Artificial intelligence and business: A hybrid genetic algorithm for e-business strategic planning and performance evaluation

Anastasia-Dimitra Lipitakis
Department of Informatics and Telematics
Harokopio University, Athens, Hellas

Evangelia A.E.C. Lipitakis
Kent Business School, University of Kent, Canterbury, United Kingdom

Key-words
Artificial intelligence, businesses, e-business, strategic planning, performance evaluation, algorithmic Methodologies, evolution process, genetic algorithms

Abstract

In this research study the topic of Artificial Intelligence and Business with emphasis on Genetic Algorithmic approach for e-business strategic planning and performance evaluation is examined. Several AI methodologies and their usage in large organizations in various applications, such as staffing, careers, promotions, hiring and hiring predictions are discussed. The concept of genetic algorithms with their several important aspects is discussed and various related applications in a wide spectrum of topics are presented. An e-business strategic planning and performance evaluation model has been synoptically presented and a modified genetic e-business model has been introduced. Various genetic programming implementations with the corresponding software products are given and several AI software packages, genetic algorithmic methodologies and related applications are presented. The proposed models can be used by practitioners who may use them for increasing their organization's performance and also by theoreticians for academic purposes by developing e-business strategic management, strategic planning and performance evaluation.

1. Introduction

Artificial Intelligence (AI) software provides appropriate solutions for several organizations and businesses by speeding processes of online simulations of what software includes multiple choice questions and captures several human qualities using natural language processing and machine learning (ML) methodologies for constructing psychological profiles of suitable candidates. ML methodologies and related predictive algorithms can be used as tools for identifying best candidates and assessing certain human qualities by analyzing special characteristics such as word choice, gestures, emotional traits, facial expressions, body language, social media posts etc. Furthermore, human judgment, face to face interaction and personal communication can be used. Better communication with potential applicants can be achieved by special AI programs allowing the scanning of millions of job opening s for uncovering connections between job performance and certain attributes by applying ML models and analytics methods. The resulting search results can be matched the intents of job seekers and special software can be applied by doing searchers on the related pages of internet. The optimal result of algorithmic and ML methodologies usage is to match job applicants to the available jobs in a better way than actual human endeavor can do.

Genetic algorithms can be used for e-business problems, particularly strategic planning and performance evaluation, leading to improved overall performance of large organizations. A new e-business strategy planning and performance evaluation scheme based on adaptive algorithmic modelling techniques, the so-called *Lipitakis-Phillips model* (Lipitakis and Phillips, 2013) is used in conjunction to Genetic Algorithms for improved performance.

The structure of this research work is as follows: in section 2, the topic of AI algorithms and Organization staffing, careers and hiring is examined. In section 3, several basic elements of Artificial Intelligence Algorithms and Business are discussed, while in section 4 Genetic Algorithms and Business with generic and genetic algorithms are discussed. In section 5, an e-business model for strategic planning and performance evaluation *(Lipitakis and Phillips model)* is synoptically presented and in section 6 a genetic modified e-business algorithmic scheme for strategic planning is introduced. Finally, in section 7, the Genetic Algorithms and applications demonstrating both the applicability and efficiency of GA's, genetic programming implementations and software products are presented, and the relationship of GA's and evolution process is discussed.

AI Algorithms and Organization Interoperability: staffing, careers and hiring

It is reported that Google for the production of appropriate software scanned millions of job openings in order to uncover several connections between certain attributes and job performances, then applied ML methodologies and applied analytics. This software can make certain postings more visible to people doing several Internet searches. Several researchers support that job interviews can be useless undercutting the impact of more valuable information about interviewees and most interviews are considered to be waste of time because most of the time is spent trying to confirm the impression formed in the first few seconds of the interview. Some of the reviews of this approach have been tested in elite academic schools (MIT, Stanford) and the internal evidence showed that test scores, grades and school's pedigree weren't an appropriate predictor of job success. Based on that Google started using a special algorithmic procedure (qDroid program) that help identifying relevant characteristics, such as intellectual humility, cognitive abilities, learning abilities, skills, related experience, provided by applicant's data according to predetermined interviewing questions.AI methodologies have the abilities to deal with such data examining multiple variables and finding various patterns that common interviewers might not trace. Note that the supervised learning can use variables that should be weighted based on certain qualities of high performers. AI methodologies can use computing power for processing huge quantities of data, automate analysis of job candidates identities, detecting online certain personal characteristics and world views. Although the candidate social media information about certain personal information, such as political affiliation, race, religion, sexual orientation is illegal leading to hiring discrimination, these may be of critical importance to employment decisions. AI methodologies can quickly trace many social media posts and web articles by analyzing them (by using comparative analytics), reading relevant data, texts, images, videos, while shielding employers from liabilities.

AI methodologies can be used by several organizations and companies in various applications, such as law, travel, retail, pharmaceuticals, call centers etc. Can be also used in order to predict if prospective candidates will be adapt with organization's culture and/or remain with the organization for significant time. Special AI services can be used for predicting individual turnover and performance based on several words and phrases used by people listed as references, who are presenting with online series of behavioral questions tailored to specific jobs. The input of such procedures is then graded, averaged and the results compared with databases of many other candidates for the same position providing useful insights into how the applicant is compared with other applicants (Skill Survey AI services).

