

Test Data Algoritma Kromosom Pada Sidik Jari Menggunakan Jaringan Syaraf

Melyani

Universitas Bina Sarana Informatika

Jalan Dewi Sartika No.8 , Cawang – Jakarta Timur, Indonesia

Email ; * melyani.myn@bsi.ac.id

Abstract Biometric features that can be used for identification include iris, voice, DNA and fingerprints. Fingerprints are the most widely used biometric feature because of their uniqueness, universality and stability. Fingerprint recognition can be grouped into two different forms of problems, namely verification and identification. Verification is comparing one fingerprint with another fingerprint. Meanwhile, identification is matching an input fingerprint with fingerprint data in the database. Thus, identification can be interpreted as an extension of verification carried out by comparing one fingerprint to many fingerprints. Identification is inherently more complex than verification. The problem increases as the number of fingerprint datasets increases, resulting in an increase in the time required for the identification process. However, there is a way to overcome this complexity, namely classification. Apart from that, evolutionary algorithm optimization can also be carried out. The Chromosome Algorithm is an improvement on the evolutionary algorithm with a separate local search process. The memetic or chromosomal algorithm is a simple algorithm with reliable performance that can provide accurate solutions to problems in the real world. The current challenge is with the increasing growth of datasets (more than 106) which include the process of clustering text analysis, molecular DNA simulation, feature selection, and forecasting, handling large-scale optimization such as complex simulations, data mining, quantum chemistry, spectroscopic analysis, geophysical analysis, drug discovery, and fingerprint recognition studies. Chromosome algorithms have proven to be very competitive in large-scale optimization because they are based on stochastic algorithms that do not require gradient information.

Keywords: Neural Network Data Test, Fingerprint, Chromosome Algorithm

Abstrak Fitur biometric yang dapat digunakan untuk identifikasi diantaranya adalah iris mata, suara, DNA dan sidik jari. Sidik jari merupakan fitur biometric yang paling banyak digunakan karena keunikan, universal dan stabilitasnya. Pengenalan sidik jari dapat dikelompokkan dalam dua bentuk masalah yang berbeda yaitu verifikasi dan identifikasi. Verifikasi adalah membandingkan satu sidik jari dengan satu sidik jari yang lain. Sedangkan identifikasi adalah mencocokkan satu sidik jari inputan dengan data sidik jari yang ada didalam database. Dengan demikian, identifikasi dapat diartikan sebagai perluasan dari verifikasi yang dilakukan dengan membandingkan satu sidik jari ke banyak sidik jari. Identifikasi pada dasarnya lebih kompleks dari verifikasi. Masalah bertambah seiring dengan banyaknya dataset sidik jari yang mengakibatkan semakin besarnya waktu yang diperlukan untuk proses identifikasi. Namun ada cara untuk dapat mengatasi kompleksitas tersebut yaitu klasifikasi selain itu juga dapat dilakukan optimasi algoritma evolusi Algoritma Kromosom merupakan peningkatan dari algoritma evolusi dengan proses pencarian lokal yang terpisah. Algoritma memetika atau kromosom termasuk algoritma yang sederhana dengan performa yang dapat diandalkan dapat memberikan solusi akurat untuk mengatasi permasalahan di dunia nyata. Tantangan saat ini adalah dengan pertumbuhan dataset yang semakin meningkat (lebih dari 106) yang diantaranya ada pada proses peng-klausteran analisis teks, simulasi DNA molekuler, seleksi fitur, dan peramalan, penanganan optimasi skala besar seperti simulasi kompleks, data mining, kimia kuantum, analisis spektroskopi, analisis geofisika, penemuan obat, serta studi pengenalan sidik jari. Algoritma kromosom telah terbukti sangat kompetitif dalam optimasi skala besar karena algoritma ini didasarkan pada algoritma stokastik yang tidak memerlukan informasi gradient.

Kata Kunci: Test Data Jaringan Syaraf , Sidik Jari , Algoritma Kromosom

PENDAHULUAN

Identifikasi personal menjadi salah satu masalah yang penting dalam masyarakat terkait dengan kendali akses, kriminalitas dan identifikasi forensik, perbankan maupun sistem komputer (Jain, Bolle & Pakanti, 2005). Fitur biometric yang dapat digunakan untuk identifikasi diantaranya adalah iris mata, suara, DNA dan sidik jari. Berdasarkan penelitian

Received Desember 12, 2023; Accepted Januari 27, 2024; Published Februari 28, 2024

* Melyani, melyani.myn@bsi.ac.id

(Maltoni, Maio, Jain & Prabhakar, 2009) sidik jari merupakan fitur biometric yang paling banyak digunakan karena keunikan, universal dan stabilitasnya. Sidik jari banyak digunakan sebagai fitur keamanan untuk pengenalan forensic, akses gedung, otentikasi ATM atau pembayaran. Selain itu, menurut (Pakanti, Prabhakar & Jain, 2002) sidik jari banyak digunakan karena keunikan, ukuran dan ke-khas-annya.

Identifikasi sidik jari secara otomatis telah menjadi topik penelitian yang menarik dalam dua dekade terakhir (Jain & Feng, 2011). Alat pengenalan sidik jari sangat mudah didapatkan sekarang ini, atas dasar tersebut banyak perusahaan dan lembaga yang menggunakannya, seiring dengan bertambahnya jumlah orang yang harus diidentifikasi secara personal (Jain, Bolle & Pakanti, 2005).

Pengenalan sidik jari dapat dikelompokkan dalam dua bentuk masalah yang berbeda yaitu verifikasi (Jain, Hong & Bolle, 1997) dan identifikasi (Jain, Hong, Pakanti & Bolle, 1997). Verifikasi adalah membandingkan satu sidik jari dengan satu sidik jari yang lain. Sedangkan identifikasi adalah mencocokkan satu sidik jari inputan dengan data sidik jari yang ada didalam database. Dengan demikian, identifikasi dapat diartikan sebagai perluasan dari verifikasi yang dilakukan dengan membandingkan satu sidik jari ke banyak sidik jari.

Identifikasi pada dasarnya lebih kompleks dari verifikasi. Masalah bertambah seiring dengan banyaknya dataset sidk jari yang mengakibatkan semakin besarnya waktu yang diperlukan untuk proses identifikasi. Namun ada cara untuk dapat mengatasi kompleksitas tersebut yaitu klasifikasi (Galar, Derrac, Peralta, Triguero, Paternain, Lopez-Molina, Garc'ia, Benitez, Pagola, Barrenechea, Bustince & Herrera, 2015) dan peng-indeks-an (Wang, Wang, Cheung & Yuen, 2015), selain itu juga dapat dilakukan optimasi algoritma evolusi (Bäck, Fogel & Michalewicz, 1997) dengan memadukannya bersama metode pencarian lokal. Algoritma evolusi merupakan algoritma yang sangat baik digunakan untuk masalah optimasi (Lastraa, Molinab & Benítez, 2015).

Algoritma memetika (Moscato, 1989); (Moscato, 1999) merupakan paningkatan dari algoritma evolusi dengan proses pencarian lokal yang terpisah (Lin & Chen, 2011). Algoritma memetika termasuk algoritma yang sederhana dengan performa yang dapat diandalkan (Merz & Freisleben, 1999); (Ong, Lim, Zhu & Wong, 2006), dapat memberikan solusi akurat untuk mengatasi permasalahan di dunia nyata (Caponio, Cascella, Neri, Salvatore & Sumner, 2007); (Jiao, Gong, Wang, Hou, Zheng & Wu, 2010); (Urselmann, Barkmann, Sand & Engell, 2011). Tantangan saat ini adalah dengan pertumbuhan dataset yang semakin meningkat (lebih dari 106) yang diantaranya ada pada proses peng-klasteran analisis teks (Lu, Wang, Li & Zhou, 2009); (Bai, Liang, Dang & Cao, 2011), simulasi DNA molekuler (Lang, Drouvelis, Tafaj,

Bastian & Sakmann, 2011); (Zhao, Sheong, Sun, Sander & Huang, 2013); (Bahmann & Kortus, 2013), seleksi fitur (Hong & Cho, 2006), dan peramalan (Niu, Wang & Dash Wu, 2010), penanganan optimasi skala besar seperti simulasi kompleks (Chen, Fujishiro & Nakajima, 2002), data mining (Agarwal, Merugu, 2007), kimia kuantum (Shao et.al, 2006), analisis spektroskopi (Roy & Gerber, 2013), analisis geofisika (Blum, Dimet & Navon, 2009), penemuan obat (Dudley, Schadt, Sirota, Butte & Ashley, 2010), serta studi genom (Shi, Wahba, Irizarry, Bravo & Wright, 2012). Algoritma memetika telah terbukti sangat kompetitif dalam optimasi skala besar karena algoritma ini didasarkan pada algoritma stokastik yang tidak memerlukan informasi gradient. Dengan menggunakan metode pencarian lokal kinerja algoritma dapat ditingkatkan secara cepat. Kecepatan kinerja ini akan sangat berguna pada optimasi skala besar (Lastraa, Molinab & Benítez, 2015). Pada mulanya, para peneliti berusaha menggunakan algoritma evolusi sebagai solusi untuk klasifikasi, pengenalan pola dan pengolahan gambar (Coello, 2006). Kemudian digunakanlah algoritma genetika yang dapat mendeteksi dan menyesuaikan setiap rotasi gambar dan mengembalikannya ke bentuk aslinya atau posisi awal. Dengan algoritma genetika para peneliti dapat mengambil sudut rotasi dengan akurasi yang hampir 100% (Ibrahiem & El-Kareem, 2008). Algoritma genetika dapat diterapkan pada skala besar dan data berdimensi tinggi seperti pengenalan wajah, pengenalan sidik jari dan pengenalan iris (Jayaram & Fleyeh, 2013).

