Jurnal Sains & Informatika *ISSN: 2460-173X*
Volume 2,Nomor 2, Nopember 2016

Penerapan *Hybrid Bat Algorithm* (BA) dengan *Differential Evolution* (DE) untuk Mengoptimasi Model Multiobjektif

Veri Julianto

Jurusan Teknik Informatika, Politeknik Negeri Tanah Laut
Jalan A. Yani Km 6 Desa Panggung, Pelaihari
verijulianto@gmail.com

Abstrak – *Optimasi merupakan suatu proses untuk memaksimalkan atau meminimalkan suatu permasalahan tertentu agar diperoleh kondisi optimal. Dalam mengoptimasi suatu permasalahan terkadang model kasus yang tersedia bersifat muliobjektif, yang artinya ada lebih dari satu fungsi model yang harus diselesaikan secara bersamaan. Pada model multiobjektif keoptimalan satu fungsi dipengaruhi fungsi-fungsi yang lain. Sehingga untuk mendapatkan solusinya lebih sulit jika dibandingkan dengan mengoptimalkan satu fungsi saja. Salah satu metode dalam penyelesaian permasalahan multiobjektif yaitu dengan menggunakan metode metaheuristik. Metode ini mudah digunakan dan tidak memerlukan informasi gardien dari model yang akan dioptimalkan. Salah satu metode metaheuristik adalah Bat Algorithm (BA). Pada penelitian ini BA akan dihybrid-kan dengan menggunakan metode Differential Evolution (DE) yang sudah sejak lama powerfull dalam pencarian solusi optimal suatu persamaan. Kedua algoritma ini digabungkan dengan menggabungkan local search pada BA dengan DE. Sehingga keunggulan pada BA dan keunggulan pada DE dapat dioptimalkan dalam satu algoritma hybrid. Berdasarkan hal tersebut maka hybrid dari BA dan DE akan diuji dengan menyelesaikan model-model fungsi yang bersifat multiobjektif. Hasil yang didapat menunjukan bahwa algoritma hybrid ini mampu menyelesaikan fungsi-fungsi yang memuat permasalahan multiobjektif.*

Kata Kunci*: Hybrid, Multiobjektif, Optimal, Bat Algorithm (BA), Differential Evolution (DE)*

1. PENDAHULUAN

Permasalahan-permasalahan optimasi di bidang teknik sering kali menjumpai model fungsi yang sulit untuk diselesaikan, seperti bersifat *non linier*, memiliki banyak kendala (*constraint*), *multimodal*¸ dan juga memiliki fungsi tujuan lebih dari satu (*multiobjektif*). Jika model permasalahan berupa persamaan yang *single objective* maka akan mudah untuk diselesaikan dan didapatkan solusinya. Akan tetapi, model permasalahan yang lebih mendekati kenyataan dalam bidang teknik yaitu berupa permasalahan yang multiobjektif atau memiliki lebih dari satu fungsi tujuan. Permasalahan pada optimasi multiobjektif yaitu banyaknya fungsi yang harus diselesaikan sekaligus, dikarenakan satu fungsi yang akan dioptimalkan maka akan berdampak pada fungsi-fungsi tujuan yang lain.

Metode yang digunakan untuk mengoptimasi suatu fungsi dari suatu permasalahan dibidang teknik sudah bermacam-macam, tergantung pada kerumitan kasus yang ada. Misalkan, kasusnya sederhana bisa dikerjakan dengan menggunakan metode eksak, akan tetapi apabila kasus rumit dan memiliki kompleksitas tinggi maka perlu menggunakan metode komputasi dalam menyelesaikannya. Metode komputasi dalam permasalahan optimasi juga cukup banyak. Sebelum diperkenalkan metode metaheuristik, banyak orang menggunakan metode turunan atau *gradient* seperti *Newthon Rhapson*, *Bisection*, dan lain-lain. Metode ini bagus untuk fungsi yang sederhana, akan tetapi metode ini mempunyai kelemahan yaitu menghendaki fungsi memiliki turunan kontinyu, memiliki tebakan awal yang tepat agar konvergen dan terkadang terjebak pada kondisi minimum atau maksimum lokal. Sehingga diperlukan metode lain untuk menyelesaikan permasalahan ini, salah satunya metode metaheuristik.