AI methodologies can be used in similar way for predicting which college/university graduates will be hiring as investment bankers and ensure diversity fitting its culture and examining the likelihood of staying with the organization. This procedure is using predictive analytics software for a few minutes survey (Citigroup/Koru AI software). Such AI software packages can examine several related variables revealing certain qualities, such as applicant persistence, patterns in between the available data, organization past tendencies concerning recruitments. These collected data on retention, hiring, performance can be added (as learning process) in the AI software data bases. Several other related qualities, such as emotions, and other characteristics concerning intents, habits, special qualities, personality can be included for comparative purposes with an 'atlas of emotions' containing thousands of facial expressions available in fractions of a second (HireVue AI algorithms).

Machine Learning-Genetic Algorithms and Business

Introductory Remarks

In recent times the field of Artificial Intelligence (AI) has significant impact on financial services sectors and global financial markets comparing with sophisticated classes of algorithms for trading by creating a new interesting applied research topic for traders and regulators (McGrath, 2016). It is reported that algorithmic trading systems are handling about 75% of the volume of global trades (Thomson Reuters, 2016-2017) with industrial predictions showing a constant increase mainly due to the following reasons: (i) huge product demand and several innovations, (ii) greater automation of trades in several asset classes (EU-MiFID II), (iii) retail trading market expansion worldwide opening new algorithmic trading and increasing demand for technological advancement.

Algorithmic procedures can be efficiently used for computing, data processing; automated reasoning and sophisticated algorithmic applications play a significant role in sifting through masses of data and have a widespread use in everyday life (Hickman, 2013). Note that certain computer algorithms are designed for allowing computers to learn on their own facilitating machine learning (ML) and including data mining, pattern recognition and other related applications (Finley, 2014). The inventor of World Wide Web (WWW) stated characteristically that 'Data is a precious thing and will last longer than the systems themselves'.

The field of natural language processing (NLP), an interesting area of ML, uses several applications ranging from text understanding to processing and comprehending the contents of given photographs. ML algorithmic methodologies are given 'data teaching sets' and then are using these data for answering certain questions. ML can continue to add to its teaching sets aiming to completion of their tasks over time. The learning process is related to the following topics: Data security, personal security, financial trading, fraud detection, financial trading, healthcare, marketing personalization, recommendations, natural language processing, online search, smart cars.

3.2 The structure of a simple Genetic Algorithm

Genetic algorithms, a branch of evolutionary algorithms originated from the Darwinian evolution principles, are heuristic modelling techniques inspired be natural evolution and can be efficiently applied to wide spectrum of real world of significant complexity. These computational techniques can lead to optimized solutions of certain complex problems (McCall, 2005).

Let us consider a given well defined given problem for solution and a bit string representation for candidate solutions. Then a simple genetic algorithm can be presented as follows:

Algorithm GA1 (GPC, FCP, ISO, RNP, BSGA)

Purpose: This algorithm describes the basic steps of a Generic Algorithm

Input: randomly generated population chromosomes GPC, fitness of population chromosome FCP, iterative steps for offspring ISO, replacement of new population RNP

Output: basic steps of a GA (BSGA)

Computational Procedure:

Step 1: chose a randomly generated population of (n-1) bit chromosomes (i.e. candidate solutions) GPC

Step 2: compute the fitness f(x) of each chromesome x in the population FCP.

Step 3: repeat the following iterative steps until n-offspring ISO have been created:

Step 3.1: chose a pair chromosomes from current population with selection probability being an increasing function of fitness. Note that the selection process can be done with replacement, i.e. the same chromosome can be selected more than one time in order to become parent.

Step 3.2: by considering the crossover probability (crossover rate) pc, cross over the selected pair at a randomly chosen point (chosen with uniform probability) to form two offspring. In the case that there is no crossover then form two offspring that are exact copies of their respective parents. The crossover probability is defined as the probability that two parents will cross over in single point. The genetic algorithm can have multi-point crossover versions in which the crossover rate for a pair of parents is the number of points at which a crossover takes place.

Step 3.3: mutate the two offspring at each locus with mutation probability pm and transfer the resulting chromosomes in the new population. If n is odd, then one new population member can be randomly discarded.

Step 4: replace the current population with the new population NEP.

Step 5: has the computational work finished?

If yes, then go to step 6, otherwise go to step 2

Step 6: form a list of the basic steps of a Generic Algorithm BSGA

The GA1 algorithm describes the basic steps of a Genetic Algorithm. Note that each iteration of this iterative process is called a generation.

Genetic Algorithms and Business

4.1 Introductory Remarks

A genetic algorithm is a computational method for solving constrained and unconstrained optimization problems that is based on natural selection, i.e. the process that drives biological evolution. Genetic algorithms can repeatedly modify certain populations of individual solutions by selecting individuals at random from current populations to be parents and then uses them for producing children for the next generation. The populations with successive generations are evolving towards optimal solutions and genetic algorithms can be used for solving complex problems with discontinuous, stochastic, non-differentiable and highly nonlinear objective functions, as well as for solving mixed integer programming problems.

The genetic algorithms use three main types of rules at every step for creating the next generations from current populations, i.e. selection rules (selection of parents), crossover rules (combining two parents to form children of next generation) and mutation rules (random changes to parents for forming children). Note that genetic algorithms (GA) differ from classical algorithms (CA) (derivative based optimization algorithms) in the following: (i) GA generate populations of points at every iteration, with the best point approaching optimal solutions, while CA generate single points at every iteration with sequences of points approaching optimal solutions, (ii) GA select the next populations by computing which use random processes, while CA select next points in sequences by deterministic computations.