Dua algoritma evolusi yaitu Algoritma Memetika (MA) dan Algoritma Genetika (GA), telah banyak digunakan untuk sistem pengenalan wajah (Kumar, Kumar & Rai, 2009). Kedua algoritma ini digunakan untuk mengurangi dimensi waktu dan meningkatkan pengenalan. Hasil penelitian menunjukkan algoritma ini mengungguli sistem pengenalan dengan metode konvensional. Hasil penelitian juga menunjukkan Algoritma Memetika lebih baik dari Algoritma Genetika (Cordon, Herrera & Lozano, 1995). Untuk mempersingkat waktu proses perhitungan dan meningkatkan kualitas hasil yang diperoleh algoritma memetika dapat diparalelkan (Bozejko & Wodecki, 2011). Berbagai algoritma optimasi yang diusulkan biasanya terkendala pada waktu penyelesaian yang lambat atau memori computer yang menjadi terbatas pada saat proses komputasi. Solusi untuk masalah tersebut adalah dengan mengembangkan prosedur algoritma heuristic, seperti algoritma memetika. Namun masalah baru muncul karena biasanya terkendala pada prasyarat komputasi yang tinggi sehingga dibutuhkan sistem komputasi berkinerja tinggi (HPC). Algoritma kromosom atau memetika adalah perluasan dari algoritma genetika. Algoritma genetika bekerja secara parallel dengan melakukan pencarian menggunakan banyak individu, sehingga algoritma genetika nyaris tidak mungkin terjebak dalam ekstrem local. Namun algoritma genetika harus melakukan verifikasi

kelayakan solusi pada setiap iterasi sehingga akan menambah waktu komputasi, artinya kinerja algoritma ini lebih lambat jika dibandingkan metode lain. Kekurangan algoritma genetika ini dapat di atasi dengan menambah fitur pencarian local yang selanjutnya dikenal dengan algoritma memetika. Algoritma memetika merupakan metode pencarian heuristic hasil penggabungan algoritma genetika dengan metode pencarian local yang dapat meningkatkan kualitas pencarian solusi (Moscato, 1989). Fitur pencarian local pada algoritma memetika dapat diterapkan sebelum atau sesudah proses seleksi, crossover dan mutasi selain berguna juga untuk memperkecil ruang pencarian. Algoritma memetika memberikan solusi yang lebih baik dari algoritma genetika namun waktu komputasi lebih lama atau lebih lambat (Ali, 2013). Menurut Kieiarova (2017), waktu komputasi algoritma memetika rata-rata sekitar 5.3 detik dan algoritma genetika sekitar 1.9 detik. Algoritma memetika memerlukan waktu komputasi 5 kali lebih lama daripada algoritma genetika karena algoritma memetika melakukan dua set pencarian lokal pada saat proses evolusi.

Pada sistem High Performance Computing (HPC) atau komputasi berkinerja tinggi, untuk mendapatkan waktu proses yang efisien, ada dua cara yang biasa dilakukan yaitu dengan membuat prosesor yang kencang dan dengan menghitungnya secara paralel dengan multi-prosesor. Jika cara pertama dilakukan, perusahaan elektronik pembuat prosesor harus mengurangi bagian-bagian elektronik yang akan menjadikan produknya lebih kecil dan sinyal yang dihasilkan menjadi lebih pendek. Teknologi semikonduktor, saat ini menggunakan teknik litografi yang sudah hampir mencapai batasnya. Chip prosesor terbaru dibuat dengan teknologi fabrikasi 45nm dan jika dikurangi lagi kemungkinan kesalahan proses manufaktur menjadi lebih tinggi juga keandalannya akan berkurang. Oleh karena itu, peluang untuk meningkatkan kecepatan komputasi yang masih memungkinkan adalah dengan cara komputasi paralel (Assiroj, Hananto, Fauzi & Warnars, 2018).

1. Biometrik

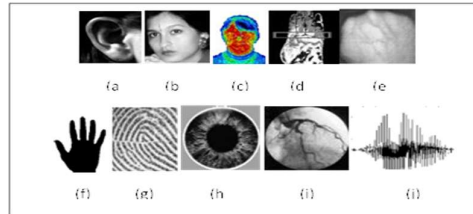
Sistem biometric dikelompokkan dalam dua bentuk yang berbeda yaitu verifikasi (Jain, Hong & Bolle, 1997) dan identifikasi (Jain, Hong, Pakanti & Bolle, 1997). Verifikasi adalah membandingkan karakteristik biometric inputan yang ditangkap oleh alat dengan template yang sudah disimpan dalam system. Artinya melakukan perbandingan satu ke satu. System verifikasi dapat menolak atau menerima data inputan. Sedangkan system identifikasi akan mengenali seseorang dengan cara mencari kecocokan dari seluruh data yang tersimpan, melakukan perbandingan satu ke banyak. Dalam sistem identifikasi, sistem menetapkan identitas subjek (atau gagal jika subjek tidak terdaftar dalam database sistem) tanpa subjek harus mengklaim identitas. Fitur fisiologi manusia dapat digunakan sebagai cara identifikasi

biometric jika memenuhi persyaratan berikut:

- Universalitas, artinya setiap orang harus memiliki fitur biometric;
- Khas, artinya setiap orang mempunyai ciri fitur biometric yang berbeda;
- Permanen, artinya tetap atau invariant. Keadaan setiap ciri fitur biometric harus tetap selama periode waktu tertentu.

Beberapa contoh fitur biometric yang dapat diidentifikasi terlihat pada gambar 1

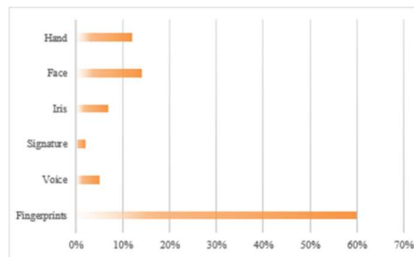
Gambar 1 : Fitur biometric



Sumber: Jain, A.K., Bolle, R.M., Pankanti, S. Biometrics: Personal Identification in Networked Society. Springer International (2005)

Pengenalan fitur biometric sidik jari adalah salah satu yang paling handal dan cocok untuk aplikasi skala besar. Penggunaannya sudah meluas sampai pada perangkat bergerak seperti telepon pintar saat ini. Pada tahun 2012, pendapatan industri yang berkaitan dengan penggunaan fitur biometric ini mencapai 60% dari total semua fitur biometric. Pengenalan wajah ada di urutan kedua dengan 14% seperti terlihat pada gambar 1 berikut:

Gambar 2. Fitur International Biometric



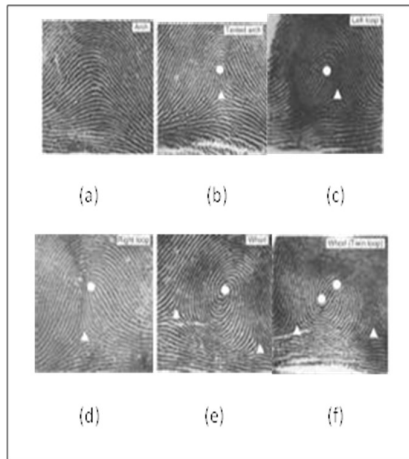
Sumber: Perbandingan pendapatan produk dengan fitur biometric International Biometric Group, (2012)

2. Sidik Jari (Fingerprints)

Penemuan ini merupakan dasar dari pengenalan sidik jari modern. Pada akhir abad ke sembilan belas, Sir Francis Galton meneliti lebih lanjut tentang sidik jari. Galton memperkenalkan fitur-fitur kecil untuk mencocokkan sidik jari tahun 1888.

Tingkat kemandirian terpenting dalam pengenalan sidik jari terjadi pada tahun 1899, dimana Edward Henry membuat "System Henry" yang terkenal dengan klasifikasi sidik jarinya seperti terlihat pada gambar 1.3 Sehingga pada awal abad ke dua puluh, bentuk sidik jari sudah dapat dipahami dengan baik.

Gambar 3 Klasifikasi Sidik Jari



Sumber : Le, H.H., Nguyen, N.H., Nguyen, T.T. Exploiting GPU for large scale fingerprint identification (2016)

Kebanyakan penulis membagi masalah pengenalan sidik jari menjadi dua varian yang merupakan masalah berbeda-beda:

- a. Verifikasi terdiri dari menentukan apakah dua sidik jari gambar P1 dan P2 milik orang yang sama, melakukan perbandingan 1:1 Output sistem adalah penerimaan atau penolakan identitas yang diklaim tergantung pada kesamaan level (disebut skor) dari kedua sidik jari.
- b. Identifikasi bertujuan untuk menemukan sidik jari yang cocok dengan input sidik jari dalam database, sehingga pemiliknya bisa diidentifikasi. Database sidik jari adalah seperangkat template T of N sidik jari $T = \{T_1; T_2; \dots; T_N\}$ yang digunakan sebagai referensi untuk identifikasi. Jadi, identifikasi adalah masalah perbandingan 1: N karena sidik jari input I perlu dibandingkan dengan semua T_i templat sidik jari (dengan $i = 1; 2; \dots; N$) untuk menemukan kecocokan yang memberikan skor tertinggi. Skor ini disebut m_{best} . Ini didefinisikan dalam Persamaan. (1), di mana $Q(I; T_i)$ adalah fungsi yang cocok (lihat Bagian 2.3). Jika m_{best} lebih rendah dari ambang tertentu ϕ , maka sistem dapat mempertimbangkan bahwa sidik jari input tidak memiliki template yang sesuai dalam database. Oleh karena itu, output system dapat berupa identitas yang cocok, pemberitahuan "tidak ditemukan", atau satu set identitas kandidat. Makalah ini berfokus pada sistem itu hanya mempertimbangkan skor maksimum, jadi kasus terakhir tidak rinci, seperti yang ditunjukkan pada Persamaan. (2)

$$= \max \{Q(I, T_i) | i \in \{1, 2, \dots, N\}\} \quad (1)$$

$$Q(I, T_i) = \{ \dots, > \phi \}$$

$$(i, j) \in \{1, \dots, N\}, h$$

3. Algoritma Genetika

Algoritma Genetika merupakan algoritma evolusioner untuk memecahkan masalah optimisasi berdasarkan pada ide-ide evolusi seleksi alam (Sahid, 1997). Algoritma ini mensimulasikan proses evolusi dengan menghasilkan populasi solusi layak dan menerapkan beberapa operator genetik untuk menghasilkan populasi baru berdasarkan aturan seleksi. Ide mengenai algoritma evolusioner diperkenalkan pertama kalinya oleh Rechenberg pada tahun 1973 (Kralev, 2009). Pada awal tahun 1992 pemrograman mengenai genetika mulai diperkenalkan oleh