Metode metahuristik adalah metode optimisasi yang dilakukan dengan memperbaiki kandidat penyelesaian secara iteratif sesuai dengan fungsi objektifnya (Talbi, 2009). Metode ini mampu menghasilkan penyelesaian yang baik dalam waktu yang cepat (*acceptable*), tetapi tidak menjamin bahwa penyelesaian yang dihasilkan merupakan penyelesaian terbaik (*optimal*). Metaheuristik ini menekankan pada proses eksplorasi (pencarian global) dan eksploitasi (pencarian lokal).

Metode metaheuristik pada masa modern ini sering diinspirasi dari kejadian-kejadian alami baik dari tingkah laku hewan, tumbuhan dan juga konsep evolusi. Metode metaheuristik yang menggunakan konsep evolusi diinspirasi dari hukum Darwin, bahwa yang bertahan adalah yang paling kuat atau yang paling baik. Teknik evolusi digunakan pada algoritma genetika (*Genetic Algorithm*) dan *Differential Evolution* (DE) untuk membantu menemukan solusi optimal suatu permasalahan. Adapun algoritma yang didasarkan pada teknik *swarm intelligence* yaitu seperti halnya *Particle Swarm Optimization (PSO), Ant Colony*

Optimization, Artificial Bee Colony, Chuckoo Search, Firefly, and Bat Algorithm.

Pada penelitian ini akan digunakan Algoritma Kelelawar (*Bat Algorithm (BA)*) yang akan dikombinasikan dengan teknik evolusi dengan mengunakan konsep *Differential Evolution* (DE). Algoritma kelelwar sendiri diinspirasi dari tingkah laku terbang kelelawar dalam menemukan mangsa dan menghindari rintangan. Fenomena inilah yang menginspirasi Yang (2010) dalam mengembangkan algoritma kelelawar ini. Algoritma kelelawar sangat bagus dalam menyelesaikan permasalahan kontinyu dan diskrit, kemudian dikembangkan kembali oleh Yang (2012) untuk menyelesaikan permasalahan multiobjektif. Penelitian ini akan menggabungkan (*hybrid*) Algoritma Kelelawar dengan *Differential Evolution (DE)*, penggabungan ini pernah berhasil dikerjakan oleh Fister (2013) untuk model satu fungsi (*single objective*). Selanjutnya dalam penelitian ini, penggabungan algoritma kelelawar dan *Differential Evolution* ini akan digunakan untuk menyelesaikan permasalahan yang bersifat banyak fungsi (multiobjektif).

2. TINJAUAN PUSTAKA

2.1 *Differential Evolution* (DE)

Differential Evolution (DE) adalah sebuah metode yang dikembangkan oleh Kenneth Price dan dipublikasikan pada Oktober 1994 dalam majalah Dr. Dobb's Journal (Price dkk, 2005). Metode ini merupakan metode optimasi matematis fungsi multidimensional dan termasuk dalam kelompok *evolutionary algorithm*.

Sebuah populasi dapat diinisialisasi, *upper* dan *lower bounds* untuk setiap parameter harus ditentukan, yaitu dengan vector inisialisasi dimensi $b_L$dan b_U. L menunjukan *lower* dan U menunjukan *upper*. Berikutnya adalah membangitkan bilangan acak untuk setiap parameter j dan vektor i pada integrasi g. Misal nilai inisial ($g = 0$) :

$$x_{j,i,0} = rand_j(0,1).\left(b_{j,U} - b_{j,L}\right) + b_{j,L} \qquad (1)$$

Bilangan acak (1) dibangkitkan berasarkan distribusi uniform pada rentang $[0,1)$ atau $0 \leq rand_j(0,1) < 1$.

Setelah diinisialisasi, DE akan memutasi dan me-rekombinasi populasi awal untuk menghasilkan populasi baru. Mutasi pada beberapa kamus bahasa menunjukkan pengertian berubah dan dalam konteks genetika mutasi berarti perubahan dengan elemen acak. Berikut ini adalah persamaan yang menunjukkan bagaimana membentuk vektor mutan, $v_{i,g}$:

$$\boldsymbol{v_{i,g} = x_{r0,g} + F.\left(x_{r1,g} - x_{r2,g}\right)} \qquad (2)$$

Pada Persamaan (2) $r0, r1, r2$ adalah indeks acak, integer, dan berbeda. Indeks basis vektor $r0$, dapat ditentukan dengan berbagai cara antara lain acak, permutasi, stokastik, dan acak offset. Sedangkan untuk $r1$ dan $r2$ dipilih secara acak sekali untuk setiap mutan.

