(IJNCAA)

ISSN 2220-9085(ONLINE)
ISSN 2412-3587(PRINT)

INTERNATIONAL JOURNAL OF NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

Volume 8, Issue 3,
2018

SDIWC

www.sdiwc.net

International Journal of
NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

Editor-in-Chief

Editorial Board

Overview

The SDIWC International Journal of New Computer Architectures and Their Applications (IJNCAA) is a refereed online journal designed to address the following topics: new computer architectures, digital resources, and mobile devices, including cell phones. In our opinion, cell phones in their current state are really computers, and the gap between these devices and the capabilities of the computers will soon disappear. Original unpublished manuscripts are solicited in the areas such as computer architectures, parallel and distributed systems, microprocessors and microsystems, storage management, communications management, reliability, and VLSI.

One of the most important aims of this journal is to increase the usage and impact of knowledge as well as increasing the visibility and ease of use of scientific materials, IJNCAA does NOT CHARGE authors for any publication fee for online publishing of their materials in the journal and does NOT CHARGE readers or their institutions for accessing the published materials.

Publisher

The Society of Digital Information and Wireless Communications
20/F, Tower 5, China Hong Kong City, 33 Canton Road, Tsim Sha Tsui,
Kowloon, Hong Kong

Further Information

Website: http://sdiwc.net/ijncaa, Email: ijncaa@sdiwc.net,
Tel.: (202)-657-4603 - Inside USA; 001(202)-657-4603 - Outside USA.

Permissions

International Journal of New Computer Architectures and their Applications (IJNCAA) is an open access journal which means that all content is freely available without charge to the user or his/her institution. Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the articles in this journal without asking prior permission from the publisher or the author. This is in accordance with the BOAI definition of open access.

Disclaimer

Statements of fact and opinion in the articles in the *International Journal of New Computer Architectures and their Applications (IJNCAA)* are those of the respective authors and contributors and not of the *International Journal of New Computer Architectures and their Applications (IJNCAA)* or *The Society of Digital Information and Wireless Communications (SDIWC)*. Neither *The Society of Digital Information and Wireless Communications* nor *International Journal of New Computer Architectures and their Applications (IJNCAA)* make any representation, express or implied, in respect of the accuracy of the material in this journal and cannot accept any legal responsibility or liability as to the errors or omissions that may be made. The reader should make his/her own evaluation as to the appropriateness or otherwise of any experimental technique described.

The issue date is Sept. 2018.

IJNCAA

ISSN 2220-9085 (Online)
ISSN 2412-3587 (Print)

Volume 8, Issue No. 3

2018

CONTENTS

ORIGINAL ARTICLES

International Journal of
NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

The *International Journal of New Computer Architectures and Their Applications* aims to provide a forum for scientists, engineers, and practitioners to present their latest research results, ideas, developments and applications in the field of computer architectures, information technology, and mobile technologies. The IJNCAA is published four times a year and accepts three types of papers as follows:

1. **Research papers**: that are presenting and discussing the latest, and the most profound research results in the scope of IJNCAA. Papers should describe new contributions in the scope of IJNCAA and support claims of novelty with citations to the relevant literature.
2. **Technical papers**: that are establishing meaningful forum between practitioners and researchers with useful solutions in various fields of digital security and forensics. It includes all kinds of practical applications, which covers principles, projects, missions, techniques, tools, methods, processes etc.
3. **Review papers**: that are critically analyzing past and current research trends in the field.

Manuscripts submitted to IJNCAA **should not be previously published or be under review** by any other publication. Plagiarism is a serious academic offense and will not be tolerated in any sort! Any case of plagiarism would lead to life-time abundance of all authors for publishing in any of our journals or conferences.

Original unpublished manuscripts are solicited in the following areas including but not limited to:

- Computer Architectures
- Parallel and Distributed Systems
- Storage Management
- Microprocessors and Microsystems
- Communications Management
- Reliability
- VLSI

Solving Equations Systems Using Artificial Intelligence – a Survey

Liviu Mafteiu-Scai, Emanuela Mafteiu, Roxana Mafteiu-Scai
Computer Science Department, West University of Timisoara, Timisoara, Romania
liviu.mafteiu@e-uvt.ro, emanuela.mafteiu@e-uvt.ro, roxana.mafteiu97@e-uvt.ro

ABSTRACT

Even if solving equations systems is a very old and discussed problem in Algebra, this issue continues to be of great interest to both mathematicians and computer scientists, the proof being the large number of scientific papers about numerical methods that have appeared lately. In this work, more than 70 papers which proposed solving equations systems problem with techniques from artificial intelligence were reviewed. The convergence and the complexity of these methods are also discussed on this paper. Finally, conclusions and future directions are presented.

KEYWORDS: equations systems, artificial intelligence

1 INTRODUCTION

Even if solving equations systems is a very old and discussed problem in Algebra, we can say that this issue continues to be of great interest to both mathematicians and computer scientists, the proof being the large number of scientific papers about numerical methods that have appeared lately. Why this particular and intense interest in solving equation systems? The answer is simple: real world can be easier and accurate described, understood and modelled using equations systems.

Before computers, only mathematicians were primarily concerned with the study of solving the equation systems. Their interest in the issue comes from antiquity, when in *The Nine Chapters on the Mathematical Art*, a Chinese book composed by several generations of scholars from the 10th–2nd century BCE, a technique for solving equations systems very close to Gaussian elimination is presented. Also, ancient Greek mathematicians' contributions must be taken into consideration. Other important contributors to this issue were Michel Rolle (1690), Newton (1720), Nathaniel Hammond (1742) and Thomas Simpson (1755).

The computer that emerged in the last century has become the primary tool in the implementation of equations systems solving methods, equations systems that were increasingly bigger, practically prohibitive to solve them with the pencil on the paper. More, the computers generate a second stage in the development of methods of solving the equation systems. The next stage begins with using parallel computers, which required the adaptation of the known solving methods, allowing the real-time solving of large and very large systems (millions of equations).

The Artificial Intelligence and its results in recent decades have led to a new stage in solving equation systems. Different from traditional mathematical approach, this stage is not easily accepted by mathematicians.

Nowadays, mathematicians consider two main types of methods for solving equation systems: *direct* methods and *iterative* methods.

Mainly, direct methods consist of converting the system of equations into an equivalent triangular system that is easier to solve. Direct methods make it possible to determine the exact solution of the system only in the ideal case when there are no rounding errors due to the finite representation of numbers in computer memory. Major problems arise when systems with more than 100 equations are solved with direct methods, when these methods are unusable due to the accumulation of rounding errors that seriously alter the solution. From the complexity point of view, in general, the number of arithmetic operations performed is $O(n^3)$, which is not very encouraging for a software developer. The most known methods of this type are: Cramer, Gaussian elimination, Gauss-Jordan, LU factorization (decomposition) etc.

The iterative methods determine the solution on the basis of an iterative process, starting from an initial approximation of the solution, in the conditions in which the equations system is well conditioned. In practice, the iterative process is stopped after a preset number of steps (unknown a priori) or after certain conditions have been met. By iterative methods, the solution of a system of linear equations is obtained by generating a string that tends to its exact solution. A major advantage of iterative methods is that in practice rounding errors and truncation errors may be insignificant. We note that even if the solution is affected by truncation errors that are not encountered in direct methods, the iterative solution may have better accuracy than the solution obtained through direct methods [51]. The most known methods of this type are: Gauss-Seidel, conjugate gradient, Quasi-Newton methods, Broyden, Chebyshev, etc.

A short presentation of advantages and disadvantages of some classical methods is made in Table 1.

Table 1. A short presentation of classical methods

	Advantages and Disadvantages
DIRECT METHODS	**-recommended for systems with dense matrix coefficients, but not for systems greater than 100**
1.Gaussian elimination	-applicable to upper and lower triangular systems. -in cases of sparse matrices, the bandwidth reduction improves the solving process. - numerical errors: accumulation of approximation errors in the last equation for calculating x_n; - the algorithm fails if the selected pivot is zero or has very low values; - full-pivoting is difficult to implement and is unstable;
2.Gauss-Jordan	- eliminates the substitution phase; - it is most used to determine the inverse matrix;
3.LU factorization	- recommended for repeated solving systems that differ only in terms of column free; - the elimination phase is more complex; - the number of operations is greater than in Gaussian elimination;; - numerical errors.
4. Cholesky factorization	- for symmetric matrices, the computational effort is reduced to half compared with Gauss or LU methods; - it is recommended for systems with positively defined matrices; - pivoting is not required; - only the lower (or upper) triangle of matrices must be stored, requiring only $n(n + 1)/2$ memory locations.
5. QR factorization	- it can be applied in the case of the singular matrix; - it is more stable than LU factorization; - can also solve systems with rectangular matrices; - ensures a good accuracy. - it is recommended for ill-conditioned systems. -fails if matrix does not have full rank (all equations must be linearly independent)
ITERATIVE METHODS	**-are recommended for large systems of equations: 10^3-10^6;** **- negligible rounding errors;** **- negligible truncation errors;** **- simpler implementation;**
1.Gauss-Seidel	- better convergence than Jacobi method; - the implementation requires little memory space; - converges even when the Jacobi method does not converge; - very good accuracy; - it is recommended for diagonal-dominant systems. - sufficient convergence conditions: - symmetric, diagonal-dominant and positive matrix - sometimes it does not converge
2. Jacobi	- easy to understand and implement; - very good accuracy; - it is recommended for diagonal-dominant systems. - weak convergence or does not always converge.
3. Southwell	- for systems with positively defined and nonsingular matrix and all elements on the main diagonal equals.
4. Conjugate Gradient	-recommended for systems with sparse matrices and each equation has a certain internal regularity; - small errors; - fast convergence for well-conditioned systems; - arbitrary, often weak, convergence for ill conditioned systems; - good convergence only for positively defined and symmetric systems. - generalized variants of method lead to lower convergence.
5. Preconditioned Conjugate Gradient	-recommended for poorly conditioned systems with high number of conditions;
6. BiConjugate Gradient	- for systems with non-symmetrical and non-singular matrices - sometimes the convergence is arbitrary / irregular and the method may fail.
7. Newton methods	- good convergence even when the condition number of the matrix is very high
8. Broyden	- very good convergence, 2n steps for linear systems with size n
9. Chebyshev	- very good convergence; - it can also be applied to non-symmetrical systems. - only for systems with positively defined matrices.

In conclusion, we can say that the right choice of a classical numerical method is not simple, and more, improper selection can lead to wrong solutions.

Besides the classical numerical methods for solving equations systems, in last the decades new methods that use artificial intelligence techniques to determine the solutions were proposed. The hybridization of these new methods with classical methods has also led to promising results.

2. TEHNIQUES FROM AI FOR EQUATIONS SYSTEMS SOLVING

In this section, short descriptions of the most used methods from artificial intelligence used for solving equations systems are made.

Monte Carlo method

Monte Carlo method (MCM) is an old search algorithm, dating back to the 1940. MCM was used for the first time in physics, after which it was used in a lot of fields such as computational biology, astrophysics, microelectronics, robot planning, species conservation, weather forecasting, air traffic control, optimization problems and many more. MCM is a probabilistic method, that uses random numbers to either simulate a stochastic behavior or to estimate the solution of a problem. From computer science point of view, Monte Carlo tree search (MCTS) is a heuristic search algorithm used in decision processes, based on the analysis of the most promising moves in the decision space and building a search tree according to the results. Expanding this search tree is based on random sampling of the search space. At the heart of MCM, it is a uniform random number generator that produces an infinite stream of random numbers inside interval (0, 1). MCM is naturally parallel because many independent samples are used to estimate the solution. These samples can be computed in parallel, thereby speeding up the solution finding process. We note that in the case of an optimization problem, there is a risk of getting stuck in local minima. A detailed description of MCM can be found in book [54].

Artificial neural networks

An artificial neural network (ANN) is a computing system inspired by the biological neural networks that constitute human's brain. Such systems learn by examples as the human brain often does. The basic elements of ANN are artificial neurons. An ANN is similar to a directed graph in which nodes are represented by artificial neurons and edges with weights, that are connections between input and output neurons. Each input has a weight, ~~an~~ information used by the ANN to solve a problem. On a standard neural network, a lot of artificial neurons are arranged in three layers: input, hidden and output. In this standard model, we have a fully connection i.e. each hidden neuron is fully connected to the every neuron in its previous layer (input) and to the next layer (output). An ANN learns by adjusting its weights and bias (threshold) iteratively to yield desired output, after a process called *training.* The training process is performed using defined set of rules, i.e. a learning algorithm. In practice we have many ANN architectures, most often determined by the hidden layer's connection mode: Perceptron, Radial Basis Function Network, Multilayer Perceptron, Recurrent Neural Network, Hopfield Network, Boltzmann Machine Network, Convolutional Neural Network, Modular Neural Network, etc.

Genetic algorithms

Genetic Algorithms (GA) mimic the processes observed in natural evolution, like natural selection and genetics, following the principles of first laid down by Charles Darwin of "survival of the fittest". In general, the scope of GA is to solve optimization problems. The father of the GA is considered to be John Holland, in the early 1970's [39]. GAs are iterative processes, that use a specific population called generation, who exists on each iteration. At each iteration, for all individuals, the fitness value is computed, i.e. the value of the objective function in the optimization problem being solved. To improve the fitness function values, operations like crossover, mutation and selection are used on each iteration. We note that the elitism selection process is the basic of most GAs. The process is finished after a predefined number of iterations, a solution was found or if some other conditions are full-filed. Regarding to “solution was found” as a stopping criterion, there is a problem of GAs, because the current solution is compared only with the previous solution, so, we do not know if the algorithm is not into a local optima. Another limitation of GAs is their bad complexity, especially when we have large search space size, big population and complex problems to be solved.

Particle swarm optimization (PSO)

This method, proposed by Kennedy and Eberhart [52], is a population-based on stochastic approach, designed for solving optimization problems, a method that simulates the "social behavior" of living communities. In PSO, software agents called particles, move in the search space of the optimization problem. The position of each particle represents a candidate solution of the problem and in order to improve this, each particle searches better positions in the search space by changing its velocity. The main difference between GA and PSO methods is that PSO does not have crossover and mutation operators, but the particles adapt their positions (i.e. improve the fitness) using self-internal velocity, self-memory and a sharing mechanism for information. As advantages, PSO is easy to implement and has only a few parameters to adjust, compared to GA. The major disadvantage is the tendency to a premature convergence in mid optimum points. Much more about PSO method can be found in book [17].

Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a metaheuristic optimization based on a probabilistic technique, proposed by Marco Dorigo in his PhD thesis in 1992 [21]. The algorithm is inspired by ants' life, i.e. searching an optimal path between their colony and source of food. In time, three basic models are proposed: Ant System, Ant Colony System (ACS), and MAX-MIN Ant System (MMAS). The most important advantages of ACO are: inherent parallelism and possibility to use it in dynamic applications. The main disadvantages are: the probability distribution can be changed by iterations and an uncertain time to converge, even if the convergence is guaranteed.

3. METHODS FROM AI FOR SOLVING EQUATIONS SYSTEMS

A brief description of the proposed methods for solving equation systems using intelligent artificial techniques is made in this section.

3.1 Monte Carlo Methods (MCM)

A method that use Monte Carlo algorithm is proposed in paper [45] to solve the nonlinear systems. The proposed method is very simple, because it does not require the differentiation of left side functions of the system. Another advantage is that the initial estimation of the solution is not necessary. The main disadvantages consist in arbitrary accuracy of the solutions, its applicability only for smaller systems, a slower convergence and a large number of generation needed for large systems.

In technical report [34], an improved solving process for large -linear and nonlinear- equations systems that uses sequential Monte Carlo technique is presented.

A Monte Carlo algorithm for solving the equations with polynomial nonlinearity and systems of algebraic quadratic equations is proposed in work [23].

A parallelization of quasi-Monte Carlo method for solving very sparse systems of linear equations is proposed in paper [73]. The associated matrix of the equations system is fully stored on each processor, a scheme that allows each processor to generate independently of each other the random walks. An advantage of the proposed method consists in possibility to solve equations systems that differ only by right-hand side vectors.

In order to solve systems of linear equations using Monte-Carlo method, three methods and their average complexity for generating the trajectories of the Markov chain associated to the linear equations system were proposed in [95].

For solving nonlinear and linear systems of equations, in paper [16] Monte Carlo method was used. It was used queueing networks for Monte Carlo method combined with some results from game theory. Examples with full explanations of the proposed method are also presented in the paper.

An improvement of the performance of the Monte Carlo algorithm from time complexity point of view by reusing the random walks to estimate the other unknowns of the equations system is proposed in paper [47].

3.2 Artificial Neural Networks (ANN)

In paper [15] some artificial neuronal networks circuits architectures that can be implemented on VLSI circuits, are investigated for solving systems of linear equations. Simulating the proposed architectures on a computer, using a set of relevant examples, validates the proposed architectures.

A method for solving linear equations systems by artificial neural network is proposed in paper [71]. In this approach, the used neural network is a Hopfield network with a Lyapunov energy function.

A method based on a neural network for solving nonlinear equations systems for which the computing time does not depend on the size of the system, is proposed in work [83].

In paper [75] recurrent neural networks are proposed for solving nonlinear equations systems, by approximating them with a multilayer perceptron (MLP). The iterative method that has been simulated in neural network was Newton's method. The backpropagation learning algorithm was used in this work.

A recurrent neural network for solving systems of inequalities and equations systems is proposed in [109]. Their approach is a generalization of the method proposed in [11]. Also, the digital realization of the proposed recurrent neural networks (continuous time and discrete time), that use a one-dimensional systolic array of processors is presented. We note that the number of neurons in proposed approach is increasing only linearly with the problem size, which is an important advantage.

In paper [72] a neural network to solve nonlinear equations systems is proposed. The learning algorithm uses is a back propagation algorithm. The networks for 2x2 and 3x3 equation systems are exemplified and finally the general case, *n*x*n* system, is described.

A neural network algorithm for solving nonlinear equations systems using a gradient descent rule with variable step-size is proposed in work [59]. The convergence of the proposed algorithm is proved in this paper.

In paper [32] two neural network algorithms are proposed for solving nonlinear systems: first for finding singular and the second for finding multiple roots. The convergence of proposed algorithm was proved in this work.

A neural network approach to solve a large linear equations system is proposed in paper [99]. In their approach, the authors consider that there are no known details about the internal structure of the equations system, i.e. a Black- Box approach. The main part of the proposed method consists in identification of system details, more exactly, finding a relationship model, determining the system orders and approximation of the unknowns function by neural network model, using a multi-layer feed-forward artificial neural networks.

A feed-forward neural network capable of solving nonlinear algebraic systems with polynomials equations is proposed in paper [29]. The proposed method is validated by solving a 3x3 nonlinear equations system and comparing the obtained results with those obtained by Mathematica software package. The proposed neural network is capable of finding all roots of a nonlinear equations systems – eight in the case of reported example- even if this requires more iterations/steps.

In paper [30] a neural network architecture that uses a back-propagation algorithm and an adaptive learning rate procedure for solving nonlinear equations systems, is proposed.

3.3 Genetic Algorithms (GA)

A parallel evolutionary algorithm for solving systems of nonlinear equations is proposed in paper [121]. The results are obtained using a simulation of parallelism on a serial computer, so, there are missing useful indicators like speedup or scalability.

Methods for solving non-linear equations using GAs are proposed in paper [74]. Square function, absolute function and a function for minimizing a linear combination of the equations are used to transform the equations system into a suitable problem for genetic algorithms.

In paper [31], the proposed method transforms the nonlinear equations systems into a multi-objective optimization problem. It is considered a multi-objective optimization problem because each equation is considered an objective. An evolutionary algorithm is used for solving the problem. A solution is a vector whose length is equal to the number of system‘s unknowns. To obtain new candidate solution, crossover and mutation operators are used. In order to compare the solutions between them, the Pareto dominance [18] relationship is used. The process is iterative, ending after a preset number of iterations. Real-world examples presented in the paper (application from interval arithmetic benchmarks, application from neuropsychology, chemical equilibrium application, kinematic application, combustion application and economics modelling), with experimental data and graphical representations, give a clear and accurate picture of the algorithm.

A simple genetic algorithm for solving linear equations systems is proposed in paper [19]. The algorithm uses a standard fitness function that must be minimized by genetic algorithm.

A genetic algorithm for solving linear systems of equation is proposed in paper [42]. The approach is classical in terms of genetic algorithms, i.e. the equations system is rewritten in terms of objective functions that must be minimized in terms of their absolute values. To compute the fitness of each chromosome, the *coefficient of multiple determination (R^2)* [27] was used, also called the *squared multiple correlation coefficient*. The value of R^2 (equal with 1 or very close) or a preset number of iteration are the criteria to stop the algorithm. In the reported experiments, the solutions obtained with proposed algorithm are compared with the results obtained with Gaussian elimination. It can be seen from these results that the genetic algorithm always finds solutions, and more, finds multiple sets of solutions (when available) for a given system of equations.

In paper [43], the effects of working parameters of genetic algorithms in solving linear equations systems are studied.

An adaptive genetic algorithm for solving ill-conditioned equations system is proposed in [60]. The main changes were made to penalty function, population migration and elite individuals reservation. The new proposed fitness function is a combination between a classical fitness function and a penalty function.

In paper [56] a rugged genetic algorithm for solving simultaneous nonlinear equations is proposed. Unlike the classic approach where a chromosome is a string, in the proposed approach a chromosome is a string-ring, i.e. the last bit is connected to the first and the crossover operator is annular. A constraint handling strategy, described in [55], was used to improve the algorithm. In their experiments, the authors use two different binary encodings: *Gray* and *Weighted binary* strategies, the errors in case of first seem to be less.

To improve the population diversity in a genetic algorithm used to solve equations systems, in paper [94], pairs of symmetric and harmonious individuals are introduced.

3.4 Particle Swarm Optimization (PSO)

A PSO algorithm is proposed in [9] to solve systems of unconstrained equations. The proposed algorithm is able to find multiple roots of the system in only one run using a modified fitness function.

In paper [118] a modified PSO algorithm through the control of the search dynamics using a Proportional–Integral–Derivative (PID) controller for solving systems of equations is proposed.

The Bacterial Foraging Algorithm (BFA) [84] was improved using PSO in order to solve nonlinear equations systems in paper [65].

In paper [4] a PSO method for solving nonlinear equations systems is proposed. In this paper, it is demonstrated that a system of equations can be transformed into an optimization problem. It is also showed that if the swarm gets bigger, the number of required iterations on PSO algorithm goes down. The experimental data also shows the preference of the PSO method of finding roots which are closer to the starting point of iterations.

A PSO method is proposed in [44] for solving systems of nonlinear equations. In the proposed algorithm, new relationships are introduced for updating the particles. The reasons for the changes in basic PSO algorithm were to diminish the tendency to stay in local optima and to increase the convergence speed. The major modification consists in introducing a set of four random variables and a greater dependence of working parameters on the values of the best individuals/particles. Moreover, the worst particles are used in the new generation process. Five standard benchmark functions are used to test the performance of the modified PSO algorithm. The results show a good convergence, about 100 iterations. The results in solving nonlinear systems are compared with those reported in other works.

A PSO algorithm for solving nonlinear equations systems -transformed into an optimization problem- is proposed in paper [108]. Chaotic maps are used to improve the effectiveness of the proposed algorithm and the *Logistic map* [103] seems to be the optimal choice. Nine known examples of nonlinear systems are used in experiments and the results are compared with results obtained by many other methods from literature.

For a real problem that consists in analysis of flow in water distribution networks using *Content Model*, solving equations systems (linear and nonlinear) is needed. In paper [79] a PSO method is used instead of Newton method. The major advantage of the proposed PSO method is that is able to find the global optimum in an uncertain problem, while Newton' methods cannot do this in the case of non-convex problems. The PSO method is modified for constrained optimization problem. The experiments are made using the modified PSO method and the General Gradient Algorithm (GGA), an iterative method for solving equations systems. The experiments show that PSO is recommended in case of non-convex problems while GGA is better in case of convex problems.

A hybridization between PSO algorithm and Artificial Bee Colony (ABC) [49] algorithm is proposed in [48] for solving nonlinear equations systems.

In paper [122] a hybridization between PSO and Artificial Fish Swarm (AFS) [58] algorithm is proposed for solving ill-conditioned linear systems of equations. The proposed algorithm combines good local convergence of PSO with global convergence performance of the AFS.

3.5 Ant Colony Optimization (ACO)

Based on basic ACO algorithm, in paper [77] an improved variant of ACO for solving nonlinear equations systems is proposed. Experimental results show a decreasing in number of iterations required to obtain the solution.

A robust method based on ACO for solving ill-conditioned linear equations systems is proposed in work [33].

For solving nonlinear equations systems, in paper [110] an improved ACO algorithm is proposed. The improvement consists in using quantum rotation gate [35] to represent and update the ants' pheromone.

3.6 Other Metaheuristics Used

An evolutionary algorithm for solving equations systems inspired from political life -very close to what we call today *globalization* - named Imperialist Competitive Algorithm (ICA) [5] is proposed in [3]. In the beginning, the individuals in evolutionary process are countries and finally we have only one empire that represents the solution of the equations system. In experimental part, comparisons of the obtained results with other known methods are made.

Using Imperialist Competitive Algorithm (ICA) with a multi-population technique, a parallel algorithm for solving systems of nonlinear equations is proposed in work [66]. In this multi-population approach, each processor has its own population and working parameters. The migration technique is also used, i.e. each processor can select the best or worst individuals and sent to the other processors. The authors reported a super linear performance of parallel algorithm proposed.

Using a combination between three different mutation strategies in a differential evolution algorithm, a method to solve nonlinear equations systems is proposed in [91].