J.H. Koza (Kralev, 2009). Algoritma genetika menggunakan analogi secara langsung dari kebiasaan yang alami yaitu seleksi alam. Dalam ilmu biologi, sekumpulan individu yang sama, hidup dan berkembang bersama dalam suatu area, disebut dengan populasi. Algoritma genetika bekerja dengan suatu populasi yang terdiri atas individu-individu, yang masing-masing individu mempresentasikan solusi yang mungkin bagi suatu permasalahan. Algoritma genetika memiliki performansi yang baik untuk masalah- masalah selain optimisasi kompleks dari satu variabel atau multivariabel. Salah satu aplikasi algoritma genetika adalah pada permasalahan optimisasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi, seperti optimisasi penjadwalan mata kuliah. Pada algoritma genetika, kandidat solusi (yang selanjutnya disebut 'solusi') dari suatu masalah direpresentasikan sebagai kromosom. Kromosom ini merupakan representasi dari solusi yang ada. Kumpulan kromosom disebut sebagai populasi. Masing-masing kromosom pada populasi akan dievaluasi menggunakan fungsi fitness, yaitu fungsi yang mengukur secara kuantitatif suatu kromosom untuk bertahan pada populasi. Pembentukan populasi baru dilakukan dengan menerapkan operator-operator genetika secara iteratif sampai dipenuhi kriteria berhenti, misalnya diperoleh populasi yang baik dari populasi sebelumnya.

4. Algoritma Kromosom

Algoritma Memetika atau Kromosom adalah perluasan dari Algoritma Genetika. Algoritma Genetika (AG) merupakan salah satu metode heuristik yang banyak digunakan untuk menyelesaikan masalah optimisasi kombinatorial yang sulit. Konsep yang penting dalam AG adalah hereditas dan seleksi alam. Hereditas merupakan sebuah ide bahwa sifat-sifat itu dapat diturunkan kepada generasi berikutnya. Sifat-sifat individu tersebut dikodekan kedalam sekumpulan gen-gen yang disebut kromosom. Algoritma kromosom merepresentasikan salah satu area riset yang sedang berkembang pada bidang komputasi evolusioner. Istilah algoritma memetika secara luas telah digunakan untuk menggambarkan peningkatan prosedur dalam masalah pencarian baik yang dilakukan dengan pendekatan individu maupun pendekatan

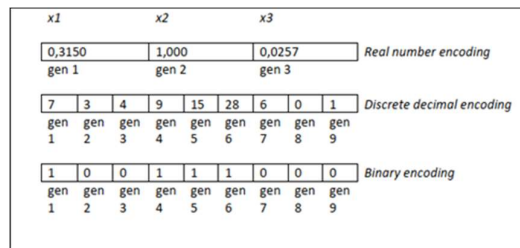
populasi.

4.1 Skema Pengkodean Kromosom

Untuk bisa menggunakan Algoritma Genetika (AG), yang harus dilakukan sebelumnya adalah menentukan pengkodean kromosom yang cocok untuk masalah yang akan diselesaikan. Karena kromosom harus membawa informasi dari solusi yang ia representasikan. Secara umum, suatu kromosom berbentuk:

$$\text{Kromosom} = (\text{gen1}, \text{gen2}, \dots, \text{genn})$$

Gambar 4. Pengkodean Kromosom

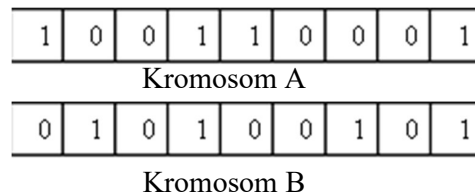


Sumber : Sheng, W., Howells, G., Fairhurst, M. & Deravi, F. (2007). A memetic fingerprint matching algorithm, IEEE Trans. Inf. Forensics Secur., vol. 2, no. 3, pp. 402–411, 2007.

Terdapat tiga skema pengkodean kromosom yang paling umum digunakan, yaitu:

1. Binary encoding

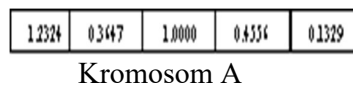
Binary encoding adalah pengkodean yang paling sering digunakan. Karena penelitian pertama kali tentang AG, menggunakan pengkodean ini dan pemakaiannya relatif sederhana. Pada Binary encoding nilai gen adalah 0 atau 1. Contoh kromosom dengan Binary encoding: Contoh kromosom dengan Binary encoding:



Sumber : masalah yang menggunakan Binary encoding : Knapsack problem

2. Value encoding

Value encoding digunakan untuk masalah yang sulit jika direpresentasikan dengan binary encoding . Pada value encoding, setiap kromosom adalah barisan dari beberapa nilai. Nilai disini dapat disesuaikan dengan masalahnya, misalnya bilangan real, huruf, atau objek tertentu. Contoh kromosom dengan value encoding:



Kromosom B ABDJEIFJDHDIERJFDLDFLFEGT

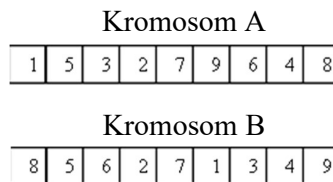
Kromosom C (back), (back), (right), (forward), (left)

Sumber : masalah yang menggunakan value encoding : Menentukan bobot untuk neural network.

3. Permutation encoding

Permutation encoding dapat digunakan pada ordering problem. Pada permutation encoding, setiap kromosom terdiri dari angka yang merepresentasikan urutan pengerjaan atau aktivitas.

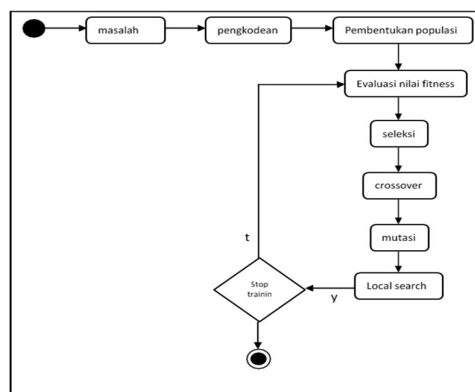
Contoh kromosom dengan permutation encoding:



Sumber : masalah yang menggunakan value encoding Travelling Salesman Problem dan Permutation Flow Shop Scheduling Problem.

Algoritma Kromosom menghasilkan sebuah solusi berdasar pada pencarian lokal. Tujuan utama algoritma memetika ini adalah dengan diberikan di awal suatu peraturan dalam kegiatan untuk ditempatkan pada jadwal mingguan sedemikian rupa sehingga dapat menghasilkan solusi dengan nilai terbaik akan solusi yang dihasilkan. Algoritma memetika dapat digambarkan dalam diagram alir seperti ditunjukkan pada gambar 5

Gambar 5 Diagram Alir Algoritma Kromosom



Sumber : Kielarova, S.W. Development of Hybrid Memetic Algorithm and General Regression Neural Network for Generating Iterated Function System Fractals in Jewelry Design Applications.(2017)

4.2 Jaringan Syaraf

Jaringan Syaraf merupakan salah satu representasi dari otak manusia yang mencoba mensimulasikan proses pembelajaran pada otak manusia . Jaringan Syaraf ini pertama kali

dipresentasikan oleh Warren McCulloch dan Walter Pitt pada tahun 1943 . Jaringan Syaraf memiliki beberapa fitur atraktif antara lain:

- Ketahanan dan toleransi kesalahan. Rusaknya salah satu sel tidak mempengaruhi performance secara signifikan.
- Fleksibel. Jaringan dapat menyesuaikan diri dengan lingkungan baru tanpa perlu menggunakan program terinstruksi. - Memiliki kemampuan komputasi yang kolektif.
- Kemampuan dalam berhubungan dengan variasi data.

4.3 Propagasi Mundur

Proses algoritma kromosom pada pengenalan sidik jari menggunakan jaringan syaraf untuk proses peramalan ini adalah sebagai berikut:

1. Inialisasi count = 0 , fitness = 0 , jumlah siklus
2. Generasi populasi awal . Kromosom individu dirumuskan sebagai urutan gen berturut-turut, masing-masing pengkodean input.
3. Desain jaringan yang cocok
4. Menetapkan bobot
5. Melakukan training dengan propagasi mundur
6. Cari kesalahan kumulatif dan nilai fitness. Kemudian dievaluasi berdasarkan nilai fitness.
7. Jika fitness sebelumnya < nilai fitness saat ini, simpan nilai saat ini
8. Count = count +1
9. Seleksi : Dua induk dipilih dengan menggunakan mekanisme wheel roulette
10. Operasi kromosom: crossover,mutasi dan reproduksi untuk menghasilkan fitur baru set
11. Jika (jumlah siklus <= count) kembali ke nomor empat
12. Pelatihan jaringan dengan fitur yang dipilih
13. Studi kinerja dengan data uji.

METODE PENELITIAN

Metode Penelitian ini bersifat ekperimental, diawali dengan pembangunan model komputasi yang kemudian diimplementasikan dengan memanfaatkan literature review rancangan metode Sidik Jari dengan menggunakan algoritma kromosom. Terdapat beberapa dataset sebagai data uji eksperimen, dalam hal ini berupa data gambar sidik jari.

Berikut adalah metode pada disertasi ini:

1. Metodologi menggunakan evolutionary prototyping
2. Pembacaan data fingerprint menggunakan Java Fingerprint Reader Framework utk menterjemahkan image jd data byte

3. Data byte finger print di compare menggunakan memetic algoritm dlm bahasa java
4. Memasukan proses nomor 3 dlm multithread proses
5. Mengukur performance multithread.

