KOMBINASI ALGORITMA *DIFFERENTIAL EVOLUTION DENGAN ITERATED GREEDY* UNTUK PERMASALAHAN *TRAVELING SALESMAN PROBLEM* (TSP)

Ong Andre Wahju Riyanto*

ABSTRAKSI

Penelitian ini ditujukan untuk memperbaiki kelemahan algoritma *differential evolution* (DE) yang sering menghasilkan solusi dini (konvergensi prematur) dalam menyelesaikan persoalan optimasi kombinatorial. Penelitian ini mengkombinasikan antara algoritma *differential evolution* (DE) dengan *iterated greedy* (IG) kemudian menerapkan kombinasi algoritma baru ini untuk menyelesaikan permasalahan *traveling* salesman problem (TSP). Inti kombinasi tetap didasarkan pada algoritma DE, sedangkan algoritma IG digunakan untuk memperbaiki hasil solusi algoritma DE. Eksperimen dilakukan dengan membandingkan kinerja kombinasi algoritma *differential evolution-iterated greedy* (DE-IG) dengan empat algoritma lain, yaitu: 1)*Particle swarm optimization-simulated annealing* (PSO-SA), 2)*Genetic algorithm (GA), 3)Simulated annealing (SA), dan 4)Ant colony optimization* (ACO) dalam rangka menyelesaikan permasalahan TSP. Hasil eksperimen menunjukkan bahwa hasil solusi menggunakan kombinasi algoritma *differential evolution* (DE) dengan *iterated greedy* (IG) lebih baik dibandingkan dengan empat algoritma tersebut.

Kata kunci: *differential evolution; iterated greedy; traveling salesman problem*; PSO;SA ;ACO

1. PENDAHULUAN

Permasalahan *traveling salesman problem* (TSP) telah dikaji secara intensif sejak beberapa dekade sebelumnya. *traveling salesman problem* (TSP) merupakan persoalan optimasi kombinatorial klasik dalam area riset operasi. Terdapat beberapa penggunaan praktis TSP, misalnya untuk persoalan *vehicle routing problem* (VRP) standar, maupun VRP yang mengandung pembatas,

* Ong Andre Wahju R. adalah dosen PNS Kopertis Wilayah VII Jawa Timur pada Univ. Wijaya Putra

seperti kapasitas kendaraan Laporte, G. (1992). Dalam bentuk jaringan, TSP bisa diformulasikan sebagai berikut: misalkan kita mempunyai graph G dengan N node/simpul dengan label 1, 2, ...,N. Simpul ini mewakili kota sedangkan garis penghubung mewakili sambungan antar kota. Setiap sambungan dari simpul i ke j mempunyai bobot atau ongkos c_{ij} yang mewakili panjang jalan atau jarak. Permasalahannya adalah menemukan rute atau perjalanan yang mengunjungi setiap kota hanya sekali (kecuali kota pertama) dengan jarak total minimum menurut Santosa, Budi dan Willy, Paul (2011). TSP selama ini telah digunakan sebagai persoalan dasar untuk perbandingan atas perbaikan beberapa teknik optimasi metaheuristik seperti *genetic algorithm* (GA), *simulated annealing* (SA), *ant colony optimization* (ACO) maupun *particle swarm optimization-simulated annealing* (PSO-SA).

Differential evolution (DE) adalah salah satu algoritma metaheuristik yang pemakaiannya cukup luas dalam bidang rekayasa. Differential evolution (DE) pertama kali diperkenalkan oleh Ken dan Storn di tahun 1995. Sejak diusulkan tahun 1995, Santosa, Budi dan Willy, Paul (2011) menyatakan DE mendapatkan reputasi yang bagus sebagai metode optimasi global yang efektif. Akhir-akhir ini beberapa peneliti banyak menggunakan algoritma DE untuk menyelesaikan persoalan-persoalan optimasi kombinatorial. Keunggulan algoritma DE adalah kecepatan dalam menemukan titik-titik solusi baru. Akan tetapi untuk persoalan-persoalan optimasi kombinatorial, kelemahan algoritma differential evolution adalah sering menghasilkan titik-titik solusi dini (konvergensi prematur), dimana deviasi hasil solusi algoritma DE masih relatif besar terhadap solusi optimal. Oleh karena itu Penelitian ini mengajukan suatu strategi mengkombinasikan algoritma DE dengan algoritma lain sebagai upaya memperbaiki kinerja algoritma DE. Algoritma yang dipilih adalah algoritma iterated greedy (IG). Algoritma IG telah terbukti efektif untuk persoalan optimasi kombinatorial.

