertia weight* dapat meningkatkan performansi dalam beberapa aplikasi, beberapa penelitian menunjukkan bahwa ketika menggunakan *inertia weight* faktor *velocity* maksimum dapat menjadi sederhana yang diset dengan nilai pada rentang dinamis dari setiap variable biasanya antara 0.4 sampai 0.9.

Konsep *inertia weight* ( $w$ ) dikembangkan oleh Shi dan Eberhart pada tahun 1998 menjadi suatu parameter kontrol yang baik dari lingkup penelusuran yang dimotivasi keinginan untuk mengurangi  $V_{\max}$ . Persamaan menggambarkan *velocity* dan posisi *particle* yang dimodifikasi dengan parameter *inertia weight*.

## 2.5 Constriction Factor

Parameter lain yang dikenal didalam algoritma PSO adalah *constriction factor*, parameter ini diperkenalkan oleh Clerc dengan tujuan dapat menjamin suatu penelusuran dalam algoritma PSO untuk konvergen lebih cepat.

*Coeffisient constriction* yang paling sederhana biasa disebut oleh Clerc sebagai type 1 yang memerlukan aplikasi dari koefisien untuk kedua bentuk persamaan *velocity*.

Pada umumnya peneliti menerapkan metode *constriction* dengan menetapkan nilai  $\phi = 4.1$  maka nilai-nilai untuk  $c_1$  dan  $c_2$  diset sama dengan 2.05 sehingga nilai  $C = 0.729$ . Dengan metode *constriction* ini mengakibatkan konvergensi dapat terjadi dari waktu ke waktu dimana lintasan osilasi amplitudo semakin berkurang .

## 2.6 Modified Improved Particle Swarm Optimization (MIPSO)

Persamaan (1)-(3) adalah persamaan dasar algoritma PSO yang dimodifikasi dengan menggunakan *Inertia Weight Aproach* (IWA). *Inertia weight* diperkenalkan untuk menyeimbangkan kemampuan antara penelusuran

*global* dan *local* . Clerc memperkenalkan parameter lain yang disebut dengan *Constriction Factor Approach* (CFA) yang digunakan untuk memodifikasi algoritma IPSO yang ada yang disebut dengan *Modified Improved Particle Swarm Optimization* (MIPSO). Parameter ini dapat memberikan performansi yang lebih baik pada algoritma MIPSO.

Pada umumnya peneliti menerapkan *constriction factor* pada algoritma PSO dengan mengeset nilai  $c_1$  dan  $c_2 = 2.05$  sehingga diperoleh nilai  $C = 0.729$ . Secara aljabar nilai ini ekuivalen dengan menggunakan *inertia weight* ketika  $w=0.729$  dan  $c_1 = c_2 = 1.49445$ , semakin besar nilai  $\phi$  maka  $C$  menjadi semakin kecil dan efek peredaman menjadi lebih baik. Karakteristik konvergensi dan perilaku pada sistem dapat dikontrol oleh  $\phi$  .

Berbeda dengan metode *evolutionary* komputasi (EC) yang lain, CFA pada algoritma MIPSO dipastikan konvergen pada penelusuran yang didasarkan pada teori matematika. Algoritma MIPSO dengan *Constriction Factor Approach* (CFA) dapat menghasilkan solusi yang baik dibandingkan dengan algoritma IPSO yang menggunakan *Inertia Weight Approach* (IWA), meskipun CFA hanya memperhitungkan perilaku dinamik dari *particle* atau *agent* dan pengaruh dari interaksi antara *particle-particle*, dimana persamaan-persamaan itu telah dikembangkan dengan posisi terbaik dengan  $Pbest$  dan  $Gbest$ , meskipun dapat berubah selama penelusuran di dalam persamaan dasar PSO.

## 3. BIAYA PEMBANGKITAN

Biaya operasi dari suatu sistem tenaga listrik merupakan biaya terbesar dalam pengoperasian suatu perusahaan listrik. Biaya yang dikeluarkan oleh suatu perusahaan listrik untuk menghasilkan energi listrik dalam suatu sistem tenaga listrik ditentukan oleh biaya investasi dan biaya operasi atau biaya produksi. Besar biaya investasi tidak bergantung pada besar daya keluaran pembangkit tetapi bergantung pada besar kapasitas daya terpasang pembangkit.