HASIL DAN PEMBAHASAN

Berdasarkan hasil dari metode penelitian dan penelitian-penelitian sebelumnya. Dan berikut dibawah ini table Flowchart Peramalan.

Gambar 6 Flowchart Peramalan



Langkah pertama tentunya kita menginisialisasi semua nilai yang akan digunakan selama proses pelatihan. Kemudian kita mengambil data-data yang digunakan untuk proses pelatihan yang diambil dari database. Setelah itu menormalisasi data tersebut untuk menyesuaikan antara range input-output dengan fungsi aktivasinya. Langkah selanjutnya adalah masuk ke backpropagation sebagai berikut :



Gambar 7. Proses Backpropagation

Pada subproses ini, dilakukan penghitungan nilai neuron di hidden layer dan ditambah dengan biasnya. Kemudian hasil yang didapat diberi fungsi aktivasi sigmoid. Hal ini juga dilakukan di layer output. Proses tersebut merupakan fase pertama dari backpropagation. Fase

kedua adalah mencari δ di layer output dan menghitung suku perubahan ke layer output. Setelah itu, juga mencari δ di hidden layer dan menghitung suku perubahan ke hidden layer. Lalu meng-update ke hidden layer dan ke layer output.

Beikut ini adalah gambarnya dalam bentuk flowchart.

Gambar 8: Peopagasi dengan Peramalan



IV. Testing & Implementasi

Testing dilakukan untuk menentukan parameter- parameter yang tepat untuk digunakan dalam aplikasi ini. Caranya adalah dengan melakukan uji coba terhadap parameter-parameter data – data sidik jari. Parameter-parameter yang tepat adalah parameter- parameter yang saat dilakukan pelatihan dan testing memiliki nilai error yang terkecil dengan data validasi.seperti yang di tampilkan dalam Tabel di bawah ini.

Tabel 1 Data Learning

Bulan1	Bulan2	Bulan3	Bulan4	Bulan5	Bulan6	Bulan7	Bulan8	Bulan9	Bulan10	Bulan11	Bulan12	Target
2816	2992	3366	3960	4048	4576	5324	5258	4774	4554	3690	3168	2508
2992	3366	3960	4048	4576	5324	5258	4774	4554	3690	3168	2508	3916
3366	3960	4048	4576	5324	5258	4774	4554	3690	3168	2508	3916	4026
3960	4048	4576	5324	5258	4774	4554	3690	3168	2508	3916	4026	4334
4048	4576	5324	5258	4774	4554	3690	3168	2508	3916	4026	4334	5478
4576	5324	5258	4774	4554	3690	3168	2508	3916	4026	4334	5478	5852
5324	5258	4774	4554	3690	3168	2508	3916	4026	4334	5478	5852	6600
5258	4774	4554	3690	3168	2508	3916	4026	4334	5478	5852	6600	5874
4774	4554	3690	3168	2508	3916	4026	4334	5478	5852	6600	5874	5082
4554	3690	3168	2508	3916	4026	4334	5478	5852	6600	5874	5082	4598
3690	3168	2508	3916	4026	4334	5478	5852	6600	5874	5082	4598	3872
3168	2508	3916	4026	4334	5478	5852	6600	5874	5082	4598	3872	3520
2508	3916	4026	4334	5478	5852	6600	5874	5082	4598	3872	3520	2684
3916	4026	4334	5478	5852	6600	5874	5082	4598	3872	3520	2684	4094
4026	4334	5478	5852	6600	5874	5082	4598	3872	3520	2684	4094	4378
4334	5478	5852	6600	5874	5082	4598	3872	3520	2684	4094	4378	4752
5478	5852	6600	5874	5082	4598	3872	3520	2684	4094	4378	4752	5682
5852	6600	5874	5082	4598	3872	3520	2684	4094	4378	4752	5682	5984
6600	5874	5082	4598	3872	3520	2684	4094	4378	4752	5682	5984	6798
5654	5082	4598	3872	3520	2684	4094	4378	4752	5682	5984	6798	6116

Berdasarkan data – data yang ada, terdapat 36 data yang membuat 24 pola data. 24 pola data dibagi menjadi dua bagian yaitu 20 pola data (tabel 1) untuk data pelatihan dan 4 data digunakan untuk testing (tabel 2)

Tabel 2. Testing Data

Bulan1	Bulan2	Bulan3	Bulan4	Bulan5	Bulan6	Bulan7	Bulan8	Bulan9	Bulan10	Bulan11	Bulan12	Target
5082	4598	3872	3520	2684	4094	4378	4752	5684	5984	6798	6116	5214
4598	3872	3520	2684	4094	4378	4752	5684	5984	6798	6116	5214	4686
3872	3520	2684	4094	4378	4752	5684	5984	6798	6116	5214	4686	4202
3520	2684	4094	4378	4752	5684	5984	6798	6116	5214	4686	4202	3612

Testing dimulai dari menentukan arsitektur neural network. Jaringan memiliki tiga bagian, yaitu input, hidden layer, dan juga output. Jadi kita menentukan jumlah neuron terbaik yang nantinya akan digunakan untuk aplikasi ini. Tahap kedua adalah menetapkan parameter-parameter genetika. Parameter algoritma genetika akan diuji coba pada tahapan berikutnya sehingga pada tahap ini nilainya belum diubah-ubah. Ketika sudah mendapatkan arsitektur jaringan terbaik, maka dilakukan proses uji coba untuk parameter algoritma genetika. Proses uji coba tahap dilakukan dengan menentukan jumlah neuron kemudian melakukan proses pelatihan dengan cara inisialisasi bobot dengan angka random.

Kemudian dihitung nilai neuron dan juga bobotnya dengan menggunakan data pelatihan yang telah dinormalisasi terlebih dahulu. Proses tersebut merupakan proses yang dilakukan pada tahap fungsi evaluasi dalam algoritma genetika. Kemudian masuk ke tahap seleksi, crossover, dan mutasi. Setelah proses pelatihan, maka dilakukan testing feedforward dan dihitung nilai error antara data yang dihasilkan program dengan data validasi. Variabel awal algoritma genetika yang digunakan adalah enam individu, sepuluh generasi, 0,6 peluang crossover, 0,1 peluang mutasi.

Tabel 3. Percobaan Mencari Parameter

Learning Rate	Arsitektur	Error Peramalan
0,05	12 – 3 – 1	24,1 %
0,1	12 – 3 – 1	26,9 %
0,2	12 – 3 – 1	22,5 %
0,05	12 – 5 – 1	19,3 %
0,1	12 – 5 – 1	21,4 %
0,2	12 – 5 – 1	17,5 %
0,05	12 – 7 – 1	20,5 %
0,1	12 – 7 – 1	28,7 %
0,2	12 – 7 – 1	29,3 %
0,1	12 – 5 – 2 – 1	18,6 %
0,2	12 – 5 – 2 – 1	18,4 %
0,05	12 – 5 – 2 – 1	19,5 %
0,2	12 – 5 – 3 – 1	19,2 %
0,1	12 – 5 – 3 – 2 – 1	18,1 %
0,2	12 – 5 – 3 – 2 – 1	17,8 %
0,2	12 – 5 – 3 – 3 – 1	17,3 %
0,1	12 – 5 – 3 – 3 – 1	19,4 %

Berdasarkan tabel diatas dapat disimpulkan bahwa arsitektur yang memiliki error yang terkecil adalah arsitektur dengan konfigurasi jaringan 12 – 5 – 3 – 3 – 1 yang memiliki tingkat akurasi mencapai 17,3 %.

Setelah melakukan tahap pertama, masuk ke tahap berikutnya yaitu ujicoba variabel algoritma genetika terakhir mempunyai tiga neuron, dan satu buah output dengan laju pembelajaran yang digunakan adalah 0,2. Parameter yang digunakan algoritma genetika adalah sepuluh individu, 0,8 peluang crossover, 0,1 peluang mutasi, dan 50 generasi. Dari uji coba yang dilakukan, aplikasi peramalan ini memiliki tingkat akurasi peramalan mencapai 86%

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa aplikasi peramalan produksi mie untuk Omega Mie Jaya dengan menggunakan algoritma genetika dan

neural network berhasil dilakukan. Aplikasi peramalan ini memiliki arsitektur jaringan berupa 12 input, tiga jaringan hidden layer dimana layer pertama terdiri dari 5 neuron, layer kedua memiliki tiga neuron dan lapisan

DAFTAR PUSTAKA

- Agarwal, D. & Merugu, S. (2007). Predictive discrete latent factor models for largescale dyadic data, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07, ACM, New York, NY, USA, 2007, pp. 26–35. Aguilar, J. & Colmenares, A. (1998). Resolution of pattern recognition problems using a hybrid genetic/random neural network learning algorithm. *Pattern Analysis and Applications*. 1 (1): 52–61. doi:10.1007/BF01238026.
- Aickelin, U. (1998). Nurse rostering with genetic algorithms. Proceedings of young operational research conference 1998. Guildford, UK.
- Alsmadi M.K. (2017). An efficient similarity measure for content based image retrieval using memetic algorithm. *Egypt. Jour. Bas. App. Sci.* (2017), <http://dx.doi.org/10.1016/j.ejbas.2017.02.004>
- Ali, M.A. (2013). Algoritma Genetik Tabu Search dan Memetika Pada Permasalahan Penjadwalan. Seminar Nasional Teknologi Informasi dan Multimedia 2013.
- Aranha, C. & Iba, H. (2009). The Memetic tree-based genetic algorithm and its application to Portfolio Optimization. *Memetic Comp.* (2009), vol. 1, no. 2. pp. 139–151 DOI 10.1007/s12293-009-0010-2
- Areibi, S. & Yang, Z. (2004). Effective memetic algorithms for VLSI design automation = genetic algorithms + local search + multi-level clustering. *Evolutionary Computation*. MIT Press. 12 (3): 327–353. doi:10.1162/10636560417749471535560.
- Armstrong, R., Gannon, D., Geist, A., Keahey, K., Kohn, S., McInnes, L., Parker, S. & Smolinski, B. (1999). Toward a common component architecture for high-performance scientific computing. in: Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing, pp. 115–124.
- Assiroj, P., Hananto, A.L., Fauzi, A. & Warnars, H.L.H.S. (2018). High Performance Computing Implementation: A Survey. The 1st INAPR Conference 2018, Jakarta, Indonesia. IEEE, doi: 10.1109/INAPR.2018.8627040.
- Assiroj, P., Warnars H.L.H.S., Kosala, R., Ranti, B., Supangat, S., Kistijantoro, A.I. & Abdurrachman, E. (2019). The form of High performance computing: A survey. *OP Conf. Ser.: Mater. Sci. Eng.* 662 052002, <https://doi.org/10.1088/1757-899X/662/5/052002>.
- Augugliaro, A., Dusonchet, L. & Sanseverino, E. R. (1998). Service restoration in compensated distribution networks using a hybrid genetic algorithm. *Electric Power Systems Research*. 46 (1): 59–66. doi:10.1016/S0378-7796(98)00025-X.
- Bäck, T., Fogel, D.B. & Michalewicz, Z. (1997). (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing. Ltd., Bristol, UK.
- Bahmann, S. & Kortus, J. (2013). EVO—Evolutionary algorithm for crystal structure prediction. *Comput. Phys. Commun.* 184 (6) (2013) 1618–1625.
- Bai, L., Liang, J., Dang, C. & Cao, F. (2011). A novel attribute weighting algorithm for