Algoritm IG telah sukses diterapkan pada persoalan permutasi penjadwalan job flowshop. Penjadwalan job flowshop merupakan salah satu tipe persoalan optimasi kombinatorial. Ide utama algoritma IG adalah tahapan destruktif yang dilanjutkan dengan tahapan konstruktif. Dalam tahapan destruktif, dilakukan

dengan menghilangkan beberapa komponen solusi secara acak dari solusi sebelumnya. Sedangkan tahapan konstruktif, dilakukan dengan merekonstruksi ulang solusi dengan mekanisme greedy heuristic. Dalam Penelitian ini, peneliti mengkombinasikan antara algoritma DE dengan algoritma IG kemudian menerapkan untuk menyelesaikan permasalahan TSP.

Eksperimen dilakukan sebagai upaya membandingkan kinerja kombinasi algoritma DE-IG dengan empat algoritma metaheuristik lain, yaitu: 1)Simulated annealing (SA), 2) Genetic algorithm (GA), 3)Ant colony optimization (ACO) dan 4)Particle swarm optimization–simulated annealing (PSO-SA).

TUJUAN DAN KONTRIBUSI PENELITIAN

Tujuan Penelitian

Penerapan algoritma DE pada persoalan optimasi kombinatorial sering menghasilkan solusi, dimana deviasi hasil solusi algoritma DE masih relatif besar terhadap solusi optimal. Jadi tujuan penelitian ini adalah: 1)Memperbaiki kinerja hasil solusi penerapan algoritma DE pada persoalan TSP, 2)Membandingkan kinerja solusi antara kombinasi algoritma DE-IG dengan empat algoritma lain yang telah dikaji oleh peneliti lain.

Kontribusi Penelitian

Kontribusi penelitian adalah sebagai berikut: 1)Memperbaiki kelemahan algoritma DE untuk menyelesaikan persoalan optimasi kombinatorial, 2) Mengembangkan suatu kombinasi algoritma baru yang efektif untuk persoalan optimasi kombinatorial, dan 3)Menemukan kombinasi algoritma baru yang lebih baik dibandingkan dengan kombinasi algoritma-algoritma yang telah dikembangkan sebelumnya.

2. TINJAUAN PUSTAKA

A. Formulasi Matematis Permasalahan TSP

Formulasi matematis permasalahan TSP adalah sebagai berikut:

Minimasi $z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij}$, (1)

Dibatasi oleh:

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ , j = 1,2,..., n,} \quad (2)$$

$$\sum_{j=1}^{n} x_{ij} = 1, i=1,2,\ldots,n \text{ ,} \quad (3)$$

$$x_{ij} \in \{0,1\} \text{ , i,j = 1,2…, n} \quad (4)$$

$\tilde{x}$ adalah bentuk Hamiltonian cycle (5)

Dimana, c_{ij} adalah indeks biaya dan x_{ij} adalah variabel keputusan yang berkaitan dengan penugasan elemen i ke posisi j. Jika x_{ij} = 1, maka elemen i dihubungkan dengan elemen j, yang berarti terbentuk rute dari kota i ke kota j. Fungsi tujuan z pada persamaan (1) merepresentasikan biaya total yang harus diminimasi. Pembatas pada persamaan (2) memastikan bahwa setiap posisi j diisi hanya oleh satu kota. Pembatas (3) menjamin bahwa setiap kota i ditugaskan tepat pada satu posisi. Pembatas (4) memastikan bahwa x_{ij} merupakan anggota dari himpunan biner, 0 atau 1. Pembatas (5) memastikan bahwa rute dari setiap kota hanya dikunjungi sekali.