Untuk melengkapi strategi pencarian *differential mutation*, DE menggunakan *crossover* dengan tujuan meningkatkan diversitas parameter populasi. *Crossover* membangun vektor uji dari nilai parameter yang telah dikopi dari dua vektor yang berbeda. Persamaan (3) untuk vektor uji adalah sebagai berikut:

$$\boldsymbol{u_{i,g+1} = \left(u_{1i,g+1}, u_{2i,g+1}, \ldots, u_{ni,g+1}\right)} \qquad (3)$$

Dimana :

$$u_{i,g+1} = \begin{cases} v_{j,i,g+1}\ if (rand_j(0,1) \leq CR\ or\ j = j_{rand} \\ x_{j,i,g}\ yang\ lainnya\ j = 1,2,3, \ldots, n \end{cases} \qquad (4)$$

Menurut Price dkk (2005), pada dasarnya ada dua tahapan dalam proses evolusi yang menggunakan seleksi yaitu *parent selection* dan *survivor selection*.

Berikut ini adalah penjelasan mengenai kedua seleksi tersebut :

a. *Parent selection*

Vektor yang terpilih ditandai dengan nilai fungsi terbaik dan probabilitas seleksi tertinggi. Metode ini dalam memberikan probabilitas seleksi membutuhkan tambahan asumsi tentang bagaimana menggambarkan nilai fungsi tujuan menjadi probabilitas.

b. *Survivor selection*

Metode ini juga bias disebut *repelcment*. Untuk mengetahui apakah vector menjadi anggota generasi $g + 1$, maka vector uji $u_{i,g+1}$ dibandingkan dengan vector target $x_{i,g}$menggunakan kriteria *greedy*. Jika vektor $u_{i,g+1}$ menghasilkan fungsi biaya yang lebih kecil daripada $x_{i,g}$ maka $x_{i,g+1}$ akan diatur menjadi $u_{i,g+1}$, dan bila sebaliknya maka nilai $x_{i,g}$ yang lama dipertahankan. Apabila penjelasan diatas ditunjukan dalam persamaan , maka hasilnya adalah sebagai berikut :

$$x_{i,g+1} = \begin{cases} u_{i,g}\ if\ f\left(u_{i,g}\right) \leq f(x_{i,g}) \\ x_{i,g}, yang\ lainnya \end{cases} \qquad (5)$$

Pada penelitian sebelumnya algoritma DE digunakan oleh Fahmiari pada tahun 2010 untuk menyelesaikan permaslahan kompleks pada portofolio. Algoritma DE berhasil memberikan berbagai alternatif portofolio yang risikonya rendah pada ekspektasi return yang telah ditentukan dibandingkan dengan *generalized random gradient* (GRG2).

 ISSN: 2460-173X

2.2 *Bat Algorithm* (BA)

Konsep dasar pengembangan Algoritma Kelelawar yaitu:

a. Semua kelelawar menggunakan kemampuan ekolokasi untuk mengetahui jarak, mereka juga dapat membedakan antara mangsa dan benda-benda di sekitar mereka.
b. Kelelawar terbang secara acak dengan kecepatan $\mathbf{v_i}$ dan dengan posisi $\mathbf{x_i}$ dengan frekuensi tetap f_{min}, dengan variasi panjang gelombang λ dan kenyaringan A_0 saat mencari mangsa.
c. Mereka secara otomatis dapat menyesuaikan panjang geombang dari sinyal yang merka pancarkan dan menyesuaikan tingkat sinyal $r \in [0,1]$, dan tergantung pada target mereka.
d. Tingkat kekerasan suara diasumsikan bervariasi dari A_{min} hingga A_0.

