

A Survey on Evolutionary Computation: Methods and Their Applications in Engineering

Morteza Husainy Yar¹, Vahid Rahmati² & Hamid Reza Dalili Oskouei³

¹ Cando Network Security Institute, Tehran, Iran

² Ayandeh Samin Electronics, Tehran, Iran

³ Shahid Sattari Aeronautical University of Science and Technology, Tehran, Iran

Correspondence: Vahid Rahmati, Ayandeh Samin Electronics, Tehran, Iran. Tel: 98-912-702-3373. E-mails: vahidrahmati9170@gmail.com/vahid.rahmati@icdst.org/mortezahuseinyar@gmail.com/h_oskouei@yahoo.com

Received: June 22, 2016

Accepted: July 7, 2016

Online Published: August 9, 2016

doi:10.5539/mas.v10n11p131

URL: <http://dx.doi.org/10.5539/mas.v10n11p131>

Abstract

Evolutionary computation is now an inseparable branch of artificial intelligence and smart methods based on evolutionary algorithms aimed at solving different real world problems by natural procedures involving living creatures. It's based on random methods, regeneration of data, choosing by changing or replacing data within a system such as personal computer (PC), cloud, or any other data center. This paper briefly studies different evolutionary computation techniques used in some applications specifically image processing, cloud computing and grid computing. These methods are generally categorized as evolutionary algorithms and swarm intelligence. Each of these subfields contains a variety of algorithms and techniques which are presented with their applications. This work tries to demonstrate the benefits of the field by presenting the real world applications of these methods implemented already. Among these applications is cloud computing scheduling problem improved by genetic algorithms, ant colony optimization, and bees algorithm. Some other applications are improvement of grid load balancing, image processing, improved bi-objective dynamic cell formation problem, robust machine cells for dynamic part production, integrated mixed-integer linear programming, robotic applications, and power control in wind turbines.

Keywords: evolutionary computation, swarm intelligence, cloud computing, signal processing, power control

1. Introduction

Evolutionary computation, abbreviated as EC, is a subfield of artificial intelligence (AI), machine learning, and smart systems. The EC is almost a new and interesting area of research which uses biological evolution principles in order to solve computational problems with different complexities. A *computational problem* that must be solved by EC often requires the system to follow some convergence criteria expressed by the problem or its conditions. *Evolution* is defined as optimization process that tries to advance the ability of an organism to live longer in rapidly changing and unstable environments. Now if systems with computational models of evolutionary environments and processes are solved by a systematic approach, hence it's referred to as EC. The processes are: natural selection, reproduction, and survival. The theory of natural selection means that all living creatures on earth are the result of adaptation (and upgrade) to the demands of environment to date. If an organism has a successful adaptation to such environments, then it has higher fitness as it has more chance to exit and to outlast. Survival of fitness also shows that entities that are more fit to fluctuations of environment have more chances to survive longer, therefore more chances to spread their genotype to future generation by reproduction (Eiben & Smith, 2003; Fogel, 2007).

Evolutionary computation in this case employs some procedures based on evolutionary principles and techniques as the soul of EC. Generally, solutions of many problems are searchable in a domain—for example large integer factorization—but the problem is that covering the whole range of solutions and probably testing them for verification can be challenging. Clearly, optimum solution is achieved when the number of possible correct solutions is less, and on the other hand, if search space is not limited, the findings are based on exhaustive searches that may either lead to an answer or not. This is not just for EC, and many other topics such as

mathematics, signal processing, artificial neural networks (ANNs) and more suffer from such conditions that lead to many answers verifiable after extensive searches (Rahmati, Yar, Khalilpour, & Malekijavan, n.d.; Vahid Rahmati, Yar, & Malekijavan, 2014). Search is one way to find the solution, therefore, a collection of different searching methods developed with regard to the extent of spaces. Among these searching methods, the natural evolution methods—usually using meta-heuristic procedures—are the best choices at the moment. Despite applications, many of the resources categorize methods into two main branches: 1- evolutionary algorithms (EAs) 2- swarm intelligence (SI) (Back & Schwefel, 1996; Fogel, 2007; Wierstra, Schaul, Glasmachers, Sun, & Schmidhuber, 2011).