A hybridization between *Harmony Search* (HS) metaheuristic [25] and *Differential Evolution* (DE) [102] method is used for solving systems of nonlinear equations in work [92]. HS method is used for global search and is helped by DE through updating the HS's parameters to improve the efficiency and robustness of the first. We must mention that HS is critically analyzed in the paper [120], where it is proved that this metaheuristic is only a special case of evolution strategies.

In [93] a method to find multiple roots of nonlinear equations systems based on Harmony Search (HS) optimization is proposed. To avoid previously computed solutions, in proposed algorithm a penalty function was implemented to avoid the convergence to already located solutions.

A Memetic Algorithm (MA) [82] for solving linear equations systems is proposed in paper [67]. In proposed approach, the problem of solving linear system of equations is transformed into a multi-objective optimization problem, where each equation is used to define an objective function. The proposed algorithm is able to find solutions of the given linear system of equations, even in cases where traditional methods fail~~.~~ The experimental results were validated by *Mathematica* software.

In paper [69]a metaheuristic inspired from human brainstorming (BSO) [12] combined with concepts from graph theory for solving nonlinear systems of equations, is proposed. The *creativity* needed in brainstorming optimization process is modelled like a search space using *k*-

*partite graph*s. In the cases when no exact solution exists, an approximate solution is good and it can be obtained by the proposed method, i.e. solutions whose fitness is very close to 0. Also, systems with multiple solutions can be solved; the proposed method has been able to find all the solutions inside of a given interval preset by user.

An improved variant of *Cuckoo Search*, algorithm (CS) [112] named *Double Mutation Cuckoo Search* Algorithm (DMCS) for solving systems of nonlinear equations was proposed in paper [14]. The basic idea is to make the mutations according to certain probability values, to avoid the sensitivity of the basic CS algorithm to initial values, to improve local search and to increase search space, the final effect being to improve the convergence of the DMCS algorithm. Also, the proposed algorithm is not sensitive to continuity or differentiability problems of equations.

A modified Cuckoo Optimization Algorithm [COA] for solving systems of nonlinear equations is proposed in [2]. The focus on this work is on the main disadvantage of the standard COA method [90] in solving nonlinear equation systems related to its reduced ability to find an accurate solution or to fall into a local optimum, even if the method converges very rapidly to the vicinity of the global optimum. The experimental results show a less number of fitness function evaluations compared to other evolutionary methods such as PSO, GA and COA.

Solving nonlinear equations systems is transformed into an optimization problem and, *Multistart* and *Minfinder* global optimization methods was used to solve the system in paper [107]. Multistart and Minfinder are effective due to their ability to locate both the global optimum and all the local minimum of the objective function. Thus, experimental results reported by authors indicate that all possible roots of the equations systems can be found. More, the proposed method does not depend on the problem formulation.

A self-adaptive *levy mutation* operation [57] is used for solving nonlinear equations systems in method proposed in paper [61]. In this approach, each equation represents an objective function and as a result, the entire equations system is seen as a multi-objective optimization problem.

Greedy Randomized Adaptive Search Procedure (GRASP) [24] is used to solve nonlinear equations systems in work [38]

The nonlinear equations system is transformed into a bi-objective optimization method in work [28]. The experimental results obtained with proposed method are compared with other multi-objective and single-objective methods. The results show a good scalability and robustness of the proposed method.

To find multiple roots of nonlinear systems a merit function that creates repulsive regions in the neighborhoods of the previous roots is used in method proposed in [37]. Solving the nonlinear system is seen as a global optimization, and for this, the simulated annealing (SA) algorithm is chosen. Experimental examples include real-world applications, more precisely heavy issues from chemistry.

A hybridization between Glowworm Swarm Optimization (GSO) [53] and Hooke-Jeeves pattern search [80] used for solving system of nonlinear equations is proposed in [114]. In proposed algorithm, GSO method is used to obtain the initial value of Hooke-Jeeves method.

Combining the methods Glowworm Swarm Optimization (GSO) and Simplex Search method [81] led in work [88] to a new algorithm for solving nonlinear equations systems. The proposed algorithm can find multiple solutions if they exist and can be parallelized. More, no restriction for equations, i.e. equations does not need to be continuous and differentiable.

An improved Glowworm Swarm Optimization (GSO) algorithm is proposed in [123] for solving nonlinear equations systems. The improvement consists of adding the *leader* concept, i.e. the best individual on a generation. Unlike other techniques in evolutionary algorithms, in the proposed algorithm, after each generation, all individuals move to the leader area, improving the global search.

In paper [124] a hybrid method based on Invasive Weed Optimization (IWO) [76] and Differential Evolution (DE) [102] for solving nonlinear equations systems is proposed. Experimental results obtained for small equations systems compared with results obtained with EA and PSO methods validate the proposed method and show smaller errors of solutions obtained with it.

The method proposed in [86] uses Invasive Weed Optimization (IWO) for solving nonlinear systems. The method can find the complex roots of the equations system.

For solving systems of nonlinear equations with higher dimension, a method that combines Firefly Algorithm (FA) [113] and Pattern Search (PS) [40] technique is proposed in work [119]. The first method is used in global exploration phase while the second is for exploitation phase.

Using an improved Spiral Dynamics Inspired Optimization (SDIO) [104] metaheuristic, in paper [101] a method that can find all roots of a nonlinear equations system, is proposed. We note that the method has the ability to find all the roots within a given bounded domain, as it can be seen from the experiments reported by the authors.

An algorithm for solving nonlinear equations systems using a new Probabilistic-Driven Search algorithm (PDS) is proposed in [41]. In proposed method, the nonlinear equations system is transformed into a single-objective optimization problem. The main advantage of the proposed probabilistic method is the ability to overcome local optimal solutions. The experimental results are compared with those reported in [31].

In paper [20] a method based in Quantum Computing [6] for solving nonlinear systems of equations is proposed. A modified Grover's quantum search algorithm is proposed, resulting in a more efficient computing process. Each variable of equations system is represented using a register in the quantum computer.

3.7 Hybrid Methods

In the last years, a lot of hybrid methods are proposed for solving equations systems. These methods combine metaheuristics with performant iterative methods of system solving, the results being methods with good spatial and temporal complexity and accurate solutions.

In paper [50] a hybridization between a genetic algorithm (GA) and an iterative method for solving a system of nonlinear equations, is proposed. GA is used to locate initial guesses for Newton method.

One of the methods for solving equations systems is the method of Successive Over-Relaxation (SOR), a faster convergent variant of the Gauss–Seidel method [97]. Regarding to relaxation factor, - which greatly influences the convergence rate of the SOR method- it is very difficult to estimate the optimal value of it. The goal of the evolutionary component in hybrid method proposed in paper [36] is to find the best value for the relaxation factor. The convergence of the proposed hybrid algorithm is also proved. From evolutionary process, the proposed algorithms use a deterministic recombination and a random mutation. In first experiments, the hybrid algorithm was tested for solving linear systems of equations. The results show a great advantage of the proposed algorithm which consists in a few orders of magnitude difference in errors.

In paper [46], a hybridization between Jacobi method and a time variant adaptive evolutionary algorithm is proposed for solving linear equations systems. The convergence of the hybrid algorithm is proved, and validated theoretically and experimentally in this paper. The role of the evolutionary part is to self-adapt the relaxation factor used in Jacobi method. With respect to the evolutionary part of the algorithm, the recombination involves all individuals within a population and a mutation is achieved by performing one iteration of Jacobi method. The proposed hybrid algorithm is also very simple and easy to implement both in sequential and parallel.

Paper [96] proposes a preconditioning technique based on a genetic algorithm (GA) to solve a non-linear equations system by a fixed-point method. More exactly, the genetic algorithm establishes a correct order of the equations of a non-linear system to be solved by the fixed-point method. The objective function of GA evaluates the number of equations which each individual of the population is able to solve. The detailed demonstration of the applicability of the proposed method is made using a real world problem.

A hybrid approach between chaos optimization algorithm (COA) and quasi-Newton method for solving systems of nonlinear equations is proposed in paper [64]. The COA is used to search a feasible initial guess for Newton method while the last method is used for its high speed of convergence in obtaining an accurate solution.

A hybridization between Conjugate Direction (CD) method (also known as Powell's method) [87] and Particle Swarm Optimization (PSO) is proposed in [78] for solving systems of nonlinear equations. The CD method helps PSO to jump over local minima. PSO is a stochastic algorithm that helps CD with a good initial solution, i.e. each component helps the other, which leads to a more global efficiency of the proposed hybrid algorithm (CDPSO). The experimental results reported show a better convergence of CDPSO compared to CD, PSO and Newton methods.

In paper [22], the authors propose a hybrid method for solving systems of nonlinear equations that uses GA. If in classical methods solving a system of nonlinear equations often involves transforming it into a linear system, in proposed method, this transformation is replaced by a genetic algorithm. In first part of the proposed algorithm, *Gauss–Legendre Integration* [89] scheme is used, a tool to approximate definite integrals. Using n-point formula, a system of non-linear equations of dimension *2n* is obtained. Because it is improper to use a Newton method to solve this equations system, a genetic approach is used. More exactly, a fitness function must be minimized by a genetic algorithm, i.e. the maximum absolute value of each equation in the system. In our opinion, the major inconvenience of the proposed algorithm consists in transposing a given system of nonlinear equations into a problem of form *Gauss–Legendre Integration.*

In paper [106] a hybridization between *Electromagnetic Meta-Heuristic* method (EM) [8] and *Newton-GMRES* method is proposed in order to solve nonlinear equations systems. After the transformation of equations system into an unconstrained minimization problem, the EM method is used to find a good initial guess for *Newton-GMRES* method. Large and sparse systems of equations are used in experiments and the results obtained with the proposed method are compared with other known iterative methods.

A hybrid method based on Genetic Algorithms (GAs) and Artificial Neural Networks (ANN) for solving ill-conditioned system of linear equations is proposed in [98]. The GA component is responsible for finding an initial solution for the equations system, while ANN has the role of refining the primary solution.

Paper [68] proposes a simplified form of a memetic algorithm to improve the convergence of iterative methods for solving systems of equations, i.e. the memetic algorithm is used to determine an initial vector favorable to a rapid convergence for iterative methods. The experimental results obtained with other iterative methods like conjugate gradient, Newton, Chebyshev and Broyden, recommend the proposed hybridization, that is a better alternative in comparison with that random chosen values.

A hybridization between a genetic algorithm and an augmented Lagrangian function to solve equations systems is proposed in [85]. In fact, the original constrained problem is replaced by a sequence of unconstrained sub-problems.

In paper [1] is proposed a hybrid method for solving ill-conditioned equations system for which the associated matrix has the condition number greater than 1. In the proposed algorithm, there is a hybridization between an iterative method and an evolutionary method, more precisely, conjugate gradient (CG) [97] and flower pollination algorithm (FPA) [115]. That is in fact a combination between a faster convergence and a faster global search. The main steps of the proposed algorithm are: convert the equations system into a fitness function, use the FPA part to obtain a good initial guess and finally use the CG part to solve the system. The reported results show a better accuracy of the solution obtained with proposed algorithm compared with other methods.

Figure 1 shows researchers' concern in solving equation systems using artificial intelligence techniques, reflected in the number of published scientific papers. An increase in this interest can be seen especially in the last decade. Also in the last decade one can observe a dominance of metaheuristic methods and an increase of the interest for hybrid methods.

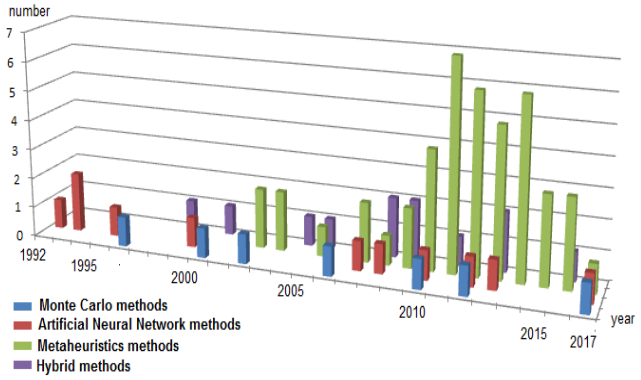


Figure 1:Papers about solving equations systems using AI

4. CONVERGENCE AND COMPLEXITY

It is known that there is a skepticism of mathematicians in solving numerical problems by other techniques than purely mathematical techniques, i.e. methods that use techniques from artificial intelligence. They claim in particular that there are no rigorous mathematical demonstrations regarding to the convergence of non-traditional methods, and unfortunately they are right in many cases. Nevertheless, there are a few published papers in which, for the particular case of the proposed method, the convergence of the method is mathematically demonstrated.

In paper [4] it is proved that solving a nonlinear equations system can be transformed into an optimization problem. The authors point out that the demonstration is valid only if the system of equations is consistent, that is, at least one solution. The demonstration is based on the description of the system of equations in the form of coordinate functions and their minimization in the case which a solution exist. As it was seen in the previous section, most approaches transform the solving of a system of equations into an optimization problem, so, we can take into consideration the mentioned paper.

In [7] the authors prove the convergence of Monte Carlo method in solving sparse linear systems for some particular cases of the associated matrix of the system. There are identified classes of associated matrices with guaranteed convergence: strictly diagonally dominant, generalized diagonally dominant and block diagonally dominant. With regard to the Monte Carlo method, in same paper the difficulty of accurately establishing the order of complexity due to the stochastic nature of the method was shown. At first sight we can speak of a complexity class O(n), but a strong dependence of complexity on the size of the system of equations, is marked in the mentioned paper. Also, a dependence of the complexity on the number of histories and the length of random walk it is showed.

In [34] the author shows the conditions in which Monte Carlo method used in solving nonlinear equations systems has quadratic convergence as in Newton's method.

In paper [109] rigorous proofs on the global convergence of the proposed neural networks for solving systems are given.

The convergence of the hybrid algorithm proposed in paper [36] is proved there. The convergence demonstration is done both for recombination and for mutation of the evolutionary component of the hybrid algorithm.

In [59] a theorem that proves the convergence of the proposed neural-network algorithm for solving equations systems, is given and proved.

In [32] the convergence of the neural-network algorithms proposed is proved by the theorem given. For this, an error function and a Lyapunov function are defined.

The following papers refer to the convergence of evolutionary algorithms, which even if they do not demonstrate their convergence in solving the systems of equations in particular, should be taken into consideration because solving equations systems is most often seen as a problem of optimization.

Brain storm optimization (BSO) algorithm is analyzed from Markov model point of view in paper [125]. The proposed Markov model gives the theoretical probability of the occurrence of each possible population as the number of generation count goes to infinity. More, using the proposed model, the convergence of the BSO method is analyzed.

In paper [62] the extension of genetic algorithms with a probabilistic Boltzmann reduction operator is considered and for this case, the proving of their convergence to the optimum using Markov chain is made. Another theoretical part related to genetic algorithms convergence is developed in work [100]. In paper [26] the convergence properties of genetic algorithms are discussed and it is shown that running a genetic algorithm for a sufficiently long time, its convergence to global optimum is guaranteed. More, the theoretical aspects can be used to obtain efficient implementations of genetic algorithms. In paper [63] Hamming distance is used to predict time convergence and average fitness of the genetic algorithms. Also, the worst case time complexity is provided.

A review of convergence analysis of PSO algorithms is done in paper [105]. There are some modes reviewed in order to analyze the convergence of PSO: with constriction coefficients, with limits, with differential equations, with matrices, with difference equations, etc. A relationship between particle velocity and convergence of PSO is given in [116]. In [117], convergence and rates of convergence of PSO method are analyzed using stochastic approximation methods. This was possible after the PSO procedure was rewritten as a stochastic approximation type iterative algorithm.

The ACO algorithm convergence is analyzed in paper [10]. Regarding to the ACO algorithm capability of solving a given problem, the term “failure probability” is used. This probability must be very close to zero for a good convergence of the algorithm. In that paper both types of convergence are analyzed: value and model, the last being stronger. The time needed for ACO convergence is also discussed and estimated in paper.

The global convergence analysis of Cuckoo Search method using Markov theory is done in paper [111]. It was proved that using a stochastic approach, the algorithm converges to the optimal state set. Also, it was shown that if two convergence conditions are satisfied, the global convergence of the algorithm is guaranteed.

The system of equations preconditioning is the transformation of the associated matrix to improve the efficiency/convergence in solving the system. The preconditioning process is applied especially in case of iterative methods to improve their convergence. The most used preconditioning technique seems to be the bandwidth reduction, for which the latest proposed methods are also based on artificial intelligence techniques [13],[70]. In paper [7] it is mentioned that the preconditioning guarantees the convergence of the Monte Carlo method, if the associated matrix of the system is transformed in strictly diagonally dominant, generalized diagonally dominant or block diagonally dominant.

5. CONCLUSION AND FUTURE WORK

In this work, more than 70 papers which proposed solving equations systems problem with techniques from artificial intelligence, were reviewed. Even there are a lot of performant iterative methods for this problem, the interest on using methods inspired from artificial intelligence was growing, especially in last decade, as it could be seen in final of section 3.

As it can be seen in Figure 1, metaheuristic methods dominate, but at the same time, hybridization between metaheuristic and iterative methods can give the best results in terms of time complexity, accuracy and convergence, as can be seen in reported results of many authors.

Unfortunately, there are not very rigorous mathematical formalisms regarding the convergence and complexity of all non-traditional methods, as it is in the case of iterative methods. Even in some papers the results obtained with these non-traditional methods are compared with the results obtained with iterative methods and they seem to be in favor of the first ones, we consider that the phenomenon cannot be generalized as long as there is no theoretical approach to prove this, at least for some classes of equation systems. We do not want to deny the power of artificial intelligence in solving equations systems, we just say that a serious and deep theoretical approach is needed.

Thus, the study of convergence and complexity in solving equations systems of some non-traditional methods will be one of our future concerns. Of course, a complete study of hybrid methods must be done, especially from global running time, i.e. the sum between time needed for iterative method and time needed for metaheuristic in case of hybridization is less than or greater than either of the two. A study on the influence of preconditioning on the convergence speed for metaheuristic methods and hybrid methods will be another concern, because we consider that the parallelization of these methods could be influenced by preconditioning.

REFERENCES

[1]Mohamed Abdel-Baset, Ibrahim M. Hezam, *A hybrid flower pollination algorithm for solving ill-conditioned set of equations*, Int. J. Bio-Inspired Computation, 8/4, 2016

[2] Mahdi Abdollahi, Asgarali Bouyer, Davoud Abdollahi, *Improved cuckoo optimization algorithm for solving*

systems of nonlinear equations, J Supercomput 72:1246–1269, Springer 2016

[3] Abdollahi M, Isazadeh A, Abdollahi D, *Imperialist competitive algorithm for solving systems of nonlinear equations*, Elsevier Comput Math Appl 65:1894–1908, Elsevier 2013

[4] I. Amaya, J. Cruz, and R. Correa, *Real Roots of Nonlinear Systems of Equations Through a Metaheuristic Algorithm*, Revista Dyna, 78/170, 15-23, 2011.

[5] E. Atashpaz-Gargari, C. Lucas, *Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition*, in: IEEE Congress on Evolutionary Computation, 2007, 4661–4667, 2007

[6] Benioff Paul, *The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines*, Journal of statistical physics. 22 (5): 563–591, Springer 1980

[7] Michele Benzi, T. M. Evansm, S. P. Hamilton, M. L. Pasini, S.R. Slattery, *Analysis of Monte Carlo accelerated iterative methods for sparse linear system*s, Numer Linear Algebra Appl. 2017

[8] Birbil, S.I., Fang, S.C.: *An electromagnetism-like mechanism for global optimization*, J. Glob. Optim. 25, 263–282, 2003

[9] R. Brits, A.P. Engelbrecht, F. van den Bergh, *Solving systems of unconstrained equations using particle swarm optimization*, IEEE International Conference on Systems, Man and Cybernetics 2002, ISBN: 0-7803-7437-1, IEEE CPS 2013

[10] Lorenzo Carvelli, Giovanni Sebastiani, *Some Issues of ACO Algorithm Convergence*, Ant Colony Optimization - Methods and Applications, https://www.intechopen.com/ , 2011

[11] Y. Censor and T. Elfving, *New method for linear inequalities*, Linear Alg. Appl., vol. 42, 199–211, 1982.

[12] Shi Cheng, Yifei Sun, Junfeng Chen , Quande Qin, Xianghua Chu, Xiujuan Lei, and Yuhui Shi, *A Comprehensive Survey of Brain Storm Optimization Algorithms*, Evolutionary Computation (CEC), 2017 IEEE Congress, 5-8 June 2017 San Sebastian Spain, DOI:10.1109/CEC.2017.7969498, IEEE 2017

[13] P. Z Chinn, J. Chvátalová, A.K. Dewdney, N.E. Gibbs, *The bandwidth problem for graphs and matrices—a survey*, Journal of Graph Theory, October 2006, DOI: 10.1002/jgt.3190060302, 2006

[14] Chiwen Qu, Wei He, *A Double Mutation Cuckoo Search Algorithm for Solving Systems of Nonlinear Equations*, International Journal of Hybrid Information Technology 8/12, 433-448, 2015

[15] A. Cichocki, R. Unbehauen, *Neural networks for solving systems of linear equations and related problems*, IEEE Trans. on Circuits and Systems-I, vol. 39, no. 2, Feb. 1992, 124-138, 1992

[16] Daniel Ciuiu, *Solving nonlinear systems of equations and nonlinear systems of differential equations by the Monte Carlo method using queueing networks and games theory*, Analele Universitatii Bucuresti, Seria Informatica 1, 111-125., 2010

[17] Maurice Clerc, *Particle Swarm Optimization*, Wiley, ISBN: 978-1-118-61397-9, 2013

[18] Coello Coello**,** Carlos, Lamont**,** Gary B., van Veldhuizen, David A., *Evolutionary Algorithms for Solving Multi-Objective Problems*, ISBN 978-0-387-36797-2, Springer 2007

[19] Al Dahoud Ali, Ibrahiem M. M. El Emary, Mona M. Abd El-Kareem, *Application of Genetic Algorithm in Solving Linear Equation Systems*, MASAUM Journal of Basic and Applied Science, ½, 2009

[20] L.A. Daoud, *Quantum Computing for Solving a System of Nonlinear Equations over GF(q)*, The International Arab Journal of Information Technology, 4/3, 201-205, 2007

[21] M. Dorigo, V. Maniezzo, A. Colorni, *Ant system: optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 26/1, February 1996, 29-41, IEEE 1996

[22] El-Emary I.M.M., Abd El-Kareem M.M., *Towards Using Genetic Algorithm for Solving Nonlinear Equation Systems*, World Applied Sciences Journal 5/3,. 282-289, 2008

[23] Ermakov S., Kaloshin I., *Solving the Nonlinear Algebraic Equations with Monte Carlo Method*, In: Balakrishnan N., Melas V.B., Ermakov S. (eds) Advances in Stochastic Simulation Methods. Statistics for Industry and Technology. Birkhäuser, Boston, MA, 2000

[24] T.A. Feo and M.G.C. Resende, *Greedy randomized adaptive search procedures*, Journal of Global Optimization, 6:109–133, 1995

[25] Z.W. Geem, J.H. Kim and G. Loganathan, *A new heuristic optimization algorithm: harmony search*, Simulation, 76/2, 2001, 60-68, 2001

[26] David Greenhalgh, Stephen Marshall, *Convergence Criteria for Genetic Algorithms*, SIAM Journal on Computing, 30/1, 269-282, 2000

[27] Glantz, Stanton A.; Slinker, B. K. (1990). *Primer of Applied Regression and Analysis of Variance*. McGraw-Hill. ISBN 0-07-023407-8., 1990

[28] Wenyin Gong, Yong Wang, Zhihua Cai, and Shengxiang Yang, *A Weighted Biobjective Transformation Technique for Locating Multiple Optimal Solutions of Nonlinear Equation*, IEEE Transactions on Evolutionary Computation - October 2017

[29] K. Goulianas, A. Margaris, M. Adamopoulos, *Finding all real roots of 3x3 nonlinear algebraic systems using neural networks*, Applied Mathematics and Computation 219, 4444–4464, Elsevier 2013

[30] K. Goulianas, A. Margaris, I. Refanidis, K. Diamantaras, *Solving polynomial systems using a fast adaptive back propagation-type neural network algorithm*, European Journal of Applied Mathematics, https://doi.org/10.1017/S0956792517000146, 2017

[31] Grosan C., Abraham A., *A New Approach for Solving Nonlinear Equations Systems*, IEEE Transaction on Systems, Man and Cybernetics, part A: Systems and Humans, 38/3, 2008

[32] Xiangde Guo, Zhezhao Zeng, *The Neural-Network Approaches to Solve Nonlinear Equation*, Journal Of Computers, 5/3, March 2010

[33] Duan Haibin, Wang Daobo, Zhu Jiaqiang, *Novel method based on ant colony optimization for solving ill-*

conditioned linear systems of equations, Journal of Systems Engineering and Electronics, 16/3, 2005

[34] John H. Halton, *On Accelerating Monte Carlo Techniques for Solving Large Systems of Equations*, TR96-041, Dept. of Computer Science University of North Carolina at Chapel Hill Chapel Hill, NC 27599-3175, 1996

[35] K.H. Han and J.H. Kim, *Quantum-inspired evolutionary algorithm for a class of combinatorial optimization*, IEEE Trans. Evol. Comput., 6/6, 580-593, Dec. 2002.