Dalam jurnal yang ditulis oleh Yang (2010) membahas mengenai pergerakan kelelawar dalam menghasilkan sebuah algoritma sehingga dapat dijadikan sebagai metode pencarian solusi suatu fungsi objektif. Pada jurnal tersebut dikatakan bahwa kelelawar terbang dengan kecepatan $\mathbf{v_i}$ pada posisi $\mathbf{x_i}$ di ruang pencarian pada dimensi d. Posisi $(\mathbf{x_i^t})$ dan kecepatan $(\mathbf{v_i^t})$ yang baru pada waktu t diberikan oleh :

$$\mathrm{f_i} = \mathrm{f_{min}} + (\mathrm{f_{max}} - \mathrm{f_{min}})\beta, \quad (6)$$
$$\mathbf{v_i^t} = \mathbf{v_i^{t-1}} + (\mathbf{x_i^t} - \mathbf{x}_*)\mathrm{f_i}\,, \quad (7)$$
$$\mathbf{x_i^t} = \mathbf{x_i^{t-1}} + \mathbf{v_i^t}, \quad (8)$$

dengan $\beta \in [0,1]$ merupakan vektor acak dari distribusi uniform dan $\mathbf{x}_*$ merupakan solusi global terbaik yang diperoleh dengan membandingkan dengan seluruh solusi diantara n kelelawar.

Dalam pencarian lokal, setiap satu solusi didapatkan diantara solusi terbaik saat itu. Solusi yang baru untuk setiap kelelawar dibangkitkan secara lokal menggunakan *random walk.*

$$\mathbf{x_{baru}} = \mathbf{x_{lama}} + \epsilon A^t\,, \quad (9)$$

dengan $\epsilon \in [-1,1]$merupakan suatu bilangan acak dan $A^t = \langle A_i^t \rangle$ adalah rata-rata dari tingkat Kekerasan suara dari seluruh kelelawar pada waktu t. Pada penelitian ini *random walk* di rubah dengan menggunakan hasil penelitian Yuanbin, M. ,Xinquan, Z. , dan Shujian, X. (2013) yaitu sebagai berikut :

$$\mathbf{x_{baru}} = \mathbf{x_{lama}} + c \cdot rand \cdot (\mathbf{x}_* - \mathbf{x_{lama}}) \quad (8)$$

dengan $\mathbf{x}_*$ adalah $\mathbf{x}$ terbaik sekarang untuk kelelawar ke-i dan nilai c adalah konstanta positif . Sedangkan $rand \in [0,1]$ merupakan bilangan acak.

Tingkat kekerasan A_i dan laju emisi gelombang suara r_i harus diperbaharui di setiap iterasi. Tingkat kekerasan suara biasanya menurun seiring kelelawar menemukan mangsanya sementara laju emisi gelombang suara meningkat. A dapat bervariasi dari $A_0 = 1$ hingga $A_{min} = 0$. Untuk setiap iterasi, maka A_i dan r_i diperbaharui dengan formula berikut :

$$A_i^{t+1} = \alpha A_i^t,\ r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)], \quad (10)$$

dengan $0 < \alpha < 1$ dan $\gamma > 0$. untuk $t \to \infty$ kita perhatikan bahwa :

$$A_i^t \to 0, \quad \text{dan} \quad r_i^t \to r_i^0.$$

2.3 Multiobjektif BA

Pada kasus multiobjektif dengan menggunakan Algoritma Kelelawar akan lebih rumit jika dibandingkan dengan menyelesaikan satu fungsi tujuan (*single objective*). Untuk menyelesaikan permasalahan multiobjektif dengan menggunakan Algoritma Kelelawar Multiobjektif atau *Multiobjective Bat Algorithm* (MOBA) diperlukan konsep pareto optimal dalam menemukan himpunan solusi-solusinya. Berikut ini adalah algoritma MOBA dengan menggunakan metode bobot jumlah untuk membuat fungsi tujuannya menjadi tunggal.

