International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 1 | ISSN : 2456-3307

A Complementary Review on Various Artificial Intelligence Techniques

V. Ramesh*[1], G Raju[2]

[1]Assistent Professor CSE, Sri Indu College of Engineering And Technology, Hyderabad, Telangan, India

2Assistent Professor CSE, Vaagdevi College of Engineering, Hyderabad, Telangan, India

ABSTRACT

Since scheduling process is a very important and complex method, several programmers are searching and working on this issue for years. Still, several researchers within the educational institutes try to seek out the simplest resolution. As time is cash, time improvement is that the most significant purpose, that makes the researchers develop a system for programming in the simplest manner by applying the simplest resolution. Once inspect the assembly line of a works or a number of categories and lecture rooms in a university, shows that having a plan in these places not solely helps regulate things, however conjointly it helps optimize consumption of resources like time and energy among the constraints and limitations. This paper explains and reviews the three techniques that have antecedently been applied to programming domain by researchers and developers among many artificial intelligence techniques. These various techniques i.e. genetic formula, Neural Network and fuzzy logic are outlined, mentioned and compared in terms of some measures.

Keywords: Artificial Intelligence; Scheduling Problem; Genetic Algorithm; Neural Network, Fuzzy logic;

I. INTRODUCTION

Artificial intelligence is technology that is designed to learn and self-improve. It is typically used to solve complex problems that are impossible to tackle with traditional code. In some cases, artificial intelligence research and development programs aim to replicate aspects of human intelligence or alternate types of intelligence that may exceed human abilities in certain respects. The following are common types of artificial intelligence.

We using various types application using through **artificial intelligence Activity Recognition** Determining what humans or other entities such as robots are doing. For example, a car that can see its owner approaching with a heavy bag of groceries may decide to open an appropriate door automatically.

Affective Computing AI that seeks to read and use emotion. Is a type of artificial intelligence that seeks to understand and use emotion? It is the machine equivalent of emotional intelligence in humans. Affective computing has numerous applications related to improving computer-human interactions and interpreting human actions and behaviours.

Artificial intelligence can potentially detect human emotions using cues such as facial expressions, tone of voice, word usage or other inputs such as the way someone is typing or their body temperature. As with any powerful tool, this has potential to improve quality of life or make it worse depending on how it is applied and governed. Affective computing has broad privacy implications as your toaster could theoretically record your emotions and make this information available to companies and/or governments. Alternatively, emotions could be unrecorded and simply used to improve user interfaces such as a synthetic voice that can reflect a range of emotions in its tone.

Artificial Creativity AI applications in fields that are viewed as creative such as music, art and design. Also applies to human-like abilities such as including humour in a conversation.

Artificial Immune System Techniques of machine learning that mimic processes of immune system learning and memory.

Artificial Stupidity The ability for an artificial intelligence to perform sub optimally in certain contexts. For example, a game boot may adapt its intelligence to the skill of the other players in a game.

Automation of decisions or physical tasks using machinery such as robots.

Blockhead A program that simulates intelligence by retrieving information from a data repository. In some cases, products claim to be artificially intelligent as a marketing approach when their software has a tradition design that does not learn in a dynamic way.

BotA robot that takes no physical form.

Chatterbot An artificial intelligence that can talk to humans, often over text chat. Typically designed to pass a Turing test.

Committee Machine A committee machine is a neural network that combines results or learning from other neural networks. It is an approach to scaling AI.

Computer Automated Design Most computing is currently focused on automation and calculation. Artificial intelligence may support expansion of this role into areas that are thought to require creativity such as design and architecture.

Computer Vision Analysing and understanding visual information is a reasonably complex task that typically requires artificial intelligence.

Decision Support System The use of artificial intelligence to support human decision-making. For example, a tool that determines what information you may need in a given situation.

Embodied Agent An artificial intelligence that has a body either physical or simulated.

Ensemble Learning A machine learning technique that uses multiple learning algorithms.

Evolutionary Algorithms A class of algorithms inspired by processes of biological evolution such as reproduction, mutation, recombination and selection.

Friendly Artificial Intelligence The study of techniques to make AI safe and useful. It is commonly argued that artificial intelligence represents an existential risk.

Intelligence Amplification AI tools that are designed to augment human intelligence. For example, a tool might identify options for problem solving.