This paper introduces both of these categories briefly with current real world examples and algorithms that show promising future for such methods. Management of immense amount of information available on PCs and web servers, use of cloud/grid computing, and efficient signal/image processing are among top applications of EC. The subfields for the EA are therefore: 1- differential evolution and search, 2- evolutionary strategy and programming, and 3- genetics (expressing) programming and algorithm. SI also includes: 1- ant colony, 2- bees algorithm, 3- cuckoo bird algorithm, and 4- particle swarm (Eberhart & Yuhui, 2001; Eberhart & Kennedy, 1995; Fogel, 2007; Guo, Zhao, Shen, & Jiang, 2012; Marinos & Briscoe, 2009; Sun, Wierstra, Schaul, & Schmidhuber, 2009).

2. Main Applications of EC

As mentioned in introduction, smarter image and signal processing are among important applications of EC where some regions in the image must be extracted in order for later processing. However, filtering and de-noising are also done by some algorithms. These applications are also introduced in this paper (Felzenszwalb & Huttenlocher, 2004; Horng, 2010; James & Eberhart, 1995; Kirchmaier, Hawe, & Diepold, 2013).

The most important and recently introduced concept in the ever growing field of information industry is cloud computing (CC) that gives privilege of request sending to some immense computational resources available on the web. The main fact proposed by CC states: *it's not necessary for every organization or company to build up its own data centers and high-tech infrastructures while it's possible to use CC on a pay per use basis*. The main problem for companies was building and maintaining such infrastructures, which without any doubt needs expenditure of money with too much efforts for future benefits of the companies. High amounts of investment are not feasible for middle to low income companies permitting them to use CC instead. This makes a brighter future for CC, thereupon a better prospect for upcoming EC methods and techniques (Arya & Verma, 2014; Ma & Wang, 2012; Subrata, Zomaya, & Landfeldt, 2007; Younge, Laszewski, Wang, Lopez, & Carithers, 2010).

Data centers with (super) computers provide different types of resources and services for CC, accordingly different types of CC services appeared in the market. Abbreviation paradigm of such services uses the suffix –aaS which represent “*as a service*”. These are: 1- API¹ as a service (AaaS) 2- Data... (DaaS) 3- Infrastructure... (IaaS) 4- Platform... (PaaS) 5- Software... (SaaS) 6- Storage... (SEaaS) 7- Security... (SECaaS) 8- Test environment... (TEaaS). Some CC service suppliers are Amazon EC2, Google compute engine, Google app engine (GAE), Microsoft Azure, Google apps, IBM blue cloud, and more that provide public, private (enterprise), hybrid, and community clouds (Furht & Escalante, 2010).

Although CC is growing and transfer of data done by secure communication protocols with public key encryptions, there are some companies and individuals that are still reluctant to use CC services. In the question of “*high cost or high risk?*” many of them reply *high risk* as the data is available to CC providers (Shaikh & Haider, 2011; Zissis & Lekkas, 2012).

Despite cluster computing, if local and global resources together are used at the same time to solve a particular problem, then such a method is called grid computing (GC) which is itself a kind of distributed computing (DC). However, it's useful to use local and global computing units concurrently in order to compute, but this itself makes some problems arose from mismanagement of nodes in such networks which are solvable via genetic algorithms (Berstis, 2002; Furht & Escalante, 2010).

EAs can also be employed in military applications (Vahid, Tal, & Ghasem, 2014), however, there are few papers considering such growing number of applications in smart electronic warfare capabilities. Another application of EAs and ANNs can be software's developments total cost evaluation (Vahid, 2016b).

3. Evolutionary Algorithms (EAs)

The motivation behind EAs is essentially to employ the natural mechanisms of life, biological evolution including mutation, selection, and more—that are used by living creatures to solve a problem—in a computational manner. EAs are part of a broader algorithm set (EC) and are based on random searches and

meta-heuristics. The accuracy of optimized candidate solution by using EAs are generally improved by more iterations. However, more iteration cannot always reduce error (Fogel, 2007).