Fungsi objektif $f_1(\mathbf{x}), \dots, f_p(\mathbf{x}),\ \mathbf{x} = (x_1, \dots, x_n)^T$
Inisiasi populasi kelelawar $\mathbf{x_i}$ $(i = 1,2, \dots, n)$ *dan* v_i
Definisikan frekuensi f_i *pada* $\mathbf{x_i}$
Inisiasi laju emisi gelombang r_i *dan tingkat kekerasan* A_i
For j=1 :n *(titik-titik pareto front)*
Bangkitkan p bobot $w_k \geq 0, \sum_{k=1}^{p} w_k = 1$
Bentuk fungsi tunggal $f = \sum_{k=1}^{p} w_k f_k$
while *(*$t < iterasi\ maksimum$*)*
Bangkitkan solusi baru dengan mengatur frekuensi
Perbaharui kecepatan dan lokasi $[persamaan\ (6) sampai\ (8)]$
    if ($\boldsymbol{rand > r_i}$***)***
      pilih solusi diantara solusi terbaik
    bangkitkan solusi lokal diantara solusi terbaik (persamaan 9)
    end
    if ($\boldsymbol{rand < A_i}$ & $\boldsymbol{f}(\mathbf{x_i}) < \boldsymbol{f}(\mathbf{x}_*)$***)***
      terima solusi yang baru
      perbaharui r_i *dan* A_i *persamaan* (10)
    end
Urutkan setiap kelelawar dan pilih $\mathbf{x}_*$ *yang baru*
End while
Simpan $\mathbf{x}^*$ *sebagai non dominated solution*
End

Gambar 1. *Peseoudocode* Algoritma Kelelawar

2.4 Pareto Optimal

Berikut ini akan diberikan definisi mengenai pareto optimal yang dijadikan sebagai alat untuk menentukuan solusi dari permasalahan multiobjektif.

Definisi 1. Pareto Optimal

"*Sebuah titik* $\mathbf{x}_* \in R^n$ *disebut sebagai pareto optimal jika dan hanya jika tidak ada titik yang lain,* $\boldsymbol{x} \in R^n$ *memenuhi* $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}_*)$, $i = (1,2,3, \dots, p)$.

Berdasarkan definisi 1 bisa dikatakan bahwa $\boldsymbol{x}_*$ adalah pareto optimal jika tidak terdapat vektor keputusan (*vector feasible*) yang dapat menurunkan beberapa fungsi objektif tanpa menaikan fungsi objektif yang lain sekaligus. Tidak seperti masalah optimasi dengan satu fungsi tujuan, multiobjektif akan langsung mendapatkan himpunan dari solusi. Misalkan, vektor keputusan $\boldsymbol{x}_*$ pada himpunan solusi disebut *non-dominated* dan himpunan dari solusi yang *non-dominated* dari fungsi objektif (tujuan) dinamakan pareto front (*pareto frontier*).

3. METODE PENELITIAN

Penelitian ini dilakuakan berdasarkan tahapan-tahapan penting yang dikerjakan dengan berorientasikan kepada indikator keberhasilan dalam menggabungkan algoritma kelelawar dengan *Differential Evolution* (DE) sehingga dapat digunakan untuk menyelesaikan permasalahan multiobjektif. Untuk dapat mencapai, indikator tersebut, maka tahapan-tahapan penelitian ini adalah sebagai berikut :

1. **Analisa masalah**, dalam hal ini kebutuhan menganalisa permasalahan yang akan diteliti mengenai dua algoritma yaitu algoritma kelelawar (BA) dan *Differential Evolution*.
2. **Analisa kebutuhan**, dalam hal ini segala kebutuhan dalam meneliti baik dari jurnal, buku, literatur-literatur, alat dan bahan.
3. **Mendesain sistem** yang akan dibangun dengan menggabungkan BA dan DE sehingga menjadi algoritma baru.
4. **Membuat program** hasil dari desain algoritma penggabungan antara DE dan BA.
5. **Menguji program** yang dibuat dengan mengubah-ubah parameter untuk didapatakn kondisi optimal.
6. **Menguji program** pada permasalahan multiobjektif.

4. HASIL DAN PEMBAHASAN

4.1 Algoritma Hybrid Differential evolution dengan Algoritma Kelelawar

Hasil uji coba penggabungan algoritma *DE* dan *BA* dalam mendapatkan solusi optimal untuk kasus multiobjektif.