B. Algoritma Differential Evolution (DE)

Algoritma DE pertama kali diperkenalkan oleh Storn dan Price di tahun 1995 menurut Santosa, Budi dan Willy, Paul (2011), untuk menyelesaikan permasalahan optimasi kontinyu.Algoritma DE didasarkan pada solusi berbasis populasi. Algoritma DE diawali dengan pembangkitan inisialisasi populasi individu solusi di ruang pencarian solusi. DE mampu menemukan solusi global optima dengan menggunakan informasi jarak dan arah menurut perbedaan diantara populasi. Di setiap generasi, operator mutasi dan kawin-silang

diaplikasikan terhadap individu-individu solusi untuk membangkitkan populasi individu baru. Kemudian dilakukan seleksi untuk memilih individu-individu solusi yang terbaik. Individu terbaik akan menjadi individu populasi yang baru. Parameter kunci pada algoritma DE adalah ukuran populasi, fraksi mutan dan tingkat kawin-silang (*crossove*r). Saat ini terdapat beberapa varian DE, format DE yang umum adalah DE/x/y/z. Dimana x adalah vektor dasar yang dipertubasi, y adalah jumlah vektor yang berbeda yang digunakan untuk pertubasi vektor x, dan z adalah tipe operator kawin-silang (*crossover)* yang digunakan (tipe bin: *binomial* atau tipe exp:*exponential*).

Langkah 0: menentukan ukuran populasi yang dinotasikan dengan PS, menentukan fraksi mutan (F), menentukan tingkat kawin-silang (CR), menentukan jumlah generasi atau iterasi maksimum (K_{max}).

Let population size be PS and ith individual in the N-dimensional search space at generation k be $X_i(k) = [x_{i1}, x_{i2}, ... , x_{iN}]$, where i=1,2,...,PS. For a minimization problem, DE's basic procedure, which is denoted as DE/rand/1/bin, can be given below:

Langkah 1: inisialisasi populasi awal, yang dibangkitkan melalui persamaan berikut ini:

$$x_{i,j}(0)=\text{rand}(0,1)*(x_{i,jmax}-x_{i,jmin})+x_{i,jmin} \quad j=1,2,...,N \qquad (6)$$

Langkah 2: Mengevaluasi nilai tujuan pada seluruh individu populasi, $X_i(k)$, i=1, 2,...,PS. Individu terbaik adalah individu yang memiliki nilai tujuan yang paling minimum pada populasi di generasi ke-k. Kemudian tetapkan k=1.

Langkah 3: Fase mutasi. Setiap individu $X_i(k)$, dibangkitkan suatu vektor mutan $V_i(k+1) = [v_{i,1}(k+1),...,v_{i,N}(k+1)]$ melalui persamaan berikut ini:

$$V_i(k+1)=X_{r1}(k)+F*(X_{r2}(k)-X_{r3}(k)) \qquad (7)$$

dimana, r1, r2, r3 $\in$ {1, 2,...., PS}. Individu r1, r2 dan r3 dipilih secara acak dari populasi $X_i(k)$ dengan $i \neq r1 \neq r2 \neq r3$. $F \in [0,1]$.

Langkah 4: Fase kawin-silang. Untuk setiap populasi target, $X_i(k)$, suatu vektor percobaan $U_i(k+1)=[u_{i,1}(k+1),...,u_{i,N}(k+1)]$ dibangkitkan melalui persamaan berikut ini:

$$u_{i,j}(k+1) = \begin{cases} v_{i,j}(k+1) & \text{if } (rand(j) \leqslant CR) \text{ or } j = randn(i), \\ x_{i,j}(k) & \text{otherwise,} \end{cases} \quad j = 1, \ldots, N, \quad (18)$$

dimana rand(j) adalah adalah angka acak berdistribusi seragam dalam interval [0,1] untuk dimensi ke-j. Randn(i) adalah indeks dari himpunan {1, 2,...., N} yang diambil secara acak. CR $\in$ [0,1] adalah konstanta yang mengendalikan diversitas populasi.

Langkah 5: Fase seleksi. Untuk setiap individu di populasi target $X_i(k+1)$, individu populasi percobaan $U_i(k+1)$ dibandingkan dengan individu populasi target $X_i(k)$ dengan menggunakan kriteria seleksi turnamen berikut ini:

$$X_i(k+1) = \begin{cases} U_i(k+1) & \text{if } f(U_i(k+1)) < f(X_i(k)), \\ X_i(k) & \text{otherwise,} \end{cases} \quad (19)$$

Dimana f adalah fungsi tujuan dan $X_i(k+1)$ adalah individu populasi yang baru.

Langkah 6: Memperbaharui nilai tujuan terbaik di iterasi ke-k. Kemudian tetapkan k = k + 1.