Genetic algorithms are heuristic searches that mimic the process of natural selection using basic methods, such as mutation and crossover for generating new genotypes aiming to computation of acceptable solutions to given problems. ML algorithms can be used for improving the performance of genetic and evolutionary algorithms (Zhang et al., 2011). GA's can be efficiently used on mixed (continuous and discrete) combinatorial problems, but they tend to be computationally expensive. By using genetic algorithms the solution of the original problem is represented as a *genome (chromosome)* and then a population of solutions is created and genetic operators (mutation, crossover) are applied for evolving the solutions in order to find the best ones.

Genetic algorithms can be efficiently used by considering the following important aspects: (i) definition of the objective functions, (ii) definition and implementation of generic representations and (iii) definition and implementation of generic operators. Several variations for improving performance, computing multiple optima and parallelizing these algorithms can be tested for solving the considered problems.

4.2. Genetic and Generic algorithms

In computer science and operations research, a genetic algorithm is a meta-heuristic procedure inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms. Genetic algorithms are heuristic search methods used in artificial intelligence and computing. They can be used for finding optimized solutions to search problems based on the theory of natural selection and evolutionary biology. Genetic algorithms are excellent for searching through large and complex data sets. They are considered capable of finding reasonable solutions to complex issues as they are highly capable of solving unconstrained and constrained optimization issues. The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation.

Generic programming is a style of computer programming in which algorithms can be written in terms of types to be specified later. This programming approach allows writing common functions or types that differ only in the set of types on which they operate when used, thus reducing duplication. These software entities are known as generics in computer languages, such as Ada, C#, Delphi, Eiffel, F#, Java, Objective-C, Visual Basic and others. Generic programming follows the idea of abstracting from concrete, efficient algorithms to obtain generic algorithms that can be combined with different data representations in order to produce a wide variety of useful software.

Genetic algorithms can be efficiently used for computing solutions of complex search problems. They can be used in several fields, such as engineering for creating incredibly high quality products due to their ability to search a through a huge combination of parameters to find the best match. They can search through different combinations of materials and designs for finding the perfect combination of both which could result in stronger, lighter and overall, better final products. They can also be used for designing computer algorithms, scheduling different tasks, and solving other optimization problems. Genetic algorithms are based on the process of evolution by natural selection which has been observed in nature. They can basically replicate the way in which life uses evolution for computing solutions to real world problems. Genetic algorithms are currently considered to be prominent computational models of evolution in artificial life systems. These decentralized models can be used as basis for understanding other complex systems and phenomena in the world. The generic algorithms can be used for studying how learning and evolution processes interact and furthermore for modelling various interesting systems, such as ecosystems, immune systems, cognitive systems and social systems (Winder et al., 1995; Whitley and Vose, 1995; Louis, 1993). GA's can be efficiently used on mixed (continuous and discrete) combinatorial problems, but they tend to be computationally expensive. By using genetic algorithms the solution of the original problem is represented as a *genome (chromosome or strings of ones and zeroes or bits [genes])* and then a population of solutions is created and genetic operators (mutation, crossover, inversion) are applied for evolving the solutions in order to find the best ones.

GA' can be efficiently used by considering the following important aspects: (i) definition of objective functions, (ii) definition and implementation of generic representations and (iii) definition and implementation of generic operators. Several variations for improving performance, computing multiple optima and parallelizing these algorithms can be tested for solving the considered problems (Grefenstette and Baker, 1989; Altenberg, 1994; Vose, 1993).The main difference between genetic programming and genetic algorithms is the representation of solutions, i.e. genetic programming creates computer programs following the schemes of computer languages as solutions, while genetic algorithms create strings of numbers that represent solutions(Whitley and Vose, 1995; Mitchell, 1999).

Genetic algorithms can be considered as *weak* optimization methods and their search procedures can be used for solving a wide range of computational problems without any domain specific knowledge. These search procedures are based on mechanics of natural selection and natural genetics using various evolution theories as tools for solving computational problems in science and engineering. This can be achieved by evolving populations of candidate solutions to particular problems using operations based on natural selection and natural genetic variation (Goldberg, 1989). GA's have been originally proposed for studying phenomena of adaptation as they occur in nature and for developing various ways in which mechanisms of natural adaptations that can be imported into computer systems (Holland, 1975). Genetic algorithms can use combinations of exploitations and explorations by exploiting appropriate solutions by combining different solution parts in order to form new candidate solutions (crossover process), as well as creating new candidate solutions by changing randomly parts of old solutions (mutation process). There are numerous variants of genetic algorithms and their implementation can be made as generic as possible. This implementation can be based on the description in "*Evolutionary algorithms in theory and practice*" (Back, 1996) and "*Evolution in time and space*" (Langevin, 1911). In the latter work the two basic conceptions of universe, i.e. the rational mechanics of Newton and Galileo, and the advanced electromagnetic theory, i.e. Maxwell, Hertz and Lorentz, are discussed.

Genetic Programming algorithms are evolutionary algorithms based on methodologies inspired by biological evolution for finding that can perform several user defined tasks. These algorithms are machine learning techniques used for optimizing populations of computer programs according to fitness landscape determined by programming abilities for performing predetermined computational tasks. The term Meta-Genetic Programming refers to meta-learning techniques of evolving genetic programming systems using generic programming itself. The process is a recursive terminating algorithm that allows avoiding infinite recursion (Schmidhuber, 1987).