Fungsi objektif $f_1(\mathbf{x}), \dots, f_p(\mathbf{x})$, $\mathbf{x} = (x_1, \dots, x_n)^T$
Inisiasi populasi kelelawar $\mathbf{x_i}$ $(i = 1,2, \dots, n)$ *dan* v_i
Definisikan frekuensi f_i *pada* $\mathbf{x_i}$
Inisiasi laju emisi gelombang
r_i *dan tingkat kekerasan* A_i
For j=1 :n *(titik-titik pareto front)*
  Bangkitkan p bobot $w_k \geq 0, \sum_{k=1}^{p} w_k = 1$
  Bentuk fungsi tunggal $f = \sum_{k=1}^{p} w_k f_k$
while $(t <$ *iterasi maksimum*$)$
  Bangkitkan solusi baru dengan mengatur frekuensi
  Perbaharui kecepatan dan lokasi
  $[$*persamaan* (6)*sampai* $(8)]$
 if $(\boldsymbol{rand} > \boldsymbol{r_i})$
   pilih solusi diantara solusi terbaik
   membangkitkan solusi local dengan Differential Evolution.
 end
 if $(\boldsymbol{rand} < \boldsymbol{A_i}\ \&\ \boldsymbol{f}(\mathbf{x_i}) < \boldsymbol{f}(\mathbf{x}_*))$
   terima solusi yang baru
   perbaharui r_i *dan* A_i *persamaan* (10)
 end
Urutkan setiap kelelawar dan pilih $\mathbf{x}_*$ *yang baru*
End while
Simpan $\mathbf{x}^*$ *sebagai non dominated solution*
end

Gambar 2. *Peseoudocode Hybrid* Algoritma

4.2 Uji Coba *Hybrid* Algoritma

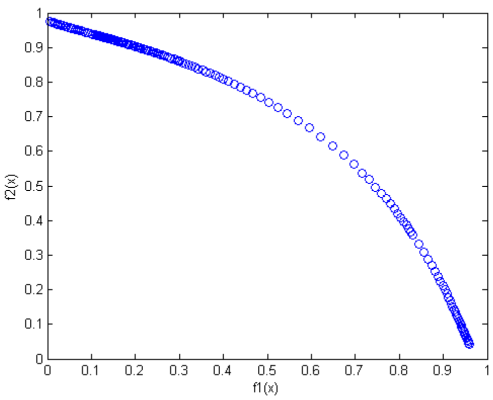
Hasil pembuatan program dengan penggabungan Algoritma *DE* dan *BA* akan diujicobakan kepada beberapa fungsi berikut:

A. Fungsi Fonseca dan Fleming (FON)

Fungsi Fonseca dan Felming (FON) adalah fungsi yang mempunyai 2 fungsi tujuan dengan batas yaitu [-6,6]. Bentuk fungsi ini yaitu sebagai berikut

$$\text{Minimum FON} = \begin{cases} f_1(x) = 1 - \exp\left(-\sum_{i=1}^{d}\left(x_i - \frac{1}{\sqrt{d}}\right)^2\right) \\ f_2(x) = 1 - \exp\left(-\sum_{i=1}^{d}\left(x_i + \frac{1}{\sqrt{d}}\right)^2\right) \end{cases}$$
$$\text{dengan } -6 \leq x_i \leq 6$$

Dengan menggunakan algoritma *Hybrid* maka didapatkan solusi yaitu sebagai berikut ini, solusi ini akan diperoleh suatu himpunan *non-dominated* yang berbentuk *pareto front* seperti gambar dibawah ini:



Gambar 3. Solusi *Pareto Front* Fungsi FON

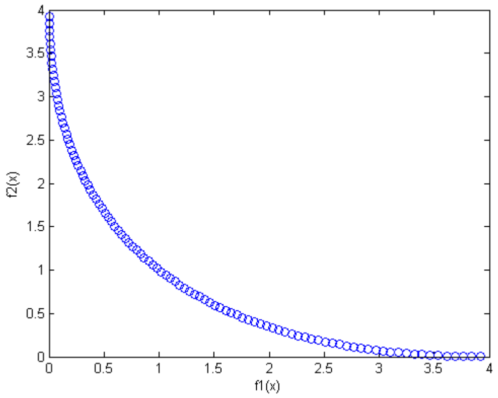
Tabel 1. Beberapa *pareto front* dari fungsi FON