- clustering high-dimensional categorical data, *Pattern Recogn.* 44 (12) (2011) 2843–2861.
- Bao, Hu & Xiong. (2013). A PSO and pattern search based memetic algorithm for SVMs parameters optimization. *Neurocomputing*, vol. 117, pp. 98–106, 2013. DOI: 10.1016/j.neucom.2013.01.027
- Barrientos, R.J., Gómez, J.I., Tenllado, C., Matias, M.P., Marin, M. (2011). kNN query processing in metric spaces using GPUs. In: Jeannot, E., Namyst, R., Roman, J. (eds.) *Euro-Par 2011. LNCS*, vol. 6852, pp. 380–392. Springer, Heidelberg (2011).
https://doi.org/10.1007/978-3-642-23400-2_35
- Bereta, M(2019).Baldwin effect and Lamarckian evolution in a memetic algorithm for Euclidean Steiner tree problem. *Memetic Comput.*, vol. 11, no. 1, pp. 35–52. Springer, Heidelberg (2019).
- Bhanu, B., Tan, X. (2001). A triplet based approach for indexing of fingerprint database for identification. In: Bigun, J., Smeraldi, F. (eds.) *AVBPA 2001. LNCS*, vol. 2091, pp. 205–210. Springer, Heidelberg (2001). <https://doi.org/10.1007/3-540-45344-X>
- Bhatt, H.S., Bharawaj, S., Singh, R. & Vasta, M. (2012). Memetically Optimized MCWLD for Matching Sketches With Digital Face Images. *IEEE Transactions On Information Forensics And Security*, vol. 7, no. 5, pp. 1522–1535, 2012.
- Blum, J., Le Dimet, F. X. & Navon, I. M. (2009). Data assimilation for geophysical fluids, in: P.G. Ciarlet (Ed.), *Handbook of Numerical Analysis, Handbook of Numerical Analysis*, vol. 14, Elsevier, 2009, pp. 385–441.
- Botzheim, J., Toda, Y. & Kubota, N. (2012). Bacterial memetic algorithm for offline path planning of mobile robots. *Memetic Computing*, vol. 4 No. 1, pp. 73-86. Springer Intl. DOI: 10.1007/s12293-012-0076-0
- Bozejko, W. & Wodecki, M. (2011). The methodology of parallel memetic algorithms designing, doi:10.5220/0003186006430648, in *Proceeding of the 3rd International Conference on Agents and Artificial Intelligence (ICAART-2011)*, pages 643-648, SCITEPRESS (Science and Technology Publications, Lda).
- Buck, A.R., Keller, J.M. & Skubic, M. (2013). A memetic algorithm for matching spatial configurations with the histograms of forces. *IEEE Transactions on Evolutionary Computation*. Vol. 17-4. Pp. 588-604. doi: 10.1109/TEVC.2012.2226889
- Burke E. K., Gendreau M., Hyde M., Kendall G., Ochoa G., Özcan E. and Qu R. (2013). "Hyper-heuristics: A Survey of the State of the Art". *Journal of the Operational Research Society*. 64 (12): 1695–1724.
- Burke, E. & Smith, A. (1999). A memetic algorithm to schedule planned maintenance for the national grid. *Journal of Experimental Algorithmics*. 4 (4): 1–13. doi:10.1145/347792.347801.
- Cabido, R., Montemayor, A.S. & Pantrigo, J.J. (2012). High performance memetic algorithm particle filter for multiple object tracking on modern GPUs. *Journal Soft Comput* (2012) vol. 16, no. 2, pp. 217–230. DOI: 10.1007/s00500-011-0715-2.
- Cao, K., Liu, E., Jain, A.K. (2014). Segmentation and enhancement of latent fingerprints: a coarse to fine ridgestructure dictionary. *IEEE Trans. Pattern Anal. Mach. Intell.* 36(9), 1847–1859 (2014).

- Caponio, A. & Neri, F. (2012). Handbook of Memetic Algorithms, SCI 379, pp. 241–260. Springer-Verlag Berlin Heidelberg 2012, DOI: 10.1007/978-3-642-23247-3_15.
- Caponio, A., Cascella, G.L., Neri, F., Salvatore, N. & Sumner, M. (2007). A fast adaptive memetic algorithm for off-line and on-line control design of PMSM drivers, IEEE Trans. Syst. Man Cybern. — Part B, Special Issue Memetic Algorithms 37 (1) (2007) 28–41.
- Cappelli, R., Ferrara, M., Maltoni, D. (2010). Minutia cylinder-code: a new representation and matching technique for fingerprint recognition. IEEE Trans. Pattern Anal. Mach. Intell. 32(12), 2128–2141 (2010).
- Cappelli, R., Ferrara, M., Maltoni, D. (2011). Fingerprint indexing based on minutia cylinder-code. IEEE Trans. Pattern Anal. Mach. Intell. 33(5), 1051–1057 (2011)
- Cappelli, R., Maio, D. (2004). The state of the art in fingerprint classification. In: Ratha, N., Bolle, R. (eds.) Automatic Fingerprint Recognition Systems, pp. 183–205. Springer, New York (2004). https://doi.org/10.1007/0-387-21685-5_9.
- Carey, P. (2015). Supercomputers a Hidden Power Center of Silicon Valley, The San Jose Mercury News, May 7, 2015, http://www.mercurynews.com/business/ci_28071868/supercomputershiddenpowercenter-silicon-valley.
- Chavez, E., Navarro, G. (2005). A compact space decomposition for effective metric indexing. Pattern Recogn. Lett. 26(9), 1363–1376 (2005)
- Chavez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.L. (2001). Searching in metric spaces. ACM Comput. Surv. 33(3), 273–321 (2001).
- Changchit, C. Chuchuen, C. (2018). Cloud computing: An examination of factors impacting users' adoption. Journal of Computer Information Systems, Vol. 58, 1-9
- Chen, L., Fujishiro I. & Nakajima, K. (2002). Parallel performance optimization of large-scale unstructured data visualization for the earth simulator, In: Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization, EGPGV '02, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association, pp. 133–140.
- Chen, X. S., Ong, Y. S., Lim, M. H. & Tan, K. C. (2011). "A Multi-Facet Survey on Memetic Computation". IEEE Transactions on Evolutionary Computation. 15 (5): 591–607. doi:10.1109/tevc.2011.2132725.
- Chen, X. S., Ong, Y. S. & Lim, M. H. (2010). "Research Frontier: Memetic Computation - Past, Present & Future". IEEE Computational Intelligence Magazine. 5 (2): 24–36. doi:10.1109/mci.2010.936309.
- Chi, Y & Liu, J. (2014). Learning large-scale fuzzy cognitive maps using a hybrid of memetic algorithm and neural network. IEEE International Conference on Fuzzy Systems, 1036-1040. doi: 10.1109/FUZZ-IEEE.2014.6891604
- Coello, C. (2006). Evolutionary Algorithms: Basic Concepts and Applications in Biometrics, Synthesis and Analysis in Biometrics, World Scientific Publishing, pp 1-34, 2006.
- Cordon, O., Herrera, H. & Lozano, M. (1995). A Classified review on the combination fuzzy logic- genetic algorithms bibliography, Tech. Report 95129, Department of Computer Science and AI, Universidad de Granada, Granada, Spain, 1995, Available at : <http://decsai.ugr.s/~herrera/flga.html>.
- Costa, V. G, Barrientos, R.J., Marin, M., Bonacic, C. (2010) Scheduling metric- space queries

