Academic Journal of Nawroz University (AJNU), Vol.10, No.3, 2022

https://doi.org/10.25007/ajnu.v11n3a1414

The Genetic Algorithm (GA) in Relation to Natural Evolution

Pawan Shivan Othman[1], Rasheed Reber Ihsan[2], Reving Masoud Abdulhakeem[3]
[1,3]Department of Computer Science, Nawroz University, Kurdistan Region – Iraq
[2]Department of Computer Engineering and Communication, Nawroz University, Kurdistan Region –Iraq

ABSTRACT

For optimizing search global solution for complicated issues, the Genetic Algorithm (GA)is a famous evolutionary computation technique that plays an important role in finding meaningful solutions to hard problems with a huge search space could be a process based on genetic selection ideas. In addition, it supports machine learning causes, as well as study and evolution. However, developing genetic processes that were formerly significant to a random population, which might be started by biology for chromosomal production with factors like selection, crossover, and mutation. The aim of going through this GA process is to find a solution for consecutive generations. In individual production there has been an extent success instantly in ratio to fitness which is suited for it, as a result successive generation will be better in one condition, which is ensuring the quality. Furthermore, John Holland is considered as being the funding father of the initial genetic algorithm, with a funding date in the 1970s. in this paper we have explained what a genetic algorithm is, its key operations, and how it works as well as its features and applications.

KEYWORDS: Optimization Natural Evolution, Genetic Algorithm, Mutation, Crossover, Selection.

1. Introduction

Evolutionary computation is a branch of computer science that deals with computational issues and biological progress. Most of those issues need a lot of research work due to having a huge number of possibilities, such as hardware circuit layouts for configuration that are responsible of producing desire behavior, to find a set of equations that will predict the ups and downs of financial market, or a set of rules that controls a robot as it navigates its environment. These kinds of problems always require a process to be controlled that is, to continue in a changing environment and perform well [2].

Behavior that are complicated and hard to be resolved such as intelligence could be manifested from aligned application and the reaction of

the rules. One of the examples of this philosophy is Neural Networks, and evolutionary commutation is another one. An irresistible source of inspiration for representing demanding computational problems is Biological Evolution. Also, searching through out a vast number of possibilities is called the Evolution Method.

The Genetic Algorithm (GA) is an adaptive search algorithm that replicates parts of the evolution processes: selection, fitness, reproduction, crossover (also known as recombination), and mutation. It is based on Darwin's theory of evolution [1][3].

In an artificial environment, a population of people (organisms, strategies, individuals, objects...) is characterized by a genetic sequence that induces physical features.

Mutations and exchanges of genetic material search new individual qualities, whereas the fittest people reproduce and spread their genes. the majority of GAs has been progressed for optimization purpose; despite being originally established for the analysis of adaptation in natural systems (J. Holland, 1975). (Whitley, 1994). They've been used in a wide range of fields, as shown in table 1.

TABLE 1: GA in various fields

Research Domains:	Name Of Authors:
Biology	Street,1999
Economics	Chatterjee Et Al., 2018
Finance	Lwin Et Al., 2014
Operational Research	Della Croce Et Al.,
Game Theory	Axelrod Et Al., 1987, Vi´e, 2020b
Deep Learning and Neural Networks	Stan- Ley Et Al., 2019, Chung and Shin, 2020
Forecasting	Packard, 1988, C. Ahn and Ramakrishna, 2003
Optimization	Wiransky, 2020, Dhunny Et Al., 2020
Computer Science and Algorithms	Koza, 1992
Data Science	Yang And Honavar, 1998
Healthcare	Tao Et Al., 2019

Aside from the hardness of the application problem to be solved with Genetic Algorithms (GAs), there's also the issue of the solution's quality, or the computational resources required to find it [5]. it, is dependent on the properties of the Genetic Algorithm. Genetic Algorithms (GAs) are a powerful technique for resolving optimization issues. To use Genetic Algorithms to solve a problem, you must first answer a few questions concerning the starting population, the likely and type of crossover, the likelihood and type of mutation, the terminating criteria, the type of selection operator, and the fitness function to be used [7].