Machine Learning Algorithms that learn from data are a fundamental difference between AI and traditional software.

Natural Language Processing The ability to recognize interprets and synthesizes speech.

Neural Network Artificial neural networks are an AI approach that was originally inspired by biological neural networks. With time, their resemblance to biology has decreased and they are now typically based on statistics and signal processing techniques.

Neuro Fuzzy A neural network designed to think in fuzzy logic.

Recursive Self-Improvement Software that writes its own code in cycles of self-improvement. Considered somewhat risky.

Self-Replicating Systems Software easily self replicates with operations such as file copy. It is also possible for robots to self-replicate. This has practical applications such as space exploration.

Sentiment Analysis Tools that determine the general opinion, emotion or attitude in content such as a comment in social media.

Strong Artificial Intelligence Research into human-like artificial intelligence.

Super human an ability where machines surpass the greatest human achievements. For example, AI can generally beat humans at chess.

Super intelligence A theoretical AI that surpasses human intelligence in a comprehensive way.

Swarm Intelligence Large numbers of small self-organizing units that act locally to generate collective intelligence.

Symbolic Artificial Intelligence An old-fashioned approach to AI that restricts computation to a set of symbols that resemble a language of logic.

Technological Singularity A hypothetical scenario whereby artificial intelligence rapidly improves by many orders of magnitude using techniques such as recursive self-improvement. Generally thought to result in a super intelligence that humans cannot understand or control.

Weak Artificial Intelligence Practical applications of artificial intelligence that typically focus on solving a problem with software that can learn from real world data.

II. LITERECTURE SURVE

There are many intelligent techniques, that are developed and researched by an enormous range of researcher and developer.Definitely the developed techniques, that are desired by biological evolution method and human brain, have far better proficiency to propose and notice a better resolution. That is the reason, that Genetic algorithmic rule, fuzzy logic, and Neural Network is designated to review on the programming domain, though there are different techniques with this feature likewise. "In several domains restricted resources got to be optimally utilized and also the construction of fine schedules is commonly the principal suggests that of achieving this goal [1].

As explicit by le Pape, (1994), programming is that the method of distribution activities to resources in time. A method involves intensive decision-making, in brief the method of formulating a schedule. This method is affected and influenced by variety of constraints. These constraints govern and sometimes reduce the space of permissible solutions [2].

Commonly, programming issues are NP-hard, suggests that however there are notable algorithms to find optimum solutions in multinomial time. There are unit algorithms to solve the problem but they take too long once the matter size increases or some additional constraints add. Therefore, most researches are taken to either simplifying the scheduling problem to the purpose wherever some algorithms will notice solutions, or to create economical heuristics to seek out appropriate solutions. In some cases, the matter could contain of merely finding a possible resolution, and regularly having a possible resolution might not be assured [3]. There is a unit range of application of programming drawback in industries. [4] Planned a way for determination berth allocation drawback with biological process algorithms.

Genetic algorithms are universally ascribed to Holland [5] and his students within the Nineteen Seventies, though evolutionary computation dates back more, that Fogel [6] consummated an in depth review on. Genetic algorithms are random meta-heuristics that imitate some options of biological evolution. Canonical GAs were not meant for operate optimization, as mentioned by DE Erica Jong [7]. However, slightly changed versions tested winning. Several samples of winning implementations are found in Back [8], Chaiyaratana and Zalzala et al [9]. In the late years, Genetic Algorithms have been increasingly well liked to solve difficult optimization problems such as

those found within the areas of programming or timetabling. There is no pre-defined path that consist constraints into GAs.

This is probably one of their biggest weaknesses, as it does not build them promptly Amenable to most real world optimization issues.However, as noted by Michalewicz [10], their application and success is problem specific [11].

Genetic algorithm as a solution method and technique has been applied on programming domain by many researchers, as the nature of scheduling issues is very complicated and synthetic.

GA has capability to solve nonlinear and combinatorial issues and typically work well on the issues within which the target and /or search area mix each distinct and continuous variables. They additionally use for looking out massive, multimodal spaces effectually as they care for a population of people instead of one individual (solution) and use no gradient or different problem-specific data [3].