Biaya investasi meliputi biaya pembangunan pusat pembangkit, jaringan transmisi dan distribusi serta peralatan sistem lainnya, sedangkan biaya operasi atau biaya produksi merupakan semua biaya yang dikeluarkan dalam pengoperasian suatu pembangkit. Untuk sistem yang sudah ada (telah beroperasi) biaya investasi telah tertentu besarnya.

Beberapa komponen biaya yang terkait dengan optimisasi pembangkit adalah, antara lain:

- a. Biaya Tetap
- b. Biaya Bahan Bakar
- c. Biaya Star-Up
- d. Biaya Produksi
- e. Biayan Daya Cadangan

Meminimumkan biaya operasi pembangkitan adalah merupakan optimisasi, sehingga optimisasi pembangkitan dapat didefinisikan sebagai suatu proses pembangkitan yang bertujuan untuk mengoptimalkan daya dan meminimumkan biaya pembangkitan.

### 3.1 Karakteristik Input-Output Pembangkit

Karakteristik input output pembangkit termal adalah karakteristik yang menggambarkan hubungan antara input bahan bakar (liter/jam) dan output yang dihasilkan oleh pembangkit (MW).

Pada umumnya karakteristik input output pembangkit termal didekati dengan fungsi polinomial orde dua yaitu:

$$H_i = \alpha_i + \beta_i P_i + \gamma_i P_i^2 \quad (4)$$

dengan :

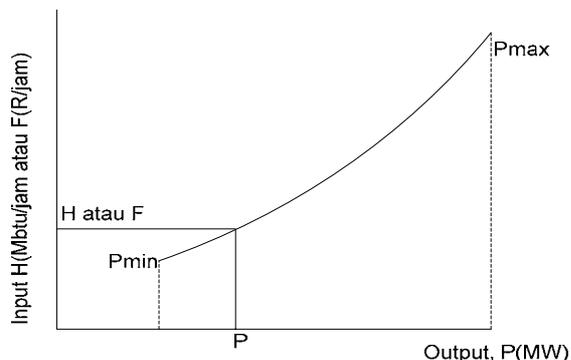
$H_i$  = Input bahan bakar pembangkit termal ke- $i$  (liter/jam)

$P_i$  = Output pembangkit termal ke- $i$  (MW)

$\alpha_i, \beta_i, \gamma_i$  = Konstanta input-output pembangkit termal ke- $i$ . Penentuan parameter  $\alpha_i, \beta_i$ , dan  $\gamma_i$  membutuhkan data yang berhubungan dengan input bahan bakar  $H_i$  dan output pembangkit  $P_i$ .

Output listrik dari sistem pembangkit termal selain disalurkan melalui jaringan transmisi pada suatu sistem tenaga listrik juga digunakan pada sistem tenaga bantu (*auxiliary power system*) pada suatu pusat pembangkit. Unit turbin uap membutuhkan 2-6 % dari output kotor untuk tenaga penggerak turbin (*boiler*), pompa, kipas, lampu dan sebagainya.

Untuk menggambarkan karakteristik input-output, input kotor direpresentasikan sebagai input total yang diukur dalam rupiah per jam dan output bersih pada suatu plant adalah output daya listrik dalam MW yang disediakan oleh sistem pembangkit tenaga listrik.



Gambar 1. Karakteristik Input-Output unit pembangkit termal (ideal)

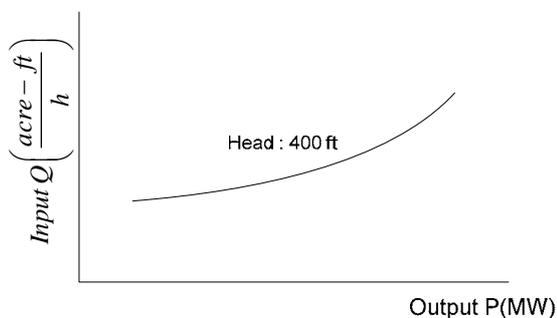
Gambar 1 adalah karakteristik input-output unit pembangkit termal dapat dinyatakan sebagai berikut:

1. Input dari pembangkit dinyatakan dalam :  
 $H = \text{Mbtu/jam}$  (energi panas yang dibutuhkan),  
 atau  
 $F = \text{R/jam}$  (total biaya bahan bakar)
2. Output dari pembangkit dinyatakan dalam :  
 $P = \text{MW}$  (daya).

Data karakteristik input-output biasanya diperoleh dari hasil perhitungan desain atau dari hasil pengukuran. Jika data yang digunakan adalah data dari hasil pengukuran maka akan diperoleh kurva yang tidak kontinyu (*smooth*).