All of these characteristics and operators have an impact on the performance of a GA and are interconnected, forming a system. Plus, it's showed that it's a highly recommended that the ''rules'' of evolution are remarkably simple Genetic algorithms (GAs), which will be the main focus of this review, that is virtually widely used from of evolutionary Along with genetic programming, evolution methods, and evolutionary programming, the genetic algorithm is part of the evolutionary algorithm family [9]. The term "evolutionary algorithm" refers to a large category of stochastic optimization approaches.

A population of candidate solutions for the problem at hand is maintained by an evolutionary algorithm. After that, a collection of stochastic operators is applied iteratively to the population to evolve it [15]. Mutation, recombination, and selection, or something similar, are commonly included in the set of operators. We introduce genetic algorithms, their main principles, and algorithm processes in this study [13].

We begin by discussing their properties as a search strategy in order to comprehend the rationale for their wide-ranging and effective applications in optimization and modeling.

2. Literature Review

GA was reviewed according to the PRISMA criteria. A comprehensive search of Google Scholar and PubMed for research papers connected to GA was conducted. This document also includes the important research works discovered during the manual search. Keywords like "Genetic Algorithm," "Application of GA," "operators of GA," "representation of GA," and "variants of GA" were used during the search.

Let's have a look at some of the publications that have been published in the past that are relevant to the Genetic Algorithm. In the early 1960s, in Holland's papers (e.g. Holland 1962) the idea of Genetic Algorithm was found in a brief sight, and it is a great use for finding out the path encoding in graph which changes into chromosomes through shorted path via application [8].

Cavicchio's thesis (Cavicchio 1970) observed these concepts as a type of adaptive search and examined them experimentally on challenging search problems including subroutine selection and pattern recognition. Some of the early research on elitist types of selection and concepts for modifying the rates of crossover and mutation may be found in his work[2].

The increased interest in GAs prompted a series of conversations and preparations for the inaugural International Conference on Genetic Algorithms (ICGA), which took place in Pittsburgh, Pennsylvania, in 1985. Around 75 people attended, and they presented and discussed a wide spectrum of recent developments in GA theory and implementation (Grefenstette 1985) [9].

The International Society for Genetic Algorithms (ISGA) is a non-profit organization whose mission is to provide a vehicle for conference funding as well as to assist in the coordination and facilitation of GA-related activities. One of its first business moves was to support a proposal for a FOGA (Foundations of Genetic Algorithms) theory workshop in Bloomington, Indiana (Rawlins 1991) [4].

Jang Sung Chun investigated the use of genetic algorithms as a means of searching for optimization problems in a variety of ways (1998). On several optimization tasks, the evolutionary algorithm and genetic algorithm were compared, and the results revealed that the genetic algorithm outperformed the evolutionary algorithm.

From 1990 to the present, the GA community has experienced tremendous growth and diversity, as evidenced via the numerous conference events for example; ICGA, and FOGA [6]. New GA applications are being developed all the time, covering a wide spectrum of topics from engineering design to operations research to autonomous programming.

3. GENETIC ALGORITHM AND ITS WORKFLOW

According to Darwin principal of survival of the fittest, Genetic Algorithm (GA) that is relevant to the types that are endangered by predators or environmental changes and risks [10]. Those who are not suited and fittest cannot always survive this. On the other hand, those who are fittest have a better chance to survive, because they have the ability to change situations, and also mostly their traits and skills are inherited by their kids and even sometimes more developed, which means the future generations could be even fitter which is mentioned in figure 1.

In addition, some species randomly get genetic mutations, and their fit individuals and their evolutionary descendants surviving for a long time could be increased by these mutations [14]. To have a full optimization the GA serves as inherent solution while each chromosome is individually generated.