A mixture of strength and suppleness is that the reason of GA success at a massive and ever growing varies of programming drawback. The strength comes from the experimental proved ability of evolutionary algorithms to effectively notice globally competitive optimum in massive and complex search areas. The desirable scaling of biological process algorithms as a function of the dimension of The search space makes them exclusively efficient in compare to different search algorithms for the massive search spaces typical of universe programming [12].

Since programming issues are terribly process and complicated problems that mostly get

highly domain dependent features, they require highly domain Particular optimization algorithms. Genetic algorithmic rules anyhow separate the optimization algorithm from domain data that has caused promising and helpful results in the programming domain. The approach GAs use is separating a roughly domain freelance and extremely sturdy optimization technique (mutation and crossover operators apply to population of individuals) from the domain special aspects of a tangle (the analysis operate for the chromosome) [13].

As noted by Zhang and Huang(1999),the Simplicity of Neural Networks and their capability to perform distributed process in conjunction with their ability for

learning and generalization, have made them a popular Methodology, allowing them to be used in several real life applications [14].Hand written letters recognition, Facial expressions, finding an optimum route for a commercial traveler problem and programming a job-shop area unit simply a number of instances of issues that may be sole with success victimization NN neural networks. Cheung (1994) numerous NNs architectures such as looking out network (Hopfield net), error correcting network (Multi-Layer Perception), probabilistic network (Simulated Annealing),competing network and self - organizing network were represented by Cheung (1994) to solve the programming issues [15]. because the general programming drawback is associate NP-exhausting combinatorial optimization drawback, COP,(Pullman 1975), massive issues still gift large sensible difficulties. Fi attack complex combinatorial optimisation problems (Hopfield 1982, 1984;

Hopfield & Tank 1985, 1986; Tank & Hopfield 1986; see but Wilson & Pawley 1988) redoubled the priority within the potential use of those networks for programming. Neural networks are inherently parallelizable and in principle can be used to solve massive issues and have been thought-about for a selection of programming issues consisting adjective management of packet switched pc communication networks (Mars 1989), integrated scheduling of producing systems (Dali & Lamer 1989), optimization of parallelizing compilers (Kalahari 1990), designing and programming in region comes (Ali 1990). Ultimately however not least time period control systems for manufacturing applications (Smith et beer 1988) and area mission programming (Gasping 1989) [1].

To solve the programming drawback, very different programs are developed. Linear programming generally uses to specify the value operate supported the actual programming drawback. The job-shop programming translated into an applied mathematics format by Willems and Rooda. Then mapped into associate applicable NN structure to make an answer [16].

In addition, Foo and Takefuji utilize integer linear programming neural networks for programming drawback determination by minimizing the whole starting times of all the roles with a priority constraint [17].Furthermore, a neural network technique derived from linear programming by Zhang, Yan, and Chang, within which preventative jobs area unit scheduled based mostly on their priorities and point in time [18].Moreover, Hamada and Ohnishi [19] based mostly on neural network,created a parallel algorithm for pre-emptive Task scheduling issues via authorizing a task transfer across achiness. Finally, a classical native search Heuristic algorithmic rule embedded into the TSP optimization neural network y Park [20].

Hopfield and Tank used neural network likewise to solve the optimization problems [21]. Basically the Hopfield neural networks hand in glove decide vegetative cell output state data based mostly on the state input information from a community of neurons. Per neuron swaps data with different neurons within the network. The neurons exert this data to bring the network to achieve convergence. Inherently, the Hopfield neural network Operation is a relaxation method that permits associate energy operates to realize associate optimized resolution with less computation. The mentioned energy operate is associate applicable Lyapunov operate [22]. Fuzzy pure mathematics has been applied to make hybrid-scheduling approaches [23].

Fuzzy logic may be a figure of ambiguous logic that derived from fuzzy pure mathematics to cope with approximate reasoning instead of precise.
In contrast with "crisp logic", wherever binary sets have binary logic, the formal logic variables could have a membership worth of not solely zero or 1that is, the degree of truth of noun cement will vary between zero and one and is not forced to the two truth-values of classic propositional logic. A fuzzy logic approach application will optimize multiplex goals and then reach a nearer similarity to the important world [24].