3.1 General Steps in an EA

EAs usually follow some steps in each iteration to find the best solution (Eiben & Smith, 2003; Fogel, 2007). This is as:

- 1- Fitness of population of individuals grows after natural selection due to environmental pressures.
- 2- Evaluation of each individual is done by a fitness function (given by the problem).
- 3- Parents are selected among individuals based on fitness values.
- 4- New individuals are generated from the parents (step 3) by recombination.
- 5- The fitness value of new individuals and old candidates are compared to select next generation.
- 6- If error of solution is more than expected go to step 1 else finish iteration.

Now that the main steps in an EA are known, it's useful to review some of the most famous algorithms.

3.2 Differential Evolution (DE)

Minimization and maximization of functions are generally a mathematical topic. Minimization of several variable functions can be very complex but practical as it can exhibit error function. DE algorithm thus, as a stochastic optimization method, invented to minimize an objective function under certain given constraints. DE is capable of finding true global solution by few control parameters and fast convergence (Bäck & Schwefel, 1993; Das, Abraham, & Konar, 2008; Vahid, 2015; Vahid, 2016a).

3.3 Differential Search Algorithm (DSA)

DSA utilizes migration of super organism by stable motioning for real valued numerical optimization problems. This multi way method uses the mid-point in journey of the migrating super organism (Das et al., 2008).

3.4 Genetic Programming (GP)

Specific user tasks on computers can be discovered or evaluated by biological evolution methods. GP works because it's possible to optimize a computer population (by fitness function) in order to perform user tasks. GP cannot solve all problems as it's complex, however, it has produced excellent outcomes in computing (Kumar, Husian, Upreti, & Gupta, 2010; Whitley, 1994).

3.5 Evolutionary Programming (EP)

Unlike GP that follows fixed structures, EP doesn't obey any constant pattern because all parents are generators due to extension of membership concept in this method (Bäck & Schwefel, 1993).

3.6 Evolution Strategy (ES)

In ES, mutation and selection are searchers, and normal parameters of problems are used for representations. Evolution of generations in an iterative manner where it's stopped by termination criterion (usually, the expected level of accuracy) forms this strategy (Wierstra, Schaul, Peters, & Schmidhuber, 2008).

3.7 Genetic Algorithm (GA)

Process of natural selection with a heuristic search forms GA by natural evolution characteristics that has applications such as automatic electronic circuit generating (Fogel, 2007; Whitley, 1994; Sachs, Russell, & Hollowell, 2002; Vahid & Ghashghavi, 2016).

3.8 Gene Expressing Programming (GEP)

Tree structures are used in data compression, searching and information processing algorithms. GEP also uses complex trees in order to make adaptable programs by changing almost all properties. Because GEP can use its genome to distribute genetic information, it's called genotype. Ability of adapting to the new environment also classifies GEP as phenotype (Kumar et al., 2010).

4. Swarm Intelligence Algorithms (SIAs)

Previous part briefly discussed EAs and their properties. This part focuses on SI algorithms including 1- ant colony 2- bees algorithm 3- cuckoo bird algorithm 4- particle swarm (Krause, Cordeiro, Parpinelli, & Lopes, 2013; Kennedy & Eberhart, 1995a).

4.1 Ant Colony Optimization (ACO)

ACO is a meta-heuristic optimization procedure primary invented for finding an optimal path in a graph which is based on observations of ants seeking colonies and sources of food. In summary, when one of the members of the colony discovers a short (and possibly safe) path according to its experience, other members converge to that path—as they have already tested and benefited from that path—which results in concentrated rally of ants (Kennedy & Eberhart, 1995b; Mahor, Prasad, & Rangnekar, 2009).

4.2 Bees Algorithm (BA)

Similar to ACO, BA is inspired by bees searching for food and sharing secret places as sources of food with other members as long as they contain enough food. Clearly, these bees will stop advertising those discovered locations if the source of foods has neither enough high quality nor richness (optimized). Therefore, BA is a population-based search algorithm used for combinatorial and continuous optimizations using neighborhood and global searches simultaneously (Kennedy & Eberhart, 1995a).

4.3 Cuckoo Bird Search Algorithm (CSA)

CSA is a simple optimization algorithm inspired by the way tricky cuckoos fighting to reproduce by laying their eggs in the nests of other birds with similar patterns on eggs exploiting the host birds to protect all eggs and therefore increase the chance of Cuckoos' eggs to become chicks (Poli, Kennedy, & Blackwell, 2007).