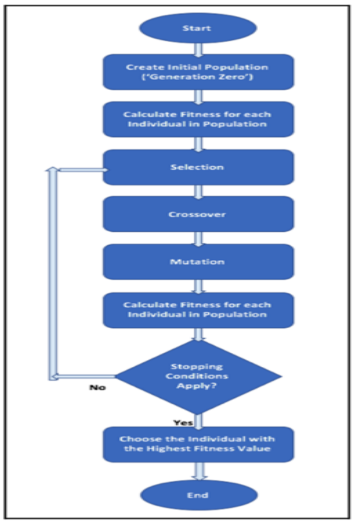

Figure 1. Basic flow chart of a genetic algorithm

4. HOW DO GENETIC ALGORITHMS WORK?

For genetic algorithm their fitness classification is what gives them the capacity to live and survive despite all the environmental risks. Each generation is made up by parent population and their offspring, which has all the surviving individuals (chromosomes) from the previous generations. The two genetic operators that activate and create offspring or progeny are Crossover and Mutation, and apparently this is known as fresh solution [12].

Parents are determined for creating a new generation so that their chance of being picked up could be corresponding near their fitness value. The better this fitness rating goes, the more opportunity of life and surviving to reproduce is given. The table 2 represents the GA's characteristics [15]. Mostly regular GA starts with some random individuals to generate a population of some possible solutions. After measuring the individual fitness, according to their fitness marks, some of them will be chosen as parents.

Table 2: GA characteristics.

General Algorithm	Genetic Algorithm
Process Of Generating New Solution	Genetic Operators

Selection	Surviving Parents
Initial Solution	Random Chromosome
Fitness Function	Quality Of Individual
Best Solution	Elite
New Solution	Children (Offspring)
Old Solution	Parent
Solution	Chromosome (Individual)
Decision Variable	Gene Of Chromosome

5. Creating an Initial Population

One of the most significant problems to consider in evolutionary computation is population scaling [10], [18]. Researchers frequently claim that a "small" population size can lead to bad solutions [17], [18], [12], and that a "big" population size can cause the algorithm to take longer to find a solution [15], [14], [10], [12]. So, we're dealing with a trade-off that may be approximated by feeding the algorithm "enough" chromosomes [24] to get "excellent" results. For us, "enough" is linked to instances in the search space and diversity.

One of the chosen solutions to the GA's optimization problem is chromosome. According mathematical formulation and also the studies have shown us that each

A chromosome (also known as a genotype) is a group of factors that define a possible solution to the problem the genetic algorithm is trying to solve. The population is the collection of all solutions [10].

In an N-dimensional optimization problem, a chromosome is a 1N-dimensional array. The following is the array's definition:

$$\text{Chromosome} = X=(x_1, x_{2,} \dots, x_i, \dots\dots, x_n) \qquad (1)$$

If X indicates a practical optimization problem solution, x_i indicates the ith decision variable (or gene) of solution X, and N denotes the number of decision variables. M stands for the population size, or the number of alternative solutions. A matrix of chromosomes of size $M \times N$ shows the population of created alternative solutions:

$$Population = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_j \\ \vdots \\ X_M \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,i} & \cdots & x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,i} & \cdots & x_{2,N} \\ & & & \vdots & & \\ x_{j,1} & x_{j,2} & \cdots & x_{j,i} & \cdots & x_{j,N} \\ & & & \vdots & & \\ x_{M,1} & x_{M,2} & \cdots & x_{M,i} & \cdots & x_{M,N} \end{bmatrix}$$

Where X_J is the jth solution (or chromosome), $x_{i,j}$ is the ith solution's jth decision variable (or gene), and M is the population size.

5.1 Selection

Selection (Reproduction); The phase of selection is dedicated to offer additional reproductive chances to those population members that are the fittest, identifying the proper selection technique is a vital process [19].

Many selection techniques are introduced in the literature, including: Elitism, Rank, Roulette Wheel, Tournament and Stochastic Universal Sampling (SUS), see Table 3.

Table 3: GA selection operators

Selection Techniques	Pros	Cons
Elitism	Preserve Best Individual in Population	Best Individual Can Be Lost Due to Crossover and Mutation Operators
Rank	Free From Bias Preserve Diversity	Slow Convergence Sorting Required Computationally Expensive
Roulette Wheel	Easy To Implement Simple	Risk Of Premature Convergence Depends Upon Variance Present in The Fitness Function
Tournament	Preserve Diversity Parallel Implementation No Sorting Required	Loss Of Diversity When the Tournament Size Large

Stochastic Universal Sampling (Sus)	Fast Method	Premature Convergence

5.2 Population

(Whitley, 1989) conducted that we take population into account by using Selective Pressure and Diversity that are the GA's search methods for population. The two named methods are contrarily connected, and as a result raising one lowers the other one. The two methods can highly affect the chromosome in different ways [20].