Modeling of incorrect programming data is permissible by fuzzy logic, with linguistic variables, which outlined via membership functions showing the accuracy degree of information the info the information} and also the reasoning regarding the wrong data by using fuzzy rules [25].Fuzzy logic that as introduced by Sade (1965) has been utilized to many industrial problems [26]. Fuzzy logic system approach privilege is incorporating each numerical result from a previous resolution or simulation and the programming experience from experiences or observation, and that is

convenient to implement. Numerous programming systems supported fuzzy logic have freshly been created, though direct comparisons between them do not seem to be straightforward thanks to their totally different implementations and objectives [27].

III. COMPARATIVE ANALYSIS

In this section the selected techniques have been explained and discussed in terms of their performance, accuracy in making decision and velocity as following.

A. Genetic Algorithm

GA as a subfield of Evolutionary Algorithms is a quickly growing area of Artificial Intelligence, which has been inspired by the process of natural evolution.

In genetic algorithm to solve the problem, a set of stochastic operators iteratively applies on a population of candidate solutions, which are known as individuals to get the better solutions. They are also called creatures or phenotypes.

In this technique, after generating a primary population of candidate solutions, at the first step, the fitness of all solutions gets apprised then modifying would be done by applying the stochastic operators, which are reproducing the most successful solution (Selection), recombining discrete solutions (Crossover) and randomly mutate solutions (Mutation). The fitness of improved solutions apprises as well and then a new population produces and the same procedure is applied on [20].

Generally, this process ends when either a maximum number of generations has been generated, or a required fitness rate has been attained for the population. If the algorithm has closed because of a maximum number of generations, a most successful solution may or may not have been achieved. Genetic Algorithm specifically was reputed and developed by John Holland in the early 1975, and in the University of Michigan [28].

This algorithm is well suited to apply on hard problems such as optimization problem and in various domains like Scheduling, Business, Engineering, etc., because unlike heuristic technique, GA is applied on a population of candidate solutions (individuals) rather than a single solution (individual).

In genetic algorithms, often the solution times significantly enhance or the quality of final solutions reduces as they operate on large and complex problem. This weakness, necessitate the creating a well-suited GA which is able to solve complicated problems fast, reliably and accurately [29].

In other hand, due to GAs inherent parallelism, they apply on a population of candidate solutions rather than a single solution and that is the reason, which however this algorithm is suitable to utilize on complex and hard problems such as optimization problem, and this could be very helpful to decrease the length of time and increase the quality. Genetic algorithm parallel implementations are held in common, and they do well

To decrease the needed time to get acceptable solutions. But, as the parallel GAs are governed by many parameters that their effects on the quality of their search and on their proficiency are not deep understood, the capability to develop an accurate and fats parallel genetic algorithm that achieve the optimal solution in the shortest time, is restricted [30].

Commonly, either genetic algorithms process gets over when a maximum number of generations have been generated, or a required fitness rate has been attained for the population. If the algorithm has closed because of a maximum number of generations, a most successful solution may or may not have been achieved. This lengthens the taken time to reach the solution and makes GAs to be considered as a time consuming procedure [31].

Many efforts have been done to make genetic algorithm quicker, and one of the good and promising alternative is using parallel implementation of genetic algorithms, which has got essential to achieve high quality solution in shortest time possible. Since the parallel nature of genetic algorithms identified, a large number of developers have applied it successfully to decrease the needed time to approach an acceptable solution for hard and complicated problem. Genetic algorithms can frequently be used as optimizers, and are not guaranteed to converge optimal solution, but the chance of success is

increasable where there is an input parameters' accurate manipulation as well as enough representation for problem. The population size is very significant as it effects on GAs ability to achieve good solution and on the taken time to approach. If population size were very large, GA would waste computational sources to process unessential and irrelevant individuals. In case of having too small population size, the solution space would not be able to be sampled appropriately and will be hard to recognize the good solution [32].

B. Neural Network

Since centuries ago, human has been attempting to realize the bio physiology of human brain as the features such as intelligence, creativity, learning ability, flexibility, and generalization.

Whereof the algorithmic methods were not suited for implementing these traits, there was a basic need to have a well-suited method for implementation, which is designed, based on human behaviours and imitates natural neural networks in different systems analysis, decision making and solving the problems.

Neural Networks, which have analysis and learning power, composed of a huge anthology of many connected parallel processors, which are called neurons. Artificial neurons calculate its input weighted sum then if the acquired content exceeds threshold (bias), it gets active. The neurons operate coordinately to solve the problems and transfer the information using synapses. In fact, synaptic connection between neurons is used to store the obtained knowledge.