- processing on multi-core processors. In: 18th Euromicro Conference on Parallel, Distributed and Network-based Processing (PDP 2010), pp. 187–194. IEEE Computer Society, Pisa (2010)
- Costa, D. (1995). An evolutionary tabu search algorithm and the NHL scheduling problem. *Infor* 33: 161–178.
- Data from Top500.org, “The List: November 2015,” <http://www.top500.org/lists/2015/11/>
- Datta, A. & Soundaralakshmi, S. (2003). Fast parallel algorithm for distance transform. *IEEE Transactions on System, Man, Cybernetics A, System, Humans* 33 (2003) 429–434.
- Di Gesù, V., Lo Bosco, G., Millonzi, F. & Valenti, C. (2008). A memetic algorithm for binary image reconstruction. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4958 LNCS, pp. 384–395, 2008.
- Dworak, K & Boryczka, U. (2012). Differential Cryptanalysis of Symmetric Block Ciphers Using Memetic Algorithms. *ACIIDS: Asian Conference on Intelligent Information and Database Systems. 4th Asian Conference, 2012 Proceedings, Part II*. Springer International Publishing. 2012. DOI: 10.1007/978-3-642-28490-8.
- Dudley, J.T., Schadt, E., Sirota, M., Butte, A.J. & Ashley, E. (2010). Drug discovery in a multidimensional world: systems, patterns, and networks, *J.Cardiovasc. Trans. Res.* 3 (5) (2010) 438–447.
- Ergun, H., Hertem, D.V., & Belmans, R. (2012). Transmission system topology optimization for large-scale offshore wind integration, *IEEE Trans. Sust. Energy* 3 (4) (2012) 908–917.
- Fernandez, E., Grana, M. & Cabello, J.R. (2004). An instantaneous memetic algorithm for illumination correction. *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)* pp. 1105–1110, 2004. DOI: 10.1109/CEC.2004.1330985.
- Feng, L., Tan, A.H., Lim, M.H. & Jian, S.W. (2012). Band selection for hyperspectral images using probabilistic memetic algorithm. *Journal Soft Computing*, vol. 20, no. 12, pp. 4685-4693, 2012. DOI: 10.1007/s00500-014-1508-1
- França, P., Mendes, A. & Moscato, P. (1999). Memetic algorithms to minimize tardiness on a single machine with sequence-dependent setup times. *Proceedings of the 5th International Conference of the Decision Sciences Institute*. Athens, Greece. pp. 1708–1710.
- Galar, M., Derrac, J., Peralta, D., Triguero, I., Paternain, D., Lopez-Molina, C., Garc'ia, S.J., Benitez, M., Pagola, M. Barrenechea, E., Bustince, H. & Herrera, F. (2015). A survey of fingerprint classification Part I: Taxonomies on feature extraction methods and learning models, *Knowledge-Based Systems* 81 (2015) 76–97.
- Galinier, P., Boujbel, Z. & Fernandes, M. C. (2011). An efficient memetic algorithm for the graph partitioning problem. *Ann. Oper. Res.*, vol. 191, no. 1, pp. 1–22, 2011. DOI: 10.1007/s10479-011-0983-3.
- Gálvez, A. & Iglesias, A. (2018). Modified Memetic Self-Adaptive Firefly Algorithm for 2D Fractal Image Reconstruction. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, 2018, pp. 165-170. doi: 10.1109/COMPSAC.2018.10222.
- Ghosh, M., Kundu, T., Ghosh, D., & Sarkar, R. (2019). Feature selection for facial emotion

- recognition using late hill-climbing based memetic algorithm. *Multimed. Tools Appl.* doi:10.1007/s11042-019-07811-x.
- Ghosh, M., Malakar, S., Bhowmik, S., Sarkar, R. & Nasipuri, M. (2017). Memetic Algorithm Based Feature Selection for Handwritten City Name Recognition. *CICBA 2017, Part II, CCIS 776*, pp. 599–613. Springer Nature Singapore Pte Ltd, 2017. DOI: 10.1007/978-981-10-6427-2.
- Ghosh, M., Malakar, S., Bhowmik, S., Sarkar, R. & Nasipuri, M. (2005). Feature Selection for Handwriting Word Recognition Using Memetic Algorithm. *Advances in Intelligent Computing*, vol. 3644, no. November. Springer Singapore, 2005. doi: 10.1007/11538059.
- Goldberg, E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- Gong, Y.J., Ge, Y.F., Li, J.J., Zhang, J. & Ip, W.H. (2016). A splicing-driven memetic algorithm for reconstructing cross-cut shredded text documents. *Appl. Soft Comput. J.*, vol. 45, pp. 163–172, 2016. DOI: 10.1016/j.asoc.2016.03.024.
- Haas, O., Burnham, K. & Mills, J. (1998). Optimization of beam orientation in radiotherapy using planar geometry. *Physics in Medicine and Biology*. 43 (8): 2179– 2193. doi:10.1088/0031-9155/43/8/013. PMID 9725597.
- Haque, M.N., Mathieson, L & Moscato, P. (2018). A memetic algorithm for community detection by maximising the connected cohesion. *2017 IEEE Symp. Ser. Comput. Intell. SSCI 2017 - Proc.*, vol. 2018-Janua, pp. 1–8, 2018.
- Harmon, A. (2014). *HPC Matters: Funding, Collaboration, Innovation*. ScienceNode, November 26, 2014, <https://sciencenode.org/feature/hpc-matters-funding-collaboration-innovation.php>.
- Harris, S. & Ifeachor, E. (1998). Automatic design of frequency sampling filters by hybrid genetic algorithm techniques. *IEEE Transactions on Signal Processing*. 46(12): 3304–3314. DOI:10.1109/78.735305.
- Hart, W., Krasnogor, N. & Smith, J.E. (2005). *Memetic evolutionary algorithms*, pp. 3–27 (2005).
- Hart W. (1994). *Adaptive global optimization with local search*. Ph. D. Thesis, University of California, San Diego
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
- Hong, J.H., Min, J.K., Cho, U.K. & Cho, S.B. (2008). Fingerprint classification using one-vs-all support vector machines dynamically ordered with Naïve Bayes classifiers. *Pattern Recogn.* 41(2), 662–671 (2008).
- Hong, J.H. & Cho, S.B. (2006). Efficient huge-scale feature selection with speciated genetic algorithm, *Pattern Recogn. Lett.* 27 (2) (2006) 143–150.
- Huang, K.W., Wu, Z.X., Peng, H.W., Tsai, M.C., Hung, Y.C. & Lu, Y.C. (2018). A memetic particle gravitation optimization algorithm for solving image segmentation. *Proceedings of 4th IEEE International Conference on Applied System Innovation 2018, ICASI 2018*. pp. 82-85. DOI: 10.1109/ICASI.2018.8394392.
- Huang, K.W., Wu, Z.X., Peng, H.W., Tsai, M.C., Hung, Y.C. & Lu, Y.C. (2019).

- Memetic particle gravitation optimization algorithm for solving clustering problems. *IEEE Access*, vol. 7, pp. 80950-80968, 2019. DOI: 10.1109/ACCESS.2019.2923979
- Hwang, B.W., Kim, S., Lee, S.W., Huang, D.S., Zhang, X.P., & Huang, G.B. (2005). Feature Selection for Handwritten Word Recognition Using Memetic Algorithm. *Advances in Intelligent Computing*, vol. 3644, no. November. Springer Singapore, 2005. IBM White Papers. (2015). <https://www-03.ibm.com/support/techdocs/atsmastr.nsf/Web/WhitePapers> (diakses 10 Januari 2020).
- Ibrahiem, E.E. & El-Kareem, M. (2008). On the application of Genetic Algorithm in Finger Print Recognition, *World Applied Sciences Journal*, 5(3), pp 276-281, 2008.
- Ichimura, T. & Kuriyama, Y. (1998). Learning of neural networks with parallel hybrid GA using a royal road function. *IEEE International Joint Conference on Neural Networks*. 2. New York, NY. pp. 1131–1136.
- IDC. High Performance Computing in the EU: Progress on the Implementation of the European HPC Strategy (Brussels: European Commission DG Communications Networks, Content & Technology, 2015), 17, <http://knowledgebase.eirg.eu/documents/243153/246094/High+Performance+Computin+g+in+the+EU+Progress+on+the+Implementation+of+the+European+HPC+Strategy.pdf>
/b0adf617-3f50-4a6f-9217- 4e0fbb5edd09.
- Jain, A.K., Bolle, R.M., Pankanti, S. (2005). *Biometrics: Personal Identification in Networked Society*. Springer International.
- Jain, A.K., Feng, J. (2011). Latent fingerprint matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 88–100.
- Jain, A.K, Flynn, P., Ross, A.A. (2007). *Handbook of Biometrics*. Springer, New York (2007). <https://doi.org/10.1007/978-0-387-71041-9>.
- Jain, A.K., Hong, L., & Bolle, R. (1997). On-line fingerprint verification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997) 302–314.
- Jain, A.K., Hong, L., Pankanti, S. & Bolle, R. (1997). An identity-authentication system using fingerprints, *Proceedings of IEEE* 85 (1997) 1365–1388.
- Jayaram, M.A. & Fleyeh, H. (2013). Soft Computing in Biometrics: A Pragmatic Appraisal, *American Journal of Intelligent Systems* 2013, 3(3): 105-112. doi: 10.5923/j.ajis.20130303.01.
- Jiao, L., Gong, M., Wang, S., Hou, B., Zheng, Z. & Wu, Q. (2010). Natural and remote sensing image segmentation using memetic computing, *IEEE Comput. Intell. Mag.* 5 (2) (2010) 78–91.
- Jat, S.N & Yang, S. (2011). A Memetic Algorithm for the University Course Timetabling Problem. *Systems Man and Cybernetics Part C: Applications and Reviews IEEE Transactions on*, vol. 41, pp. 93-106, ISSN 1094-6977.
- Johnston, D. (2014). HPC Matters to Our Quality of Life and Prosperity. *Scientific Computing*, November 11, 2014, <http://www.scientificcomputing.com/articles/2014/11/hpc-matters-our-quality-life-andprosperity>