In the wide spectrum of genetic algorithms applications are included the following topics:
Optimization (numerical optimization and combinatorial optimization-circuit layout and job scheduling)
Automatic programming (automata, sorting networks)
Machine learning (prediction in weather forecasting, protein structure, neural networks, sensors for robots)

Economics (modelling, bidding strategies, economic markets)
Immune Systems: (modelling natural immune systems)
Ecology (modelling biological processes, symbiosis, host parasite co-evolution)
Population genetics (viability of genes)
Social systems (evolutionary behaviour of social systems, evolution of co-operation and communication in multi-agent systems).

An e-Business Strategic Planning and Performance Evaluation model (the LP model)

An efficient e-Business model, the so called *Lipitakis-Phillips* (LP) model, for strategic planning and performance evaluation has been recently proposed (Lipitakis and Phillips, 2014).
The proposed e-business models and corresponding hypotheses have been statistically tested by using explanatory and confirmatory factor analysis, correlation between independent and dependent variables and regression analysis. In such cases the application of regression analysis it has been found that the independent variables of basic components can be used for predicting the dependent variables of financial and non-financial performance (Lipitakis, 2013).

A class of adaptive algorithms for solving e-business problems has been also recently presented (Lipitakis and Lipitakis, 2013). The basic components for estimating the strategic planning parameters of Formality, Participation, Sophistication and Thoroughness can be defined as follows:
Formality: the explicit and systematic procedures, policies and goals.
Participation: the involvement of senior and middle management. Improvement of communication and development of a shared vision for the direction of the firm.
Thoroughness: the extent to which a firm uses internal and external experience, and ensures adequate time is devoted to the strategic planning process.
Sophistication: use of a wide range of managerial techniques. Having a short or long term approach. Coordination of e-business across the organisation and having an appropriate budget for e-business.

These independent strategic planning variables are accompanied with the dependent variables of finance and non-finance. In this research work a new adaptive algorithmic modelling for e-business strategic planning and evaluation, based on an e-business performance model (Lipitakis, 2013), is presented.

The proposed e-business strategic planning model is based on a modified version of the LP e-business model, using a predetermined number of independent and dependent variables, and can be efficiently used for computing the best performance measurements and solving a wide class of e-business and strategic management problems under uncertainty conditions. Furthermore, a predetermined number of independent strategic planning variables and two dependent variables of financial and non-financial performance of organizations are being considered.

6. A Hybrid Genetic Algorithmic Scheme for e-Business Strategic Planning and Performance Evaluation

The genetic algorithm GA1 can be used for searching solutions, as indicated in the following algorithmic procedure. An adaptive algorithmic scheme for computing the performance evaluation of e-business and strategic management problems has been recently presented (Lipitakis and Lipitakis, 2013). A modified adaptive algorithmic modelling (MADAM) scheme using a hybrid genetic algorithm and a set of dependent and independent variables, given in eleven computational modules, can be described by following the above adaptive scheme can be describe as follows:

Algorithm MADAM-1 (GA1, FNFP, FPST, ε_{ST} STR, ε_{LE} LEA, ε_{PC} PCU, ε_{CO} COH, ε_{KN} KNO, ε_{AL} ALL, ε_{AD} ADM, ε_{UN} ADAMS)
Purpose: describes a modified Adaptive Algorithmic Modeling Scheme for computing the best performance measurements and solving a wide class of e-business and strategic management problems under uncertainty conditions. The genetic algorithm GA1 is applied for appropriate solutions.
Input: genetic algorithm GA1, Formation FORM, Participation PART, Sophistication SOPH, Thoroughness THOR, Finance Performance FINP, Non-Finance Performance NFIP, Structure STR, Leadership LEA, People and Culture PCU, Coherence COH, Knowledge KNO, Alliances ALL, Agility & Decision Making ADM, sp-parameters ε_{FO} , ε_{PA} , ε_{SO}, ε_{TH}, ε_{FP} , ε_{NF} and ε_{ST}, ε_{LE}, ε_{PC}, ε_{CO}, ε_{KN}, ε_{AL}, ε_{AD} , and uncertainty factor parameter ε_{UN}

Output: The (optimized) Modified Adaptive Algorithmic Model Solution (MADAMS)
Computational Procedure:
Step 1: apply CA1 searching for a feasible solution
Step 2: If there is no feasible solution then go to module 1, otherwise go to step 11.1
Module 1- Estimate the independent variables FPST (FORM, PART, SOPH, THOR)*Module 2*- Estimate the dependent strategic planning variables FNFP (FINP, NFIP)
Module 3- Determine input sp-parameters
Module 4- Apply CA1 and Design Structure STR (MRE, POR, SAR, DBF)
Module 5: Apply CA1 and Improve Leadership LEA (TCH, LAD, LAC, LEIS)
Module 6: Focus on People and Culture PCU (REW, RCR, LRE, RTR, ICO)
Module 7: Emphasize on Coherence COH (MPE, III, SIN, DDS, CCS)
Module 8: Comment on Knowledge KNO (KDA, KFO, KEM, KAC, KSH)
Module 9: Apply CA1 and Determine Alliances ALL (ART, APE, CRI)
Module 10: Apply CA1 and Focus on Agility & Decision Making ADM (IRE, MSR, TRTO, PMA, MSA, ADE)
Module 11: Form the e-business solution
Step 11.1: Determine the uncertainty parameter ε_{UN}
Step 11.2: Form the solution (ε_{UN} MADAMS)

The values of the sp-parameters affecting the corresponding input variables of the optimized algorithm MADAM-1 can be determined experimentally or approximately from corresponding appropriate mathematical model. In the special case that the sp-parameters take the values

$\varepsilon_{FO} = \varepsilon_{PA} = \varepsilon_{SO} = \varepsilon_{TH} = \varepsilon_{FP} = \varepsilon_{NF} = 1$ and

$$\varepsilon_{ST} = \varepsilon_{LE} = \varepsilon_{PC} = \varepsilon_{CO} = \varepsilon_{AL} = \varepsilon_{AD} = 1 \qquad (6.1)$$

A simplified form of the algorithm can be obtained, while the selection of the appropriate sp-parameters leading to (nearly) optimized solutions is dependent on the nature of the considered problem and often requires extensive experimentation. Further information about the algorithmic modules and input parameters can be found in a recent research work (Lipitakis and Lipitakis, 2013).