As a matter of fact, neural networks are able to learn from experience to progress their proficiency as well as adapting to environment fluctuations. A network once is created for a specific application, gets able and ready for training. This process begins by randomly choosing the elementary weights. The training or learning will start using two approaches, supervised learning and unsupervised learning.

Nowadays, in the developed countries, everything is based on electronic and automatic systems and the paper-based process are roughly obsolete. For instant, in manufacturers to check the workers, paper-based processes are not used anymore and electronic based systems have replaced them [28] [22].

Artificial Intelligence connectionist systems are based on interconnected artificial neurons networks, which imitate brain neuronal networks [33].

Artificial neural networks are largely parallel divided processors, which are created of simple processing units. They are inherently able to adapt to particular problems, learn, and generalize. The computational potency and capability of neural networks are raised of this ability as well as their parallel-distributed structure. These traits make artificial neural networks able to solve complicated problem [34].

High quality decision making is an important goal of the Navy but they need a working model that is capable of making decisions similarly to a human detailer under time pressure, uncertainty, and is able to learn/evolve over time as new situations arise and new standards are created. For such task, clearly, an intelligent agent and a learning neural network are better suited"[35].

Neural network can be a well-suited tool for decision making as a classification problem. They are able to learn making human-like decision and naturally follow alterations within data set as the environment variations [35].

The taken time to process and reach the result depends on the size of neural network. If the neural networks were massive, high processing time would be required. Also sometime the training time takes longer [36].

C. Fuzzy Logic

FL which was introduced by Lofti Zadeh, a professor at the University California, in his paper "Fuzzy Sets'' in 1965, is a problem-solving control system methodology and one of the strongest tools to scheme the independent intelligent systems [26]. It is very useful for solving the problems that are not easy to model mathematically. FL can be operated on problem that is based on vague, imprecise and incomplete data, and this is its very positive point.

In a Fuzzy Logic process, a crisp set of input data congregates and transforms to a fuzzy set with a reliable set of inference rules during fuzzification step,

and then during the defuzzification process the generated outputs get converted into a crisp set using a membership function [37].

FL is well suited for many control system applications as it mimics human control logic. In fact, to control the problems imitates how a man makes decision, only speedier. In FL unlike classical logic, a statement can have a truth value between 1 and 0, all real numbers from 0 to 1, rather than having a truth value of either 1 for true or 0 for false. The article "A partly true story", which is written by Ian Stewart, is very helpful to get more about the Fuzzy Logic and its concepts [38].

Fuzzy Logic is one of strongest tools to develop autonomous intelligent systems, and well suited to solve the problems which are hard to model mathematically. The most potency of fuzzy logic is upon its capability to pull out the outcome and produce responses in terms of vague, imprecise, and defective qualitative data. Furthermore, fuzzy logic is able to perform well to reach load balancing in both simulation and real-time implementation [39].

Generally, load balancing consists three parts. The first one is information rule that demonstrates the set and storing processes of applied information to make the decision. The next one is transferring rule that decides when to intern an effort to transfer a task or not to transfer a task. The last one is location rule that select the processing elements to and from which task would be transferred. Achieving good load balancing will increase the accuracy of making decision and decrease the taken time to approach the solution [40].

Moreover, FL is capable to reach stable situation in a shorter time distance and less values, rules, and decisions are required [41].

IV. CONCLUSION

This paper represented a study on the programing problems using 3 techniques of artificial intelligence (AI). Genetic algorithm (GA), fuzzy logic (FL), and Neural Network (NN), is incredibly complicated and attractive techniques that are chosen to be into thought during this work on programing domain. As a matter of truth, improvement issues like scheduling downside has been into account by a huge variety of researchers and developers because the importance of your time and creating the correct call is incredibly touchable. Time improvement during this age which period is money may well be a lot of facilitates to extend the potency of all sections that are attached time constraints. Therefore, solving this problem that is NP-Complete and a hot domain needs intelligent powerful techniques that use difficult and onerous search area. Still, there are miles to go for locating the best solution using completely different intelligent techniques and customised algorithm. Sure enough AI ways will be terribly effective and helpful to succeed in the target purpose. The out there techniques don't have all the benefits and strengths along and some disadvantages cut back their potency. Scheduling issues build the likelihood and potential for innovative solutions by intelligent techniques like GA, NN, FL, and etc. further techniques should be ready to meet all requirements at the same time to offer the best solution for the shortest time period.