4.4 Particle Swarm Optimization (PSO)

PSO is an iterative meta-heuristic computational method that solves a problem by improving the possible correct solution (known as candidate solution) with respect to a measure of quality that leads to searching very large spaces of possible solutions with even no assumption about the problem. PSO was largely inspired by birds and fishes. Population of candidate solutions corresponds to swarm of particles that change their position according to some formulas in order to guide the whole swarm for an uncertain and possibly optimized solution. Indeed, swarms try to move to center while trying to match the neighbors' velocities and avoiding collisions (Poli et al., 2007).

5. Classic Real World Applications of EAs and SIAs

However, there exist many real world applications for EC, in this part, some real world contributions of EAs and SIAs are provided in compliance with definitions and statements presented already.

5.1 CC Scheduling by GA

In CC, scheduling is vital for balancing the loads between virtual machine (VM) resources (Vahid & Hedayat, 2016). Usual schedulers in CC follow a static manner considering only current status of the system bringing about imbalances and unequal distribution of tasks and finally failures. GA chooses the best solution, with regard to past and current states of data, resulting in migration's cost reduction (Gan & Huang, 2010).

5.2 CC Scheduling by ACO and BA

Similar to GA based task scheduling in CC, ACO has been proven to be a good load balancer algorithm (LBACO). However, combing ACO and BA, as their swarm structures suggest, can lead to less operational costs as power consumption is also minimized. In fact, the mixture of these two results in more efficient resource management. In such methodologies, service rescheduling which is the appropriate management of under-loaded CPUs is done by BA, and power consumption management which controls idle CPUs is done by ACO.

When several (mostly independent and unrelated to initial) tasks are waiting in a queue in a CC, the loads are not uniformly balanced over all VMs. In this case, honey bee behavior for load balancing (HBB-LB) invented to reduce waiting and execution times (Kun, Gaochao, Guangyu, Yushuang, & Wang, 2011; Dhinesh & Venkata, 2013; Vahid & Hedayat, 2016).

5.3 Grid Load Balancing by ACO

ACO has been used for grid load balancing which protects the nodes from being overloaded by distributing the loads uniformly among all nodes. It means that similar groups of ants (loads) are absorbed by corresponding colonies (nodes) (Ma & Wang, 2012), (Ludwig & Moallem, 2011).

5.4 Image Processing by ACO and BA

Obtaining image features, reducing image segmentation computations, and enhancing signal to noise ratio (SNR) are among applications of ACO and BA (Huang, Cao, & Luo, 2008; Akay, 2013). Transform-invariant pattern recognition by EAs and ANNs is also studied already by the authors in Vahid & Hedayat (2014).

6. Newly Introduced Trends

Last part introduced some of the main and some-how well established applications of EC, but there are some other applications and improvements of older algorithms used in newly introduced trends.

6.1 Improved Image Segmentation Via K-Means Clustering and PSO

Image segmentation can be very practical when data mining procedures need to extract specific features and information from a set of images. Vast amount of visual resources available arises the need to use smart methods on these information as quickly as possible. K-means clustering is a fast algorithm that can build up enough clusters for image segmentation using implementation with low complexities at expense of lower visual qualities. Recently, the use of both dynamic PSO—as a global optimizer—with K-means resulted in better enhancements compared to usual PSO with K-means (PSOK). This method is called dynamic PSOK (DPSOK) (Li, He, & Wen, 2015).

6.2 Scheduling Algorithm for Load Balancing (SALB) in CC Based on ACO

It's possible to enhance grid load balancing via ACO explained already, however the classic algorithm can be improved by some various parameters. SALB tries to improve ACO by considering SLA violation, minimum overhead, and power consumption in multiple nodes balancing of dynamic workloads (Khan & Sharama, 2014).