Multiple iterative applications of the proposed adaptive algorithm on a set of selected e-business strategic planning performance evaluation scheme, using multiple-point Likert scale measurement in every iterative step, can lead to comparable numerical results for evaluating the performance of the adaptive algorithmic application for the strategic planning performance of each organization/business at national/international region levels for comparative purposes.

The main advantage of the proposed algorithmic approach is twofold. Firstly, the adaptive algorithms can be efficiently used for solving a wide class of e-business and strategic management problems, and secondly the dynamical choice of the sp-parameter values, which can be related to both quantitative and qualitative nature of the input parameters (data) of the given problem, can lead to (near) optimum solutions. The evaluation of a firm e-business performance as a time-dependent problem the investigation of the performance stability over a certain period of time seems to be a challenging future research problem (Coltman et al., 2008; European Commission, 2004).

7. Genetic Programming Implementations and Software Products

7.1 Genetic Languages and Implementations

In this section several genetic programming implementations with the corresponding related software products are given. Genetic programming implementations can be achieved by several languages and their variants, such as indicated in the following:

MATLAB [GPLAB (Silva, 2007); GPTIPS: Searson et al., 2010)
Python [Pyvolution (Panchapakesan, 20120; deap (Fortin et al., 2012); pySTEP (Khoury, 2009); Pyevolve, Perone, 2014)]
Java [EvoJ (Keijzer et al., 2002); PMDGP (Van der Meulen et al., 2001); ECF (Georgopoulos et al., 2009); EOEvolutionary Computation Framework (Keijzer et al., 2002)]
C# [Heuristic Lab (Wagner, 2009); GPdotNET (Bahrudin, 2009)]
Prolog [DCTG-GP (Ross, 2001)]
Ruby [DRP (Nuanain and O'Sullivan, 2014)]

Perl [PerlGP (McCallum, 2006)]
.NET [GPE (Manec, 2016)].

Further related information can be found in several Genetic Programming Implementations (Goldberg, 1989; Mitchell, 1999; Silva, 2007; Manec, 2016).

7.2 Genetic Algorithms and Evolution Process

Genetic algorithms are considered to be heuristic computational techniques for searching or optimization, originally motivated by the evolution principle through genetic- selection process. The genetic algorithms by using highly abstract versions of evolutionary processes in order to evolve solutions to given problems, by operating on certain populations of artificial chromosomes, i.e. strings in a finite alphabet (binary). Note that each chromosome represents a solution of the problem having fitness, i.e. a real number measuring how good the obtained solution is to the given problem. A random generated population of chromosomes is selected and a genetic algorithm, with a highly modular nature, is using a process of fitness-based selection and recombination in order to produce a successor population (next generation). This iterated process produces a sequence of successive generations and is terminated by a predetermined stopping criterion, evolving to a good (best) solution to the original problem.

The genetic algorithms, initially proposed by Holland (1975), can be efficiently used in a wide spectrum of scientific, engineering and industrial problems, particularly in computational intelligence problems including topics such as, neural networks, artificial immunology, particle swarm optimization, ant colony optimization (Barbosa, 2013). Genetic algorithms manipulate several populations of chromosomes, i.e. abstractions of biological DNA chromosomes, such as strings of letters from the set {A, C, G, T}. Note that a particular position (*locus*) in a chromosome is referred to as *gene*, while the letter occurring at that point in the chromosome is referred to as *allele* (value). Each bit-string representation used is referred to as the genetic algorithm encoding of the problem. The fitness function is the computed value for the quality of chromosome as a solution to the given problem (Engelbrecht, 2002).

Conclusions

Artificial Intelligence methodologies, particularly the application of Genetic Algorithms in Business has been considered. The usage of Genetic Algorithms in several organizations in various applications concerning business activities, such as staffing, careers, hiring as well as hiring predictions have been discussed. The concepts of genetic and generic algorithms with their several important aspects are discussed and related applications in a wide spectrum of topics have been presented. An e-business strategic planning and performance evaluation model has been synoptically presented and a modified genetic e-business model has been introduced. Various genetic programming implementations with corresponding related software products have been given, several AI software packages, genetic algorithmic methodologies and applications have been also presented.

Future research work includes the application of genetic algorithms in a wider spectrum of applications, such as enterprise information systems, computing information technology, financial engineering business intelligence, digital information management, knowledge management and e-learning services will be studied, as well as their application on time-dependent large scale complex computational problems in parallel computing environments.