V. REFERENCES

[1]. M. D. Johnston and H.-M. Adorf, "Scheduling with neural networks—the case of the Hubble Space Telescope," Computers & operations research, vol. 19, pp. 209-240, 1992.

[2]. P. Baptiste and C. Le Pape, "A theoretical and experimental comparison of constraint propagation techniques for disjunctive scheduling," in IJCAI (1), 1995, pp. 600-606.

[3]. M. B. Wall, "A genetic algorithm for resource-constrained scheduling," Massachusetts Institute of Technology, 1996.

[4]. M. M. Kashani, and M. Ayob, "Reduction Operators for Magnetic Optimization Algorithm," Journal of Applied Sciences, vol 14, p. 3446-3454, 2014.

[5]. J. H. Holand, "Adaptation in natural and artificial systems," Ann Arbor: The University of Michigan Press, 1975.

[6]. D. B. Fogel, Evolutionary computation: the fossil record: Wiley-IEEE Press, 1998.

[7]. K. A. De JOD, "Genetic Algorithms Are NOT Function Optimizers."

[8]. T. Back, "Applications of Evolutionary Algorithms, extended edition," University of Dortmund, 1993.

[9]. N. Chaiyaratana and A. Zalzala, "Recent developments in evolutionary and genetic

algorithms: theory and applications," in Genetic Algorithms in Engineering Systems: Innovations and Applications, 1997. GALESIA 97. Second International Conference On (Conf. Publ. No. 446), 1997, pp. 270 -277.

[10]. Z. Michalewicz, "A Survey of Constraint Handling Techniques in Evolutionary Computation Methods," Evolutionary Programming, vol. 4, pp. 135-155, 1995.

[11]. U. Aickelin and K. A. Dowsland, "An indirect genetic algorithm for a nurse-scheduling problem," Computers & operations research, vol. 31, pp. 761-778, 2004.

[12]. D. Montana, M. Brinn, S. Moore, and G. Bidwell, "Genetic algorithms for complex, real-time scheduling," in Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on, 1998, pp. 2213-2218.

[13]. H.-L. Fang and D. Bu, "Investigating genetic algorithms for scheduling," 1992.

[14]. Y.-M. Huang and R.-M. Chen, "Scheduling multiprocessor job with resource and timing constraints using neural networks," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 29, pp. 490-502, 1999.

[15]. A. S. Jain and S. Meeran, "Job-shop scheduling using neural networks," International Journal of Production Research, vol. 36, pp. 1249-1272, 1998.

[16]. T. Willems and J. Rooda, "Neural networks for job-shop scheduling," Control Engineering Practice, vol. 2, pp. 31-39, 1994.

[17]. F. Y.-P. Simon and Y. Takefuji, "Integer linear programming neural networks for job-shop scheduling," in Neural Networks, 1988., IEEE International Conference on, 1988, pp. 341-348.

[18]. C.-s. Zhang, P.-f. Yan, and T. Chang, "Solving job-shop scheduling problem with priority using neural network," in Neural Networks, 1991. 1991 IEEE International Joint Conference on, 1991, pp. 1361-1366.

[19]. A. Hanada and K. Ohnishi, "Near optimal jobshop scheduling using neural network parallel computing," in Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON'93., International Conference on, 1993, pp. 315-320.

[20]. J. G. Park, J. M. Park, D. S. Kim, C. H. Lee, S. W. Suh, and M. S. Han, "Dynamic neural network with heuristics," in Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on, 1994, pp. 4650-4654.

[21]. J. J. Hopfield and D. W. Tank, "“Neural” computation of decisions in optimization problems," Biological cybernetics, vol. 52, pp. 141-152, 1985.

[22]. R.-M. Chen and Y.-M. Huang, "Competitive neural network to solve scheduling problems," Neurocomputing, vol. 37, pp. 177-196, 2001.

[23]. D. W. Sellers, "A survey of approaches to the job shop scheduling problem," in Southeastern Symposium on System Theory, 1996, pp. 396-396.

[24]. R. Ramkumar, A. Tamilarasi, and T. Devi, "Multi Criteria Job Shop Schedule Using Fuzzy Logic Control for Multiple Machines Multiple Jobs."