6.3 Improved Bi-Objective Dynamic Cell Formation Problem (DCFP) by Non-Dominated Sorting GA (NSGA-II)

Dynamic cellular manufacturing system (DCMS) is an important production system fulfilling the needs of flexibility, quickness and efficiency in Industrial Revolution 4.0 standards, however, DCMS ignores some social and other factors. Recently, an innovative method (NSGA II-MOSA) to tackle such issues developed using both hybrid meta-heuristic based on the NSGA-II plus multi-objective simulated annealing (MOSA) is proposed (Deb, Pratap, Agarwal, & Meyarivan, 2002).

6.4 Improvement of Robust Machine Cells for Dynamic Part Production (RMCDPP) by GA

A GA based heuristic can be used for cost reduction by several industrial parameters; for instance, capacity of production, different development routes, working systems ability, and more (Deep & Singh, 2015).

6.5 Integrated Mixed-Integer Linear Programming (MILP) Model to DCMS Problem with Uncertainty

Many of the parameters in control systems, manufacturing, and engineering are well known which may be referred to as deterministic parameters—that were mostly considered in early versions of swarm and other smart methods (Clerc, 1999)—however, there exist multi stochastic parameters with much uncertainty studied in recent papers that can change the path of system completely to failures, high system breakdown and relocation costs, and etc. To deal with such latent parameters in DCMS, MILP model can be employed to assess different levels of uncertain factors and protection during system planning (Sakhaii, Tavakkoli, Bagheri, & Vatani, 2015).

6.6 Robotic Applications

Many tasks assigned to robots include ordinary man's obligations that must be done efficiently by faster means. Industries use robotic arms, smart measuring arrangements, and integrated manufacturing course of decision making processes. These processes are instrumentally interdependent, for that reason, deliberating multi-task considerations can be complicated even by human designer or smart robotic tools. SIAs are employed to handle the processes in different categories, but main studies concern applications on solving by a group of independent robots in a distributed (and not centrally managed similar to swarm) situations. Main lines of control in this case are: 1- accumulation, 2- grouping, 3- searching, 4- object clustering and sorting, 5- system navigation, 6- path shaping, 7- task assigning (Bayindir, 2016). The communication in industries may suffer from noise and other disturbances, particularly when heavy machines with inductors are used such as rotary motors. This problem is solvable by localizing regions in the factory and acquiring data from nearby sensors (Bayindir, 2016; Li, Xu, & Zhao, 2015).

6.7 Power Control in Wind Turbines by GA and PSO

Using wind power as a promising source of alternative energy—by constructing wind farms and appropriate positioning of the turbines in order to maximize capturing of wind energy and generating electricity—is now a well-recognized green method (Lindman & Söderholm, 2015). The art of designing such wind farms is based on minimizing the number of turbines used and maximizing output power by GA and PSO (Lindman & Söderholm, 2015). Wind variations and turbine vibrations during the conversion of wind energy to the electricity can cause resonations leading to unsmooth DC voltage and, because of the reason given, incompatible sinusoidal current on the grid side. More recently, the task of smooth controlling assigned to grid side converter (GSC) can be done

by PSO to engage in On-Off controlling scheme based on maximum power point tracking for the rotator side (Kahla, Soufi, Sedraoui, & Bechouat, 2015). Other conventional control mechanisms methods (Ghasem Khani, Tal'ati, & Rahmati, 2014; Vahid Rahmati, 2013, 2014) can also be done using GA and PSO by means of optimization algorithms.

7. Conclusion

This paper introduced EC, which includes EAs and SIAs, as an important computational tool currently being used largely. It also presented a summary for different EAs and SIAs together with their real world applications that mainly deal with cloud computing, distributed computation, image processing, power control, robotic applications, DCMS, RMCDPP, and DCFP problems. Finally, the paper discussed classic and new ways of applying these algorithms to specific problems such as service scheduling in CC, load balancing in grids, and image segmentation and enhancements.