Langkah-7: Kriteria pemberhentian iterasi atau generasi. Jika $k < K_{max}$, maka langkah kembali ke langkah 3, fase mutasi. Jika sebaliknya proses iterasi berhenti, selanjutnya menampilkan nilai tujuan terbaik yang diperoleh di iterasi K_{max}.

C. Algoritma Iterated Greedy (IG)

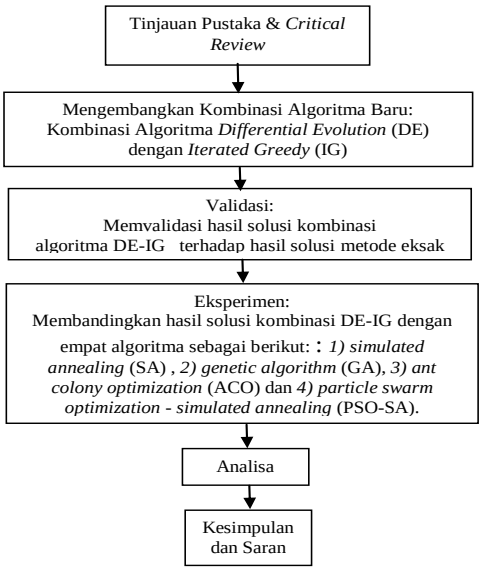
Algoritma IG telah berhasil diterapkan untuk permasalahan permutasi penjadwalan *job flowshop*. Ide utama algoritma IG adalah tahapan destruktif yang dilanjutkan dengan tahapan konstruktif. Dalam tahapan destruktif, dilakukan dengan menghilangkan beberapa komponen solusi secara acak dari solusi sebelumnya. Sedangkan tahapan konstruktif, dilakukan dengan merekonstruksi ulang solusi dengan mekanisme greedy heuristic.

Prosedur kunci algoritma IG adalah destruktif dan konstruktif. Untuk permasalahan TSP dengan n kota misalnnya terdapat 5 kota dengan individu solusi rute perjalanan 1-3-4-2-5-1. Langkah desktruktif dilakukan dengan

mengambil secara acak sejumlah d (d=2) kota dari individu solusi, sehingga rute parsial dengan 2 (n−d−1) kota yang didapatkan dinotasikan sebagai π_D sekaligus d kota dinotasikan sebagai π_R untuk diselipkan ulang kedalam π_D. Fase konstruksi dilakukan dengan cara kota pertama pada himpunan π_R diselipkan ulang kedalam seluruh kemungkinan n−d slot di himpunan π_D. Diantara sejumlah n−d selipan, selipan yang menghasilkan nilai jarak paling minimum dipilih sebagai rute parsial untuk selipan berikutnya. Kemudian kota kedua dari himpunan π_R dipertimbangkan dan seterusnya sampai himpunan π_R kosong.

3. METODE PENELITIAN

Penelitian ini mengembangkan kombinasi algoritma differential evolution (DE) dengan *iterated greedy* (IG) yang diharapkan mampu memperbaiki kelemahan algoritma DE dalam menyelesaikan permasalahan TSP. Alur kerangka kerja penelitian ditunjukkan pada Gambar 1di bawah ini.



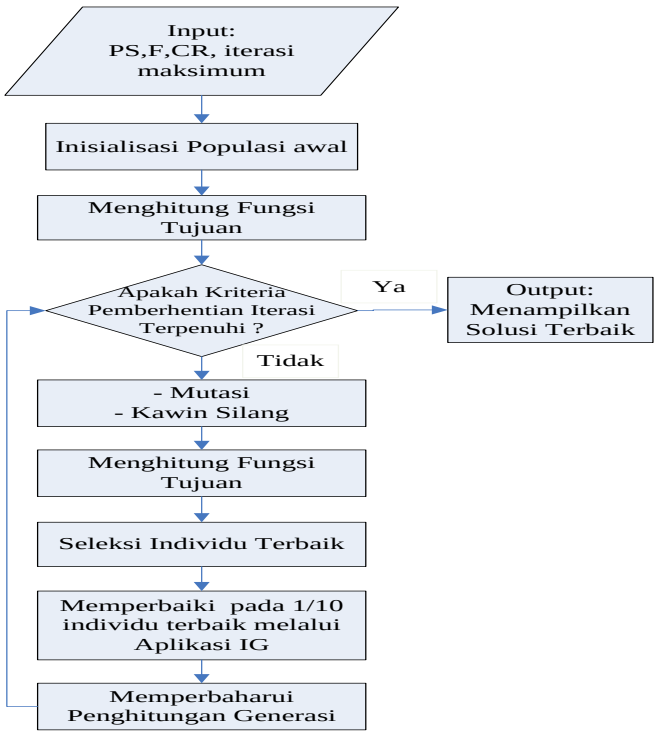
Gambar 1. Alur Kerangka Kerja Penelitian