References

Alsever J. (2017): How A.I. is changing your job hunt and what the new algorithms want, Fortune 7

Altenberg L. (1994): The evolution of evolvability in genetic programming, In K. E. Kinnear, Jr. (ed.), Advances in Genetic Programming, MIT Press

Apache Software Foundation (2006) Jakarta commons configuration project. http://jakarta.apache.org/commons/configuration/

Back T. (1996): Evolutionary Algorithms in Theory and Practice: Evolution Strategies, evolutionary programming, genetic algorithms", Oxford University Press

Backhouse R., Jansson P., Jeuring J. and Meertens L. (1999): Generic Programming — an Introduction, Advanced Functional Programming, 28-115

Bahrudin I.H. (2009): GPdotNET V4.0 – artificial intelligence tool [Computer program], *http://gpdotnet.codeplex.com*

Baily A., Ventresca M. and Berman B. (2012): Automatic generation of graph models for complex networks by genetic programming, Procs GECCO 12 conf. on Genetic and Evolutionary Computation, 711-718

Barbosa H.J.C. (ed) (2013): Ant colony optimization-techniques and applications, Int. J. Biautomation 17 (1), 119-120

Baricelli N. A. (1962): Numerical testing of evolution theories, ACTA Biotheoretica 16, 69–126

Bodenhofer U. (2003): Genetic Algorithms: Theory and Applications Lecture Notes Third Edition.

Cheong J. (2011): Controller with bare-bone Plain Old PHP Object aka POPO

Coley D.A. (1999): An Introduction to Genetic Algorithms for Scientists and Engineers, World Scientific Publishing.

Darrell Whitley and Vose M.D. (1995): Foundations of Genetic Algorithms, Vol. 3, Morgan Kaufman Publishers Inc.

Dyer D. (2010): The Watchmaker Framework for Evolutionary Computation.

http://watchmaker.uncommons.org/.

ECOS - Evolutionary and Complex Systems Group University of Coimbra Portugal

Engelbrecht A.P. (2002): Computational Intelligence: An Introduction, Wiley, New York.

European Commission, (2004): The European e-business report, Office for official publ. of European Communities, Luxembourg, 234-237.

Ferreira C. (2006): Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence, Springer-Verlag, Germany, RMIT-GP Documentation (2004):

http://goanna.cs.rmit.edu.au/~vc/research.html

Fogel D. B. (1995): Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, IEEE Press.

Fogel D.B. (1992): Evolving artificial intelligence, Ph.D. Dissertation, The University of California, San Diego.

Fortin M., De Rainville M., Gardner M. and Gagne C. (2012): DEAP: Evolutionary Algorithms Made Easy, Journal of Machine Learning Research 13, 2171-2175

GAlib: A C++ Library of Genetic Algorithm Components Author Massachusetts Institute of Technology Website http://lancet.mit.edu/g

GECCO '12 Proceedings of the 14th annual conference on Genetic and evolutionary computation, 711-718

Gene I. S. (2012): Handbook of Neuroevolution Through Erlang, Springer Verlag.

Georgiou L. and Teahan W. J. (2006): jGE - A Java implementation of Grammatical Evolution, 10th WSEAS International Conference on Systems, Athens, Greece, July 10–15, 2006.

Georgopoulos P., Sasso A., Isukapalli S., Lioy P., Vallero D., Okino M. and Reiter L. (2009): Reconstructing population exposures to environmental chemicals from biomarkers: Challenges and opportunities, Journal of Exposure Science and Environmental Epidemiology 19, 149–171

GEVA (2008): UCD's Natural Computing Research & Applications group, UCD School of Computer Science & Informatics Technical Report UCD-CSI-2008-09, http://ncra.uc.ie/geva

Goldberg D.E. and Holland J. H. (1988): Genetic algorithms and machine learning, Machine Learning 3 (2), 95–99.

Gompertz B. (1825): On the Nature of the Function Expressive of the Law of Human Mortality, and on a New Mode of Determining the Value of Life Contingencies, Phil. Trans. Roy. Soc. London 123, 513-585.

Gondro C. and Kinghorn B.P. (2007): A simple genetic algorithm for multiple sequence alignment, Genetics and Molecular Research 6 (4), 964–982.

GPTIPS: Genetic Programming and Symbolic Data Mining Platform Author Dominic Searcon Website https://sites.google.com/site/gptips4matlab/

Grefenstette J.J. (1986): Optimization of control parameters for genetic Algorithms, IEEE Trans. System, Man, Cybernetics, SMC-16 (1), 122 - 128. IEEE Press Piscataway, NJ, USA

Grefenstette J. J. and Baker J. E. (1989): How genetic algorithms work: A critical look at implicit parallelism, In J. D. Schaffer (ed.), Proceedings of the Third International Conference on Genetic