Unit pembangkit termal mempunyai batas kritis operasi minimum dan maksimum, batas beban minimum umumnya disebabkan oleh kestabilan pembakaran dan masalah desain generator, sebagai contoh beberapa unit pembangkit termal tidak dapat beroperasi di bawah 30 % dari kapasitas desain.

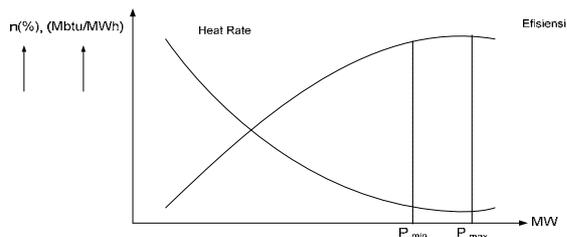
Unit pembangkit hidro mempunyai karakteristik input-output yang mirip dengan unit pembangkit termal. Karakteristik input-output pembangkit hidro menggambarkan hubungan antara input penggerak mula, berupa volume air yang dialirkan diantara sudu-sudu turbin persatuan waktu dan outputnya adalah daya listrik dari generator dalam MW. Gambar 2.8 menunjukkan karakteristik input-output untuk pembangkit tenaga air dengan head tetap.



Gambar 2. Karakteristik input-output unit pembangkit hidro

### 3.2 Batas Pembebanan Ekonomis Pembangkit Termal.

Umumnya mesin pembangkit tenaga listrik mempunyai batas pembebanan yang dibatasi oleh kapasitas ekonomis minimum dan kapasitas ekonomis maksimum dan juga keterbatasan kerja komponen-komponen mesin, sehingga dalam pembebanan pembangkit termal harus diperhatikan karakteristik efisiensi dan karakteristik kenaikan panas tiap unit pembangkit termal seperti pada Gambar 3

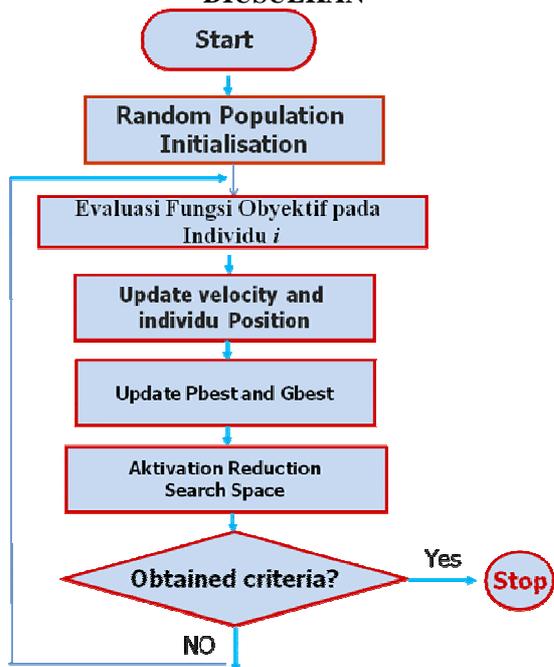


Gambar 3. Batas Pembebanan Suatu Unit Pembangkit

Rating pembebanan suatu unit pembangkit termal diambil pada daerah karakteristik heat rate hampir datar, dimana heat rate berharga minimum dan efisiensi optimum yang berkisar antara 80-90%.

Daerah pembebanan ini telah memberikan efisiensi yang cukup tinggi dengan heat rate yang rendah sehingga ditinjau dari segi ekonomisnya menguntungkan.

#### 4. FLOWCHART ALGORITMA YANG DIUSULKAN



Gambar 5 Flowchart Algoritma yang diusulkan

Pendekatan algoritma *Modified Improved Particle Swarm Optimization* (MIPSO) yang diusulkan akan diimplementasikan untuk

menyelesaikan persoalan *economic dispatch*. MIPSO memberikan suatu strategi baru untuk menemukan solusi *global* yang lebih baik.

#### 5. HASIL

Hasil perhitungan diaplikasikan pada dua contoh dengan menggunakan algoritma yang diusulkan dan untuk membandingkan keakuratan hasil maka dibandingkan dengan perhitungan menggunakan metode Iterasi Lambda. Data yang dipakai adalah persamaan karakteristik dari input-output pembangkit, nilai maksimum dan minimum dari generator dan biaya bahan bakar.