[25]. T. Bäck, D. B. Fogel, and Z. Michalewicz, Evolutionary computation 1: Basic algorithms and operators vol. 1: CRC Press, 2000.

[26]. L. A. Zadeh, "Fuzzy sets," Information and control, vol. 8, pp. 338-353, 1965.

[27]. P. Srinoi, E. Shayan, and F. Ghotb, "Scheduling of flexible manufacturing systems using fuzzy logic," International Journal of Production Research, vol. 44, pp. 1-21, 2002.

[28]. A. Khajanchi, "Artificial Neural Networks: The next intelligence," USC, Technology Commercalization Alliance. http://www. usc. edu/org/techalliance/Anthology2003/Final_Khajanch. pdf, 2003.

[29]. A. Petrovski, A. Wilson, and J. McCall, "Statistical identification and optimisation of significant GA factors," in Proceedings of the 5th Joint Conference on Information Sciences. Atlantic City, USA, 2000, pp. 1027-1030.

[30]. E. Cantu-Paz, "Designing efficient and accurate parallel genetic algorithms," 1999.

[31]. Y.-S. Chen and C. Shahabi, "Automatically improving the accuracy of user profiles with genetic algorithm," in Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing, 2001, pp. 283-288.

[32]. E. Cantu-Paz, Efficient and accurate parallel genetic algorithms vol. 1: Springer, 2000.

[33]. A. B. Porto-Pazos, N. Veiguela, P. Mesejo, M. Navarrete, A. Alvarellos, O. Ibáñez, A. Pazos, and A. Araque, "Artificial astrocytes improve

neural network performance," PloS one, vol. 6, p. e19109, 2011.
[34]. E. Laskari, G. Meletiou, D. Tasoulis, and M. Vrahatis, "Studying the performance of artificial neural networks on problems related to cryptography," Nonlinear Analysis: Real World Applications, vol. 7, pp. 937-942, 2006.
[35]. A. Kelemen, Y. Liang, and S. Franklin, "Learning high quality decisions with neural networks in" conscious" software agents," WSEAS Transactions on Systems, vol. 4, pp. 1482-1492, 2005.
[36]. D. Whitley, "Genetic algorithms and neural networks," Genetic algorithms in engineering and computer science, vol. 3, pp. 203-216, 1995.
[37]. S. Salleh, "Task Allocation In A Multiprocessor System Using Fuzzy Logic," Jurnal Teknologi, vol. 25, pp. 69-79, 1996.
[38]. I. Stewart, "A partly true story," Scientific American, vol. 268, pp. 110-112, 1993.
[39]. S. Salleh, B. Sanugi, and H. Jamaluddin, "Fuzzy logic model for dynamic multiprocessor scheduling," Matematika, vol. 15, pp. 95-109, 1999.
[40]. H.-C. Lin and C. S. Raghavendra, "A dynamic load-balancing policy with a central job dispatcher (LBC)," Software Engineering, IEEE Transactions on, vol. 18, pp. 148-158, 1992.
[41]. V. B. Rao and H. V. Rao, "Neural networks and fuzzy logic," MIS: Press, New York, 1995.
[42]. A. Marczyk, "Genetic algorithms and evolutionary computation," The Talk Origins Archive: http://www. talkorigins/faqs/genalg/genalg. html, 2004.
[43]. M. J. Heijligers, L. Cluitmans, and J. A. Jess, "High-level synthesis scheduling and allocation using genetic algorithms," in Design Automation Conference, 1995. Proceedings of the ASP-DAC'95/CHDL'95/VLSI'95., IFIP International Conference on Hardware Description Languages. IFIP International Conference on Very Large Scal, 1995, pp. 61-66.
[44]. K. Shaw, A. Nortcliffe, M. Thompson, J. Love, P. Fleming, and C. Fonseca, "Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem," in Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, 1999.
[45]. J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," Journal of clinical epidemiology, vol. 49, pp. 1225-1231, 1996.
[46]. A. K. Jain, J. Mao, and K. Mohiuddin, "Artificial neural networks: A tutorial," Computer, vol. 29, pp. 31-44, 1996.
[47]. S. Noronha and V. Sarma, "Knowledge-based approaches for scheduling problems: A survey," Knowledge and Data Engineering, IEEE Transactions on, vol. 3, pp. 160-171, 1991.