[36] Jun He, Jiyou Xu and Xin Yao, *Solving Equations by Hybrid Evolutionary Computation Techniques*, IEEE Transactions on Evolutionary Computation, 4/3, 295 - 304, ISSN: 1089-778X, 2000

[37] Henderson N, Sacco WF, Platt GM, *Finding more than one root of nonlinear equations via a polarization technique: an application to double retrograde vaporization*, Chem Eng Res Des 88:551–561, Elsevier 2010

[38] Michael J.Hirsch, Panos M.Pardalos, Mauricio G.C.Resende, *Solving systems of nonlinear equations with continuous GRASP*, Nonlinear Analysis: Real World Applications , 10/4, Elsevier 2009

[39] *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology*, Control and Artificial Intelligence, MIT Press Cambridge, MA, ISBN:0262082136, 1992

[40] Hooke, R.; Jeeves, T.A., *"Direct search" solution of numerical and statistical problems*, Journal of the Association for Computing Machinery (ACM). 8 (2): 212–229. doi:10.1145/321062.321069, 1961

[41] T. Nguyen Huu and H. Tran Van, *A new probabilistic algorithm for solving nonlinear equations systems*, Journal of Science, vol. 30, 1–14, 2011

[42] Ikotun Abiodun M., Lawal Olawale N., Adelokun Adebowale P., *The Effectiveness of Genetic Algorithm in Solving Simultaneous Equations*, International Journal of Computer Applications (0975 – 8887) 14/8, 2011

[43] A M Ikotun, A T Akinwale, O T Arogundade, *Parameter Variation For Linear Equation Solver Using Genetic Algorithm*, Journal Of Natural Sciences Engineering And Technology, 15/2, 2016

[44] Majid Jaberipour, Esmaile Khorrama, Behrooz Karimi, *Particle swarm algorithm for solving systems of nonlinear equations*, Computers and Mathematics with Applications 62, 566–576, Elsevier, 2011

[45] Aleksander Jablonski, *A Monte Carlo algorithm for solving systems of non-linear equations*, Journal of Computational and Applied Mathematics, 6/3, 171-175, Elsevier 1980

[46] A.R. M. Jalal Uddin Jamali, M. M. A. Hashem, Md. Bazlar Rahman, *Solving Linear Equations Using a Jacobi Based Time-Variant Adaptive Hybrid Evolutionary Algorithm*, Procs. of the 7th International Conference on Computer & Information Technology (ICCIT 2004), 688-693, Dhaka, Bangladesh, December 26-28, 2004

[47] Hao Ji, Yaohang Li, *Reusing Random Walks in Monte Carlo Methods for Linear Systems*, Procedia Computer Science 9, 383 – 392, Elsevier 2012

[48] Ruimin Jia, Dengxu He, *Hybrid Artificial Bee Colony Algorithm for Solving Nonlinear System of Equations*, Eighth International Conference on Computational Intelligence and Security, IEEE 2012

[49] D. Karaboga, *An idea based on bee swarm for numerical optimization*, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[50] Karr CL, Weck B, Freeman LM, *Solutions to systems of nonlinear equations via a genetic algorithm* Engineering Applications of Artificial Intelligence 1998; 11:369 –375, Elsevier 1998

[51] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, ISBN-13: 978-0-898713-52-7, Philadelphia, 1995

[52] J. Kennedy and R. Eberhart, *Particle swarm optimization*, in Proceedings of IEEE International Conference on Neural Networks, pages 1942-1948, IEEE Press, Piscataway, NJ, 1995.

[53] Krishnanand K.N., Ghose D., *Glowworm swarm optimisation: a new method for optimising multi-modal functions*, Int. J. Computational Intelligence Studies, 1/1, 93-119, 2009

[54] D.P. Kroese, T. Taimre, Z.I. Botev, *Handbook of Monte Carlo Methods*, Wiley Series in Probability and Statistics, John Wiley and Sons, New York, 2011

[55] Kuri, A., Gutiérrez, Jesús, *Penalty Function Methods for Constrained Optimization with Genetic Algorithms:A Statistical Analysis*, Lecture Notes in Artificial Intelligence, LNAI 2313, Springer, 108-117, 2002.

[56] Angel Fernando Kuri-Morales, Río Hondo No, DF México, *Solution of Simultaneous Non-Linear Equations using Genetic Algorithms*, WSEAS Transactions on Systems 2/1, 2003

[57] Lee, C. Y., Yao, X.*: Evolutionary Algorithms with Adaptive Levy Mutations*, Proceedings of the 2001 Congress on Evolutionary Computation, pp.568-575, IEEE CPS, 2001

[58] Li, X., Shao, Z., Qian, J., *An optimizing method based on autonomous animats: Fish-swarm algorithm,*. Systems Engineering and Theory and Practice 22/11, 32–38, 2002

[59] Li G, Zeng Z (2008) *A neural-network algorithm for solving nonlinear equation systems*. In: IEEE International Conference On Computational Intelligence And Security, vol 1, 20–23, 2008

[60] Li P., Tong X., *An Adaptive Genetic Algorithm for Solving Ill-Conditioned Linear Equation Group*. In: Wu Y. (eds) Software Engineering and Knowledge Engineering: Theory and Practice. Advances in Intelligent and Soft Computing, vol 114. Springer 2012

[61] Chun-an Liu, *A Special Multiobjective Evolutionary Algorithm for Solving Complicated Nonlinear Equations Systems*, Applied Mechanics & Materials, Vol. 701/702, 18-23, 2014

[62] J.A. Lozano, P. Larranaga, M. Grana, F.X. Albizuri, *Genetic algorithms: bridging the convergence gap*, Theoretical Computer Science 229, 11–22, Elsevier 1999

[63] Sushil J. Louis and Gregory J. E. Rawlins, Technical Report TR370: *Predicting Convergence Time for Genetic Algorithms*, Indiana University 1993

[64] Luo YZ, Tang GJ, Zhou LN, *Hybrid approach for solving systems of nonlinear equations using chaos optimization and quasi-Newton method*. Applied Soft Computing 8, 1068–1073, Elsevier 2008

[65] X. F. Mai and L. Li, *Bacterial Foraging Algorithm Based on PSO with Adaptive Inertia Weigh for Solving Nonlinear Equations Systems*, Advanced Materials Research, Vols. 655-657, 940-947, 2013

[66] Amin Majd et all, *Multi-Population Parallel Imperialist Competitive Algorithm for Solving Systems of Nonlinear Equations*, 2016 International Conference on High Performance Computing & Simulation (HPCS) 18-22 July 2016, IEEE CPS 2016

[67] L.O. Mafteiu-Scai, E.J. Mafteiu-Scai, *Solving liniar systems of equations using a memetic algorithm*, IJCA (0975 – 8887) 58/13, ISBN: 973-93-80870-43-5 FCS New-York, November 2012

[68] L. O. Mafteiu-Scai, *Improved the convergence of iterative methods for solving systems of equations by memetics techniques*, IJCA (0975 – 8887), 64/17, ISBN: 973-93-80873-17-5 FCS New-York, February 2013

[69]L.O. Mafteiu-Scai, *A New Approach for Solving Equations Systems Inspired from Brainstorming* , IJNCAA 5/1, 10-18, ISSN: 2220-9085, 2015

[70] L.O. Mafteiu-Scai, *The Bandwidths of a Matrix. A Survey of Algorithms*, West Univ. of Timisoara Annals, ISSN:1841-3293, DeGruyter Open, 2015

[71] K.G.Margaritis K.G, M.Adamopoulos, K.Goulianas, Solvin*g linear systems by artificial neural network energy minimisation*", 1993, Universily of Macedonia, Annals (volume XII), 502-525, (in Greek), 1993

[72] A. Margaris, M. Adamopoulos, *Solving nonlinear algebraic systems using artificial neural networks*, In Proc. of the 10th Int. Conf. on Engineering App. of Artificial Neural Networks, 107–120, 2007

[73] M. Mascagni, A. Karaivanova, *A Parallel Quasi-Monte Carlo Method for Solving Systems of Linear Equations*, Proc. of International Conference on Computational Science (ICCS02), Springer, 2002

[74] Nikos E. Mastorakis, *Solving Non-linear Equations via Genetic Algorithms*, Proceedings of the 6th WSEAS Int. Conf. on EVOLUTIONARY COMPUTING, Lisbon, Portugal, June 16-18, 24-28, 2005

[75] Karl Mathia, Richard Saeks, *Solving Nonlinear Equations Using Recurrent Neural Networks*, World Congress on Neural Networks (WCNN'95), July 17-21, 1995

[76] A.R.Mehrabian, C.Lucas, *A novel numerical optimization algorithm inspired from weed colonization*, Ecological Informatics, 1/4, December 2006, 355-366,m Elsevier 2006

[77] Zhiki Min, Yueguang Li, *A Novel Ant Colony Algorithm of Solving Nonlinear Equation Group*, IERI Procedia 2, 775 – 780, Elsevier 2012

[78] Yuanbin Mo, Hetong Liu, Qin Wang, *Conjugate direction particle swarm optimization solving systems of nonlinear equations*, Computers and Mathematics with Applications 57, 1877-1882, Elsevier 2009

[79] Moosavian N., Jaefarzadeh, M.R., *Particle Swarm Optimization for Hydraulic Analysis of Water Distribution Systems*, Civil Engineering Infrastructures Journal, 48(1): 9-22, ISSN: 2322-2093, 2015

[80] I. Moser, *Hooke-Jeeves Revisited*, IEEE Congress on Evolutionary Computation (CEC 2009), IEEE 2010

[81] Nelder, J.A. and Mead, R.. *A simplex method for function minimization*. Comput. J. 7, 1965, 308-313, 1965

[82] Ferrante Neri, Carlos Cotta, and Pablo Moscato (Eds.), *Handbook of Memetic Algorithms*, Studies in Computational Intelligence, ISBN 978-3-642-23246-6, Springer 2012

[83] Nguyen TT. *Neural network architecture for solving nonlinear equation systems*, Electronics-Letters 1993; 29/16, 1403–1405, IEEE 1993

[84] K. M. Passino, *Bacterial Foraging Optimization*, Int. Journal of Swarm Intelligence Research, 2010.

[85] Abolfazl Pourrajabian, Reza Ebrahimi, Masoud Mirzaei, Mehrzad Shams, *Applying genetic algorithms for solving nonlinear algebraic equations*, Applied Math. and Computation, 219/24, 11483-11494, Elsevier 2013

[86] Pourjafari E, Mojallali H., *Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering*, Swarm Evol Comput 4:33–43, Elsevier 2012.

[87] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical Recipes, The Art of Scientific Computing*, 3th. ed, ISBN-13 978-0-521-88068-8, Cambridge Univ. Press, 2007

[88] Liangdong Qu, Dengxu He, Jinzhao Wu, *Hybrid Coevolutionary Glowworm Swarm Optimization Algorithm with Simplex Search Method for System of Nonlinear Equations*, Journal of Information & Computational Science 8: 13, 2011

[89] Alfio Quarteroni, Riccardo Sacco, Fausto Saleri, *Numerical Mathematics*, ISBN 0-387-98959-5 SPIN 10747955, Springer 2000

[90] Rajabioun R, *Cuckoo optimization algorithm*, Appl Soft Comput 11:5508–5518, 2011

[91] Ramadas, G.C.V., Fernandes, E.M.G.P.: *Combined mutation differential evolution to solve systems of nonlinear equations.* In: 11th International Conference of

Numerical Analysis and Applied Mathematics 2013 AIP Conf. Proc., vol. 1558, 582–585, 2013

[92] Gisela C. V. Ramadas, Edite M.G.P. Fernandes, *Solving Systems of Nonlinear Equations by Harmony Search*, Proceedings of the 13th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2013 24-27 June, 2013.

[93] Ramadas G.C.V., Fernandes E.M.G.P., Rocha A.M.A.C., *Multiple Roots of Systems of Equations by Repulsion Merit Functions*, In: Murgante B. et al. (eds) Computational Science and Its Applications – ICCSA 2014, LNCS, vol 8580. Springer 2014

[94] Hongmin Ren, Long Wu, Weihong Bi, Ioannis, K. Argyros, *Solving nonlinear equations system via an efficient genetic algorithm with symmetric and harmonious individuals*, Applied Mathematics and Computation, 219/23/1, 10967-10973, Elsevier 2013

[95] Natalia Rosca, *Monte Carlo Methods For Systems Of Linear Equations,* Studia Univ. "Babes¸–Bolyai", Mathematica, LI/1, 2006

[96] Antonio Rovira, Manuel Valdes, Jesus Casanova, *A new methodology to solve non-linear equation systems using genetic algorithms. Application to combined cycle gas turbine simulation*, Int. J. Numer. Meth. Engng 2005; 63:1424–1435, Wiley InterScience DOI: 10.1002/nme.1267, 2005

[97] Saad Yousef, *Iterative methods for sparse linear systems,* (2nd ed.). Philadelphia, Pa.: Society for Industrial and Applied Mathematics, ISBN 978-0-89871-534-7, 2003

[98] M. Sammany, E. Pelican, T. A. Harak, *Hybrid neuro-genetic based method for solving ill-posed inverse problem occurring in synthesis of electromagnetic fields*, Computing, 9/4, Springer 2011

[99] N.D.K. Al-Shakarchy, E.H. Abd, *Application of Neural Equations*, Journal of K erbala University *Network For Solving Linear Algebraic*, Vol. 10 No.4 Scientific, 2012

[100] Sharapov, R.R. & Lapshin, A.V. Pattern Recognit. Image Anal. (2006) 16: 392. https://doi.org/10.1134/S1054661806030084, Springer 2006

[101] Kuntjoro Adji Sidarto and Adhe Kania, *All Solutions of Systems of Nonlinear Equations Using Spiral Dynamics Inspired Optimization with Clustering*, JACIII Vol.19 No.5 pp. 697-707doi: 10.20965/jaciii.2015.p0697, 2015

[102] Storn, R.; Price, K., *Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces*. Journal of Global Optimization. 11: 341–359, Springer 1997

[103] Tabor, M. *Chaos and Integrability in Nonlinear Dynamics: An Introduction*, New York: Wiley, 1989

[104] K. Tamura and K. Yasuda, *Spiral Dynamics Inspired Optimization*, J. of Advanced Computational Intelligence and Intelligent Informatics (JACIII), Vol.15, No.8, pp. 1116-1122, 2011

[105] Dong ping Tian, *A Review of Convergence Analysis of Particle Swarm Optimization*, International Journal of Grid and Distributed Computing,6/6,pp.117-128, http://dx.doi.org/10. 14257/ijgdc.2013.6.6.10, 2013

[106] Toutounian, F., Saberi-Nadjafi, *A Hybrid of the Newton-GMRES and Electromagnetic Meta-Heuristic Methods for Solving Systems of Nonlinear Equations*, J. & Taheri, S.H. J Math Model Algor (2009) 8: 425. https://doi.org/10.1007/s10852-009-9117-, Springer 2009

[107] I.G. Tsoulos, Athanassios Stavrakoudis, *On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods*, Nonlinear Analysis: Real World Applications 11 2465-2471, Elsevier 2010

[108] Oguz Emrah Turgut, Mert Sinan Turgut, Mustafa Turhan Cobana, *Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations*, Computers and Mathematics with Applications 68 (2014) 508–530, Elsevier 2014

[109] Youshen Xia, Jun Wang, Donald L. Hung, *Recurrent Neural Networks for SolvingLinear Inequalities and Equations*, IEEE Transactions On Circuits And Systems—I: Fundamental Theory And Applications, 46/4, 1999

[110] Y. H. Xia and Y. G. Li, *An Improved Quantum Ant Colony Algorithm of Solving Nonlinear Equation Groups*, Advanced Materials Research, Vols. 1049-1050, pp. 1363-1366, 2014

[111] He XS., Wang F., Wang Y., Yang XS. *Global Convergence Analysis of Cuckoo Search Using Markov Theory*. In: Yang XS. (eds) Nature-Inspired Algorithms and Applied Optimization. Studies in Computational Intelligence, vol 744. Springer, Cham, 2018

[112] X.-S. Yang; S. Deb (December 2009). *Cuckoo search via Lévy flights*. World Congress on Nature & Biologically Inspired Computing (NaBIC 2009). IEEE Publications. pp. 210–214. arXiv:1003.1594v1, 2009

[113] Yang, X.S.: *Nature-inspired metaheuristic algorithms*, second edition, Luniver Press, Beckington, 2010

[114] Yan Yang, Yongquan Zhou, Qiaoqiao Gong, *Hybrid Artificial Glowworm Swarm Optimization Algorithm for Solving System of Nonlinear Equations*, Journal of Computational Information Systems 6:10, 3431-3438, 2010

[115] Yang, X-S. (2012) *Flower pollination algorithm for global optimization*, Unconventional Computation and Natural Computation, pp.240–249, Springer, 2012

[116] Hongtao Ye, Wenguang Luo, Zhenqiang Li, *Convergence Analysis of Particle Swarm Optimizer and Its Improved Algorithm Based on Velocity Differential Evolution*, Comput Intell Neurosci., 2013

[117] Quan Yuan, George Yin, *Analyzing Convergence and Rates of Convergence of Particle Swarm Optimization Algorithms Using Stochastic Approximation Methods*, IEEE Transactions on Automatic Control, 60/7, 2015

[118] Wang Q., Zeng J., Jie J. *Modified Particle Swarm Optimization for Solving Systems of Equations*, In: Huang DS., Heutte L., Loog M. (eds) Advanced Intelligent Computing Theories and Applications. ICIC 2007. Communications in Computer and Information Science, vol 2. Springer 2007

[119] Xiaogang Wang, Ning Zhou, *Pattern Search Firefly Algorithm for Solving Systems of Nonlinear Equations*, 2014 Seventh International Symposium on Computational Intelligence and Design, DOI: 10.1109/ISCID.2014.222, IEEE 2014

[120] Dennis Weyland, *A critical analysis of the harmony search algorithm—How not to solve Sudoku*, Operations Research Perspectives 2 (2015) 97–10, Elsevier 2015

[121] Wu Z, Kang L. *A fast and elitist parallel evolutionary algorithm for solving systems of non-linear equations*. Congress on Evolutionary Computation, IEEE Cat. No. 03TH8674., 1026–1028, vol. 2., 2003

[122] Zhou Y., Huang H., Zhang J., *Hybrid Artificial Fish Swarm Algorithm for Solving Ill-Conditioned Linear Systems of Equations*. In: Chen R. (eds) Intelligent Computing and Information Science. Communications in Computer and Information Science, vol 134. Springer, 2011

[123] Yongquan Zhou, Jiakun Liu, Guangwei Zhao, *Leader Glowworm Swarm Optimization Algorithm for Solving Nonlinear Equations Systems*, przeglad elektrotechniczny (Electrical Review), ISSN 0033-2097, R. 88 NR 1b/2012, 2012

[124] Yongquan Zhou, Qifang Luo, Huan Chen, *A Novel Differential Evolution Invasive Weed Optimization Algorithm for Solving Nonlinear Equations Systems*, Hindawi Publishing Corporation, Journal of Applied Mathematics, Vol. 2013, http://dx.doi.org/10.1155/2013/757391, 2013

[125] Ziwei Zhou, Haibin Duan, Yuhui Shi, *Convergence analysis of brain storm optimization algorithm*, 2016 IEEE Congress on Evolutionary Computation (CEC), 24-29 July 2016, IEEE, 2016

Content Based Image Retrieval using Multimodal Data based on CCA

Ismail A. El Sayad, Mohammad A. Bazzoun, Hawraa I. Younes, Laila M. Ghoteime, Samih Abdulnabi
Department of Computer and Communication Engineering
Lebanese International University, Beirut, Lebanon
Ismail.sayad@liu.edu.lb ; mohammad.bazzoun@liu.edu.lb ; hawraa.younes@outlook.com ;
laila.ghoteime87@gmail.com ; samih.abdulnabi@liu.edu.lb

***Abstract*— The development of CBIR is based on Multimodal analysis, especially for a set of images associated with some text. Multimodality is used in multimedia to improve data retrieval from the multimedia scope. Visual features and unstructured text annotations processed, analyzed, and resolved within two data modalities simultaneously, which is referred to Multimodality. In this paper, a method is proposed for combining visual and textual data to improve the performance of image retrieval from annotated data set by representing image contents and text semantics in a Multimodal analysis space.**

Keywords—CBIR; Multimodality; CCA; BOW

I. Introduction

Nowadays, visual information plays an important role in many fields such as medical treatment, crime prevention, hospitals, engineering, journalism...etc [1]. We consume and produce a large amount of visual media with the progression of multimedia technologies and the advances in data storage. Such techniques help us for image capturing, processing, storage, and transmitting, hence enabling users to access data from any place and provide the use of digital images in different fields. However, this will raise the question of which effective methods to access, search and navigate through the data to retrieve the information needed.

Visual information retrieval has attracted great interest too, due to the evolution of image creation tools. In order to deal with this visual information, insisting requests of algorithms that can efficiently meet the needs.

In general, searching images from large database has adopted two different approaches: the first is Text-Based Image Retrieval (TBIR) systems where searching is based on annotated images, and the second is based on image content information (CBIR) which uses visual contents of the images described in the form of low-level features [2].

In TBIR systems, a user provides a query in terms of a keyword and the system will return images similar to the query [3]. The TBIR systems are fast since it applies string matching which needs less computationally time compared to CBIR. But, there are some drawbacks of this type of an image retrieval system: first, a considerable level of human labor is required for manual annotation. Second, the annotation inaccuracy due to the subjectivity of human perception [1]. Additionally, it is sometimes difficult to express in words the visual content of images, and by that decreasing the performance of the keyword-based image search [3].

In order to overcome the limitations enjoined by TBIR systems, CBIR was developed as an alternative. In a typical CBIR system as illustrated in Fig. 1, image low-level features like color, shape, texture and spatial locations are represented in the form of a feature vector. Note that in some CBIR approaches, colored images do not undergo pre-processing, as they are distorted with noise resulting from devices/sensors; thus, to improve the accuracy of retrieval, we may use effective filters to remove this noise. Pre-processing is necessary when results are used for human analysis. There are several color filters available for this purpose [6].

The feature database is formed by the feature vectors of the image. The retrieval process is initiated when a user queries the system using a query image. The query image is converted into a feature vector. The similarity measure is employed to calculate the distance between the feature vectors of the target images in the feature database and the retrieval is performed using an indexing scheme.

In addition to color and texture, spatial location can be considered as a useful factor in region clustering. For instance, ‘sky’ and ‘sea’ share the same color and texture features, yet they have different spatial locations, since sky appears at an image’s top location, while sea appears at its bottom. Spatial locations are defined as upper, bottom or top based on the location of the needed region in the testing image [7]. The search focused on the region centroid and its minimum bounding rectangle to provide the retrieval system with spatial location information, where the spatial center of a region was used to represent the same spatial location [8].

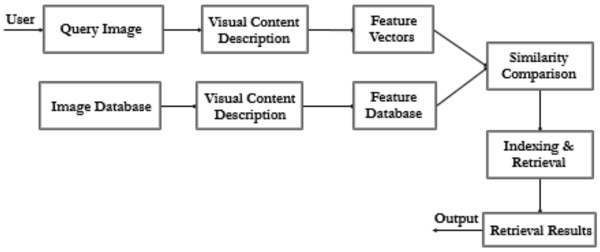


Fig. 1- Typical CBIR System

Whereas there are also some drawbacks for CBIR system where it doesn't support textual queries and it doesn't capture "semantics" by mean that it's possible to answer query 'Red ball' with 'Red Rose' [4]. Thus, retrieval systems tend to incorporate user’s relevance feedback to further improve the retrieval process and produce more meaningful retrieval results [5] and thus reducing the 'semantic gap' between low-level features and high-level features.

The paper is organized as follows: Section II presents the Multimodality information retrieval methodology based on the proposed work. Section III introduce the proposed method of Image Retrieval utilizing Multimodality. Section IV shows and interprets the result of the proposed approach. Finally, our conclusions are presented in Section VI.

II. Multimodality in Information Retrieval

Human interaction is the major difference between CBIR and TBIR systems. Humans tend to use high-level features (concepts), such as tags, text descriptors, and keywords, to interpret images and measure their similarity [10]. The difference between the limited descriptive power of low-level image features and the richness of user semantics is referred to as the 'semantic gap' [1].

One way of bridging this gap is the use of both visual and textual approach which is introduced to increase the system's performance by using both information. The goal is to join TBIR and CBIR systems into one system known as CBIR system for Multimodal data or Multimodality. Early fusion and late fusion were used to perform this also, but they came up with some drawbacks, where early fusion integrate both data modalities before a user request is received [11].

For the late fusion, it refers to those methods that preserve each data modality separately where. Moreover, late fusion combines the scores of the confidences calculated for the models composed of different features, in such a way that scores represent the possibility of classifying a test sample into the positive class by one specific model [15]. Moreover, late fusion approach lies in its failure to utilize the feature level correlation among modalities [12].

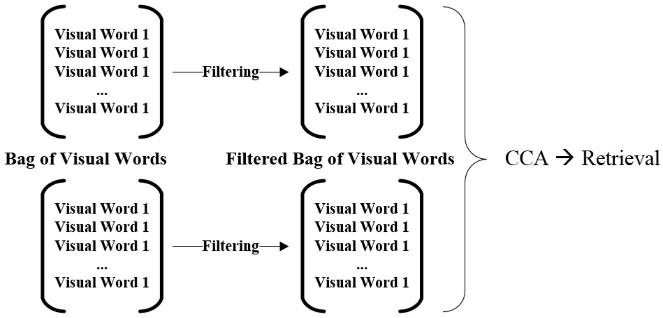


Fig. 2- Multimodal Algorithm using CCA

The approach used in this paper to fuse both visual and textual descriptors together is the Canonical Correlation Analysis (CCA) which is technically able to analyze the data involving multiple sets of variables and is theoretically consistent with that purpose as shown in Fig. 2. The description of the Multimodality algorithm using CCA is presented in the coming sections.