No. Pareto	w_1	w_2	x_1	x_2	x_3	f_1	f_2
1	0.4	0.6	-0.26674	-0.26726	-0.2666	0.882121	0.25114
2	0.48	0.52	-0.11182	-0.11187	-0.11121	0.759273	0.478308
3	0.49	0.51	-0.06432	-0.06501	-0.06469	0.709626	0.54548
4	0.495	0.505	-0.03501	-0.03485	-0.03479	0.67518	0.586384
5	0.5	0.5	-0.00134	-0.00075	-0.00088	0.633376	0.630863
6	0.505	0.495	0.03428	0.034552	0.035222	0.586649	0.674945
7	0.51	0.49	0.065055	0.063985	0.06531	0.545327	0.709748
8	0.515	0.485	0.090156	0.090329	0.090949	0.508913	0.737626
9	0.52	0.48	0.112044	0.111568	0.111618	0.478147	0.759383
10	0.6	0.4	0.266375	0.265961	0.265723	0.25232	0.881616
11	0.7	0.3	0.350059	0.351162	0.350721	0.142885	0.924494

B. Fungsi Schafer (SCH)

Fungsi Schafer (SCH) adalah salah satu fungsi multiobjektif sederhana dengan fungsi pertamanya adalah fungsi kuadrat dan fungsi keduanya adalah fungsi kuadrat juga dengan titik minimum yaitu di (2,0). Berikut ini adalah bentuk umum dari persamaan SCH yaitu :

$$\text{Minimum:}\begin{cases} f_1(x) = x^2 \\ f_2(x) = (x-2)^2 \end{cases}, \text{dengan } -4 \le x \le 4$$



Gambar 4. Solusi *Pareto Front* Fungsi SCH

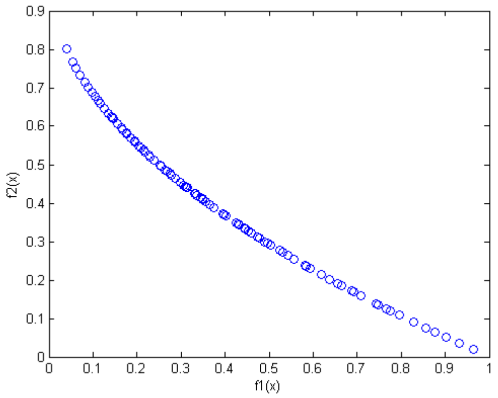
C. ZDT 1

Fungsi ZDT memiliki 6 fungsi diantaranya yaitu ZDT1. Fungsi ini mengenengahkan masalah meminimum 2 fungsi yaitu minimum $f_1(x)$, minimum $f_2(x) = g(x)h\big(f_1(x), g(x)\big)$. Pada fungsi ZDT1 ini terdapat 30 variabel (n=30) yang membentuk parito optimal yang konvek. Fungsinya yaitu sebagai berikut :

$$min - \min ZDT1: \begin{cases} f_1(x) = x_1 \\ g(x) = 1 + \dfrac{9}{n-1}\displaystyle\sum_{i=2}^{30} x_i \\ f_2(f_1, g) = 1 - \sqrt{\dfrac{f_1}{g}} \end{cases},$$

$$x_i \epsilon [0,1], i = 1, \dots, 30$$

Dengan menggunakan algortima kelelawar maka didapatkan pareto optimalnya yaitu seperti gambar berikut :



Gambar 5. Solusi *Pareto Front* Fungsi ZDT1

4.3 Pembahasan

Pengujian Algoritma Kelelawar yang digabungkan dengan *Differential Evolution* terhadap fungsi-fungsi multiobjektif berhasil dilakukan. Fungsi yang diambil untuk melakukan pengujian terhadap algoritma *Hybrid* ini yaitu fungsi-fungsi yang memiliki sifat *convex*.

Pada pengujian ini menggunakan 1000 iterasi untuk setiap jalannya program. Dengan parameter algortima yaitu jumlah populasi yang digunakan yaitu 25, parameter untuk $A = 0{,}5$ dan $r = 0{,}5$. Dan pada *Diferential Evoution* untuk parameter F untuk mutasi yaitu 0,9.