Tinjauan pustaka maupun *critical review* dilakukan pada dua aspek: pertama, menetapkan permasalahan TSP menjadi model matematis. Kedua, mempelajari beberapa metode metaheuristik yang telah digunakan oleh peneliti sebelumnya untuk menyelesaikan permasalahan TSP.

Menyusun langkah-langkah algoritma DE maupun langkah-langkah algoritma IG. Kemudian mengkombinasikan antara algoritma DE dengan algoritma IG dengan skema kombinasi algoritma DE-IG yang ditunjukkan oleh Gambar 2 di bawah.

Validasi hasil solusi algoritma DE-IG terhadap solusi optimal dengan jumlah 7 kota dilakukan untuk menilai performansi kombinasi algoritma DE-IG. Ukuran problem 7 kota dipilih dengan pertimbangan ukuran problem yang bisa ditemukan seluruh kombinasi rute kota yang mungkin tanpa terkendala waktu komputasi yang sangat lama.

Upaya eksperimen dilakukan untuk mengetahui kinerja kombinasi algoritma DE-IG dalam menyelesaikan permasalahan TSP terhadap empat algoritma lain. Hasil solusi hasil solusi kombinasi algoritma DE-IG dibandingkan dengan kombinasi algoritma PSO-SA oleh Fang, dkk



Gambar 2. Skema Kombinasi DE-IG

4. ANALISA HASIL EKSPERIMEN

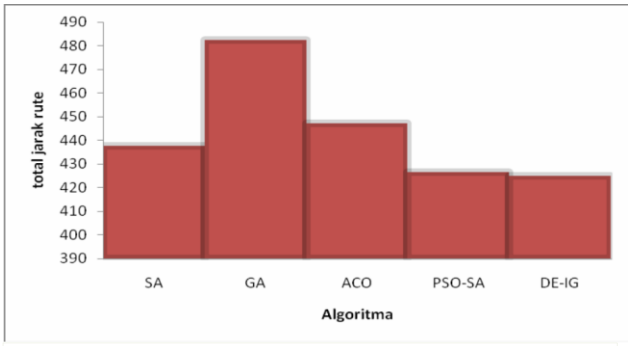
Untuk perbandingan, peneliti membandingkan hasil solusi kombinasi algoritma DE-IG dengan hasil penelitian Fang, dkk [3] yang mengembangkan kombinasi algoritma PSO-SA untuk menyelesaikan pemasalahan TSP (kasus: Oliver30 dan Att48). Konfigurasi parameter untuk kombinasi algoritma baru DE-IG yang digunakan oleh peneliti adalah sebagai berikut: a) DE: PS=500, skala F=0.5, CR=0.7, maksimum generasi=3000, b)IG:maksimum iterasi=1000. Kombinasi algoritma DE-IG diterjemahkan ke dalam perangkat lunak program komputasi MATLAB 7 dan dieksekusi pada PC *Pentium Dual Core* RAM 1 GB. Eksperimen replikasi dilakukan sebanyak 20 kali baik untuk kasus Oliver30 maupun kasus Att48. Hasil eksperimen ditunjukkan di Tabel 1, Tabel 2, Gambar 3 dan Gambar 4.