A. *Visual Features: Bag-Of-Visual Words*

Define Image can be represented by all words that can be generated from it using the so-called visual vocabulary or bag-of-words. The Bag-Of-Visual Words (BOW) [13] is extracted from the whole image, but it's better to form a single bag per each segmented image to avoid mixing between background and foreground.

Researchers are recently using key/local interest points in the retrieval and classification process. Those interest points are called salient image patches that contain rich information about the image. They are then grouped into clusters, such that similar descriptors are assigned to the same cluster. Each cluster is treated as a visual word. When key points are mapped into visual words, we can represent an image as a "bag of visual words", where during the classification task it is used as a feature vector [13]. Once images are represented as bags of visual words, we can classify them by building supervised classifiers. Thus, texture feature becomes a necessary composition in setting up the high-level semantics needed for image retrieval purposes. However, this feature differs from color features as it is not so well-defined, making it not used by some retrieval systems [14].

There are basically 4 steps in order to have a basic implementation of the bag of words object classification:

1. Extracting features from a set of training images.
2. Clustering like-features together into a fixed number of clusters.
3. Constructing histograms of the frequency of features in labeled images containing objects to detect.
4. Evaluating unknown images against the histograms obtained in step 3 to classify objects in the image.

The proposed approach based on the implementation of the bag of words object classification is presented in Section III.

III. Proposed Image Retrieval Approach

The Canonical Correlation Analysis (CCA) is used to improve Image Retrieval using visual and textual data of an image. The visual data (hereby known as Visual Descriptors) are extracted using TOP-SURF [18], whilst the textual data (known as Textual Descriptors) are extracted based on the TF-IDF values of the annotated tags of the images. Using CCA, Visual Descriptors and Textual Descriptors are fused together to produce a multimodal global feature that helps in the retrieval process.

A. *Textual Features*

Another way to represent the content/semantic of an image is the textual annotations. Textual annotations are texts, tags or keywords representations that mainly describe what is in the image. These annotations might be specific or general, as much specific as they could be, as much as they help in the process of retrieval.

Text annotations are generated in two ways: one way is by manual annotation of images, which requires extra human labor but are very specific to the image, which means higher precision; another way is the use of tagged datasets such as

Flickr or KODAK datasets, which have been already annotated with different tags that could be of high/low importance to the image. Most of the tags annotated to the image, are of low importance and could be noisy, and thus a way to make these tags efficient is by filtering those tags i.e. removing less important tags and keeping tags of high importance (tags that convey the visual content of the image).

After generating tags, TF-IDF values in each image is computed. And thus, each image in the training set and query set can be represented as a vector of textual descriptors.

B. *TF-IDF Calculations*

In order to represent an image as a multimodal vector representing both visual and textual content, we need to represent the image in two separate vectors: the Visual Descriptors Vector and the Textual Descriptors Vector.

One way to represent these vectors is to use the TF-IDF values of each descriptor. TF-IDF value is the weight of a descriptor in an image with respect to the whole images. TF is the Term Frequency, which represents how frequent this term is in this image.

TF-IDF is an alternative way to evaluate the subject of an object by the words it has. With TF-IDF, words are given weights – TF-IDF measures relevance, not the frequency of occurrence [16]. That is, word counts are interchanged with TF-IDF values through the complete dataset. This measure calculates the number of times that a certain word appears in a specified document, as for words such as “and” or “the” that appear recurrently in all documents, those are analytically reduced.

$$idf\ (\mathrm{t,D}) = \log \frac{N}{|\{\mathrm{d}\varepsilon\ \mathrm{D} : \mathrm{t}\varepsilon\ \mathrm{D}\}|} \qquad \textbf{(1)}$$

IDF is the Inverse Document Frequency, which represents how frequent this term in the whole dataset/document, and it can be calculated in many ways, but the method that we used is the logarithmic IDF method as shown in **(1)**. Where *N* is the total number of images and is the number of images where the term t appears.

The TF-IDF value is calculated by multiplying the TF value with the IDF value. The method of evaluating TF-IDF is easier to be done manually for textual vectors. As for visual vectors, we use computer-based technique; i.e. TOP-SURF, to extract those descriptors and get their TF-IDF values.

C. *Canonical Correlation Analysis*

Canonical Correlation Analysis (CCA) [17] is used to combine the visual and textual features, where it applies feature level fusion. Feature fusion is the process of combining two feature vectors to obtain a single feature vector, which is more discriminative than any of the input feature vectors.

CCA is a statistical method to perform multi-view/multi-scale analysis for different data sources. It is a method for correlating linear relationships between multidimensional variables, which is finding projections (linear) for multiple data types that have maximum correlation into a high dimension feature space. CCA integrates Multimodal features to cross-modal features. It projects those features of different data modalities on multiple views into a common subspace to get the maximum correlation between visual and semantic features [12].

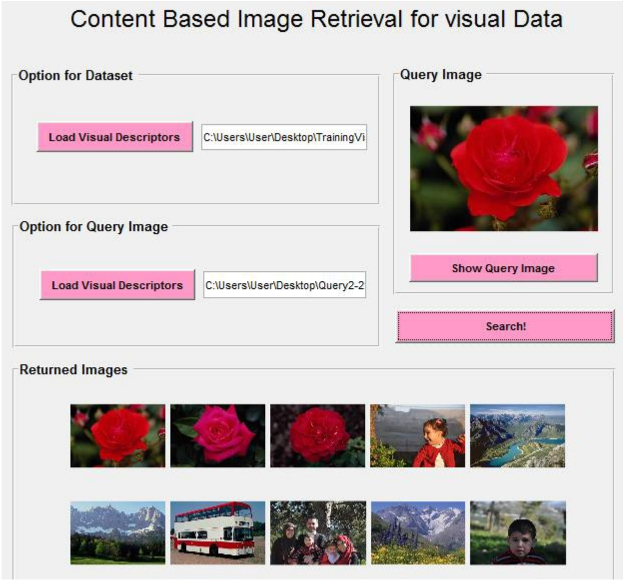
Four variables will be needed as inputs to the CCA implementation, two are related to the dataset (Visual & Textual) and the remaining two are related to the query image (Visual & Textual). The first input is "*trainX*" which is an “*n*×*p*” matrix containing the first set of training data where *n* is the number of training samples and *p* is the dimensionality of the first feature set. Then “*trainY*”, an “*n*×*q*” matrix containing the second set of training data where *q* is the dimensionality of the second feature set.

For the query image, “*testX*” is the first input which is an “*m*×*p*” matrix containing the first set of test data where *m* is the number of test samples, the second input is “*testY*”, an “*m*×*q*” matrix containing the second set of test data.

This implementation gets the train and test data matrices from two modalities X & Y and consolidates them into a single feature set Z and our proposed approach illustrated in the Results section.

IV. Results

The key objective in multimodal data is to enhance the performance of any content-based image retrieval system. To examine clearly the effect of joining two modalities (Visual and Textual), simulations were performed considering visual data alone and another simulation considering the two modalities: Visual and Textual. Our dataset (Training images) composed of 4 different categories: nature, flowers, buses and people, which were tested with several query images. All the images are stored in JPEG format.



Content Based Image Retrieval for visual Data

C:\Users\User\Desktop\TrainingVi

C:\Users\User\Desktop\Query2-2

Fig. 3-Result of Visual CBIR system

Precision and Recall shown in **(2)** and **(3)** are used to evaluate the CBIR systems. They are basic measures used in evaluating search strategies. Precision is the fraction of retrieved instances that are relevant. Whilst Recall is the fraction of relevant instances that are retrieved. In the context of Information Retrieval, the precision-recall curve becomes very useful.

$$\text{Precision} = \frac{\text{Number of relevant images retrieved}}{\text{Number of images retrieved}} \times 100 \qquad \textbf{(2)}$$

$$\text{Recall} = \frac{\text{Number of relevant images retrieved}}{\text{Number of relevant images in Dataset}} \times 100 \qquad \textbf{(3)}$$

$$MAP = \left(\sum_{q=1}^{0} AvgP(q) \right) \Big/ Q \qquad \textbf{(4)}$$

Another measure shown in **(4)**, is the Mean Average Precision (MAP) for a set of queries which is the mean of the average precision scores for each query. Precision-Recall graphs give more granular detail on how the system is performing.

An example of the retrieval for the visual system for the "Flower" query is illustrated in Fig. 3, where it can easily be seen that the retrieved results are not all accurate; from 10 images we got only 3 relevant images and the rest 7 are not relevant.

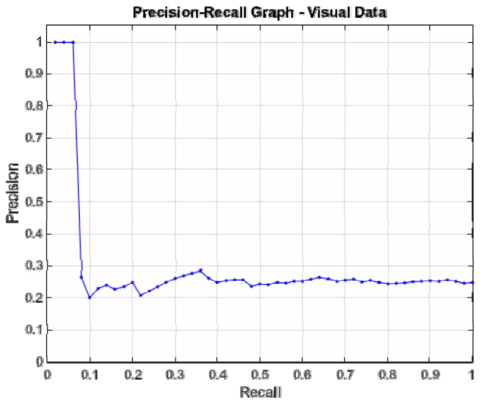
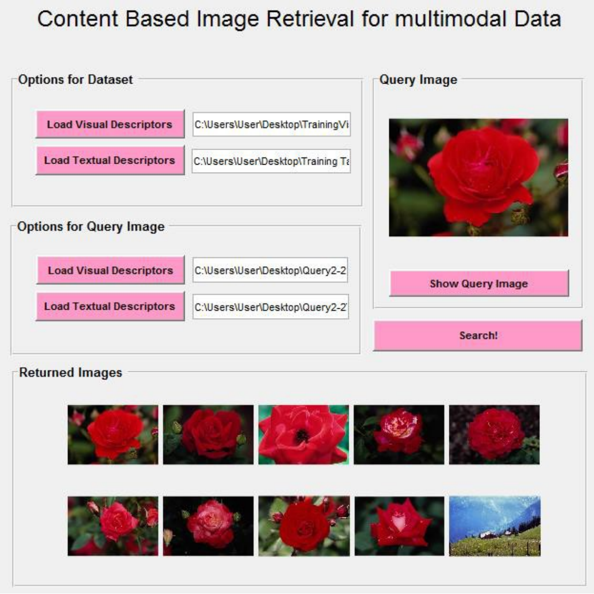


Fig. 4- Precision-Recall of visual CBIR system

We can notice obviously that the Precision value between a Recall ratios of 0.01 to 0.06 is high which is equal to 1. This is because the system is retrieving relevant images for the first three retrievals. This precision decreased to achieve only 0.25 since the system has retrieved an irrelevant image, and it stays low for the last retrieval.

Whilst when applying image retrieval for the "Flower" query in Multimodality using CCA as shown in Fig. 5, from 10 images we got only 9 relevant images and only one irrelevant retrieved image which is an example of our proposed approach that was applied to the whole dataset (Training data) and emanate pretty good results as shown in Fig. 6.


Content Based Image Retrieval for multimodal Data

Options for Dataset

Load Visual Descriptors: C:\Users\User\Desktop\TrainingVi

Load Textual Descriptors: C:\Users\User\Desktop\Training Te

Options for Query Image

Load Visual Descriptors: C:\Users\User\Desktop\Query2-2

Load Textual Descriptors: C:\Users\User\Desktop\Query2-2

Fig. 5- Result for Multimodal CBIR system

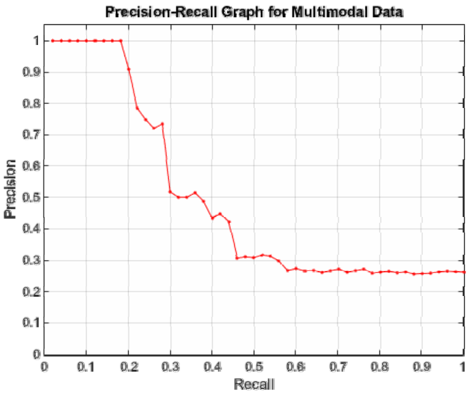


Fig. 6- Precision-Recall of Multimodal CBIR system

The Precision-Recall graph for "Flower" query shown in Fig. 6 present the retrieval of a successive chain of images without touching any of the irrelevant ones for the first 9 retrievals where the Recall ratios are between 0.01 and 0.19, and so the precision will naturally be high. But it will decrease at Recall 0.2 to a value equal to 0.91 when retrieving a non-relevant image. The precision all over the time of retrieval will be affected by the irrelevant results and by that it will still decrease to attempt 0.28 at Recall ratio between 0.58 and 1.

We have managed to retrieve a successive chain of images without changing any of the irrelevant ones for the first 9 retrievals where the Recall ratios are between 0.01 and 0.19, and so the precision will naturally be high. But the precision will decrease, when the Recall becomes 0.2, to a value equal to 0.91 when retrieving a non-relevant image. The precision all over the time of retrieval will be affected by the irrelevant results and by that it will still decrease to attempt 0.28 where the Recall ratio is between 0.58 and 1.

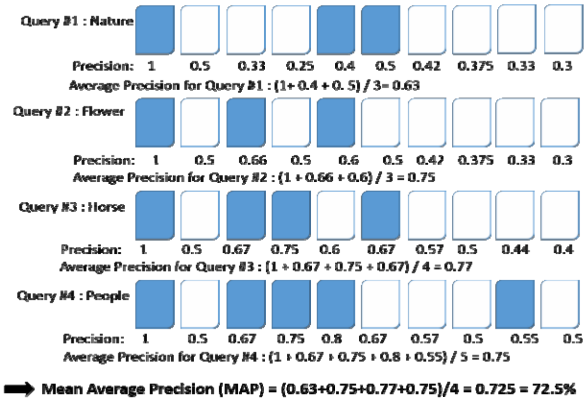

Query #1 : Nature
Precision: 1 0.5 0.33 0.25 0.4 0.5 0.42 0.375 0.33 0.3
Average Precision for Query #1 : (1+ 0.4 + 0. 5) / 3= 0.63
Query #2 : Flower
Precision: 1 0.5 0.66 0.5 0.6 0.5 0.42 0.375 0.33 0.3
Average Precision for Query #2 : (1 + 0.66 + 0.6) / 3 = 0.75
Query #3 : Horse
Precision: 1 0.5 0.67 0.75 0.6 0.67 0.57 0.5 0.44 0.4
Average Precision for Query #3 : (1 + 0.67 + 0.75 + 0.67) / 4 = 0.77
Query #4 : People
Precision: 1 0.5 0.67 0.75 0.8 0.67 0.57 0.5 0.55 0.5
Average Precision for Query #4 : (1 + 0.67 + 0.75 + 0.8 + 0.55) / 5 = 0.75
Mean Average Precision (MAP) = (0.63+0.75+0.77+0.75)/4 = 0.725 = 72.5%

Fig. 7- Mean Average Precision of Visual CBIR system

The mean average precision parameter of the system is calculated for both systems as shown Fig. 8, where the mean average precision which is used to evaluate the efficiency of the system is only 72.6% for the visual CBIR system whereas it is 88.5% for the multimodal CBIR system. Therefore the multimodal CBIR system, which is our proposed approach, present a pretty better efficiency that the CBIR system which may help in reducing time and error while applying image retrieval by decreasing the noise or unrelated retrieved images.

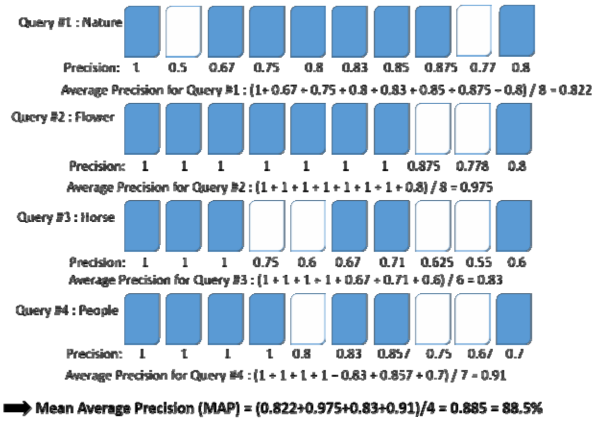

Query #1 : Nature
Precision: 1 0.5 0.67 0.75 0.8 0.83 0.85 0.875 0.77 0.8
Average Precision for Query #1 : (1+ 0.67 + 0.75 + 0.8 + 0.83 + 0.85 + 0.875 + 0.8) / 8 = 0.822
Query #2 : Flower
Precision: 1 1 1 1 1 1 1 0.875 0.778 0.8
Average Precision for Query #2 : (1 + 1 + 1 + 1 + 1 + 1 + 1 + 0.8) / 8 = 0.975
Query #3 : Horse
Precision: 1 1 1 0.75 0.6 0.67 0.71 0.625 0.55 0.6
Average Precision for Query #3 : (1 + 1 + 1 + 1 + 0.67 + 0.71 + 0.6) / 6 = 0.83
Query #4 : People
Precision: 1 1 1 1 0.8 0.83 0.857 0.75 0.67 0.7
Average Precision for Query #4 : (1 + 1 + 1 + 1 + 0.83 + 0.857 + 0.7) / 7 = 0.91
Mean Average Precision (MAP) = (0.822+0.975+0.83+0.91)/4 = 0.885 = 88.5%

Fig. 8- Mean Average Precision of Multimodal CBIR System

V. Conclusion

The use of multimodality by fusion of visual and textual data became a must in order to improve the performance and efficiency of a content-based image retrieval system. It is an efficient way which makes the accuracy and the precision of the retrieved results very high compared with any system that considers only one modality. Canonical correlation analysis CCA implementation for fusion purpose has demonstrated the high performance of the retrieved results. As for performance measures, the precision-recall graphs were very efficient proofs to confirm the importance and significance of multimodal data.

VI. References

[1] Y. L. . D. Z. . G. L. and W.-Y. M. , "A survey of content-based image retrieval with high-level semantics," Pattern Recognition, vol. 40, no. 1, p. 262–282, January 2007.

[2] S. N. and N. R. , "A NEW CONTENT BASED IMAGE RETRIEVAL SYSTEM USING GMM AND RELEVANCE FEEDBACK," Journal of Computer Science, vol. 10, no. 2, pp. 330-340, 2014.

[3] "Content- based Image Retrieval Approach using Three Features Color, Texture and Shape," International Journal of Computer Applications , vol. 97, p. 0975 – 8887, July 2014.

[4] R. F. L. F.-F. P. P. and A. Z. , "Learning Object Categories from Google's Image Search," in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, 17-21 Oct. 2005.

[5] G. W. and A. Y. , "Content Based Image Retrieval Using Enhanced Vocabulary," International Journal of Science and Research (IJSR), vol. 5, no. 5, pp. 2319-7064, May 2016.

[6] A. V. K.N. Plataniotis, "Color Image Processing and Applications," 2000.

[7] . P. K. V. and A. N. , Color Image Processing and Applications, Springer-Verlag Berlin Heidelberg, 2000.

[8] Y. S. . W. W. and A. Z. , "Automatic Annotation and Retrieval of Images," IFIP — The International Federation for Information Processing, vol. 88, pp. 267-280, 2002.

[9] V. M. I. K. and M. S. , "An ontology approach to object-based image retrieval," in Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, 14-17 Sept. 2003.

[10] M. LEW, N. SEBE, C. DJERABA and R. JAIN, "Content-Based Multimedia Information Retrieval:State of the Art and Challenges," ACM Transactions on Multimedia Computing, Communications and Applications, vol. 2, no. 1, pp. 1-19, February 2006.

[11] M. J. Huiskes and M. S. Lew, "The MIR Flickr Retrieval Evaluation," LIACS Media Lab, Leiden University.

[12] C. J. W. M. R. Z. Y. Z. and X. X. , "Cross-Modal Image Clustering via Canonical Correlation Analysis," in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[13] L. K. "Translating images to keywords: problems, applications and progress," in MIS'05 Proceedings of the 11th international conference on Advances in Multimedia Information Systems, Sorrento, Italy, September 19 - 21, 2005.

[14] I. K. S. . I. L. C. and D. S. , "Mining association rules between low-level image features and high-level concepts," in Data Mining and Knowledge Discovery: Theory, Tools, and Technology III, Orlando, April 16, 2001.

[15] J. C. Caicedo , "Multimodal information spaces for content-based image retrieval," in FDIA'09 Proceedings of the Third BCS-IRSG conference on Future Directions in Information Access, Padua, Italy, September 01 - 01, 2009.

[16] "Bag of words - TF-IDF - Deeplearning4j," Skymind. DL4J is licensed Apache 2.0., 2016. [Online]. Available: https://deeplearning4j.org/bagofwords-tf-idf.html#bag-of-words--tf-idf.

[17] . A. Pradeep K., M. A. Hossain, A. El Saddik and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," Springer-Verlag, 4 April 2010, p. 345–379.

[18] B. T. E. M. B. and M. S. L. , "TOP-SURF: a visual words toolkit," in Proceedings of the 18th ACM International Conference on Multimedia, Firenze, Italy, 2010.

International Journal of New Computer Architectures and their Applications (IJNCAA) 8(3): 125-128
The Society of Digital Information and Wireless Communications, 2018 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

A New Visual Vocabulary for Object Recognition

Ismail A. El Sayad, Mohammad A. Bazzoun, Ola S. Shamieh, Samah B. El-Kashoua, Samih Abdulnabi
Department of Computer and Communication Engineering
Lebanese International University, Beirut, Lebanon
Ismail.sayad@liu.edu.lb ; mohammad.bazzoun@liu.edu.lb ; ola.shamieh92@gmail.com
samah_kashoua@hotmail.com ; samih.abdulnabi@liu.edu.lb

Abstract— **Images can be represented at different levels, many approaches in the domain of image representation were developed to move from low-level to high-level image representation like Collocation patterns (moving from Visual Words to Visual Phrases), Descriptive Visual Words and Phrases (DVWs & DVPs) for Image applications and Recognition Using Visual Phrases. Moreover, many relevant articles have tackled the problem of image representation in the case of image retrieval. These articles showed advantages and disadvantages of the used methods in addressing the problem of image representation. The main purpose of this paper is to tackle these drawbacks based on the performance of image representation. What lacks in other papers is the semantic learning and not considering the spatial location of the region. In this paper, a development is proposed for a semantic coherent visual word pattern representation that considers three main points, the neighbor among visual words, frequent item set among these visual words, and the semantic learning.**

Keywords—BOW; Image Representation; DVW; DVP; Image Processing; Image Retrieval; Visual Words

I. Introduction

There are two structures in representing an image, text-based and content based. Representing images as visual vectors, composed of distinctive and repeatable visual elements that can be compared to text words, is considered as a basic idea in the field of image representation. However, visual words may not express or illustrate the concept of an image as text words can.

In order to improve the performance of CBIR systems, there was a shift in the research focus, from designing algorithms to reduce the 'semantic gap' between the richness of human semantics and the visual features [6]. With such a representation, lots of mature techniques in information retrieval can be leveraged for vision tasks, such as visual search or recognition [4].

Many steps have been implemented toward reducing the semantic gap. Different methods were presented in our research based on different algorithms which depend on visual phrases (probability distribution, decoding, and generation of descriptive visual phrase), where the visual phrase is a collection of visual words that makes the image nearest to its semantic meaning. Besides, these algorithms have advantages and disadvantages, these disadvantages result in inefficient image retrieval, for this reason, we are going to introduce a new algorithm for a high-level representation of the image.

Object recognition in images is a topical theme that gives rise to multiple applications like automatic annotation of multimedia content, image search (e.g. Google Images), automatic classification, mobile geolocation recognition of landscape and buildings, etc.

Now a common approach for the computer representation of the visual content is to consider the images as assemblies of smaller areas (visual words), whose membership forms the entire picture. The set of visual words, which is the vocabulary, is generated by a step of quantizing the feature space. The group of visual words forms Bag-of-Visual-Words (BOW) [8]. Thus, all visually similar parts are denoted by the same word. Another approach is detecting components like "person" and "horse" independently, and then describing the relation [2]. But the main weakness of this approach is that the appearance of the objects changes after adding relations.

BOW may contain limited information which may result in an ineffective way for representing specific objects and scenes in an image due to ignoring the spatial relationship between visual words, which are very important in image representation. On the other hand, a polysemous visual word may mean different things under different contexts [3]. To resolve this ambiguity, we may tend to put them under one spatial context, known as collocation.

The previous clustering based visual word generation (k-means), may lead the cluttered background to contain non-descriptive and unnecessary visual words which result in an ineffective and compact visual word sets.

The BOW model has not been extensively tested for large image dataset, thus its performance is unclear. Performance gets better as a number of visual words increases until a certain level, where the performance will become saturated.

In this paper, we aim to find a high-level representation of the image based on Bag-of-Visual-Words that has a semantic meaning of the image closer to user's expectation.

The plan carried to achieve our paper's objective is shown in Fig. 1 as follows:

1. Extracting low-level features from an image.
2. Represent the image as a bag-of-visual-words.
3. Search for the optimal available visual vocabulary by discarding the unnecessary visual words.
4. Based on the selected visual words (after filtration), we generate a high-level representation of the image.
5. Apply a sliding window on the image to generate the transaction database.
6. Apply Apriori algorithm to generate the frequent item set.

The paper is organized as follows: Section II presents the low-level image representation. Section III describes the semantic coherent visual word representation. Section IV shows and interprets the result of the proposed approach. Finally, our Section VI concludes the paper.

II. LOW-LEVEL IMAGE REPRESENTATION

We notice the rapid increase in the size of digital images collection with the prompt evolution of Internet and cameras. For this purpose, researchers have developed many image retrieval systems. Two frameworks are considered: text-based and content-based.