- Karkavitsas, G. & Tsihrintzis, G. (2011). "Automatic Music Genre Classification Using Hybrid Genetic Algorithms". *Intelligent Interactive Multimedia Systems and Services*. Springer. 11: 323–335. doi:10.1007/978-3-642-22158-3_32.
- Kendall G., Soubeiga E. & Cowling P. (2002). Choice function and random hyperheuristics. 4th Asia-Pacific Conference on Simulated Evolution and Learning. SEAL 2002. pp. 667–671.
- Kielarova, S.W. (2017). Development of Hybrid Memetic Algorithm and General Regression Neural Network for Generating Iterated Function System Fractals in Jewelry Design Applications. Proceedings, Part II, 8th International Conference, ICSI 2017, Fukuoka, Japan, July 27 – August 1, 2017, vol. 10386, pp. 280–289, 2017. Springer International. DOI: 10.1007/978-3-319-41000-5_28
- Kollen, A. & Pesch, E. (1994). Genetic local search in combinatorial optimization, *Discrete Applied Mathematics and Combinatorial Operation Research and Computer Science* 48 (1994) 273– 284.
- Krasnogor N. & Gustafson S. (2002). Toward truly "memetic" memetic algorithms: discussion and proof of concepts. *Advances in Nature-Inspired Computation: the PPSN VII Workshops*. PEDAL (Parallel Emergent and Distributed Architectures Lab). University of Reading. horizontal tab character in [journal= at position 138 (help)
- Krasnogor N. (1999). "Coevolution of genes and memes in memetic algorithms". *Graduate Student Workshop*: 371.
- Kumar, B.V., Karpagam, G. R. & Zhao, Y. (2019). Evolutionary Algorithm With Memetic Search Capability for Optic Disc Localization in Retinal Fundus Images, no. 1. Elsevier Inc., 2019.
- Kumar, D., Kumar, S. & Rai, S. (2009). Feature Selection for face recognition: A Memetic algorithm approach. *Journal of Zhejanga University Science*, 10(8), pp 1140- 1152, 2009.
- Kumar, A. & Kwong, C. (2013). Towards contactless, low-cost and accurate 3D fingerprint identification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3438–3443 (2013).
- Lang, S., Drouvelis, P., Tafaj, E., Bastian, P. & Sakmann, B. (2011). Fast extraction of neuron morphologies from large-scale SBFSEM image stacks, *J. Comput. Neurosci.* 31 (3) (2011) 533–545.
- Lastra, M., Molinab D. & Benítez, J. M. (2015). A high performance memetic algorithm for extremely high-dimensional problems, *Information Science* 293 (2015) 35- 58, Elsevier Inc. <http://dx.doi.org/10.1016/j.ins.2014.09.018>.
- Le, H.H., Nguyen, N.H., Nguyen, T.T. (2016). Exploiting GPU for large scale fingerprint identification. In: Nguyen, N.T., Trawi' nski, B., Fujita, H., Hong, T.-P. (eds.) *ACIIDS 2016*. LNCS (LNAI), vol. 9621, pp. 688–697. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49381-6_66.
- Li, Y., Hu, J., & Jia, Y. (2014) "Automatic SAR image enhancement based on nonsubsampling contourlet transform and memetic algorithm," *Neurocomputing*, vol. 134, pp. 70–78, 2014. DOI: 10.1016/j.neucom.2013.03.068.
- Liang, B., Liu, B., Zhou, F., Guo, B., Xu, X., Kang, J., Li, J. & Liu, W. (2015). A
- Lin, J., & Chen, Y. (2011). Analysis on the collaboration between global search and local

- search in memetic computation, October 15(5) (2011) 608–623.
- Lu, Y., Wang, S., Li, S. & Zhou, C. (2009). Particle swarm optimizer for variable weighting in clustering high-dimensional data, *Mach. Learn.* 82 (1) (2009) 43–70.
- Madhavi KV, Tamilkodi R, & Sudha KJ. (2016). An innovative method for retrieving relevant images by getting the top-ranked images first using interactive genetic algorithm. *Proc Comput Sci* 2016;79:254–61.
<https://doi.org/10.1016/j.procs.2016.03.033>
- Maltoni, D., Maio, D., Jain, A.K. & Prabhakar, S. (2009). *Handbook of Fingerprint Recognition*, Springer-Verlag New York Inc.
- Marksteiner, P. (1996) High-performance computing- an overview. Vienna University Computer Center Universitätsstraße 7, A-1010 Vienna, Austria. *Computer Physics Communications* 97 16-35.
- Manacher, G.K. (1967). Production and stabilization of real time task schedules. *Journal of ACM* 14 (1967) 439–465
- Marin, M., Costa, V. G., Bonacic, C., Yates, R. B., Scherson, I.D. (2010). Sync/async parallel search for the efficient design and construction of web search engines. *Parallel Comput.* 36(4), 153–168 (2010).
- Matsui, T., Katagiri, Y., Katagiri, H. & Kato, K. (2015). “Automatic feature point selection through hybrid metaheuristics based on tabu search and memetic algorithm for augmented reality,” *Procedia Comput. Sci.*, vol. 60, no. 1, pp. 1120–1127, 2015.
- Merz, P & Freisleben, B. (1999). *Fitness Landscapes and Memetic Algorithm Design*, in: D. Corne, M. Dorigo, F. Glover (Eds.), McGraw-Hill, London, 1999.
- Merz, P. & Zell, A. (2002). Clustering Gene Expression Profiles with Memetic Algorithms. *Parallel Problem Solving from Nature — PPSN VII*. Springer. pp. 811–820. doi:10.1007/3-540-45712-7_78.
- Montazeri, M & Iran, K. (2019). Memetic Algorithm Image Enhancement for Preserving Mean Brightness Without Losing Image Features. *International Journal of Image and Graphics*. 19. 1950020. 10.1142/S0219467819500207.
- Moscato, P., Mendes, A. & Berretta, R. (2007). Benchmarking a memetic algorithm for ordering microarray data. *BioSystems*, vol. 88, no. 1–2, pp. 56–75, 2007. Elsevier Ireland Ltd. All rights reserved. doi:10.1016/j.biosystems.2006.04.005
- Moscato, P. (1989). *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetic Algorithms*. Technical Report, Caltech Concurrent Computation Program, California Institute of Technology, Pasadena, 1989.
- Moscato, P. (1999). *Memetic Algorithms: a Short Introduction*, in: D. Corne, M. Dorigo, F. Glover (Eds.), McGraw-Hill, London, 1999.
- Mu, C., Xie, J., Liu, R. & Jiao, L. 2014. A memetic algorithm using local structural information for detecting community structure in complex networks. *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*. 680- 686. doi: 10.1109/CEC.2014.6900336
- Nagy, B. & Moisi, E.V. (2016). Memetic algorithms for reconstruction of binary images on triangular grids with 3 and 6 projections, *Appl. Soft Comput. J.* (2016), <http://dx.doi.org/10.1016/j.asoc.2016.10.014>.

- Navarro, G., Paredes, R.U., (2011). Fully dynamic metric access methods based on hyperplane partitioning. *Inf. Syst.* 36(4), 734–747 (2011).
- Naveen, N & Rao, M.C. (2016). Bankruptcy Prediction Using Memetic Algorithm. 10th International Workshop, MIWAI 2016, Chiang Mai, Thailand, December 7-9, 2016, Proceedings, pp. 153-161. Springer Cham. <https://doi.org/10.1007/978-3-319-49397-8>.
- Neri, F. & Mininno, E. (2010). Memetic compact differential evolution for cartesian robot control. *IEEE Computational Intelligence Magazine* 5(2), 54–65 (2010).
- Nguyen K., Lu T., Le T. & Tran N. (2011) Memetic Algorithm for a University Course Timetabling Problem. In: Tan H. (eds) *Informatics in Control, Automation and Robotics. Lecture Notes in Electrical Engineering*, vol 132. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-25899-2_10
- Niu, D., Wang, Y. & Wu, D. D. (2010). Power load forecasting using support vector machine and ant colony optimization, *Expert Syst. Appl.* 37 (3) (2010) 2531–2539.
- Ong, Y.S., Lim, M.H., Chen, X. (2010). Memetic computation-past, present and future. *IEEE Computational Intelligence Magazine* 5(2), 24–31 (2010).
- Ong, Y.S., Lim, M.H., Zhu N, & Wong, K.W. (2006). Classification of adaptive memetic algorithms: a comparative study, *IEEE Trans. Syst. Man. Cybern.* 36 (1) (2006) 141–152.
- Ozcan, E. & Basaran, C. (2009). A Case Study of Memetic Algorithms for Constraint Optimization. *Soft Computing: A Fusion of Foundations, Methodologies and Applications*. 13(8–9): 871–882. DOI:10.1007/s00500-008-0354-4.
- Ozcan, E. (2007). Memes, Self-generation and Nurse Rostering. *Lecture Notes in Computer Science. Lecture Notes in Computer Science*. Springer-Verlag. 3867: 85–104. DOI:10.1007/978-3-540-77345-0_6. ISBN 978-3-540-77344-3.
- Ozcan, E. & Onbasioglu, E. (2006). Memetic Algorithms for Parallel Code Optimization. *International Journal of Parallel Programming*. 35 (1): 33–61. DOI:10.1007/s10766-006-0026-x.
- Ozcan E., Mohan C.K. (1998) Steady state memetic algorithm for partial shape matching. In: Porto V.W., Saravanan N., Waagen D., Eiben A.E. (eds) *Evolutionary Programming VII. EP 1998. Lecture Notes in Computer Science*, vol 1447, pp. 527-536 Springer, Berlin, Heidelberg. DOI: <https://doi.org/10.1007/BFb0040804>
- Pagacz, A & Hu, B. (2010). A Memetic Algorithm with population management for the generalized minimum vertex-biconnected network problem. *Proc. - 2nd Int. Conf. Intell. Netw. Collab. Syst. INCOS 2010*, pp. 356–361, 2010.
- Pankanti, S., Prabhakar, S. & Jain, A.K. (2002). On the Individuality of Fingerprints, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002) 1010–1025.
- Peralta, D., García, S., Benitez, J.M. & Herrera, F. (2017). Minutiae-Based Fingerprint Matching Decomposition: Methodology for Big Data Frameworks, *Information Sciences* (2017), DOI: 10.1016/j.ins.2017.05.001.
- Peralta, D., Galarc, M., Triguero, I., Paternain, D., García, S., Barrenechea, E., Benítez, J.M., Bustincec, H. & Herrera, F. (2015). A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation, *Inform. Sci.* (2015), <http://dx.doi.org/10.1016/j.ins.2015.04.013>