Algorithms. Morgan Kaufman
Holland J. H. (1975): Adaptation in Natural and Artificial Systems, University of Michigan Press. (Second edition: MIT Press, 1992.)
Hornby G.S. (2003): Generative Representations for Computer-Automated Design Systems, NASA Ames Research Center, Mail Stop 269-3, Moffett Field, CA 94035-1000 .
Jianxin J., Zhang Y. and Wang Y. (2007): A generic genetic algorithm for product family design, Journal of Intelligent Manufacturing18 (2), 233–247
Java (1996): GAlib: A C++ Library of Genetic Algorithm Components version 2.4 Documentation Revision B, Matthew Wall Mechanical Engineering Department Massachusetts Institute of Technology, http://lancet.mit.edu/ga/
JEF: Java Evolution Framework Author, Brno University of Technology, Signal Processing Laboratory Website http://spl.utko.feec.vutbr.cz/en/component/content/article/ 28-projects/258-
Keijzer M., Merelo J.J., Romero G., Schoenauer M. (2002): Evolving objects: A general purpose evolutionary computation library, Artificial Evolution 2310, 829-888.
Khoury M. (2009): pySTEP-beta 0.97.xm, http://www.opensource.org/licenses/mit-license.html
Koza J.R. (1992): Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, USA.
Koza J.R. (1994): Genetic Programming II, MIT Press, Cambridge, MA.
LAGEP: Layered Architecture Genetic Programming, Author National Chiao Tung University, Department of Computer Science Website
http://www.cis.nctu.edu.tw/gis91815/lagep/lagep.html
Finlay S. (2014): Predictive Analytics, Data Mining and Big Data: Myths, Misconceptions and Methods (Business in the Digital Economy), 2014th Edition
Hickman L. (2013): How algorithms rule the world, Guardian
Langevin P. (1911): The Evolution of Space and Time, Scientia 10, 31-54
Le Goues C., Nguyen T., Forrest S. and Weimer W. (2011): GenProg: A generic method for automated software repair. IEEE Transactions on Software Engineering
Lewars E. G. (2011): Computational Chemistry, Heidelberg, Springer.
Lipitakis A. and Lipitakis, EAEC. (2012): On the e-Valuation of Certain e-Business Strategies on Firm Performance by Adaptive Algorithmic Modelling: An Alternative Strategic Managerial Approach, Journal of Computer Technology and Application (JCTA) 3 (1), 38-46
Lipitakis A. and Lipitakis EAEC. (2013): E-Business performance and strategy planning e-valuation based on adaptive algorithmic modelling methods: Critical factors affecting e-valuation in strategic management methodologies, Universal J. of Management 2, 981-991
Lipitakis A. and Phillips P. (2014): On E-Business Strategy Planning and Performance: Recent Developments, Impacts and Assessments, Technology Analysis & Strategic Management 26 (9).
Louis S.J. (1993): Genetic algorithms as a computational tool for design, S.J. Louis
MacKay D.J.C. (2003): Information Theory, Inference, and Learning Algorithms Cambridge, Cambridge University Press, 2003.
Manec P. (2016): Genetic programming in Swift for human-competitive evolution, Department of Software and Computer Science Education, Charles University in Prague, Faculty of Mathematics and Physics
Marr B. (2016): The Top 10 AI and Machine Learning Use Cases Everyone Should Know About, https://www.forbes.com/sites/bernardmarr/2016/09/30/what-are-the-top-10-use-cases-for-machine-learning-and-ai/#5a4f59cf94c9
Maynard Smith J. (1987): When learning guides evolution, Nature 329, 761–762
McCall J. (2005): Genetic algorithms for modelling and optimisation, Journal of Computational and Applied Mathematics 184 (1), Elsevier, 205-222
McCallum R.M. (2006): Introducing a Perl Genetic Programming System – and Can Meta-evolution Solve the Bloat Problem?, Stockholm Bioinformatics Center, Stockholm University, Stockholm, Sweden
McGrath J. (2016): The rise of AI and algorithms in the financial services sector, Raconteur EU/70N

Michalewicz Z. (1996): Genetic Algorithms + Data Structures = Evolution Programs (3rd ed.), Springer-Verlag, New York
Michie D., Spiegelhalter D. J. and Taylor C. C. (1994): Machine Learning, Neural and Statistical Classification, Ellis Horwood.
Mitchell M. (1996): An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA.
Mitchell M. (1999): An Introduction to Genetic Algorithms, A Bradford Book, The MIT Press Cambridge, Massachusetts • London, England Fifth printing.
Mitchell M., Forrest S., Holland J.H. (1992): The royal road for genetic algorithms: fitness landscapes and GA performance
Murphy R. (2003): A Generic Parallel Genetic Algorithm, M.Sc. in High Performance Computing, Trinity College, University of Dublin,
Musser D. R. and Stepanov A. (1987): A library of generic algorithms in Ada, Proceedings of the 1987 annual ACM SIGAda international conference on Ada: 216–225.
Musser D. R. and Stepanov A. A. (1989): Generic programming, In P. Gianni. Symbolic and Algebraic Computation: International symposium ISSAC 1988, Lecture Notes in Computer Science 358, 13–25
Nuanain C. and O' Sullivan L. (2014): Real-time algorithmic composition with a tabletop musical interface: a first prototype and performance, Procs AM 14: A conference on Interaction with sound, article 9
Packard N. H. (1990): A genetic learning algorithm for the analysis of complex data, Complex Systems 4 (5), 543–572
Peck C. C. and Dhawan A. P. (1993): A Review and Critique of Genetic Algorithm Theories, Technical Report TR 153/6/93/ECE, College of Engineering University of Cincinnati, Department of Electrical and Computer Engineering,
Pelikan M., Goldberg D.E. and Lobo F.G. (2002): A survey of optimization by building and using probabilistic models, Computational optimization and applications 21 (1), 5-20
Pereira F. C. (2006): Creativity and Artificial Intelligence: A Conceptual Blending Approach, Applications of Cognitive Linguistics, Amsterdam, Mouton de Gruyter
PerlGP: The Open Source Perl Genetic Programming System Author Bob MacCallum Website http://perlgp.org/
Perone C.S. (2014): Pyevolve 0.6rc1, Python Software Foundation, http://pyevolve.sourceforge.net
Poli R., Nordin P., Langdon W.B. and Fogarty T.C.(editors) (1999): Genetic Programming, Proceedings of EuroGP'99, volume 1598 of LNCS, Goteborg, Sweden, 26-27 May 1999. Springer-Verlag.
Pressman R.S. (2009): Software Engineering: A Practitioner's Approach (7th ed.). Boston, Mass: McGraw-Hill.
Punnen A. P. (2002): The Traveling Salesman Problem: Applications, Formulation and Variations, In: Traveling Salesman Problem and its Variations, Gutin, G. & Punnen, (Ed.), 1-28, Kluwer Academic Publishers, Dordrecht
Reeves C.R. and Rowe J.E. (2003): Genetic Algorithms – Principles and Perspectives: A Guide to GA Theory, Kluwer Academic, Boston.
Ross B.J. (2001): Logic based Genetic Programming with definite clause translation grammars, New Generation Computing 19 (4), 313-337
Rudolph G. (1994): Convergence analysis of canonical genetic algorithms, IEEE Trans. Neural Networks 5, 96-101
Russell S. and Norvig P. (2002): Artificial Intelligence – A Modern Approach, Prentice Hall.
Samuel A.L. (1983): AI, where it has been and where it is going, In IJCAI, pages 1152–1157.
Schaffer J. D., Whitley D. and Eshelman L. J. (1992): Combinations of genetic algorithms and neural networks: A survey of the state of the art, In L. D. Whitley and J. D. Schaffer, eds., COGANN−92: International Workshop on Combinations of Genetic Algorithms and Neural Networks, IEEE Computer Society Press.
Schmidhuber J. (1987): Learning how to learn, metalearning, meta-genetic programming, credit-conserving machine learning economy, http://www.idsia.ch/~juergen/diploma.html