In text-based image retrieval (TBIR) approach, images are manually annotated by text descriptors, which arise two disadvantages from this approach: the intervention of human labor; and the subjectivity of human perception.

Content-based image retrieval (CBIR) was introduced to overcome text-based image retrieval drawbacks. In CBIR, images are indexed by their visual content such as color, texture, shapes.

The main difference between CBIR and TBIR systems is that the human interaction is an important part of the latter system. Where in CBIR, the features are extracted using computer vision techniques (color, texture, shape, spatial layout, etc.). CBIR performance is still far from user's expectations due to the presence of the 'semantic gap'.

III. REDUCING THE SEMANTIC GAP: SEMANTIC COHERENT VISUAL WORD REPRESENTATION

A. BOW Representation:

It is a kind of representation that represents only patches of an image after dividing it into regions. In this section, we will briefly review one of the approaches, bag of visual words (BOW).

In BOW approach, images are composed of local patches that are rich in information from the local interest points of the image which reveals pretty good results caused by position changes, scaling and translation.

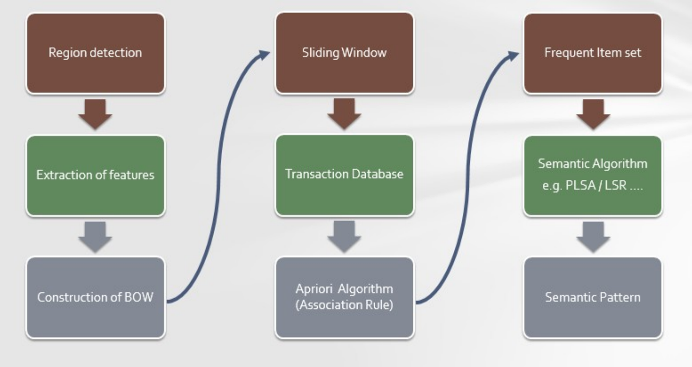


Fig. 1- Flow chart diagram that shows the main steps of SCVWP representation.

After the extraction of features from images, images can be a filename or RGB pixel array. We can display the locations of detected visual words.

B. Transaction Database:

After extracting BOW from images, we represent them as transaction database of visual words. The generation of this database is by using a sliding window, with a defined size, on each. The set of visual words in the window is a transaction row in the transaction database.

C. Finding Frequent Itemset:

After constructing BOW and presenting images as transaction database. In this section, we want to apply Apriori algorithm which is an association rule that gives the frequent visual words that occur with each other.

$$s(\mathrm{X} \rightarrow \mathrm{Y}) = \frac{\text{Number of transactions containing A\&B}}{\text{Number of total transaction}} \quad \textbf{(1)}$$

$$c(\mathrm{X} \rightarrow \mathrm{Y}) = \frac{\text{Number of transactions containing A\&B}}{\text{Number of transactions containing A}} \quad \textbf{(2)}$$

In this step, we obtain a large number of a visual pattern where each one is a set of visual words that are neighbors and frequently occur with each other. There is two kind of threshold that the number of the frequent patterns depends on, support and confidence threshold. Support determines how often a pattern is applicable to a given data set as shown in (1), while confidence determines how frequently items in Y appear in transactions that contain X as shown in (2).

After constructing the frequent item set results are saved in a text file to use it later in constructing SCVWP (Semantic Coherent Visual Word Patterns).

D. Learning the Semantic Significance of Set of Visual Words:

The TF-IDF [9] of the visual words extracted by the TOP SURF [10] will be used as the input of PLSA (Probabilistic Latent Semantic Analysis) algorithm to extract visual words that has high semantic importance in the images by using probability estimation according to latent variables (semantics or concepts that are hidden).

If the visual word has a low probability of latent variable which is below a certain threshold, PLSA will ignore it. PLSA will only preserve visual words that have a high probability to latent variables.

The output of PLSA Algorithm is saved in a text file for later use in SCVWP dictionary generation. Generation of SCVWP Dictionary that respects the main three characteristics neighbor, frequent and semantic to the same latent variable is the last step.

First, the PLSA output is filtered by removing all visual words that are under a predefined threshold. Second, in each transaction in a transaction database, visual word patterns that are related to the same latent variable above the threshold and that are neighbors and frequently occur with each other are preserved as an SCVWP in the new dictionary.

IV. RESULTS

To perform our methodology, we handle two different experiments using a predefined dictionary. This dictionary is built by using sixty images as training set and uses 5 images from 6 categories (cats, dogs, house, kid…) to obtain thirty images as testing set

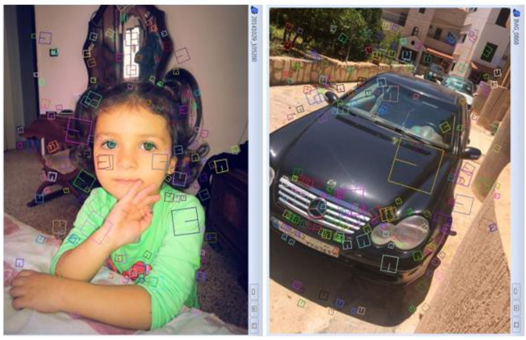
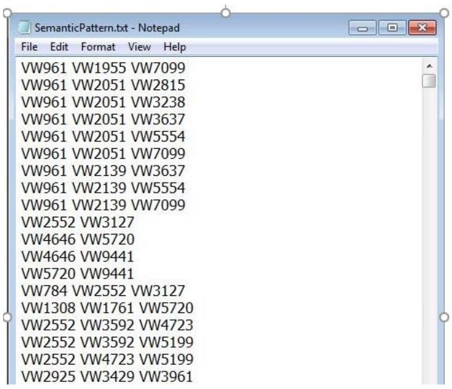


Fig. 2- Extracted Interest Points

First, interest points will be detected from each image, after that we apply the clustering (K-means) to cluster similar features into one group known as a visual word. Then the image is represented as a bag of the visual word.

To compare the result of a bag of the visual word, we will take any two images from the test and apply normalized cosine similarity or absolute distance after loading the dictionary to the top surf.

Normalized cosine similarity range between 0 and 1. If the value is near to zero then the two images are approximately similar to each other, and if the value is far from zero and near to one then the two images are not similar.

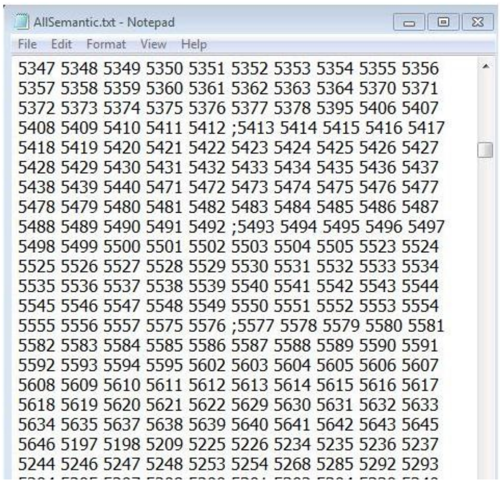


```
VW961 VW1955 VW7099
VW961 VW2051 VW2815
VW961 VW2051 VW3238
VW961 VW2051 VW3637
VW961 VW2051 VW5554
VW961 VW2051 VW7099
VW961 VW2139 VW3637
VW961 VW2139 VW5554
VW961 VW2139 VW7099
VW2552 VW3127
VW4646 VW5720
VW4646 VW9441
VW5720 VW9441
VW784 VW2552 VW3127
VW1308 VW1761 VW5720
VW2552 VW3592 VW4723
VW2552 VW3592 VW5199
VW2552 VW4723 VW5199
VW2925 VW3429 VW3961
```

Fig. 3- Semantic Pattern

In the case of absolute distance, the value represents the real distance between descriptors. When the distance is small, the images are similar to each other and when distance is large, then the images are not similar. With respect to the second experiment, we need to construct the new dictionary denoted by SCVWP (semantic coherent visual word pattern). This dictionary is characterized by spatial, frequent and semantic characteristics. There are several steps to construct this dictionary. First of all, considering the spatial position of the visual word which are the neighbor ones and this is done by moving a window across visual words of a specific width (0.15 of the scale of the image). A group of visual words that are located near enough to be included inside the window is considered as a transaction in the transaction database. As it is known, the size of transaction database increases with the increase of extracted neighbor visual words.

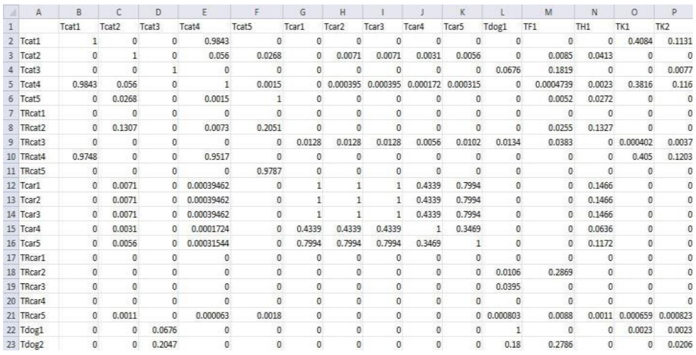
Now, this transaction database will be an input to Apriori algorithm that extracts the frequent item sets based on specific attributes required. The number of frequent words in every item set depends on the certain threshold (support and confidence thresholds). Frequent item set is saved into a text file.



```
5347 5348 5349 5350 5351 5352 5353 5354 5355 5356
5357 5358 5359 5360 5361 5362 5363 5364 5370 5371
5372 5373 5374 5375 5376 5377 5378 5395 5406 5407
5408 5409 5410 5411 5412 ;5413 5414 5415 5416 5417
5418 5419 5420 5421 5422 5423 5424 5425 5426 5427
5428 5429 5430 5431 5432 5433 5434 5435 5436 5437
5438 5439 5440 5471 5472 5473 5474 5475 5476 5477
5478 5479 5480 5481 5482 5483 5484 5485 5486 5487
5488 5489 5490 5491 5492 ;5493 5494 5495 5496 5497
5498 5499 5500 5501 5502 5503 5504 5505 5523 5524
5525 5526 5527 5528 5529 5530 5531 5532 5533 5534
5535 5536 5537 5538 5539 5540 5541 5542 5543 5544
5545 5546 5547 5548 5549 5550 5551 5552 5553 5554
5555 5556 5557 5575 5576 ;5577 5578 5579 5580 5581
5582 5583 5584 5585 5586 5587 5588 5589 5590 5591
5592 5593 5594 5595 5602 5603 5604 5605 5606 5607
5608 5609 5610 5611 5612 5613 5614 5615 5616 5617
5618 5619 5620 5621 5622 5629 5630 5631 5632 5633
5634 5635 5637 5638 5639 5640 5641 5642 5643 5645
5646 5197 5198 5209 5225 5226 5234 5235 5236 5237
5244 5246 5247 5248 5253 5254 5268 5285 5292 5293
```

Fig. 4- Semantic Indices

After that, we will extract the TF/IDF of all visual words from the whole training image, and save them into an array. Then apply PLSA (probabilistic latent semantic algorithm) with three latent variables defined. The result will be three arrays that will be saved into text file that is used in determine the semantic visual word pattern.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Tcat1	Tcat2	Tcat3	Tcat4	Tcat5	Tcar1	Tcar2	Tcar3	Tcar4	Tcar5	Tdog1	TF1	TH1	TK1	TK2
2	Tcat1	1	0	0	0.9843	0	0	0	0	0	0	0	0	0	0.4084	0.1131
3	Tcat2	0	1	0	0.056	0.0268	0	0.0071	0.0071	0.0031	0.0056	0	0.0085	0.0413	0	0
4	Tcat3	0	0	1	0	0	0	0	0	0	0	0.0676	0.1819	0	0	0.0077
5	Tcat4	0.9843	0.056	0	1	0.0015	0	0.000395	0.000395	0.000172	0.000315	0	0.0004739	0.0023	0.3816	0.116
6	Tcat5	0	0.0268	0	0.0015	1	0	0	0	0	0	0	0.0052	0.0272	0	0
7	TRcat1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	TRcat2	0	0.1307	0	0.0073	0.2051	0	0	0	0	0	0	0.0255	0.1327	0	0
9	TRcat3	0	0	0	0	0	0.0128	0.0128	0.0128	0.0056	0.0102	0.0134	0.0383	0	0.000402	0.0037
10	TRcat4	0.9748	0	0	0.9517	0	0	0	0	0	0	0	0	0	0.405	0.1203
11	TRcat5	0	0	0	0	0.9787	0	0	0	0	0	0	0	0	0	0
12	Tcar1	0	0.0071	0	0.00039462	0	1	1	1	0.4339	0.7994	0	0	0.1466	0	0
13	Tcar2	0	0.0071	0	0.00039462	0	1	1	1	0.4339	0.7994	0	0	0.1466	0	0
14	Tcar3	0	0.0071	0	0.00039462	0	1	1	1	0.4339	0.7994	0	0	0.1466	0	0
15	Tcar4	0	0.0031	0	0.0001724	0	0.4339	0.4339	0.4339	1	0.3469	0	0	0.0636	0	0
16	Tcar5	0	0.0056	0	0.00031544	0	0.7994	0.7994	0.7994	0.3469	1	0	0	0.1172	0	0
17	TRcar1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	TRcar2	0	0	0	0	0	0	0	0	0	0	0.0106	0.2869	0	0	0
19	TRcar3	0	0	0	0	0	0	0	0	0	0	0.0395	0	0	0	0
20	TRcar4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	TRcar5	0	0.0011	0	0.000063	0.0018	0	0	0	0	0	0.000803	0.0088	0.0011	0.000659	0.000823
22	Tdog1	0	0	0.0676	0	0	0	0	0	0	0	1	0	0	0.0023	0.0023
23	Tdog2	0	0	0.2047	0	0	0	0	0	0	0	0.18	0.2786	0	0	0.0206

Fig. 5- Normalized cosine similarity among every pair of images

We put a threshold, approximately the average of all values obtained from PLSA, for accepting the visual word from PLSA output to 0.002567. Now, we will search for frequent item set according to the output of PLSA for the visual word that has a frequency less or equal to 0.002567 and accepts it. After that, the new dictionary (Sematic Coherent Visual Word Pattern) is developed consisting of indexes and their assigned patterns.

Now, we have to find the semantic representation as indices for each image in the training and save them into a file that will be known as semantic indices for all the images.

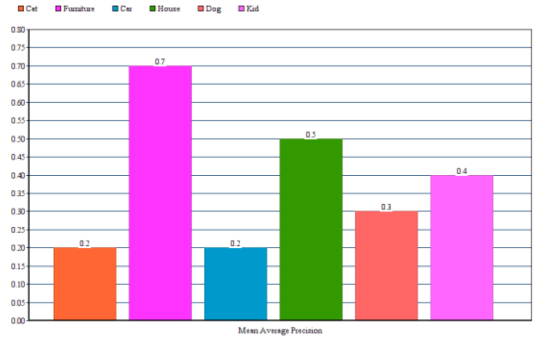


Fig. 6- Mean Average precision of BOW

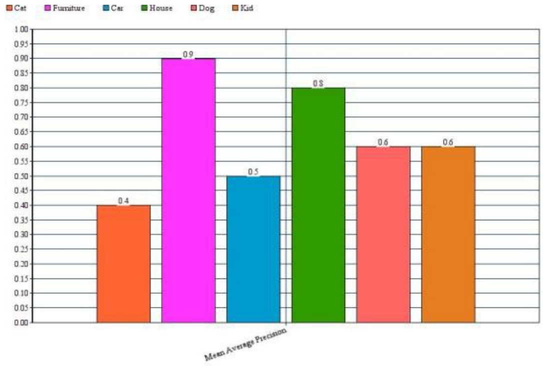


Fig. 7- Mean Average precision of SCVWP

After that, we have to find the TF-IDF for each image that will be represented as a vector. As we see, each image is represented as vector of TF-IDF. Now, we have to apply normalized cosine similarity between every pair of images.

We will calculate the MAP (Mean Average Precision) to test the performance of image representation. It is obtained after the calculation of AP (Average Precision).

By comparing the result of first experiment with this one, we can see that in certain categories, the value of AP is better than the first one. So, the performance is better when using the new representation (SCVWP).

CONCLUSION

Images can be represented by their low level features or by higher level by representing patches which is a group of pixels. Many researches in the domain of image representation have been applied to map low level to higher levels by reducing semantic gaps. Based on the results and comparison obtained we can conclude that higher level can improve the performance of image representation. These results motivate us to overcome these drawbacks in our methodology that aim to reduce the semantic gap between machine learning and human perception. Semantic Coherent Visual Word Pattern SCVWP is our new higher level image representation that contains patterns of visual word respecting three main characteristics: neighbor, frequent and semantic.

V. REFERENCES

[1] L. D. A. M. M. A. B. M. B. L.-M. E. G. and B. M. C. , "Combining local and global image features for object class recognition.," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops., 2005 Jun 25.

[2] S. M. A. and F. A. , "Recognition using visual phrases," in In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011.

[3] Y. J. Y. W. and M. Y. , "Discovery of collocation patterns: from visual words to visual phrases," in Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007.

[4] S. Z. Q. T. . G. H. Q. H. and . S. L. , "Descriptive Visual Words and Visual Phrases for Image Applications," in Proceedings of the 17th ACM international conference on Multimedia. ACM, 2009.

[5] A. R. and R. S. , "Fast algorithms for mining association rules.," in Proc. 20th int. conf. very large data bases, VLDB., 1994.

[6] Y. L. . D. Z. . G. L. and W.-Y. M. , "A survey of content-based image retrieval with high-level semantics," Pattern Recognition, vol. 40, no. 1, p. 262–282, January 2007.

[7] I. K. S. . I. L. C. and D. S. , "Mining association rules between low-level image features and high-level concepts," in Data Mining and Knowledge Discovery: Theory, Tools, and Technology III, Orlando, April 16, 2001.

[8] L. K. "Translating images to keywords: problems, applications and progress," in MIS'05 Proceedings of the 11th international conference on Advances in Multimedia Information Systems, Sorrento, Italy, September 19 - 21, 2005.

[9] "Bag of words - TF-IDF - Deeplearning4j," Skymind. DL4J is licensed Apache 2.0., 2016. [Online]. Available: https://deeplearning4j.org/bagofwords-tf-idf.html#bag-of-words--tf-idf.

[10] B. T. E. M. B. and M. S. L. , "TOP-SURF: a visual words toolkit," in Proceedings of the 18th ACM International Conference on Multimedia, Firenze, Italy, 2010.

International Journal of New Computer Architectures and their Applications (IJNCAA) 8(3): 129-141
The Society of Digital Information and Wireless Communications, 2018 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

An Enhanced Algorithm for Cloud Scheduling with Setup Cost

Redwan Al-Dilami[1], Adnan Zain Al-Saqqaf[2], Ammar Thabit Zahary[3]

[1]Lecturer. Faculty of Computing and IT, University of Science and Technology, Sana'a, Yemen
[2] Associate Prof. Faculty of Engineering, Aden University, Aden, Yemen
[3]Assistant Prof. Faculty of Computer and IT, Sana’a University, Sana'a, Yemen
[1] r.aldilami@ust.edu, [2]adnan_zain2003@yahoo.com, [3]zahary@su.edu.ye

ABSTRACT

The study aimed at investigating the problem of online task scheduling of identifying job of MapReduce on cloud computing infrastructures. It was proposed that the virtualized cloud computing setup comprised machines that host multiple identical virtual machines (VMs) under pay-as-you-go charging and booting a VM requires a constant setup time. The problem occurs when the VMs turning off after finishing processing a task and running it again for another. The goal is to limit the delay resulted from setting up the VMs, and to minimize the idle cost (VM does not find any task to possess) when VM is continuing in an active state without processing tasks. It was a constant number of VMs in an activation state continuity, and the cost of idle was distributed over existing tasks where the cost should be less than the cost of setting up one VM. The researchers' algorithm limited the delay resulted from setting up the VMs, the cost resulted from the continuing VMs in an active state without processing tasks was distributed to the current tasks fairly. The known cases was discussed where the duration of each task was known upon its arrival.

KEYWORDS

Cloud computing, clairvoyant, scheduling, Known Duration of Tasks scheduling, setup cost.

1. INTRODUCTION

Scheduling is a collection of policies and mechanisms for the sake of controlling and ordering the work of the computer system.Task scheduling is one of the basic topics discussed by many researchers towards solving the problem of scheduling. The researchers introduced a number of techniques for the sake of solving this problem [1].

Reducing the response time for cloud services will increase the confidence of customer that requests cloud services and the great motivation is minimizing the total cost of job processing.

The goal is to limit the delay resulted from setting up the VMs, and the cost resulted from continuing VMs in an active state without processing tasks was distributed to the current tasks fairly. In online scheduling, the scheduler receives jobs that arrive over time, and generally must schedule the jobs without any knowledge of the future. Cloud Computing is a new paradigm for provisioning computing instances, I.e., VMs to execute jobs in an on-demand manner. This paradigm shifts the location of the computing infrastructure from the user site to the network thereby reducing the capital and management costs of hardware and software resources [2]. Public cloud is available in a pay-as-you-go charging model that allows end-users to pay for VMs by the hour e.g., $0.12 per hour. Two-key criteria determine the quality of the provided service: (a) the dollar price paid by the end-user for activating VMs and (b) the maximum delay among all given tasks of a job which occurred by setup the VM to become active. The goal is to provide a scheduling algorithm that aims to minimize the delay and the activation's setup cost of VM.

This paper focuses on arbitrary jobs such as job identified of MapReduce. In classical scheduling problems, the number of machines is fixed, and the sequence we have to decide which job to process on which machine. However, the cloud introduces a different model in which we can activate and release machines on demand, and thus control the number of machines being used to process the jobs. This highlights the tradeoff between the number of machines used and the delay of processing the jobs. On the one hand, if the researchers does not have to pay for each machine, they could use one machine for each task of the job, and reduce the delay to a minimum. On the other hand, if we want to minimize cost, we could only use a single machine for all tasks of the jobs in a work-conserving model.

In this paper, it is assumed that all computing instances available for processing are initially active. When the jobs arrive, they inter to the VM without waiting to activate the VM. Moreover, when a VM is no longer in use, it should not be shut down, a constant time for turning off will not occur. Both setup and shutdown times are not included in the production cost of this service. Therefore, they will not be charged to the end-user. As a result, the number of VMs activated continuously (without shutdown) for a specific job which has a major impact on the total cost.

The goal is to minimize both maximum delay (setup time) and setup cost. The problem lies in finding the right balance between delay and cost.

In this study, a number of VMs are assumed in activation state continuously, so that no need to setup and shutdown VMs. This means the delay will be initially eliminated. However, there will be initially higher costs because many numbers of VMs will be in the activation state when no jobs are in process.

The remaining sections of this paper are organized as follows:

Section 2 presents the related work to setup cost scheduling. System model, system assumption, cost scheduling model, and algorithm of known duration of tasks are presented in Section 3. The results of the study and comparison are presented in Section 4 and 5, and the conclusion is presented in Section 6.

2. RELATED WORK

Due to the significance of task scheduling and being often NP-hard, these kinds of problems have been much studied. The instrument technique was surveys for scheduling algorithms and online scheduling which are found in [3],

[4], [5] and [6]. It could be the most intuitive measure of Quality of Service (QoS) received by an individual job is the flow time. That is, the flow time Fi of the i th job from the difference between its completion time and its release date. However, this measurement is used to the delay attribute in [7]. The main goal in the last reference is to give a scheduling algorithm that aims to minimize the delay and the production cost of executing a job. They assume that all computing instances available for processing are initially inactive to assign a task to any machine; It should be activated first, activation (Set up) Time: T_{setup}. To activate a machine, there is a constant duration of setup until the machine is ready to process, Shut down Time: $T_{shutdown}$. This means that Activation Time $T_s = T_{setup} + T_{shutdown}$, Both setup and shutdown times are involved in the production cost of this service, and therefore they will be charged to the end-user. In the researchers' work, the focus is on the online problem because no information is known on future arrival of tasks, but that the arrivals of tasks are independent of the scheduling. Focusing on the cases below, it is clear that they are known and unknown task duration model (i.e. clairvoyant and non-clairvoyant). In the clairvoyant case the duration of a task is known at its arrival. In the non-clairvoyant model the duration is unknown at its arrival and becomes known only once the task has been completed. The results of the researchers' work (1) the cost ratio of Algorithm Clairvoyant is at most$(1+\epsilon)$; (0< ϵ <1). Moreover a long task will have a delay of T_s. The delay of a short task is at most $2E = \frac{4T_s}{\varepsilon}$. (2) If an online non-clairvoyant algorithm is limited to a cost of $(1+\epsilon)$, then its delay ratio is $\Omega\left(\frac{\log \mu}{\varepsilon}\right)$, where μ is the ratio of the longest task duration of any job to the shortest task duration of any job.

Known Duration of Tasks: It is assumed that the duration of each task p_i is known upon its arrival. Let$E = \frac{2T_s}{\epsilon}$ (for $0 < \epsilon < 1$).

Algorithm Clairvoyant: Classify a task upon its arrival. It is a long task if $p_i \leq E$ or otherwise short.

Once the arrival of each new long task, activate a new machine. That is to say, accumulate short tasks, activate a machine to process those tasks at the earliest between case 1: the first time the volume of the accumulated tasks becomes above E . In this case, assign the tasks to a new allocated machine and restart the accumulation. Case 2: $(E - T_s)$ time passed from the earliest release time of those tasks. In this case, continue the accumulation until the machine is ready (at timeE), assign the tasks to the machine, and then restart the accumulation. If the volume of the tasks exceeds E by the time the machine is ready, stop the accumulation and go to case 1 (these tasks will be classified as case 1). Processing the tasks on their assigned machine according to their arrival order. Note that the volume assigned to a machine is below 2E and each task will start its processing within at most E time after the assigned machine is ready. Shutdown a machine once it completes tasks assigned to it.