An optimization search to stall can be caused by the low selective pressure, on the other hand, premature convergence could happen because of high selective pressure [21]. As we explained in the previous subsection, Convergence mistakes may be initiated throughout proportionate selection.

5.3 Reproduction

The Genetic algorithm should create new solutions by proceeding into a perfect solution. There is two ways, either a portion of the following generation or the parents give birth to offspring who make up the entire [22]. As a result, a mixture of parents might be produced for the next generation.

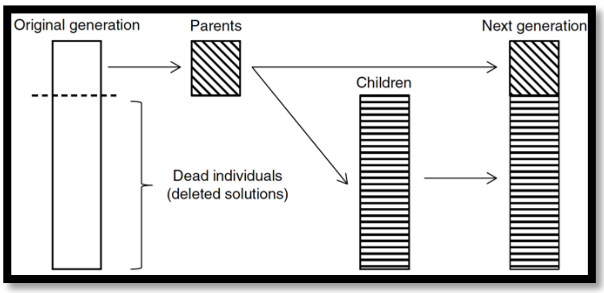

Figure 2. The procedure of a new generation being formed since the preceding one.

The parent-to-offspring ratio is a user-located metric. The technique for producing the next generation from the previous generation is shown, and explained in Figure 2. To create the next generation, there are many different approaches and methods, which define GA variations, have been devised [24].

To come up with the new solutions the chosen parents should take the responsibility, and the new solutions are the children (offspring) relying on their fitness, the selected population of the parents use R solution.

Using the crossover probability (PC) which is a user-defined parameter of method, parents are randomly chosen for this population. Each solution in the parent population needs an un-systemic Rand number that is produced from the range [0,1].

5.4 Crossover

When the solution of two parents start to interact together, Crossover will start to happen. And then new offspring are produced by transforming genes between parents.. In other words, swapping parts of the solution with another in chromosomes or solution representations [23].

The main role is to provide mixing of the solutions and convergence in a subspace. Goldberg (1989) and Michalewicz (1996) have conducted various ways of crossover, in addition to all of that, they have mentioned; one-point crossover, two-point crossover, and uniform crossover. Figure 3 shows us the three last kinds of crossings.

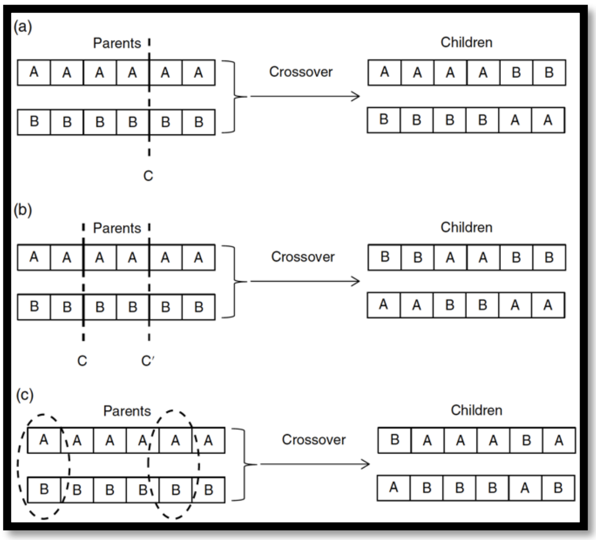

Figure 3. Crossover methods include (a) one-point crossover, (b) two-point crossover, and (c) uniform crossover.

If we look into the figure 3 we will notice that a crossover point C in figure is chosen at random, which is the result of using one-point crossover.

In the figure on one side of the point it shows that some of a child's genes are originate from one parent, and the balance of the child's genetic code.

Meanwhile, on the other side of the point, it originates from the other parent. The figure 1.5b shows that when using two-point crossover two children are born to each couple, also it shows that two crossover sites are created at random and are designated by C and C'.