Hasil dari jalannya sebuah program menunjukan bahwa algoritma *Hybrid* ini sukses dan dapat digunakan sebagai alternatif dalam menyelesaikan permasalahan multiobjektif. Salah satu hasilnya yaitu pada table 1. Algoritma *Hybrid* memberikan hasil berupa solusi dalam menyelesaikan permasalahan multiobjektif pada fungsi FON. Dengan merubah bobot maka akan didapatkan hasil pada masing-masing fungsi objektif yang optimal. Bobot tersebut menandakan seberapa penting fungsi tujuan yang dibebankan.

Pada gambar 3 dan 4 dengan melihat hasil dari literatur tes fungsi yang ada maka dapat terlihat kesamaan dari pola *pareto front* yang terbentuk. Pada gambar 3 secara umum memiliki pola seperti pada percobaan yang dilakukan oleh Yang (2012) pada permasalahan multiobjektif dengan menggunakan algoritma kelelawar dan Yang (2013) dengan menggunakan algoritma Cuckoo.

5. KESIMPULAN

Algoritma Kelelawar yang digabungkan dengan *Difrential Evolution* menjadi sebuah algoritma baru *hybrid* ini, terbukti dapat menyelesaikan kasus-kasus multiobjektif sehingga dapat dijadikan sebuah alternatif baru dalam optimasi permsalahn multiobjektif.

Untuk Penelitian selanjutnya yaitu dengan menghitung keefektifan dan keefiseinan *hybrid* Algoritma tersebut jika dibandingkan dengan hanya menggunakan algoritma kelelawar atau Diferential Evolution dalam menyelesaikan permasalahan multiobjektif.

DAFTAR PUSTAKA

Andi Hasad. 2011. Algoritma Optimasi dan aplikasinya. Sekolah pascasarjana IPB Bogor . Jawa Barat.

C. Blum, A, Roli. 2003. *Metheuristic in Combinatorial Optimization: Overview in conceptual comparison*. ACM Computing Surveys, 35(3):268-308

Deb K.,. f2001. *Multi-Objective Optimization Using Evolutionary Algorithm*, John Wiley and Sons, New York.

Fister, et all. 2013. *Hybrid Bat Algorithm,* arxiv.org/pdf/1303.6310

Price, K.V., Storn, R.M., dan Lampinen, J.A. (2005). *Differential Evolution: A Practical Approach to Global Optimization.Natural Computing Series*. Springer-Verlag, Berlin.

Sleesongsom S. 2008. *Multiobjective Optimization with Even Pareto*. Filter,In Fourth Natural Computation International Conference onNatural Computation, pp. 92-96, 2008].

Talbi. 2009. *Metaheuristics: From Design to Implementation*. Wiley. New York.

Yang. X.-S. 2010. *A new Metaheuristic Bat-Inspired Algorithm : Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)* (Eds. J.R. Gonzales et al.).Studies in Computational Intelligence, Springer Berlin, **284**, Springer, 65-74

Yang. X.-S. 2010. *Engineering Optimization : An Introduction with Metaheuristic Application*. Wiley and Son. University of Cambridge : United Kingdom.

Yang, X. S., 2011. *Bat Algorithm for Multiobjective Optimization*. Int. J. Bio-Inspired Com-putation, Vol. 3, No. 5, pp.267-274.

Yang, X. S., 2013. *Multiobjective cuckoo search for design optimization*. Int. J. Computers & Operations Research, Computers & Operations Research 40 (2013) 1616–1624.

Yuanbin, M. , Xinquan, Z. dan Shujian, X. 2013. *Local Memory Search Bat Algorithm for Grey Economic Dynamic System.* TELKOMNIKA, Vol. 11, No. 9, September 2013, pp. 4925~4934 e-ISSN: 2087-278X.

Biodata Penulis

Veri Julianto, Gunung Makmur, 1 Juli 1990. Penulis menempuh program sarjana di Jurusan Matematika, Fakultas MIPA Universitas Lambung Mangkurat dan menempuh Program Magister di Fakultas MIPA jurusan Sains Komputasi, Institut Teknologi Bandung (ITB). Keseharian penulis yaitu sebagai ketua jurusan dan dosen Teknik Informatika di Politeknik Negeri Tanah Laut**.**