Unknown Duration of Tasks: Algorithm Non-Clairvoyant Divide the time into epochs of $\mathrm{F} = \frac{4\mathrm{T_s}}{\varepsilon}$. Let $\mathrm{B_0}$ be the set of tasks given initially

(Time 0) and for $\mathrm{k} \geq 1$, let $\mathrm{B_k} = \{\mathrm{i} | (\mathrm{k}-1) \leq \mathrm{a_i} \leq \mathrm{kF}\}$. All tasks $\mathrm{B_k}$ are handled at time kF separately from tasks of $\mathrm{B_{k'}}$ for $\mathrm{k}' \neq \mathrm{k}$. Let $\mathrm{n_k} = |\mathrm{B_k}|$ is the number of tasks arrived in epoch k. Let $\mathrm{m_k} = \left\lceil \frac{\mathrm{n_k p}}{\mathrm{F}} \right\rceil$ and activate $\mathrm{m_k}$ machines. Processing tasks on machines in arbitrary order for $\mathrm{T_s} + \mathrm{F}$ time (this also includes setup times of newly activated machines). If after $\mathrm{T_s} + \mathrm{F}$ time there are still waiting tasks then activate additional $\mathrm{m_k}$ machines set $\mathrm{m_k} \longleftarrow 2\mathrm{m_k}$ and repeat this step (note that tasks that are already running will continue to run with no interruption). Shutdown a machine once it becomes idle and there are no waiting tasks.

In [8] files are divided into many small blocks and all blocks are replicated over several servers. To process files efficiently, each job is divided into many tasks and each task is allocated to a server to deal with a file block because network bandwidth is a scarce resource.

Activity Based Costing as a Solution: Activity based costing is evaluated separately for every task. It is decided on the basis of resources, space and time taken by every activity of every task.

Activity Based Costing in Cloud Computing:

One way to measure both cost of the object and its performance is the activity based costing. In this view, researchers have solved the problem like poor cost control, distorted product costs and also the starvation. They divide the task into different groups. These groups are (1) Available (dependent & independent): is the group of tasks which can be complete performed on a single data center. (2) Partially available: is the group of tasks which need resources from other data centers. Hence others classify them into cat1, cat2, cat3… and so on till N number of classification. They are done on the basis of data need. Cat1 tasks will need the data from same data centers. Similarly, cat2, cat3….cat N tasks will need the data from same data centers.

Activity Based Costing (Implementation): Researchers deal with algorithm in a tree diagram. one parent queue in which researchers stored the tasks according to their arrival time, tested for data and requested resources by the task and sorted them into two different queues, available and partially available. For available queue, they tested if the task is dependent or independent and according to that researchers stored them in their respective queues. For partially available queue, they sorted tasks in a multiple queues named cat1, cat2… and so on. To decide the priority, researchers considering four major factors i.e. time, space, resources and profit, and they have derived the following formula for deciding the priority of the task: $\sum_{\mathrm{j}=0}^{\mathrm{n}} (\mathrm{T}_{\mathrm{i,j}} + \mathrm{s_{i,j}} + \mathrm{C_{i,j}}) / \mathrm{P_i}$. The notations are explained below:

Ki : Priority of the ith task.

Ti, j: Time required to complete jth activity of ith task.

Si, j: Space needed to operate jth activity of ith task.

Ci, j: cost of jth activity in terms of resources of $i\ th$ task.
Pi : Profit from complete $i\ th$ task.
n : It is the total number of activities of any ith task.

In [9], the goal of this paper is to schedule task groups in cloud computing platform, in that resources have different resource costs and computation performance. However, researchers' algorithm measure both resource cost and computation performance, it also develops the computation/communication ratio by collecting the user tasks based on a particular cloud resource's processing capability and sends the grouped jobs to the resource. They improved Activity Based Costing (ABC)**,** their problems are reduction of makespan and reducing cost. They used CloudSim 1.0b to simulate the algorithm of task scheduling, simulated the algorithm with six nodes, five seconds of granularity time, average MI (Number of Machine Instructions) of tasks 10. They can be seen that for ABC Scheduling the time taken to complete tasks after grouping the tasks is very less when compared with time taken to complete the tasks without grouping the tasks.

In [10], researchers introduce Budget and Deadline Constrained Heuristic based upon Heterogeneous Earliest Finish Time (HEFT) to schedule workflow tasks over the available cloud resources.

In [11], researchers propose a scheduling algorithm which addresses these major challenges of task scheduling such as task completion time or task execution cost etc. The proposed model is implemented and tested on simulation tool kit. Results validate the correctness of the framework and show a significant improvement over sequential scheduling.

In [12], the goal of this study is to use the conventional scheduling concepts to merge them to provide solution for better and more efficient task scheduling which is beneficial to both user and service provider.

In [13], researchers introduce a more efficient algorithm for task scheduling based on Priority Based Scheduling in cloud computing and the implementation of it. Improvement of this algorithm should concentrate on discussing simultaneous instead of independent task scheduling in Cloud environment.

3. SYSTEM MODEL AND ASSUMPTION

3.1 System Assumption

The job input consists of multiple tasks that need to be executed. Tasks arrive over time. A task i has an arrival time a_i and maximum duration p_i, assuming the arrival time is known.

At time t, there is constant number of VMs in activation state. The cost activation is free for the first time, I.e. cost activation for each VM in the second time is with fees. Time activation for each VM is a constant (T_{setup}), the delay is denoted by ($\mathrm{D_{setup}}$) which occurred by T_{setup}, and the cost activation of each VM is constant ($\mathrm{C_{setup}}$). Each VM and its task are homogeneous. Each task runs on a single machine (instance). Each machine can run a single task at a time. Tasks are non-preemptive, i.e., a task has to run continuously without interruptions. Let $e_i = a_i + p_i$ which is the earliest possible completion time of task i . Denoted by c_i which is the actual completion time of task i and $d_i = c_i - e_i$ as the delay that the task delays for some time to find empty VM. Machine can be activated or shut down for the first time without fees otherwise with fees (i.e. activation machine again with fees). Any machine in idle state ($\mathrm{VM_{idle}}$) (i.e. VM does not find any task to possess) should not be shut down. Inactivation of a machine, there is T_{setup} time until the machine is available for processing. In shut down, there is T_{shutdown} time to turn off the machine. For simplicity, it is assumed that there is only activation time Ts = T_{setup} + T_{shutdown} and the shutdown is free. This paper focuses on the known task duration model (duration of a task is known at its arrival).

3.2 Cost Scheduling Model

At any time t, the algorithm needs to decide how much the actual idle time in machines to shut down some of them. In addition, it should decide for each task what the cost idle time to be distributed.

Goal Function:

The goal function consists of two parts: setup cost and delay that occurred by setting up VM. It is assumed that the cost charged per machine per unit of idle time ($\mathrm{T_{idle}}$) is \$ 0.002. Then the actual cost for each VM which is in idle state ($\mathrm{C_{idle}}$) is:

$$\mathrm{C_{idle}} = \sum_{\mathrm{i}} \mathrm{VM_{idle}i} * 0.002 \qquad (1)$$

Through equation No. (1), the total idle cost ($\mathrm{C_{idle}}$) of VMs can be calculated for every task for every moment. Then, that cost is distributed to current available tasks according to our model. When all VMs are decided to be in activation state, an algorithm will be available, its idle time cost should be less than cost of its setup when activation of VM occurs again. Let D be the maximum delay of the task when it waits to activate a new machine (i.e. $d_i <$ D for all tasks i). Formally, the performance of an algorithm will be described by α- the cost ratio of our algorithm ($\mathrm{C_{opt}}$) to the setup cost ($\mathrm{C_{setup}}$):

$$\alpha = \frac{\mathrm{C_{opt}}}{\mathrm{C_{setup}}} \qquad (2)$$

Where $\mathrm{C_{setup}}$ is constant.

Through equation No. (2), the cost average ratio (α) can be calculated for every task via the researchers' algorithm ($\mathrm{C_{opt}}$) comparing cost resulted from activating VM again ($\mathrm{C_{setup}}$).

and δ- the delay ratio:

$$\delta = \frac{\mathrm{D_{opt}}}{\mathrm{D_{setup}}} \qquad (3)$$

where $\mathrm{D_{setup}}$ is constant.

Through equation No. (3), the delay average ratio (δ) can be calculated for every task via the researchers' algorithm (($\mathrm{D_{opt}}$) comparing delay resulted from activating VM again ($\mathrm{D_{setup}}$)

3.3 Algorithm of Known Duration of Tasks

In this section, the researchers' algorithm of known duration of tasks is conducted. The researchers first assume that the duration of each task pi is known upon its arrival. Let $E = 2Ts$. A task is classified upon its arrival. It is a long task if $pi \geq E$, middle task if $Ts \leq pi < E$ and otherwise it will be short.

Step 1

The researchers' determined the No. of VMs which are turning on continually (N_{vm}) during the period [0,t], running on the VMs will be free for the first time.

Step2

A number of tasks of known duration are generated (for each task, certain characteristics, arrival time (T_a), service time (T_s), terminate time (T_t) …etc. Which occur in separated times during the [0, t].

Step3

The tasks are processed in the VMs as follows:

1. If the number of the tasks (N_{task}) in the period [0,t] is equal to the number of the VMs which are in the turning on state; that is $N_{vm} = N_{task}$, each task enters VM will enter and its setup cost will be equal to zero. However, the proper delay for setting up the VM will be zero because the VMs are already active:

$$D_{setup}=0$$

$$C_{setup}=0$$

2. If ($N_{task} > N_{vm}$), then additional VMs (N_{vm-add}) that will be running equal $N_{task} - N_{vm}$ (i.e. $N_{vm-add} = N_{task} - N_{vm}$), and the tasks of the VMs will process as the task of the longest size will be the first.
3. If ($N_{task} < N_{vm}$), there is a definite number of these VMs are idle. In this case, the time of idle state of all VMs should be gathered: $T_{idle} = \sum_i VM_{idle}i$, and the cost of idle (C_{idle}) will be distributed over existing tasks in accordance with the following principles:
   - a- Long task ($pi \geq E$) should cost less than the cost of setting up one VM.
   - b- Middle task ($Ts \leq pi < E$) should cost less than the two-third cost of setting up one VM.
   - c- Short task (0< $pi < Ts$) should cost less than the half cost of setting up one VM.

4- The case (3) continues until the total idle cost of VMs is greater or equal to the total cost of their setup. Hence, some VMs will be shut down in the principle of what gets idle first, should be shutdown first.

Step 4

Any VM gets turned on for the second time (i.e. activating it after being shut down), it should be turned on in line with algorithm clairvoyant in [7]. The following chart summarizes what is mentioned above:

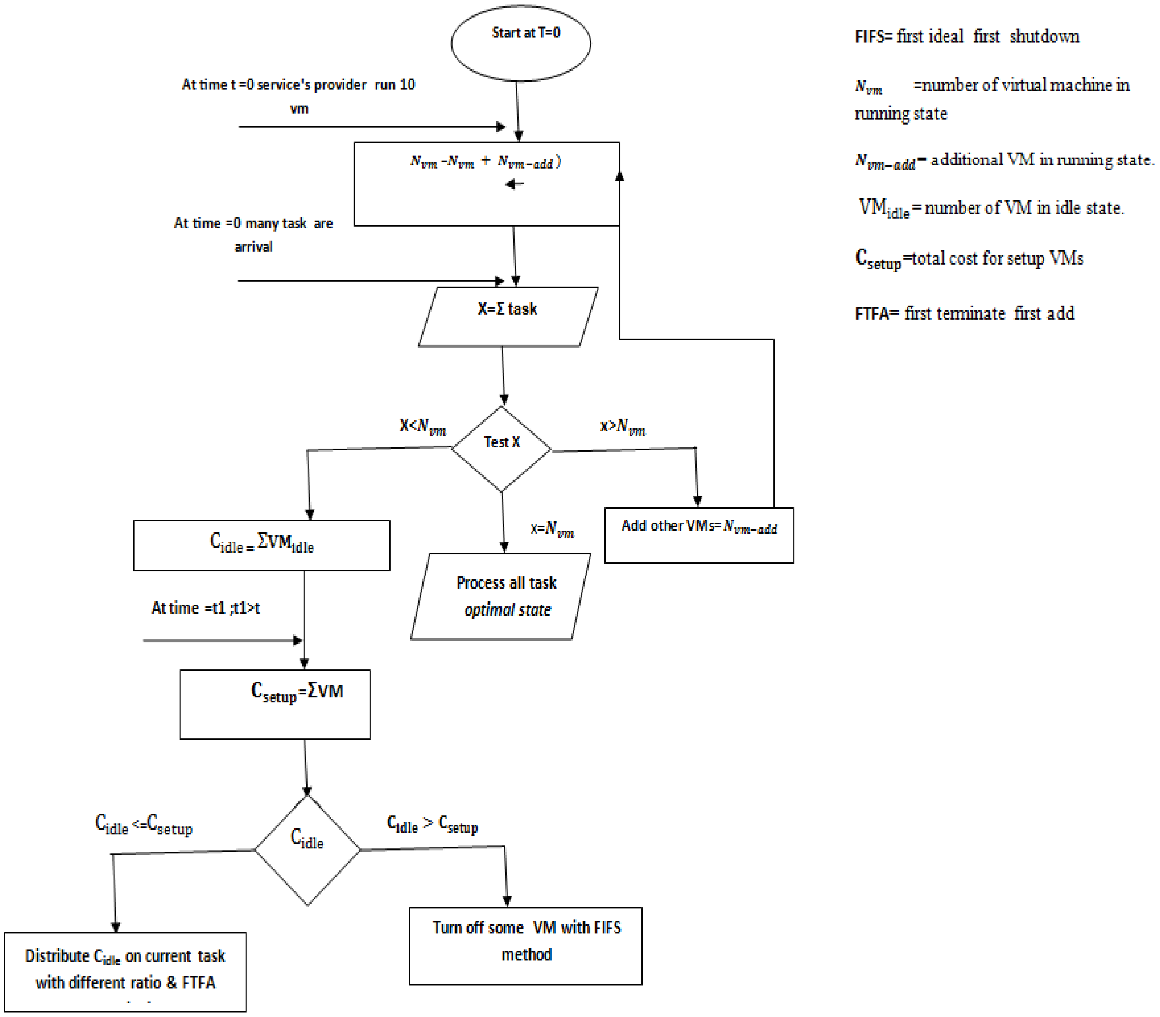

Start at T=0
At time t =0 service's provider run 10 vm
N_{vm} -N_{vm} + N_{vm-add})
At time =0 many task are arrival
X=Σ task
X<N_{vm}
Test X
x>N_{vm}
x=N_{vm}
C_{idle} = ΣVM$_{idle}$
Add other VMs=N_{vm-add}
Process all task optimal state
At time =t1 ;t1>t
C_{setup}=ΣVM
C_{idle} <=C_{setup}
C_{idle}
C_{idle} > C_{setup}
Distribute C_{idle} on current task with different ratio & FTFA
Turn off some VM with FIFS method
FIFS= first ideal first shutdown
N_{vm} =number of virtual machine in running state
N_{vm-add}= additional VM in running state.
VM$_{idle}$= number of VM in idle state.
C_{setup}=total cost for setup VMs
FTFA= first terminate first add

Figure1: Algorithm Diagram Model

4. STUDY RESULTS

The performance of an algorithm will be described by a competitive analysis where α is the cost ratio of the algorithm to the setup cost ($\alpha = \frac{C_{opt}}{C_{setup}}$) and δ ($\delta = \frac{D_{opt}}{D_{setup}}$) is the delay ratio:

1- In optimal case (I.e. $N_{task} = N_{vm}$ through time duration [0, t]) α=0 and the delay ratio δ=0

2- In the second case (when $N_{task} > N_{vm}$ through time duration [0, t]) α=0 for all jobs and additional tasks (cost activation of additional VMs is free in the first time).

Delay ratio:

$$\delta = \begin{cases} 1 & for\ addional\ \text{task} \\ 0 & otherwise \end{cases} \quad (4)$$

3- In third case (when $N_{task} < N_{vm}$ through time duration [0, t]):

$$\begin{cases} 0 \le \alpha \le 1 & for\ longest\ task\ duration. \\ 0 \le \alpha \le \frac{2}{3} & for\ middle\ task. \\ 0 \le \alpha \le \frac{1}{3} & for\ shortest\ task. \end{cases} \quad (5)$$

Delay ratio δ=0 (I.e. no arrival additional tasks to activation additional VMs).

As shown by figure 2, when the number of the tasks is equal to the number of the VMs ($N_{vm} = N_{task}$) through time duration [0,t]. That is, as far as the VM finishes processing a task, another task comes, and the VM will not turn off nor activate again. There is no delay for any neither task nor setup cost because running the VM occurs for the first time. This case is called "optimal". Notice the curves of cost and delay do not appear. In figure 3, when ($N_{task} > N_{vm}$) during the period [0, t], this case is considered rare because the problem occurs when the opposite happens. That is, services providers run a number of VMs for the additional tasks. The delay occurs because it waits for the VM to become ready as being closed earlier. However, these tasks without setup cost because these VMs occur for the first time, as shown in p15, p16, p18, p19 and p20. Notice the curves of VMs, delay appears only in the additional tasks

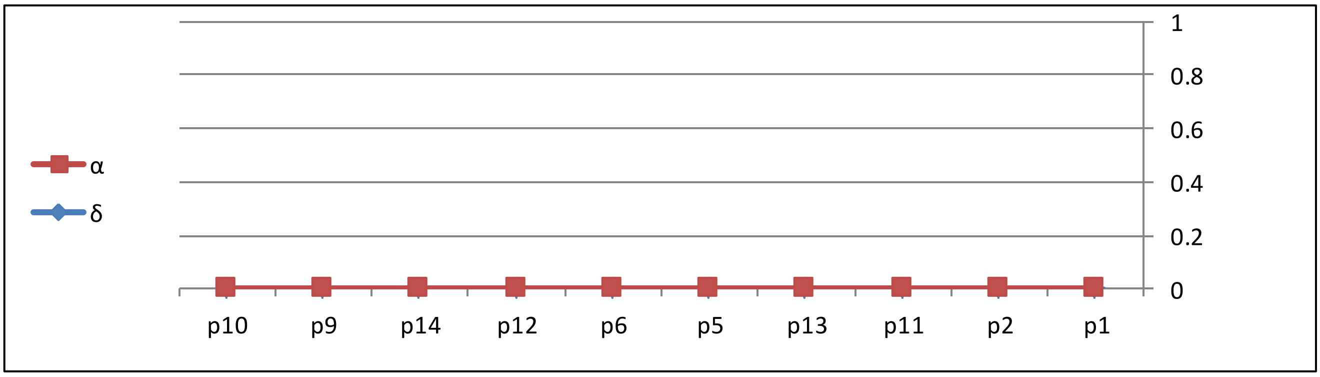


Figure 2: Optimal Case

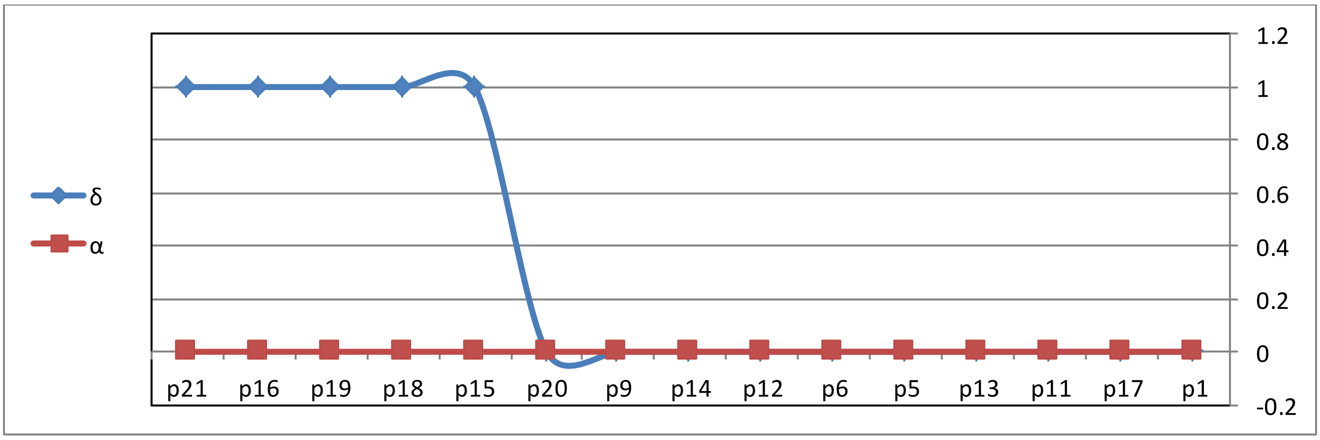


Figure 3: Delay with Additional Tasks

In figure 4, when ($N_{task} < N_{vm}$). (I.e. some VMs are idle), so, instead of turning off such VMs, the researchers should count its remaining cost when they are turning on. This cost gets distributed over all tasks during that period.

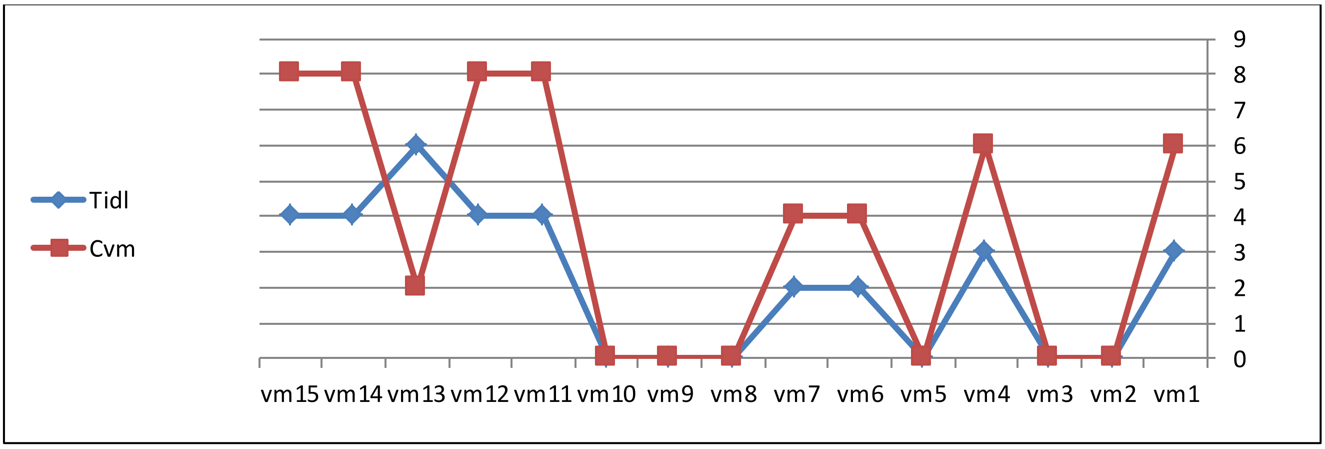


Figure 4: Cost Distribution

In table 1, it is noticed that the unit of time t=10, the total of idle cost VMs from VM1 to VM15 equals to 0.11. Thus, it shut down some VMs. It is noticed that, in the critical case, the long task will render a setup cost of one VM. In case of others, every task will render a setup cost less than that of one single VM such as VM1, VM4, VM5, VM6 etc. until the total of idle cost becomes less than the cost of activating all VMs again. Moreover, in all cases, there is no delay resulted from the time of VM's setup.

Table 1: Shutdown Some VMS

N-vm	N-task	Ta	Ts	VM_{idle}	C_{idle}	δ	α
vm1				5	0.01	0	
vm2	p2	4	8	0	0	0	0.001
vm3	p3	2	8	0	0	0	0.001
vm4				5	0.01	0	
vm5				2	0.004	0	
vm6				4	0.008	0	
vm7				4	0.008	0	
vm8				1	0.002	0	
vm9				2	0.004	0	
vm10	p10	4	6	0	0	0	0.0009
vm11				6	0.012	0	0
vm12				6	0.012	0	0
vm13				8	0.016	0	0
vm14				6	0.012	0	0
vm15				6	0.012	0	0

In all cases, it is supposed that the cost of one single time unit of the idle equals the double cost of setup unit, and the activation cost of VM is counted but the turning off is free.