In the same position the genes between the two positions in the parent solutions are kept in the genetic make-up of the offspring. As indicated in figure 3 b, the genes are located outside, and the two boundaries are exchanged to produce two children. Also, the uniform crossover schemes are presented in the figure 3 c, and that is called self-explanatory. In the rage [1,N] some integer random numbers are used to create crossover spots.

5.5 Mutation

Mutation delivers new genetic materials substance into population that's why it's a really important method to know about. The mutation operator starts to substitutes a random number of genes in an offspring. To make it easy to understand, look into the figure of 4 which shows that, in new solution random values are replaced with the values of one or more decision variables . Meanwhile, the values of the other choice variables stay unchangeable. For real-valued variables we have two methods of mutation which are known as uniform and non-uniform mutations. Uniform mutation is a method of replacing a parent gene with a randomly produced gene that is within the solutions' feasible space [5].

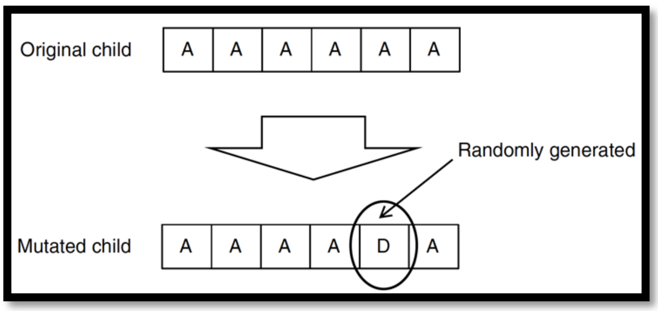

Figure 4. An example of the mutation operator

5.6 Termination Criteria

When the algorithm's iterations should come to a stop the termination criteria are specified. Choosing the genuine termination has a huge impact over the algorithm's proper convergence. Depending on the number of iterations, the amount of progress made in the target function between iterations, and the run time, the GA usually reaches the conclusion of its journey[11].

6. Applications

With high accuracy rates, genetic algorithms have been applied to a variety of NP-hard tasks. GAs have been used successfully in a few different application areas, which is mentioned in table 4.

Table 4: GA Application Areas

Operation Management	Multimedia	Wireless Networking
Facility Layout	Information Security	Load Balance
Scheduling	Image Processing	Localization
Inventory Control	Video Processing	Bandwidth & Channel Allocation
Forecasting & Network Design	Medical Imaging	
	Precision Agriculture	
	Gaming	

7. CHANLLENGES AND FUTURE POSSIBILITES

The key challenges observed during the installation of GAs are highlighted in this section, followed by relevant research topics.

7.1 Challenges

Despite the numerous benefits, some problems must be overcome in order for genetic algorithms to progress and evolve in the future. The following are some important challenges [25].

- selection of efficient fitness function
- selection of initial population

- selection of encoding schemes
- degree of mutation and crossover
- premature convergence

7.2 Future research directions

GAs have been used in a variety of fields by altering their basic structure. By addressing the current difficulty, the optimality of a solution found using GA can be improved [25]:

- Genetic algorithms imitate the process of natural evolution.
- The mapping from genotype to phenotype in real-world conditions is difficult.
- A method for selecting the optimal degree of crossover and mutation operators should be available.
- Future work can be considered for reducing the problem of early convergence.

8. CONCLUSION

The paper has presented a study of the overview of natural evolution-based GA approach to solve complicate problems. The genetic algorithm (GA) approach to optimization is based upon the concept of survival of the fittest The GA emulates the processes of evolution and is therefore an evolutionary algorithm. In such a process the strongest elements become stronger while the weakest elements are eliminated.

Consequently, the ideas of Genetic Algorithms are covered in depth in the first section of the introduction. And in the terms of chromosomal production, it's similar to biology, including factors like selection, crossover, and mutation, all of which combine to generate genetic operations that will be applied to a at initially, the population was random.

further, the genetic algorithm can be compared to natural evolution in that it is a population-oriented search and optimization tool. The operation of genetic algorithms is slowly inspired by two fundamental natural evolution ideas; one is genetic dynamics, and the other is natural selection, which is concerned with various genetic operations such as crossover, mutation, and so on.

The application of GA approach has been demonstrated including; search, optimization, decision making, machine learning, robotics, and a variety of other technologies.