Schmieder A. (2014): Swift, C++ Performance, http://www.primatelabs.com/ blog/2014/12/swift-performance/.

Searson D.P., Leahy D.E. and Willis M.J. (2010): GPTIPS:An Open Source Genetic Programming Toolbox For Multigene Symbolic Regression, Procs. of Intern. Multi Conf. of Engineers and Computer Scientists 2010, Vol. I, IMECS 2010, Hong Kong

Securities Trading Industry, Brooking-Wharton papers on Financial Services

Silva S. (2007): GPLAB, A Genetic Programming Toolbox for MATLAB, Evolutionary and Complex Systems Group, University of Coimbra Portugal Version 3

Smith G.D., Steele N.C. and Albrecht R.F. (1998): Artificial Neural Nets and Genetic Algorithms, Procs of Intern. Conf. in Norwich, UK, Springer-Verlag.

Spector L. and Helmuth T. (2014): Uniform Linear Transformation with Repair and Alternation in Genetic Programming, In Genetic Programming Theory and Practice XI, edited by R. Riolo, J. H. Moore and M. Kotanchek, pp. 137-153, New York: Springer.

TinyGP: A Contest to Design the Smallest Possible GP System Author University of Essex Website http://cswww.essex.ac.uk/staff/sml/gecco/TinyGP.html

Van der Meulen P.G.M., Schipper H., Bazen A.M. and Gerez S.H. (2001): PMDGP: A Distributed Object-Oriented Genetic Programming Environment, 7th Annual Conference of the Advanced School for Computing and Imaging, Heijen, The Netherlands

Varela F.J., Bourgine P. (Eds.) (1992): Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, MIT Press, Cambridge, MA.

Ventura S., Romero C., ·Zafra A., Delgado J.A. and · Hervás C. (2007): JCLEC: a Java framework for evolutionary computation, Soft Computing 12, 381-392, Springer-Verlag.

Vose M. D. (1993): Modeling simple genetic algorithms, In L. D. Whitley (ed.), Foundations of Genetic Algorithms 2, Morgan Kaufmann.

Vose M. D. (1993): Modeling simple genetic algorithms, In L. D. Whitley, ed., Foundations of Genetic Algorithms 2, Morgan Kaufmann.

Vose M.D. (1991): Modeling simple genetic algorithms, G. Rawlins (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, Los Altos, CA, 94-101

Vose M.D. (1999): The Simple Genetic Algorithm, MIT Press, Cambridge, MA.

Wagner S. (2009): Heuristic optimization software systems - Modeling of heuristic optimization algorithms in the HeuristicLab software environment, Ph.D. thesis, Johannes Kepler University, Linz, Austria.

Wall M. (1996): GAlib: A C++ Library of Genetic Algorithm Components Department, Massachusetts Institute of Technology.

Weise, T. (2009): Global Optimization Algorithms: Theory and Application, Retrieved from http://www.it-weise.de/projects/book.pdf

Whitley D. and Vose M. (eds) (1995): Foundations of Genetic Algorithms 3, Morgan Kaufmann.

Willis M., Hiden H., Marenbach P., McKay B. and Montague G.A. (1997): Genetic programming: An introduction and survey of applications, In A. Zalzala, editor, Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA, University of Strathclyde, Glasgow, UK.

Winder G., Periaux J. and Galan M. (eds) (1995): Genetic Algorithms in Engineering and Computer Science , John Wiley and Son Ltd

Zhang J., Zhan Z.H., Lin Y., Chen N., Gong Y.J., Zhong J.H., Chung H.S.H., Li Y., Shi Y.H. (2011): Evolutionary Computation Meets Machine Learning: A Survey, Computational Intelligence Magazine, IEEE 6 (4), 68–75.

Zhang Q. and Li H. (2007): MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Transactions on Evolutionary Computation 11 (6), 712-731

Zitzler E., Deb K. and Thiele L. (2000): Comparison of multi-objective evolutionary algorithms: empirical results, Evolut. Comput. 8 (2), 173-19