References

- Akay, B. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing*, 13(6), 3066–3091. <http://doi.org/10.1016/j.asoc.2012.03.072>
- Arya, L. K., & Verma, A. (2014). Workflow scheduling algorithms in cloud environment - A survey. *Engineering and Computational Sciences (RAECS)*, 2014 Recent Advances in, 2(5), 1–4. <http://doi.org/10.1109/RAECS.2014.6799514>
- Bäck, T., & Schwefel, H. P. (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, 1(1), 1–23. <http://doi.org/10.1162/evco.1993.1.1.1>
- Back, T., & Schwefel, H. P. (1996). *Evolutionary computation: An overview*. IEEE International Conference on Evolutionary Computation, Vol. 30, pp. 593–616. <http://doi.org/10.1109/ICEC.1996.542329>
- Bayindir, L. (2016). A review of swarm robotics tasks. *Neurocomputing*, 172, 292–321. <http://doi.org/10.1016/j.neucom.2015.05.116>
- Berstis, V. (2002). Fundamentals of grid computing. *IBM Redbooks Paper*, 1–28.
- Clerc, M. (1999). The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999* (Vol. 3, pp. 1951–1957). <http://doi.org/10.1109/CEC.1999.785513>
- Das, S., Abraham, A., & Konar, A. (2008). Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives. *Studies in Computational Intelligence*, 116, 1–38. http://doi.org/10.1007/978-3-540-78297-1_1
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. <http://doi.org/10.1109/4235.996017>
- Deep, K., & Singh, P. K. (2015). Design of robust cellular manufacturing system for dynamic part population considering multiple processing routes using genetic algorithm. *Journal of Manufacturing Systems*, 35, 155–163. <http://doi.org/10.1016/j.jmsy.2014.09.008>
- Dhinesh, B. L. D., & Venkata, K. P. (2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing Journal*, 13(5), 2292–2303. <http://doi.org/10.1016/j.asoc.2013.01.025>
- Eberhart, & Yuhui, S. (2001). Particle swarm optimization: developments, applications and resources. *Evolutionary Computation*, 2001. *Proceedings of the 2001 Congress on*. <http://doi.org/10.1109/CEC.2001.934374>
- Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 39–43). <http://doi.org/10.1109/MHS.1995.494215>
- Eiben, A. E., & Smith, J. E. (2003). *Introduction to Evolutionary Computing*. *Evolutionary Computation* (Vol. 12). <http://doi.org/10.1162/evco.2004.12.2.269>
- Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 167–181. <http://doi.org/10.1023/B:VISI.0000022288.19776.77>
- Fogel, D. B. (2007). Introduction to Evolutionary Computation. *Modern Heuristic Optimization Techniques:*