- Peralt, D., Triguero, I., Sanchez-Reillo, R., Herrera, F & Benitez, J.M. (2013). Fast fingerprint identification for large databases, *Pattern Recognition* (2013), <http://dx.doi.org/10.1016/j.patcog.2013.08.002i>.
- Peralta, D., Triguero, I., Sanchez-Reillo, R., Herrera, F., Benítez, J.M. (2014). Fast fingerprint identification for large databases. *Pattern Recogn.* 47(2), 588–602 (2014).
- Phillips, J. (2007). High Performance Computing with CUDA – Case Study: Molecular Dynamics; Super Computing Workshop.
- Pillarichie, R. & Suyanto. (2012). Algoritma Genetika Dengan Local Search Untuk Penjadwalan Kuliah, Skripsi S1, Fakultas Teknik Informatika, Telkom University.
- Poonam, G. (2009). Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm, *International Journal of Network Security & Its Applications (IJNSA)*, Vol.1, No 1.
- Potti S., Pothiraj S. (2011) GPGPU Implementation of Parallel Memetic Algorithm for VLSI Floorplanning Problem. In: Nagamalai D., Renault E., Dhanuskodi M. (eds) *Trends in Computer Science, Engineering and Information Technology*. CCSEIT 2011. *Communications in Computer and Information Science*, vol 204, pp. 432-441. Springer, Berlin, Heidelberg. DOI: [DOI: https://doi.org/10.1007/978-3-642-24043-0_44](https://doi.org/10.1007/978-3-642-24043-0_44)
- Radtke P.V.W., Wong T., Sabourin R. (2005) A Multi-objective Memetic Algorithm for Intelligent Feature Extraction. In: Coello Coello C.A., Hernández Aguirre A., Zitzler E. (eds) *Evolutionary Multi-Criterion Optimization. EMO 2005. Lecture Notes in Computer Science*, vol 3410, pp. 767-781. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-540-31880-4_53
- Ridao, M., Riquelme, J., Camacho, E. & Toro, M. (1998). An evolutionary and local search algorithm for planning two manipulators motion. *Lecture Notes in Computer Science. Lecture Notes in Computer Science*. Springer-Verlag. 1416: 105–114. DOI:10.1007/3-540-64574-8_396. ISBN 3-540-64574-8.
- Roy, T.K. & Gerber, R.B. (2013). Vibrational self-consistent field calculations for spectroscopy of biological molecules: new algorithmic developments and applications, *Phys. Chem. Chem. Phys.: PCCP* 15 (24) (2013) 9468–9492.
- Ruiz, L.G.B., Capel, M.I. & Pegalajar, M.C. (2018). Parallel memetic algorithm for training recurrent neural networks for the energy efficiency problem, *Applied Soft Computing Journal* (2018), <https://doi.org/10.1016/j.asoc.2018.12.028>
- Sahid. (1997) A Study on University Timetabling. Thesis, Department of Mathematics, The University of Queensland Australia.
- Seidenberg, M. S. & McClelland, J. L. (1989) A distributed, developmental model of word recognition and naming. *Psychological review* 96 (1989) 523–568.
- Senate Energy and Natural Resources Committee (SENRC). Next Frontier in High-Performance Computing to Usher in New Chapter in Scientific Discovery. news release, August 1, 2012, <https://www.highbeam.com/doc/1P3-2725052741.html>
- Shao, Y. et.al. (2006). Advances in methods and algorithms in a modern quantum chemistry program package, *Physical Chemistry Chemical Physics: PCCP*, 8(27), July 2006, pp. 3172–3191.

- Sheng, W., Howells, G., Fairhurst, M. & Deravi, F. (2007). A memetic fingerprint matching algorithm, *IEEE Trans. Inf. Forensics Secur.*, vol. 2, no. 3, pp. 402–411, 2007.
- Shi, W., Wahba, G., Irizarry, R.A., Bravo, H.C. & Wright, S.J. (2012). The partitioned LASSO-patternsearch algorithm with application to gene expression data, *BMC Bioinformatics* 13 (1) (2012).
- Shimpi, A.L. (2012). Inside the Titan Supercomputer: 299K AMD x86 Cores and 18.6K NVIDIA GPUs. Anand Tech, October 31, 2012, <http://www.anandtech.com/show/6421/inside-titansupercomputer-299k-amd-x86-cores-and-186k-nvidia-gpu-cores>
- Smith J. E. (2007). Coevolving Memetic Algorithms: A Review and Progress Report. *IEEE Transactions on Systems Man and Cybernetics - Part B.* 37 (1): 6–17. doi:10.1109/TSMCB.2006.883273.
- Stamatakis, A. & Ott, M. (2008) Exploiting fine-grained parallelism in the phylogenetic likelihood function with mpi, pthreads, and openmp: a performance study. in: *Proceedings of the 3rd International Conference on Pattern Recognition in Bioinformatics*, Springer-Verlag, 2008, pp. 424–435.
- Stone, H. (1992) *High-Performance Computer Architecture*. Addison-Wesley Longman Publishing Co., Inc.
- Tan KC, Lee TH & Khor EF. (2001). Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 5(6): 565–588. DOI: 10.1109/4235.974840
- Techopedia. (2016). High-Performance Computing (HPC). (diakses 10 Januari 2020) <https://www.techopedia.com/definition/4595/high-performance-computing-hpc>.
- The National Institute for Computational Sciences (NICS). What is HPC?. (diakses 10 Januari 2020),
- Tirronen, V., Neri, F., Kärkkäinen, T., Majava, K. & Rossi, T. (2008). An Enhanced Memetic Differential Evolution in Filter Design for Defect Detection in Paper Production. *Journal Evolutionary. Computation*, vol. 16, no. 4, pp. 529–555, Springer- Verlag Berlin, 2008. DOI: 10.1162/evco.2008.16.4.529
- Ulder, N.L.J., Aarts, E.H.L., Bandelt, H.J., Laarhoven, P.J.M., Pesch, E. (1991). Genetic local search algorithms for the traveling salesman problem, in: H.P. Schwefel, R. Manner (Eds.), *Parallel Problem Solving from Nature—Proceedings of 1st Workshop, PPSN I*, Lecture Notes in Computer Science, vol. 496, Springer, Berlin, Germany, 1991, pp. 109–116.
- Urselmann, M., Barkmann, S., Sand, G. & Engell, S. (2011). A memetic algorithm for global optimization in chemical process synthesis problems, *IEEE Trans. Evol. Comput.* 15 (5) (2011). 659–283.
- Varshney, K.R., Willsky, A.S. (2011). Linear dimensionality reduction for margin-based classification: High-dimensional data and sensor networks, *IEEE Trans. Signal Process.* 59 (6) (2011) 2496–2512.
- Wan, Y., Zhong, Y. & Ma, A. (2019). Fully Automatic Spectral-Spatial Fuzzy Clustering Using an Adaptive Multiobjective Memetic Algorithm for Multispectral Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, No. 4. Pp. 2324–2340. DOI: 10.1109/TGRS.2018.2872875

- Wang, Y., Wang, L., Cheung, Y.M. & Yuen, P.C. (2015). Learning compact binary codes for hash-based fingerprint indexing, *IEEE Transactions on Information Forensics and Security* 10 (2015) 1603–1616.
- Wehrens, R., Lucasius, C., Buydens, L. & Kateman, G. (1993). HIPS, A hybrid self-adapting expert system for nuclear magnetic resonance spectrum interpretation using genetic algorithms. *Analytica Chimica Acta*. 277 (2): 313–324. doi:10.1016/0003-2670(93)80444-P.
- Welekar, R. & Thakur, N.V. (2019). An Enhanced Approach to Memetic Algorithm Used for Character Recognition, *Third International Congress on Information and Communication Technology*, vol. 797. Springer Singapore, 2019.
- Yamada, T., & Nakano, R. (1991). Scheduling by genetic local search algorithms with multi-step crossover, in: H.P. Schwefel, R. Manner (Eds.), *Proceedings of 4th Conference on Parallel Problem Solving from Nature—Proceedings of 1st Workshop, PPSN I*, *Lecture Notes in Computer Science*, vol. 496, Springer, Berlin, Germany, 1991, pp. 960–969.
- Yang, S., Cheng, K., Wang, M., Xie, D. & Jiao, L. (2013). High resolution range- reflectivity estimation of radar targets via compressive sampling and Memetic Algorithm. *Inf. Sci. (Ny)*, vol. 252, pp. 144–156, 2013. <https://dx.doi.org/10.1016/j.ins.2013.06.029>
- Zhang, Y. & Zhong, Y. (2015). Sub-pixel mapping based on memetic algorithm for hyperspectral imagery, *Int. Geosci. Remote Sens. Symp.*, vol. 2015-Novem, pp. 393–396, 2015.
- Zhang, M., Ma, J., Gong, M., Li, H. & Liu, J. (2017). Memetic algorithm based feature selection for hyperspectral images classification, *2017 IEEE Congr. Evol. Comput. CEC 2017 - Proc.*, no. 2, pp. 495–502, 2017.
- Zhao, Y., Sheong, F.K., Sun, J., Sander, P. & Huang, X. (2013). A fast parallel clustering algorithm for molecular simulation trajectories, *J. Comput. Chem.* 34 (2) (2013) 95–104.
- Zhou, D., Fang, Y., Botzheim, J., Kubota & Liu, H. (2016). Bacterial memetic algorithm based feature selection for surface EMG based hand motion recognition in long-term use. *IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens, 2016, pp. 1-7. doi: 10.1109/SSCI.2016.785024
- Zhu, Z., Jia, S. & Ji, Z. (2010). Affinity propagation based memetic band selection on hyperspectral imagery datasets. *IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*. DOI: 10.1109/CEC.2010.5586533
- Zhu, Z., Ong, Y. S. & Dash, M. (2007). Markov Blanket-Embedded Genetic Algorithm for Gene Selection. *Pattern Recognition*. 49 (11): 3236–3248. doi:10.1016/j.patcog.2007.02.007.
- Zhu, Z., Ong, Y. S. & Dash, M. (2007). Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework. *IEEE Transactions on Systems, Man and* .
- Zhu, Z., Ong, Y.S. & Zurada, M. (2008). Simultaneous Identification of Full Class Relevant and Partial Class Relevant Genes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.