Tabel 1. Hasil untuk kasus Oliver30

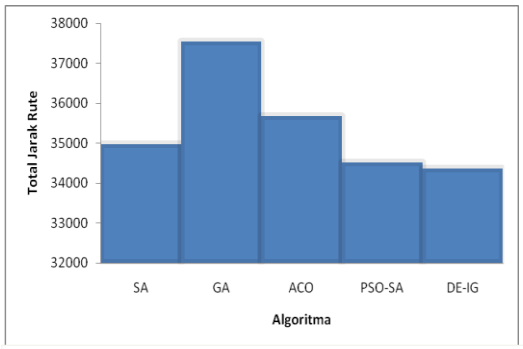
No	Algoritma	Oliver30			Penelitian oleh:
		solusi rata-rata	solusi terbaik	solusi terburuk	
1	SA	437,6632	424,9918	480,1452	Fang et,al (2007)
2	GA	482,4671	468,1532	504,5256	
3	ACO	447,3256	440,8645	502,3694	
4	PSO-SA	426,6856	423,7405	430,7122	
5	DE-IG	424,8692	424,8692	424,8692	Ong Andre (2012)

Tabel 2. Hasil untuk kasus Att48

No	Algoritma	Att48			Penelitian oleh:
		solusi rata-rata	solusi terbaik	solusi terburuk	
1	SA	34980	35745	41864	Fang et,al (2007)
2	GA	37548	36759	43561	
3	ACO	35684	34559	42256	
4	PSO-SA	34512	33966	35101	
5	DE-IG	34371	33936	34986	Ong Andre (2010)



Gambar 3. Grafik Rata-rata Jarak Rute Problem Oliver30



Gambar 4. Grafik Rata-rata Jarak Rute Problem Att48

Dari hasil eksperimen , seperti yang terlihat di Tabel 1, Tabel 2, Gambar 3 dan Gambar 4 terlihat bahwa kombinasi algoritma DE-IG memberikan hasil solusi yang lebih baik dibandingkan dengan hasil penelitian Fang, dkk [3] yang menggunakan kombinasi *algoritma* PSO-SA. Untuk persoalan Oliver30 (Tabel 1) nilai solusi rata-rata dan nilai solusi terburuk yang paling minimum mampu dicapai oleh kombinasi algoritma DE-IG, walaupun untuk nilai solusi terbaik lebih besar 0,27% dibandingkan solusi terbaik yang mampu dicapai kombinasi algoritma PSO-SA. Untuk persoalan Att48 (Tabel 2) nilai solusi rata-rata, nilai solusi terbaik maupun nilai solusi terburuk yang paling minimum mampu dicapai oleh kombinasi algoritma DE-IG.

Hal ini menunjukkan bahwa kombinasi DE-IG merupakan kombinasi algoritma yang efektif untuk menyelesaikan permasalahan traveling salesman problem (TSP).

5. SIMPULAN

Algoritma IG mampu memperbaiki hasil solusi algoritma DE dalam penyelesaian permasalahan TSP Kombinasi algoritma DE-IG mampu memberikan kinerja yang lebih baik dibandingkan dengan empat algoritma lain (SA, GA, ACO, PSO-SA) dalam penyelesaian permasalahan TSP. Oleh karena itu algoritma DE-IG dapat dikatakan sebagai algoritma yang kompetitif dalam menyelesaikan permasalahan TSP.

Usulan penelitian lanjutan akan diarahkan untuk menyelesaikan permasalahan TSP pada ukuran kasus yang lebih besar. Algoritma DE-IG bisa digunakan untuk menyelesaikan permasalahan-permasalahan optimasi kombinatorial yang lain.

DAFTAR PUSTAKA

Fang, L., Chen, P., Liu, S., "Particle Swarm Optimization with Simulated Annealing for TSP", Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, February 16-19, 2007.

Laporte, G., " The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms", Eur. J. Oper. Res. 59 (2), 345–358, 1992.

Ruiz, R., Stutzle, T., "A Simple and Effective Iterated Greedy Algorithm for the Permutation Flowshop Scheduling Problem", European Journal of Operational Research, 177, 2033-2049, 2007.

Ruiz, R., Stutzle, T., "A Simple and Effective Iterated Greedy Algorithm for the Permutation Flowshop Scheduling Problem", European Journal of Operational Research, 177, 2033-2049, 2007.

Santosa, Budi dan Willy, Paul., "Metoda Metaheuristik: Konsep dan Implementasi", edisi pertama, Penerbit Guna Widya, April 2011.

Storn, R., Price, K., "Differential Evolution-Apractical Approach to Global Optimization", Springer- Verlag, 2006.

Tasgetiren, Fatih, M., Pan, Q., Liang C.Y., "Discrete Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem", GECCO'07, London, England, United Kingdom, 2007