- Theory and Applications to Power Systems*, 1–23. <http://doi.org/10.1002/9780470225868.ch1>
- Furht, B., & Escalante, A. (2010). *Cloud Computing Fundamentals*. Handbook of Cloud Computing. <http://doi.org/10.1007/978-1-4419-6524-0>
- Gan Guo Ning, & Huang Ting Iei, G. S. (2010). Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment. *Iciss*, 60–63.
- Ghasem, K. B., Tal'ati, S., & Rahmati, V. (2014). *Distance Measurement by Delay Pulse* (Vol. 10, pp. 40–50). In Shahid Sattari University Exhibition on Electronic Warfare, Tehran, Iran.
- Guo, L., Zhao, S., Shen, S., & Jiang, C. (2012). Task scheduling optimization in cloud computing based on heuristic Algorithm. *Journal of Networks*, 7(3), 547–553. <http://doi.org/10.4304/jnw.7.3.547-553>
- Hornig, M. H. (2010). A multilevel image thresholding using the honey bee mating optimization. *Applied Mathematics and Computation*, 215(9), 3302–3310. <http://doi.org/10.1016/j.amc.2009.10.018>
- Huang, P., Cao, H., & Luo, S. (2008). An artificial ant colonies approach to medical image segmentation. *Computer Methods and Programs in Biomedicine*, 92(3), 267–273. <http://doi.org/10.1016/j.cmpb.2008.06.012>
- Kahla, S., Soufi, Y., Sedraoui, M., & Bechouat, M. (2015). On-Off control based particle swarm optimization for maximum power point tracking of wind turbine equipped by DFIG connected to the grid with energy storage. *International Journal of Hydrogen Energy*, 40(39), 13749–13758. <http://doi.org/10.1016/j.ijhydene.2015.05.007>
- Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Engineering and Technology*, 1942–1948.
- Kennedy, J., & Eberhart, R. (1995a). Particle swarm optimization. *Neural Networks, 1995. Proceedings IEEE International Conference on*, 4, 1942–1948. <http://doi.org/10.1109/ICNN.1995.488968>
- Kennedy, J., & Eberhart, R. (1995b). *Particle swarm optimization*. IEEE International Conference on Particle Swarm Optimization. <http://doi.org/10.1109/ICNN.1995.488968>
- Khan, S., & Sharama, N. (2014). Effective Scheduling Algorithm for Load balancing (SALB) using Ant Colony Optimization in Cloud Computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(2), 966–973.
- Kirchmaier, U., Hawe, S., & Diepold, K. (2013). A Swarm Intelligence inspired algorithm for contour detection in images. *Applied Soft Computing Journal*, 13(6), 3118–3129. <http://doi.org/10.1016/j.asoc.2012.07.017>
- Krause, J., Cordeiro, J., Parpinelli, R. S., & Lopes, H. S. A. (2013). A Survey of Swarm Algorithms Applied to Discrete Optimization Problems. *Swarm Intelligence and Bio-Inspired Computation*, 169–191. <http://doi.org/10.1016/B978-0-12-405163-8.00007-7>
- Kumar, M., Husian, M., Upreti, N., & Gupta, D. (2010). Genetic Algorithm: Review and Application. *International Journal of Information Technology and Knowledge Management*, 2(2), 451–454. Retrieved from [http://www.csjournals.com/IJITKM/PDF 3-1/55.pdf](http://www.csjournals.com/IJITKM/PDF%203-1/55.pdf)
- Kun, L., Gaochao, X., Guanguyu, Z., Yushuang, D., & Wang, D. (2011). Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization. *Chinagrid Conference (ChinaGrid), 2011 Sixth Annual*, 3–9. <http://doi.org/10.1109/ChinaGrid.2011.17>
- Li, H., He, H., & Wen, Y. (2015). Dynamic Particle Swarm Optimization and K-means Clustering Algorithm for Image Segmentation. *Optik - International Journal for Light and Electron Optics*, 126(24), 4817–4822. <http://doi.org/10.1016/j.ijleo.2015.09.127>
- Li, S., Xu, L. Da, & Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, 17(2), 243–259. <http://doi.org/10.1007/s10796-014-9492-7>
- Lindman, Å., & Söderholm, P. (2015). Wind energy and green economy in Europe: Measuring policy-induced innovation using patent data. *Applied Energy*. <http://doi.org/10.1016/j.apenergy.2015.10.128>
- Ludwig, S. A., & Moallem, A. (2011). Swarm Intelligence Approaches for Grid Load Balancing. *Journal of Grid Computing*, 9(3), 279–301. <http://doi.org/10.1007/s10723-011-9180-5>
- Ma, Y., & Wang, Y. (2012). *Grid task scheduling based on Chaotic Ant Colony Optimization Algorithm*. Computer Science and Network Technology (ICCSNT), 2nd International Conference on, 2012. <http://doi.org/10.1109/ICCSNT.2012.6525979>