Furthermore, the evolutionary algorithm's main work flow diagram, as well as all of its functions, are covered in detail in this paper, followed by several advantages of utilizing the genetic algorithm.

Additionally , the concerns and challenges of this study are highlighted at the conclusion of this publication.

REFERENCE

1. Abdelghany A, Abdelghany K, Azadian F (2017) Airline flight schedule planning under competition. Comput Oper Res 87:20–39
2. Abdulal W, Ramachandram S (2011). Reliability-aware genetic scheduling algorithm in grid environment. International Conference on Communication Systems and Network Technologies, Katra, Jammu, pp 673– 677
3. Abdullah J (2010) Multiobjectives ga-based QoS routing protocol for mobile ad hoc network. Int J Grid Distrib Comput 3(4):57–68
4. Abo-Elnaga Y, Nasr S (2020) Modified evolutionary algorithm and chaotic search for Bilevel program- ming problems. Symmetry 12:767
5. Afrouzy ZA, Nasseri SH, Mahdavi I (2016) A genetic algorithm for supply chain configuration with new product development. Comput Ind Eng 101:440–454
6. Aiello G, Scalia G (2012) La, Enea M. A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding Expert Syst Appl 39(12):10352–10358
7. Alaoui A, Adamou-Mitiche ABH, Mitiche L (2020) Effective hybrid genetic algorithm for removing salt and pepper noise. IET Image Process 14(2):289–296
8. Alkhafaji BJ, Salih MA, Nabat ZM, Shnain SA (2020) Segmenting video frame images using genetic algorithms. Periodicals of Engineering and Natural Sciences 8(2):1106–1114
9. Al-Oqaily AT, Shakah G (2018) Solving non-linear optimization problems using parallel genetic algo- rithm. International Conference on Computer Science and Information Technology (CSIT), Amman, pp. 103–106
10. Alvesa MJ, Almeidab M (2007) MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. Comput Oper Res 34:3458–3470
11. Arakaki RK, Usberti FL (2018) Hybrid genetic algorithm for the open capacitated arc routing problem. Comput

Oper Res 90:221–231
12. Arkhipov DI, Wu D, Wu T, Regan AC (2020) A parallel genetic algorithm framework for transportation planning and logistics management. IEEE Access 8:106506–106515
13. Azadeh A, Elahi S, Farahani MH, Nasirian B (2017) A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. Comput Ind Eng 104:124– 133
14. Baker JE, Grefenstette J (2014) Proceedings of the first international conference on genetic algorithms and their applications. Taylor and Francis, Hoboken, pp 101–105
15. Bolboca SD, JAntschi L, Balan MC, Diudea MV, Sestras RE (2010) State of art in genetic algorithms for agricultural systems. Not Bot Hort Agrobot Cluj 38(3):51–63
16. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press, Inc
17. Burchardt H, Salomon R (2006) Implementation of path planning using genetic algorithms on Mobile robots. IEEE International Conference on Evolutionary Computation, Vancouver, BC, pp 1831–1836
18. Burdsall B, Giraud-Carrier C (1997) Evolving fuzzy prototypes for efficient data clustering," in second international ICSC symposium on fuzzy logic and applications. Zurich, Switzerland, pp. 217-223.
19. Burkowski FJ (1999) Shuffle crossover and mutual information. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 1999, pp. 1574–1580
20. Chaiyaratana N, Zalzala AM (2000) "Hybridisation of neural networks and a genetic algorithm for friction compensation," in the 2000 congress on evolutionary computation, vol 1. San Diego, USA, pp 22–29
21. Chen R, Liang C-Y, Hong W-C, Gu D-X (2015) Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm. Appl Soft Comput 26:434–443
22. J.R. Cheng and M. Gen (2020) Parallel genetic algorithms with GPU computing. Impact on Intelligent
23. Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80(5), 8091-8126.
24. Logistics and Manufacturing. Cheng H, Yang S (2010) Multi-population genetic algorithms with immigrants' scheme for dynamic shortest path routing problems in mobile ad hoc networks. Applications of evolutionary computation. Springer, In, pp 562–571
25. Cheng H, Yang S, Cao J (2013) Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc net-works. Expert Syst Appl 40(4):1381–1392.