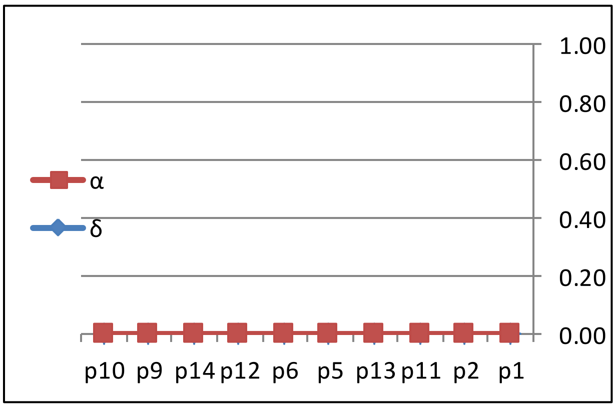
5. COMPARISON OF RESULTS

When our data is uses in others algorithms the δ and α appear as the following :

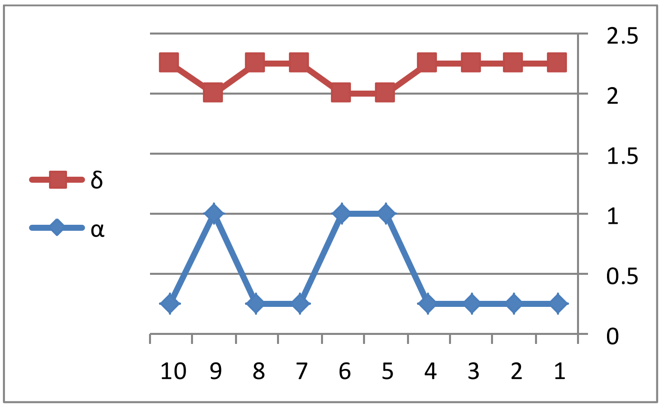
In the researchers' algorithm

In others

Case 1: $N_{task} = N_{vm}$



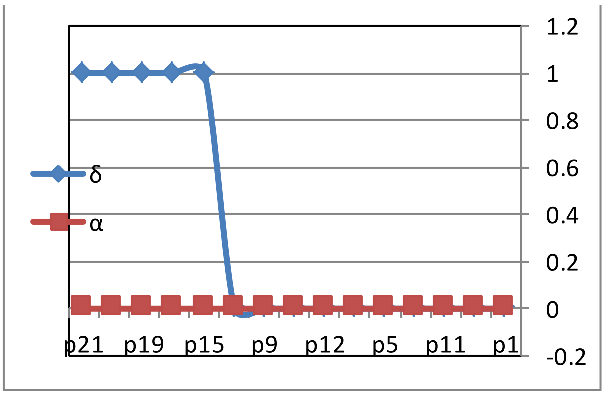
Ratio of delay (δ)=0
Ratio of setup cost (α)=0



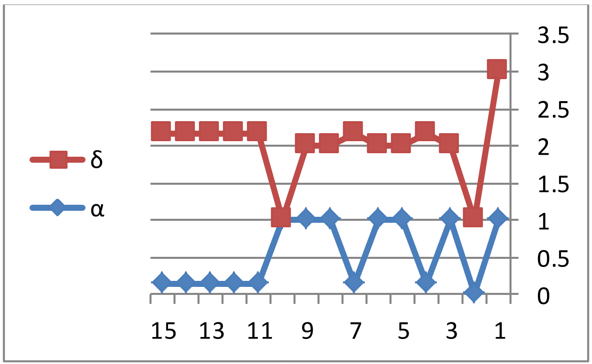
$$\text{Ratio of delay}(\delta) = \begin{cases} 2\text{Ts} & \text{for short task} \\ \text{Ts} & \text{for lareg task} \end{cases}$$

$$\text{Ratio of Setup cost}(\alpha) = \begin{cases} 0 < \alpha < 1 & \textit{forshort task} \\ 1 & \text{for lareg task} \end{cases}$$

Case2: $N_{task} > N_{vm}$



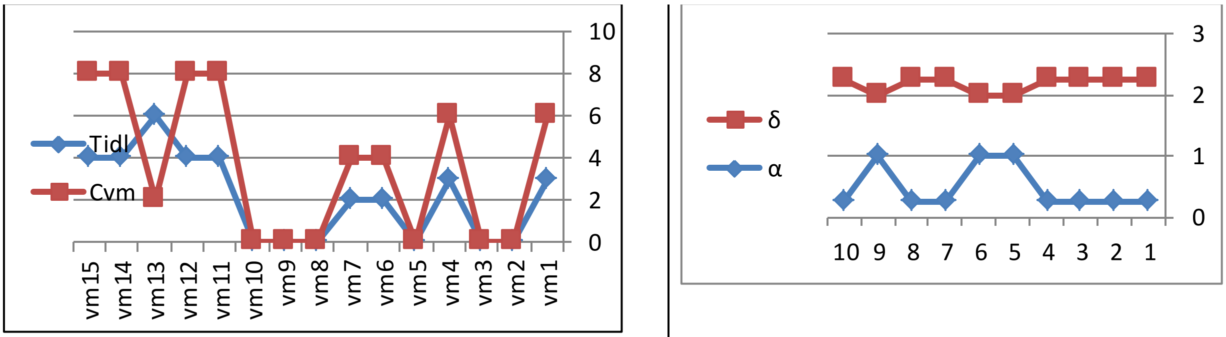
Delay =T_s only with additional task.
No setup cost for all task.



$$\text{Delay}(\delta) = \begin{cases} 2\text{Ts} & \text{for short task} \\ \text{Ts} & \text{for large task} \end{cases}$$

$$\text{Ratio of Setup cost}(\alpha) = \begin{cases} 0 < \alpha < 1 & \textit{for short task} \\ 1 & \text{for large task} \end{cases}$$

Case 3: $N_{task} < N_{vm}$



Delay =0

Idle Cost (Cvm) =

$$\begin{cases} 0 \le Cvm \le 1 & for\ longest\ task \quad . \\ 0 \le Cvm \le \frac{2}{3} & for\ middle\ task. \\ 0 \le Cvm \le \frac{1}{3} & for\ shortest\ task. \end{cases}$$

$$\text{Delay}(\delta) = \begin{cases} 2\text{Ts} & \text{for short task} \\ \text{Ts} & \text{for large task} \end{cases}$$

No idle cost but Ratio of Setup cost(α) =

$$\begin{cases} 0 < \alpha < 1 for\ short\ task \\ \quad 1 \quad \text{for large task} \end{cases}$$

Figure 5: Comparison of Results

6. CONCLUSION AND FUTURE WORK

In this paper, the researchers devised a new scheduling algorithm for cloud environment. The proposed algorithm is a generalization of clairvoyant algorithm with known duration of tasks. the researchers' algorithm is limited the delay occurs by activation and shutdown VMs. Also it distributed the idle cost of VMs over all task in a fair manner.The simulation results show that the researchers proposed algorithm out performs in terms of setup delay while producing the setup cost as good as produced by clairvoyant algorithm. In future, the researchers intend to improve this work by finding out the optimal schedule plan to decrease the idle cost of VMs.

REFERENCES

1. Gahlawat, M., Sharma, P.: Analysis and Performance Assessment of CPU Scheduling Algorithms in Cloud using Cloud Sim. International Journal of Applied Information Systems (IJAIS), Foundation of Computer Science FCS, New York, USA, 5-8 (2013).

2. Hayes, B. : Cloud computing. Communications of the ACM, 51(7): 9-11, 2008.

3. Karger, D., Stein, C., Wein, J.: Scheduling algorithms: Algorithms and Theory of Computation Handbook (1997).

4. Pruhs, K., Sgall, J., Torng, E.: Online scheduling.: pp. 115- 124 (2003).

5. Padmavathi, M., MahabbobBasha, S.,Pothapragada, S. : A Survey on Scheduling Algorithms in Cloud Computing : IOSR Journal of Computer Engineering (IOSR-JCE), 16(4), 27-32 (2014).

6. Sgall, J.: On-line scheduling. In: Developments from aJune 1996 seminar on Online algorithms: the state of the art, pp 196- 231(1998).

7. Azar, Y., Ben-Aroya, N., Devanur, N.: Cloud Scheduling with Setup Cost. In proc. SPAA'13, June 23–25, Canda (2013).

8. Ingole, A., Chavan, S., Pawde, U.: An optimized algorithm for task scheduling based on activity based costing in cloud computing. In Proc. 2011 (NCICT) published in International Journal of Computer Applications. (IJCA). 34-37 (2011).

9. Selvarani, S., SudhaSadhasivam, G.: Improved cost-based algorithm for task scheduling in cloud computing. In Proc. International Symposium on Cloud and Services Computing, IEEE, 1-5(2010).

10. Verma, A., Kaushal, S.: Cost-Time Efficient Scheduling Plan for Executing Workflows in the Cloud. Journal of Grid Computing. Springer Science, 493-506 (2015).

11. Choudhary, M., Peddoju, S.: A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment. International Journal of Engineering Research and Applications (IJERA). 2 (3), 2564-2568 (2012).

12. Chawla, Y., Bhonsle, M.: Dynamically optimized cost based task scheduling in Cloud Computing: (ITETTCS). 2 (3), 38-42 (2013).

13. Anand, P., Goswami, P.: Cost Based Algorithm Used In CloudSim. IJEDR. 2(3), 3112- 3116 (2014).

International Journal of New Computer Architectures and their Applications (IJNCAA) 8(3): 142-147
The Society of Digital Information and Wireless Communications, 2018 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Proportional Weighted Round Robin: A Proportional Share CPU Scheduler in Time Sharing Systems

Samih M. Mostafa
Math. Dept., Faculty of Science, South Valley University, Qena, Egypt.
samih_montser@sci.svu.edu.eg

ABSTRACT

In time sharing systems, many processes reside in the ready queue and compete for execution by the processor. Therefore, scheduling these processes on CPU as it can run only one process at a time is needed. In this paper, a modified version of Round-Robin (RR) called proportional weighted round robin (PWRR) is proposed. The proposed scheduler is a proportional share scheduler designed explicitly for time sharing systems. The proposed scheduler improves some scheduling criteria by minimizing turnaround times, waiting times, and context switches for the running processes. A threshold is considered to determine whether the system cannot take away the CPU from the process until it finishes or the process is interrupted due to the expiration of its time slice assigned by the RR policy. According to evaluation results, the proposed scheduler minimizes some scheduling criteria (turnaround times, waiting times, and context switches in this context)..

KEYWORDS

Process scheduling, Time sharing system, Round Robin, CPU scheduler, Round Robin.

1 INTRODUCTION

1.1 Overview

Allowing multiple processes to run concurrently in a system and share resources (e.g., CPU time) is a recent architectural technique that improves resource utilization. Such kind of systems is called time sharing systems, in which, multiple processes in memory, all compete for execution. The main objective of time sharing is to switch the CPU among processes so frequently that users can interact with each program while it is running. CPU utilization is an important concern that arose due to time sharing. So, the issue is to schedule the processes such that CPU does not sit idle. Since CPU can run a single process at a time, other processes must wait. When multiple processes are running on a single CPU, CPU time needs to be shared among all these processes. However, it seems difficult to manage this time among all the processes manually. The software that selects the processes to be scheduled according to a specific algorithm is called scheduler [1]. The scheduler that selects a process from the ready queue and dispatches to the CPU is called short term scheduler. Short term scheduler is very frequent; the process stopped temporarily from execution and is sent to the queue and competes again for execution [13].

1.2 Motivation

The behavior of a system may depend on various general criteria such as waiting time, the total time spent in the ready queue by a process, turnaround time, the time since the process was submitted to the time of completion, and so on [14]. The behavior of the system must be optimized. Different CPU-scheduling algorithms have different properties, and the choice of a particular algorithm may favor one class of processes over another. Choosing which scheduling algorithm is used for a specific system depends on which characteristics are used for comparison between CPU-scheduling algorithms. Many criteria have been suggested for comparing CPU-scheduling algorithms. The CPU scheduler is designed to allocate the resources, CPU time in this context, to all processes. The scheduling problem can be stated shortly as: which process should be moved when, to where and for how long it will run [2, 3, 4].

In general form, scheduling algorithms have been found to be NP-complete (i.e., it is believed that there is no optimal polynomial-time algorithm for them [4, 5]). Good scheduling performance may lead to a give-and-take situation, such that improving performance in one trend hurts performance in another [2]. The proposed work in this paper improves the system performance by minimizing user-based scheduling criteria (i.e., waiting time, turnaround time, and context switches). There are many different CPU scheduling algorithms. First Come First served (FCFS) algorithm is the simplest one. The policy of FCFS is managed with a First In First Out (FIFO) queue. Shortest Job First (SJF) is a different approach to CPU scheduling. The CPU is assigned to the process that has the smallest next CPU burst time. Another algorithm called Round-Robin (RR) is designed especially for time sharing systems. It is similar to FCFS scheduling, but preemption is added by defining a small unit of time, called time slice, to enable the system to switch between processes. Efficiency and effectiveness of RR are arising from its low scheduling overhead of O(1), which means scheduling the next process takes a constant time [6, 7, 8].

1.3 Problem Statement

The most important issue in RR policy is choosing the time slice. The size of time slice affects the performance of the system [15]. If the size of the time slice is too long, this will cause convoy effect (i.e., processes wait for the one big process to get off the CPU). On the other hand, if the size of the time slice is too short, more context switches occurs. Context switch time is pure overhead, because the system does no useful work while switching. Hence, the performance of the system will be degraded [9]. Allowing short processes to get more CPU time gives a share in decreasing context switches overhead. It can be concluded that the time slice must be selected in a balanced range (i.e., not be as too short or too large).

1.4 Research Contribution of this Paper

In this paper, a proportional share CPU scheduler is proposed to optimize the performance of the system by reducing some user based scheduling criteria. In this paper, the time slice assigned to a process will be proportional to its burst time, and under a predefined condition short process will gain more CPU time. In this work, a modification is implemented to RR. The modification is based on changing the time slice of each process depending on its burst time. The improvement of the proposed scheduler over RR scheduling is experimentally demonstrated, and experimental results show that the proposed scheduler can minimize the scheduling criteria.

The rest of this paper is structured as follows: section 2 presents the related research. Section 3 discusses the proposed scheduler PWRR. The experimental results are given in section 4. Section 5 presents the conclusion.

2 RELATED RESEARCHES

Various scheduling algorithms are discussed in this section.

2.1 Burst Round Robin Algorithm:

Burst Round Robin (BRR) [10] is a weighting algorithm based on burst times of processes. The higher weight, the more time slice. Short processes will leave the CPU earlier because it will be given more CPU time.

2.2 Changeable Time Quantum

CTQ algorithm [2] finds a value of time slice that gives the minimum waiting time. Depending on the burst times of running processes, the value of time slice changes in each round.

2.3 Enhanced Round Robin Algorithm

Tajwar et al., [11] proposed a dynamic round robin algorithm. In every round, the proposed algorithm assigns a new time slice equals to the mean burst time of all running processes.

2.4 Dynamic Average Burst Round Robin

DABRR [12] is a dynamic scheduling algorithm based on RR. The processes are sorted in an ascending order based on burst times.

3 THE PROPOSED ALGORITHM

3.1 PWRR Definitions

PWRR is a proportional CPU scheduler that assigns a time slice to each process proportional to its burst time. The terminology list used in this work is defined in Table i.

Table i. List of terminology.

PID	Process identification
n	Number of processes
BT[i]	Original burst time of process i
BT[i][r]	Burst time of process i in round r
W[i][r]	Weight of process i in round r
RR_TQ	Time quantum assigned by RR policy

NTQ[i][r]	New time quantum of process i in round r
TSH	Predefined threshold
WT[i]	Waiting time of process i
TAT[i]	Turnaround time of process i
AVGWT	Average waiting time
AVGTAT	Average turnaround time
CS[i]	Context switches

The design issues with the functioning of PWRR are discussed as following:

i. The queue is a FIFO queue.

ii. The weight of the process equals the process's burst time divided by summation of all burst times in the queue:

$$W[i][r] = \frac{BT[i][r]}{\sum_{i=1}^{n} BT[i]}, BT[i][1] = BT[i] \quad (1)$$

iii. The process's time slice:

$$NTQ[i][r] = (1 - W[i][r]) \times RR\_TQ \quad (2)$$

iv. The burst time:

$$BT[i][r] = BT[i][r-1] - NTQ[i][r-1] \quad (3)$$

v. The burst time of the selected process to be executed is compared with time slice assigned by the RR policy under a predefined threshold (the value of the threshold is an implementation choice).

vi. If the condition in (v) is true, this short process will complete execution until termination.

vii. If the condition in (v) is false, the running process will be interrupted by the expiration of time slice obtained from (iii) and is placed back at the tail of the ready queue.

Figure 1shows the flow chart of the proposed scheduler.

3.2 Illustrative Example

The following example simplifies the proposed consideration. Four processes arrive at the same time, each with burst time given in milliseconds (Table ii).

Table ii. Set of processes with different CPU burst times.

PID	BT[i]
P1	21
P2	20
P3	6
P4	13

a. Under RR

The following Gantt chart shows the result under RR (RR_TQ = 10ms). The scheduling criteria calculated are shown in Table iii.

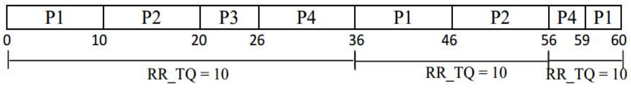


Table iii. The scheduling criteria under RR.

Process	BT[i]	WT[i]	TAT[i]	CS[i]
P1	21	39	60	2
P2	20	36	56	1
P3	6	20	26	0
P4	13	46	59	1
		AVGWT = 35.25 ms	**AVGTAT = 50.25 ms**	**No. CS = 4**

b. Under PWRR

In this work, the threshold is taken to be *TSH = 0.3 × RR_TQ*. Table iv shows the weight of the processes and the time slice for each process.

Table iv. Processes, their weights and time slices under PWRR in first round.

PID	BT[i]	BT[i][1]	W[i][1]	NTQ[i][1]
P1	21	21	0.35	6.5
P2	20	20	0.333333333	6.666666667
P3	6	6	0.1	9
P4	13	13	0.216666667	13
sum	**60**			

Process P4 achieves the condition in v, then it will continue execution until termination. The following Gantt chart shows the result under proposed scheduler:

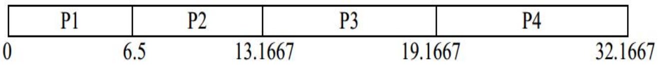


Table v shows the burst times in the second round and the time slices assigned for each process.

Table v. Processes, their weights and time slices under PWRR in second round.

PID	BT[i]	BT[i][2]	W[i][2]	NTQ[i][2]
P1	21	14.5	0.241667	7.58333
P2	20	13.3333	0.222222	13.3333
P3	6	Terminated		
P4	13	Terminated		
sum	**60**			

Process P2 achieves the condition in v, then it will continue execution until termination.

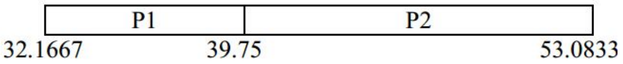
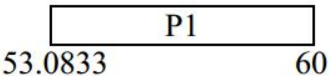


Table vi shows the burst times and the time slices in the third round.

Table vi. Processes, their weights and time slices under PWRR in third round.

PID	BT[i]	BT[i][3]	W[i][3]	NTQ[i][3]
P1	21	6.91667	0.115278	6.91667
P2	20	Terminated		
P3	6	Terminated		
P4	13	Terminated		
sum	**60**			



The scheduling criteria calculated are shown in Table vii.

Table vii. The scheduling criteria under PWRR.

Process	BT[i]	WT[i]	TAT[i]	CS[i]
P1	21	39	60	2
P2	20	33.0833	53.0833	1
P3	6	13.1667	19.1667	0
P4	13	19.1667	32.1667	0
		AVGWT = 26.1042 ms	**AVGTAT = 41.1042 ms**	**No. CS = 3**

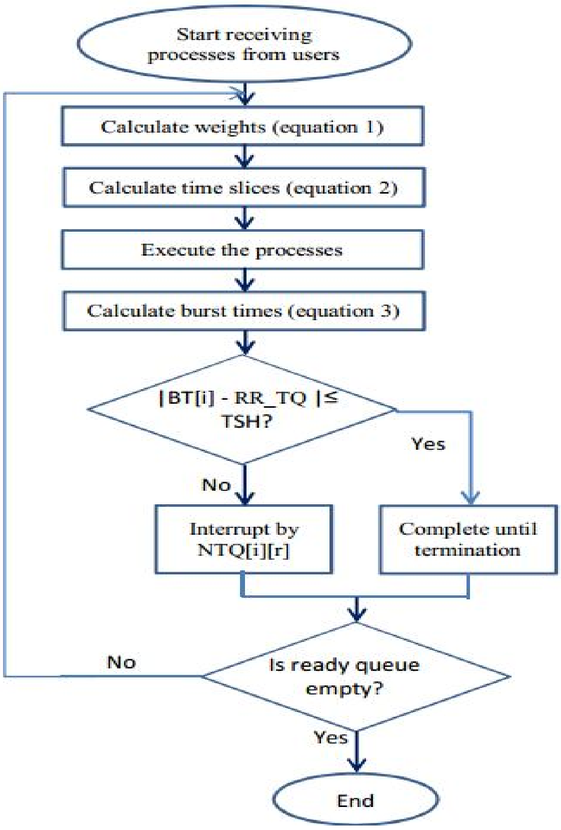

Start receiving processes from users
Calculate weights (equation 1)
Calculate time slices (equation 2)
Execute the processes
Calculate burst times (equation 3)
|BT[i] - RR_TQ |≤ TSH?
Yes
No
Interrupt by NTQ[i][r]
Complete until termination
No
Is ready queue empty?
Yes
End

Figure 1.Proposed scheduler flow chart.

4 EXPERIMENTAL RESULTS

To demonstrate the effectiveness of PWRR, some experimental data quantitatively comparing PWRR performance against RR considered on different combinations of number of processes in two scenarios. In the first scenario, the processes arrive at the same time. In the second scenario, the processes arrive at different times. The experiments were repeated many times for each scenario. In each scenario, there are two cases. In the first case, the processes are running in the order they come. The scheduler is called US_PWRR. In the second case, the processes are sorted in an ascending order. The process sets were generated by process generator routine. Each process has its id, arrival time, and burst time. The process arrival was modeled as a Poisson random process. The scheduler is called S_PWRR. The results show a significant improvement in terms of minimizing scheduling criteria (average waiting time, average turnaround time, and number of context switches). Figures 2, 3 and 4 show the average waiting times, turnaround times and number of context switches comparisons respectively in first scenario. Figures 5, 6 and 7 show the average waiting times, turnaround times and number of context switches comparisons in second scenario respectively.

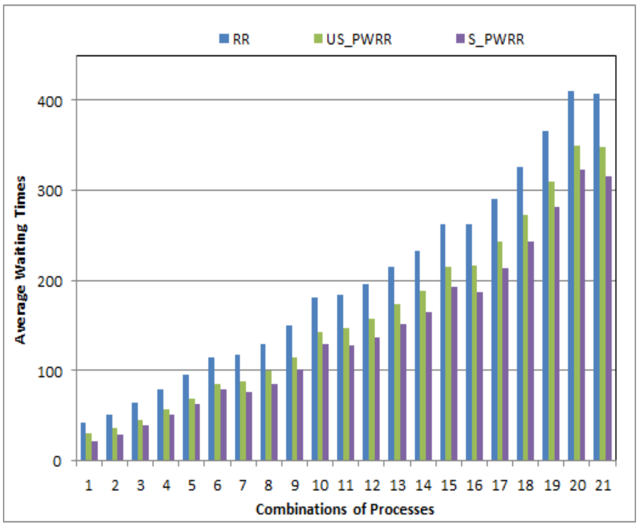


Figure 2: Average waiting times comparison in first scenario.

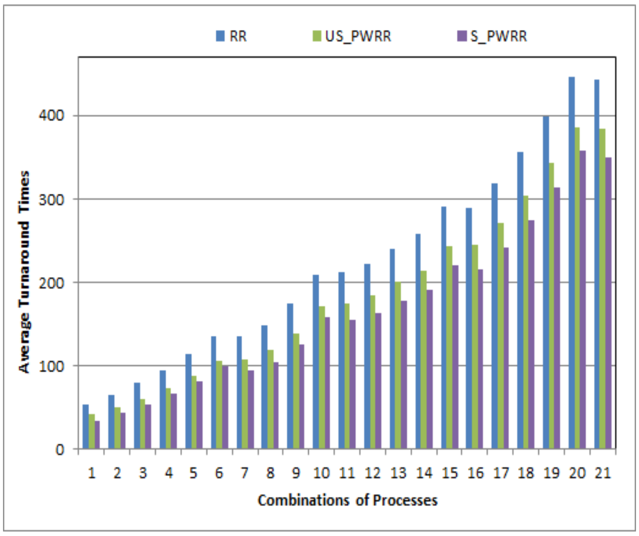


Figure 3: Average turnaround times comparison in first scenario.

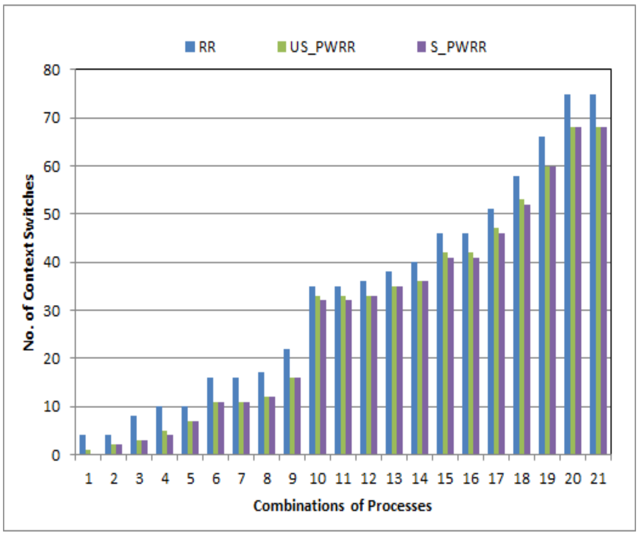


Figure 4: Number of context switches comparison in first scenario.

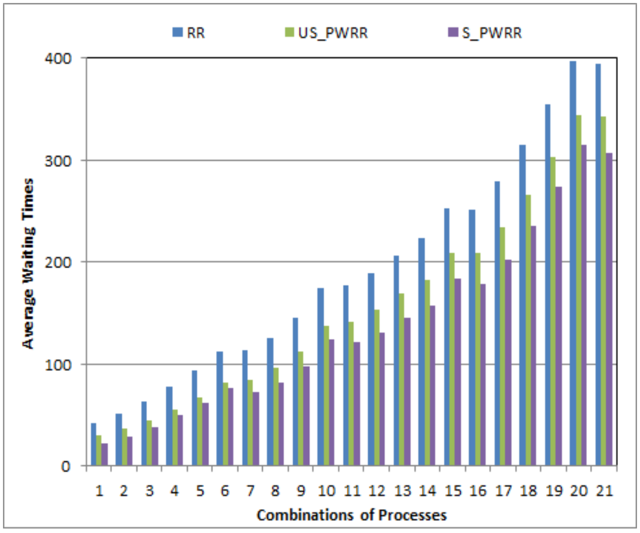


Figure 5: Average waiting times comparison in second scenario.