- Mahor, A., Prasad, V., & Rangnekar, S. (2009). Economic dispatch using particle swarm optimization: A review. *Elsevier*, 13(8), 2134–2141. <http://doi.org/10.1016/j.rser.2009.03.007>
- Marinos, A., & Briscoe, G. (2009). Community cloud computing. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5931 LNCS, 472–484). http://doi.org/10.1007/978-3-642-10665-1_43
- Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, 1(1), 33–57. <http://doi.org/10.1007/s11721-007-0002-0>
- Rahmati, V. (2013). *A New PID Based Mechanism to Control Velocity and Direction of DC Motors*. Islamic Azad University Tehran South Branch.
- Rahmati, V. (2014). *A Novel Low Complexity Fast Response Time PID Controller Based on Fuzzy Logic for DC Motors*. CAI Conference in Shahid Sattari University, Tehran, Iran, pp. 36–45.
- Rahmati, V. (2015). *Improvement of the Approximation Error for Reconstruction of Signals by Asymmetric Sampling with Presentation of a New Turbo Code*. Shahid Sattari University.
- Rahmati, V. (2016a). Improved Interpolation and Approximation through Order Manipulation. *International Journal of Emerging Computing Methods in Engineering*, 1(1), 1–10. Retrieved from <http://icdst.org/OAJ/index.php/IJECME/article/view/4/5>
- Rahmati, V. (2016b). Software Development's Total Cost Evaluation by Applying Complexity Function of Different Orders. *International Journal of Modern Engineering Technologies*, 1(1), 1–5. Retrieved from <http://www.icdst.org/OAJ/index.php/IJMET/article/view/5>
- Rahmati, V., & Ghashghavi, M. (2016). *Intelligent Methods for Analogue Electronic Circuit Design by GA*. Ayandeh Samin Electronics Group.
- Rahmati, V., & Hedayat, A. (2014). A Transform-invariant Pattern Recognition by Evolutionary Algorithms and Artificial Neural Networks. *Simorgh Institute Journal of Intelligent Methods*.
- Rahmati, V., & Hedayat, A. (2016). *How to Manage VM's Resources by GA, ACO, and BA*. Ayandeh Samin Electronics Group.
- Rahmati, V., Tal'ati, S., & Ghasem, K. B. (2014). New Technologies in Electronic Warfare (Vol. 10, pp. 11–20). Shahid Sattari University Exhibition on Electronic Warfare, Tehran, Iran.
- Rahmati, V., Yar, M. H., & Malekijavan, A. R. (2014). Neural Networks New Capacity Factor Measurement for Improvement of SOM. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 3(12), 7–10. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.7664&rep=rep1&type=pdf>
- Rahmati, V., Yar, M. H., Khalilpour, J., & Malekijavan, A. R. (n.d.). *Back Propagation Artificial Neural Network Structure Error Reduction by Defined Factor of Capacity and Algorithm Reinforcement Method*. Retrieved from https://www.academia.edu/26603028/Back_Propagation_Artificial_Neural_Network_Structure_Error_Reduction_by_Defined_Factor_of_Capacity_and_Algorithm_Reinforcement_Method
- Sachs, J. L., Russell, J. E., & Hollowell, A. C. (2002). *Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms*. *PloS one* (Vol. 6). <http://doi.org/10.1371/journal.pone.0026370>
- Sakhaii, M., Tavakkoli Moghaddam, R., Bagheri, M., & Vatani, B. (2015). A robust optimization approach for an integrated dynamic cellular manufacturing system and production planning with unreliable machines. *Applied Mathematical Modelling*, 40(1), 169–191. <http://doi.org/10.1016/j.apm.2015.05.005>
- Shaikh, F. B. F., & Haider, S. (2011). *Security threats in cloud computing*. International Conference for Internet Technology and Secured Transactions, December, 2011, 214–219. <http://doi.org/10.1016/j.sbspro.2011.08.063>
- Subrata, R., Zomaya, A. Y., & Landfeldt, B. (2007). Artificial life techniques for load balancing in computational grids. *Journal of Computer and System Sciences*, 73(8), 1176–1190. <http://doi.org/10.1016/j.jcss.2007.02.006>
- Sun, Y., Wierstra, D., Schaul, T., & Schmidhuber, J. (2009). *Efficient natural evolution strategies*. Genetic And Evolutionary Computation Conference, pp. 7. Retrieved from <http://portal.acm.org/citation.cfm?id=1569901.1569976>

- Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, 4(2), 65–85. <http://doi.org/10.1007/BF00175354>
- Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., & Schmidhuber, J. (2011). Natural Evolution Strategies. *arXiv*, 1–52. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4631255/
- Wierstra, D., Schaul, T., Peters, J., & Schmidhuber, J. (2008). *Natural Evolution Strategies*. 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 3381–3387. <http://doi.org/10.1109/CEC.2008.4631255>
- Younge, A. J., Von Laszewski, G., Wang, L., Lopez Alarcon, S., & Carithers, W. (2010). Efficient Resource Management for Cloud Computing Environments. *International Conference on Green Computing*, 357–364. <http://doi.org/10.1109/GREENCOMP.2010.5598294>
- Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. *Future Generation Computer Systems*, 28(3), 583–592. <http://doi.org/10.1016/j.future.2010.12.006>

Note

Note 1. API (Application Program Interface): interface which enables an application to communicate with the OS and other services provided by it or to create similar user interfaces within computer programs.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).