- a. Contoh pertama, diambil dalam Exp. 3.7 page 82 in "Power Generation, Operation and Control" by Allen J Wood

Unit	Unit data	in (MW)	Max (MW)	Fuel Cost (\$/Mbtu)
1	$225+8.4P_1+0.0025P_1^2$	45	350	0.8
2	$729+6.3P_2+0.0081P_2^2$	45	350	1.02
3	$400+7.5P_3+0.0025P_3^2$	47.5	350	0.9

Tabel 1. Data contoh pertama

Dengan menggunakan algoritma yang diusulkan maka diperoleh hasil:

No	Generator	MIPSO
	Load (MW)	450
1	Unit 1	206.1473
2	Unit 2	67.2187
3	Unit 3	176.6341
	Total Load (MW)	450
	Total cost (Rp. / Hour)	4485.6

Tabel 2. Hasil dari contoh pertama dengan MIPSO

Dengan menggunakan metode iterasi lambda maka diperoleh hasil:

No	Generator	Iterasi lambda
	Load (MW)	450
1	Unit 1	206.0
2	Unit 2	67.6
3	Unit 3	176.4
	Total Load (MW)	450
	Total cost (Rp. / Hour)	4485.64

Tabel 3. Hasil dari contoh pertama dengan Iterasi Lambda

b. Contoh kedua, sistem kelistrikan dengan 5 unit termal dan total load 82 MW

Unit	Unit data	M i n ( M W )	Ma x ( M W )	Fue l Co st ( R/ iter)
1	$4547.8075 + 62.8734 P_1 + 3.8639 P_1^2$	8	30	431.8
2	$4450.825 + 53.1832 P_2 + 4.181 P_2^2$	8	30	431.8
3	$345.92 + 82.923 P_3 + 10.555 P_3^2$	6	9	441.7
4	$780.7825 + 39.872 P_4 + 0.5906 P_4^2$	6	8	441.7
5	$322.5 + 156.65 P_5 + 8.625 P_5^2$	6	8	441.7

Tabel 4 Data contoh kedua

No	Generation	MIPSO
	Load (MW)	82
1	Unit 1	29.2045
2	Unit 2	28.1483
3	Unit 3	9
4	Unit 4	8
5	Unit 5	7.6472
	Total load (MW)	82
	Total cost (Rp. / Hour)	<b>174.857.063,47</b>

Tabel 5 Hasil dari contoh kedua dengan MIPSO

No	Generation	Iterasi Lambda
	Load (MW)	82
1	Unit 1	29.2
2	Unit 2	28.1
3	Unit 3	9
4	Unit 4	8
5	Unit 5	7.6
	Total load (MW)	82
	Total cost (Rp. / Hour)	<b>174.857.931,11</b>

Tabel 6 Hasil dari contoh dengan Iterasi Lambda

## 6. KESIMPULAN

1. Dalam tulisan ini masalah Optimal Generator Scheduling dapat diselesaikan dengan menggunakan MIPSO
2. Perhitungan telah diaplikasikan dalam dua sistem dan dua metode
3. Menggunakan MIPSO memperoleh hasil yang lebih baik dari metode iterasi Lambda.

## PUSTAKA

- Wood Allen J, Wollenberg Bruce F, (1996), "Power Generation, Operational, and Control", Second Edition, Jhon Wiley & Sons, Inc
- Kundur Praba (1993) "Power Sistem Stability And Control" McGraw-Hill
- Shahidehpour Mohammad, Wang Yaoyu, (2003) "Communication And Control In Electric Power Sistem" Jhon Wiley & Sons, Inc
- Saadat Hadi (2004) "Power Sistem Analysis" International Edition, McGraw-Hill
- James Kennedy and Russell C. Eberhart, with Yuhui Shi (2001) "Swarm Intelligence" Morgan Kaufmann Publishers
- Clerc Maurice (2006) "Particle Swarm Optimization" ISTE, Ltd
- Pablo E. Oñate Yumbla, Juan M. Ramirez and Carlos A. Coello Coello (Feb 2008) "Optimal Power Flow Subject to Security Constraints Solved With a Particle Swarm Optimizer" IEEE Transactions on Power Systems, vol. 23, no. 1
- Andi Syarifuddin, 2008, "Economic Dispatch For Thermal Generator In Sulawesi Selatan System Using Modified Improved Particle Swarm Optimization" ITS Surabaya