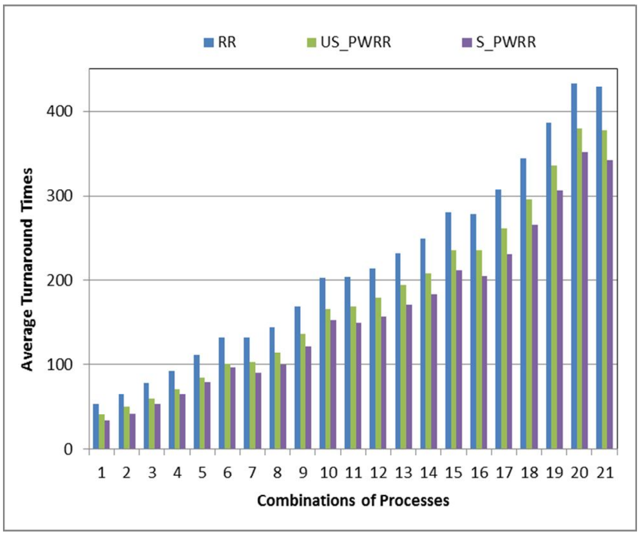


Figure 6: Average turnaround times comparison in second scenario.

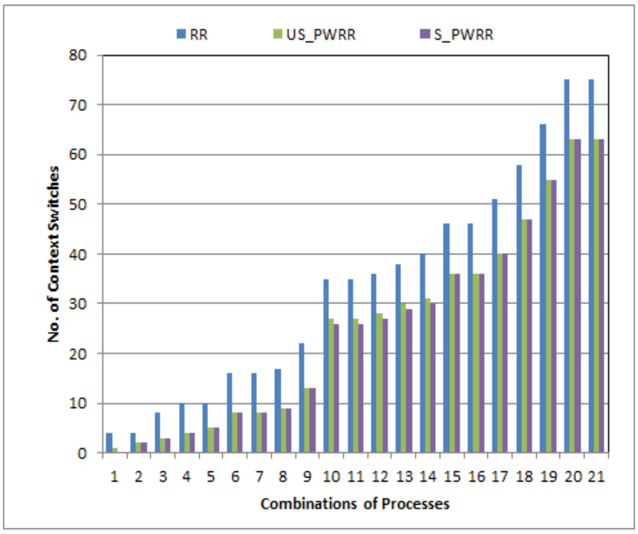


Figure 7: Number of context switches comparison in second scenario.

CONCLUSION

In this paper, a novel CPU scheduler has been proposed to benefit from low scheduling overhead of RR algorithm. The proposed scheduler based on RR scheduler because RR is designed especially for time sharing systems. The proposed algorithm assigns new time slice to each process proportional to its burst time. Short process will be given a chance to get more CPU time. The results show a significant improvement in terms of minimizing scheduling criteria. The simulation study showed that the performance of the proposed algorithm is higher than RR in general time sharing systems.

REFERENCES

1. A. Silberschatz, P. B. Galvin, and G. Gange, “Operating Systems Concepts”, John Wiley and Sons.9th Ed, International Student Version, (2013).
2. S. M. Mostafa, S. Z. Rida, and S. H. Hamad. “Finding Time Quantum of Round Robin CPU Scheduling Algorithm in General Computing Systems using Integer Programming”, International journal of Research and Reviews in Applied Sciences. (October 2010).
3. S. M. Mostafa and S. Kusakabe, “Achieving Better Fairness for Multithreaded Programs in Linux using Group Threads Scheduler”, 2013 International Workshop on ICT at Beppu, Kamenoi Hotel, Beppu, Oita, Japan, 12th to 14th. (December 2013).
4. J. D. Ullman, “Polynomial complete scheduling problems”, In Proc. of the fourth ACM symposium on Operating system principles, pp. 96 – 101, 1973.
5. K. Ramamritham and J. A. Stankovic, "Scheduling algorithms and operating systems support for realtime systems", Proceedings of the IEEE, vol. 82, no. 1, (1994).
6. L. Abeni, G. Lipari, and G. Buttazzo, “Constant bandwidth vs. proportional share resource allocation”, In Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, June 1999.
7. B. Caprita, W.C. Chan, and J. Nieh, “Group Round-Robin: Improving the Fairness and Complexity of Packet Scheduling”, Technical Report CUCS-018-03, Columbia University, June 2003.
8. B. Caprita, W.C. Chan, J. Nieth, C. Stein, and H. Zheng, “Group ratio round-robin: O(1) proportional share scheduling for uni-processor and multiprocessor systems”, In USENIX Annual Technical Conference, 2005.
9. N. Chauhan, “Principles of Operating Systems”, Oxford University Press. (2014).
10. T. Helmy and A. Dekdouk, “Burst round robin as a proportional-share scheduling algorithm”, In Proceedings of The fourth IEEE-GCC Conference on Towards Techno Industrial Innovations, pp. 424-428, 11-14, at the Gulf International Convention Center, Bahrain. (2007).
11. M. M. Tajwar, Md. N. Pathan, L. Hussaini, and A. Abubakar, “CPU Scheduling with a Round Robin Algorithm Based on an Effective Time Slice”, Journal of Information Processing Systems (JIPS) vol. 13, no. 4, pp. 941 – 950. (August 2017).
12. A. R. Dash and S. K. Samantra, “An optimized round Robin CPU scheduling algorithm with dynamic time quantum”, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT) 5 (1):7–26. doi:10.5121/ijcseit.2015.5102. (2016).
13. S. M. Mostafa, S. Z. Rida & S. H. Hamad, “Improving Scheduling Criteria of Preemptive Processes Scheduled Under Round Robin Algorithm Using Changeable Time Quantum”, Journal of Computer Science & Systems Biology (JCSB). JCSB, vol. 4.4-04-071. (2011).
14. S. M. Mostafa and S. Kusakabe, “Towards Maximizing Throughput for Multithreaded Processes in Linux”, International Journal of New Computer Architectures and their Applications (IJNCAA) 4(2): 70-78, (2014).
15. S. Elmougy, S. Sarhan, and M. Joundy, “A novel hybrid of shortest job first and round Robin with dynamic variable quantum time task scheduling technique”, J Cloud Computing 6(1):12. (2017).

International Journal of New Computer Architectures and their Applications (IJNCAA) 8(3): 148-159
The Society of Digital Information and Wireless Communications, 2018 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Comparison of Parallel Simulated Annealing on SMP and Parallel Clusters for Planning a Drone'sRoute for Military Image Acquisition

Eman Alsafi and Soha S. Zaghloul,PhD
King Saud University
435204448@student..ksu.edu.sa
smekki@ksu.edu.sa

ABSTRACT

Drones are vastly used in many civil and military applications. However, there are many factors to be highly considered in military applications.. In order to send a military drone with the aim of acquiring images from multiple sites, the mission time should be the least possible. Therefore, the minimum route plan is required.Simulated annealing (SA) algorithm is one of the metaheuristics selected to generate a feasible solution to solve this problem.This research exploits the parallelism in the simulated annealing with the aim of accelerating the time to find a suitable solution. Parallel programming divides the problem into smaller independent tasks, and then executes the sub-tasks simultaneously. Two parallel versions are therefore developed on different environment: synchronous SA on SMP, and asynchronous SA Complete Search Space (CSS) on parallel clusters. Experiments are conducted on the parallel clusters environment of the SANAM supercomputer. This research details the CSS, and compares it with the SMP SA developed in our previous study. Comparison is made in terms of speedup, efficiency, scalability, and quality of solution.

KEYWORDS

Parallel processing; Simulated annealing; Parallel Simulated Annealing; Shared-memory Processor; Parallel Cluster; SANAM

1 INTRODUCTION

Recently, drones or Unmanned Aircraft Vehicles (UAV) became very popular. This refers to their ability to undergo dangerous missions without exposing human beings' lives to any type of danger. Drones are associated with sensors and devices such as cameras, computing units, communication tools, and others. They are remotely controlled [1,2].

Drones are utilized in diverse military and civilian applications. Examples include, but are not limited to, aerial surveillance, image acquisition, remote sensing, and scientific research [2]. In addition to saving human lives, drones complete missions quickly with minimum cost [1, 3]. On the other hand, the main restriction imposed on a drone is its limited energy; and therefore, flight time. Consequently, one of the main challenges when dealing with drones is to find an effective route plan in the minimum possible amount of time [4].

As drones usually follow preloaded instructions without human intervention, the route plan may be generated either online during the flight, or offline before taking off. Moreover, drones route planning becomes more challenging when there are several geographical locations to be visited that are dispersed apart; these are called waypoints[2]. This research targets for finding a route plan that allows drones to acquire images from predefined waypoints in the least possible amount of time. Each waypoint is to be visited exactly once. Obviously, this is analogous to the well-known Travelling Salesman Problem (TSP). Finding a near-optimum route plan is necessary to minimize the drone's power consumption during the flight to cover the largest possible geographical area; and therefore, visit the largest number of targeted waypoints. In addition, achieving the mission in the minimum possible time ensures its secrecy.

However, solving TSP problem using a brute-force approach requires a significant amount of time to try every possible solution[2] . This approach is not suitable for the problem in-hands, since time and secrecy are both important factors in military applications. Therefore, a metaheuristics algorithm, the simulated annealing (SA) algorithm is used. SA is capable of finding an acceptable local optimum route plan[5]. Although SA is used to solve several complex problems, but it requires significant processing time to find a suitable solution[6]. Therefore, parallel computing is expected to positively contribute in the solution of this problem. Parallelism may minimize the execution time to fulfill the requirements of the military mission. In addition, it may increase the chance to provide a better-quality plan. However, the SA is inherently sequential as each new solution depends on the previous one. Therefore, this imposes one of the challenges associated with parallelizing SA. The improvement of the parallel computational power can overcome this challenge. In this research, we aim to study the parallel SA on SANAM supercomputer.

The performance of the parallel program is measured in terms of three metrics; namely, the speedup, the efficiency, and the scalability [7].

Therefore, the aim of this research is to design a parallel SA implementation with the purpose of generating a route plan for military drones emitted with the intention of acquiring images at multiple sites. Therefore, the program speedup, efficiency, and scalability are to be maximized; while the final distance should be at its minimum.

The layout of this paper is as follows: Section 2 exposes similar work in the literature. Section 3 explains the design of the asynchronous CSS. Section 4 reports the experiments' results. The paper is then concluded in Section 5 with a hint to our plan to future work.

2 RELATED WORK

This chapter exposes the sequential and parallel solutions to the TSP, which is similar to the drone route planning problem There are several algorithms that provide a solution for the TSP, such as LS, BB, EAs, ACO, and hybrid algorithms [7].One of these algorithms is simulated annealing which used to solve the problem in this work. The main concern of this work is comparing the parallel simulated annealing on different parallel environments methods with are SMP and parallel clusters.

Many sequential algorithms are proposed to solve the TSP, some of which are based on the ACO algorithm. The proposed solution in[8] provides a modification of the traditional ACO method; this is known as the High Performance ACO. The traditional ACO algorithm involves a single ant randomly looking for the path; whereas the updated algorithm applies the TSP on a group of ants. The authors provide a comparison between their proposed algorithm and the ant colony system algorithm on various number of nodes. They found that the proposed algorithm completes the task in less time.

Also, Local search algorithms are widely used to solve the TSP. The research in [7] provides an experimental study to test the performance of the Lin-Kernighan and the Multi-Neighborhood Search. Results show that the Lin-Kernighan provides better results than the Multi-Neighborhood Search in terms of runtime.

On the other hand, several parallel solutions are proposed in the literature to solve the TSP using diverse parallel programming platforms.In[9], the experiment is performed on a standard multicore CPU. The reported results indicate that a gained speedup of 7.3 on 8 cores. Thus, the usage of PSO algorithm is more suitable for real-time planning for the drone. Moreover, the experiments also proved that the performance of the GA is better than the PSO. The same authors improved their results in [10] by proposing a parallel hybrid algorithm that exploits the advantages of both the PSO and the GA to generate a suitable path plan for fixed-wing drones. It is found that the gained speedup is 10.7 on a 12-core SMP.

In [2], the authors planned the drone's path using parallel ACO solution on both GPU and CUDA platforms. The generated path guides the drone in disseminating keys and collecting data from wireless sensors, which are previously deployed at minimum cost. The drone launches from a station, visits all sensors in a limited period of time, then returns back to the same station it is emitted from. In their experiment, they compared the sequential performance with the parallel implementation performance. They showed that the speedup is higher when using GPU platform.

In [5], the authors generate multiple route paths for several drones simultaneously using synchronous parallel SA on the GPU. Experiments' results prove that the processing time is reduced, and a better solution is acquired, as compared to the CPU implementation.

3DESIGN AND IMPLEMENTATION

This section discusses the general design of SA. Then, the implementation of the asynchronous CSS is then discussed.

3.1 Simulated Annealing Design

Two concepts are to be defined when it comes to designing an iterative metaheuristics algorithm; namely, the solution representation and the objective function. Since SA is classified as a single-solution based metaheuristics, it requires the definition of the neighborhood. These are detailed in the following subsections [11].

3.1.1 Solution Representation

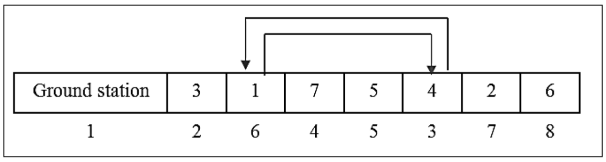
The solution representation of the drone route planning is a permutation of size *n*, where *n* is the number of the waypoints to be visited exactly once. Each permutation represents one solution as shown in Figure 4 as a sequence of nodes, where each node represents a waypoint and its index represents the corresponding order. The number of all permutations that represent the solution space taking into consideration the fixed point of start (ground station) is *(n-1)!*.

Ground station	3	1	7	5	4	2	6
1	2	3	4	5	6	7	8

Figure 1. The permutation representation of the drone route plan problem

3.1.2 Neighborhood Solution

The neighborhood of a solution is found by performing a move operator which leads to a tiny perturbations to the solution S[11]. As the drone route plan is represented by a permutation, a neighborhood is generated by the swap operator between two elements in the solution. This is illustrated in Figure 5.



Ground station	3	1	7	5	4	2	6
1	2	6	4	5	3	7	8

Figure 2. Neighbourhood solution generated by swapping the order of two waypoints

3.1.3 Objective function

The objective function is used to define the goal to be achieved by the SA. The goal of the problem in-hands is looking for the shortest route plan for a drone such that it visits each waypoint exactly once. As previously mentioned, this is similar to the TSP and has a similar objective function which is shown below:

$$f(\pi) = Min \sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)} (1)$$

where:

-π is a permutation representing a tour of the drone;

- n is the number of waypoints.

3.2 Sequential simulated annealing

As previously mentioned, the SA, like other single-solution based metaheuristics, includes two main steps. The first step is to generate the initial solution, which is constructed by using a greedy heuristic, such as the nearest neighbor algorithm or randomly. In the design of the sequential algorithm of this research, the random method is used because the greedy heuristics produces a solution in local optimum, which may not be able to provide an improved local optimum solution at the end[11].

The second step, which is the solution improvement, the design uses the swap operator between two points to generate a neighbor solution from the current solution.

In fact, the SA algorithm imitates the process of the solid hardening, which depends on the initial temperature value and the cooling rate. Therefore, the SA implementation consists of two main loops to provide a suitable solution. The outer loop, known as the cooling loop, is responsible for managing the temperature value. On the other hand, the inner loop, known as the equilibrium state loop, is responsible for constructing a neighbor solution from the current one, evaluating it, and computing the probability of the acceptance using the following formula:

$$e^{(-\Delta/T)} \quad (2)$$

where:

- Δ is the difference between the cost of the old and the new solution;
- T is the current temperature

Accordingly, the main parameters that are to be defined during SA implementation are the initial temperature, the cooling rate, and the stopping condition. The latter might be the minimum temperature or a specific number of iterations. In this research, the stopping condition is taken based on a minimum temperature. The other parameters are determined after several experiments[6]. The flowchart of the sequential algorithm is shown in Figure 3.

3.3Parallel Simulated Annealing

Metaheuristic algorithms are sequential by nature; SA is no exception. Consequently, parallelizing the SA entails a challenging problem [7]. Many approaches are proposed to parallelize SA algorithm [12]:

- Decompose the search space into smaller parts, then assign each part to a processor to find the minimum cost and share its result with other processors.
- Apply the synchronous approach, where each processor uses the same initial solution and performs parallel improvements within the same temperature. Then at each temperature value the best solution is shared between the processors to perform parallel improvement until the end. Figure 4 illustrates the synchronous approach.
- Apply the asynchronous approach, where each processor executes SA independently. The initial solution may be the same or different across the processors. Finally, compute a reduce operation to get the best solution among them. As illustrated in Figure 5.

The synchronous parallel SA on shared-memory processor (SMP) is previously studied in [13]. In this paper, the asynchronous parallel SA Complete Search Space (CSS) is investigated. The CSS algorithm starts with different initial solutions for the complete search space. The idea is illustrated in Figure 5.

3.3.1 Parallel SA approach on cluster

On the other hand, using the approach in [14] to implement SA on parallel clusters will increase the overhead of communication between nodes. This is explained by the fact that in synchronous

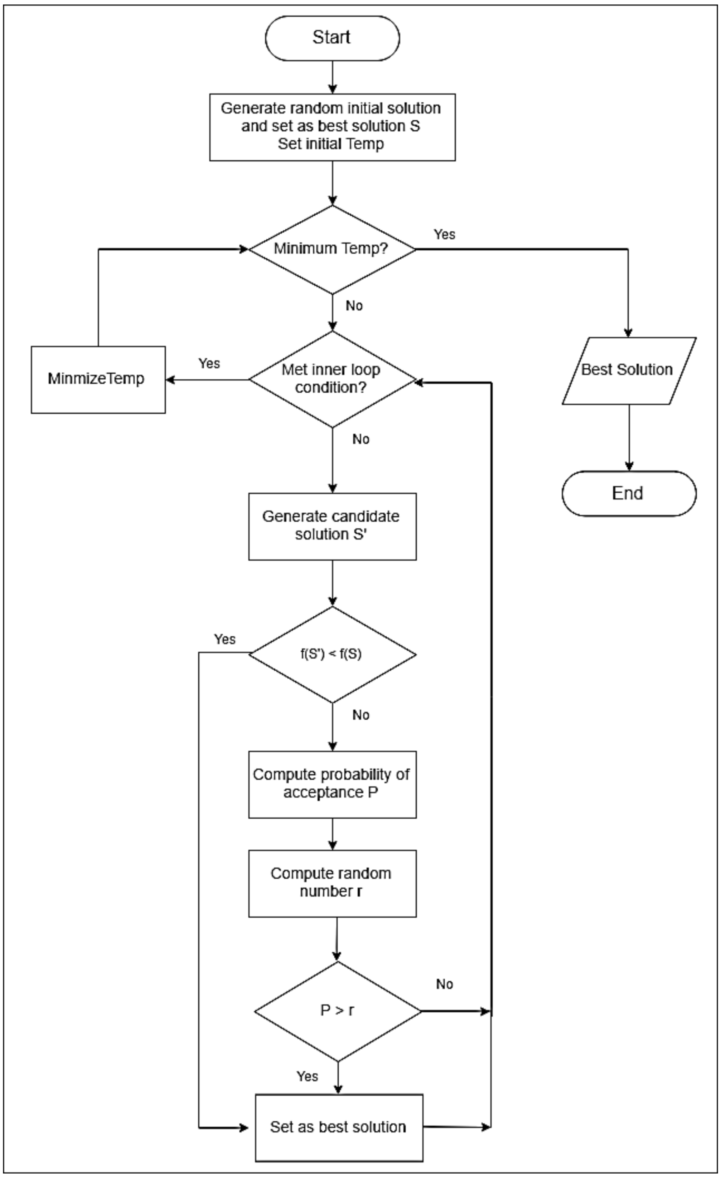

Start
Generate random initial solution and set as best solution S
Set initial Temp
Minimum Temp?
Yes
No
Met inner loop condition?
Yes
MinmizeTemp
No
Best Solution
End
Generate candidate solution S'
f(S') < f(S)
Yes
No
Compute probability of acceptance P
Compute random number r
P > r
No
Yes
Set as best solution

Figure 3. Flow chart of the SA algorithm

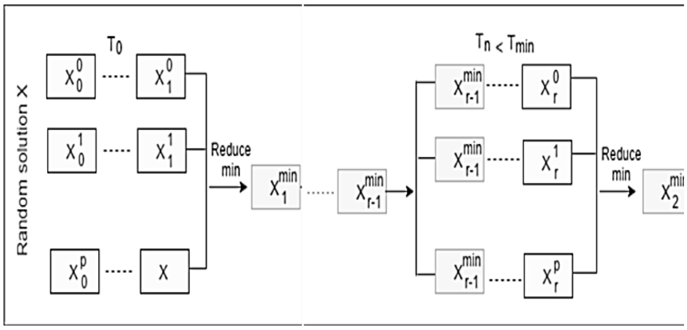


Figure 4. Synchronous SA parallel approach

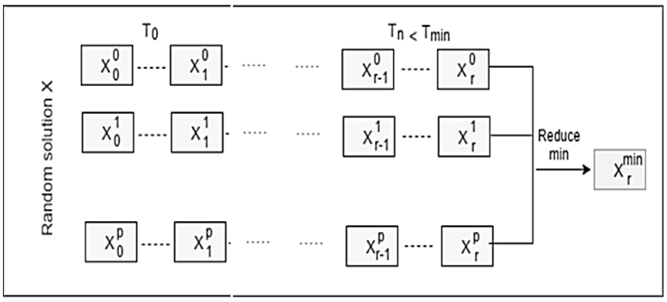


Figure 5. Asynchronous SA parallel approach

parallel SA, the processors frequently communicate with each other. Shared memory processor environment is suitable for such solution. However, in parallel clusters, communication is needed between processors after each inner loop.This is performed through message passing on parallel clusters. Message passing imposes an overhead on the program; and therefore, increases the speedup.

The asynchronous parallel SA CSS is therefore suggested to minimize the communication overhead. In this algorithm, each cluster node works on the complete search space (CSS) as illustrated in Figure 6. In the start, several initial solutions are generated and distributed over the nodes. Then each node applies the sequential SA. However, data parallelism is applied.At the end, all nodes send the produced route together with the final distance. The minimum distance with its corresponding route are then selected to be the best route plan.

3.4 Handling the Drone Energy Constraints

After generating the route plan using SA algorithm at the ground station, the route is evaluated in terms of the energy required to complete the planned mission. If the energy level is above a predefined threshold, then the drone is emitted according to the planned route. Otherwise, the mission is divided into multiple journeys. The drone is re-charged after the end of each trip, before starting a new one.

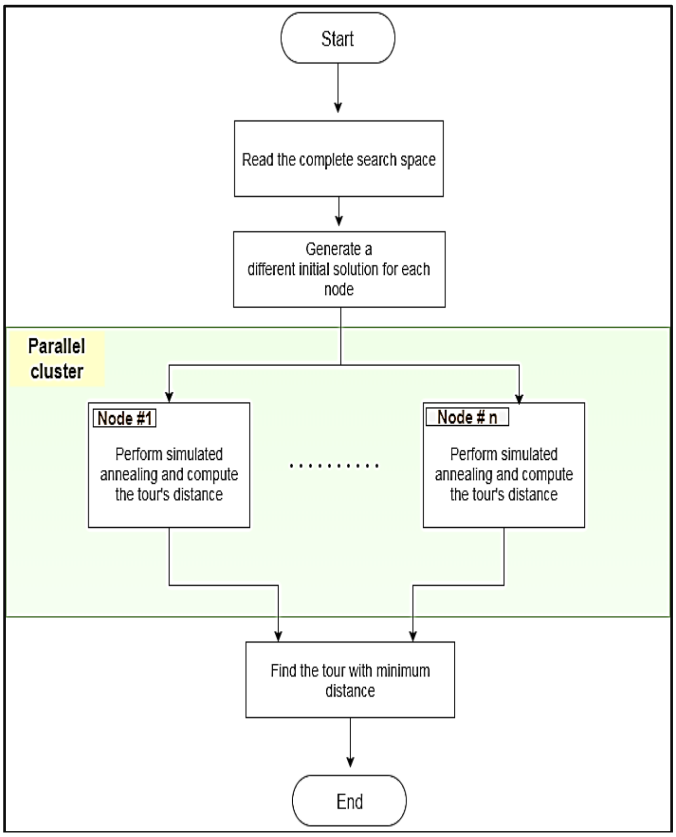

Start
Read the complete search space
Generate a different initial solution for each node
Parallel cluster
Node #1
Perform simulated annealing and compute the tour's distance
....
Node # n
Perform simulated annealing and compute the tour's distance
Find the tour with minimum distance
End

Figure 6. Flowchart of asynchronous parallel SA for complete search space approach

In fact, the drone's energy is expressed in terms of its enduring lifetime L. Therefore, the time *T*
needed to travel the final distance *D*, as planned by the SA, is calculated as follows:

$$T = D / S \qquad (3)$$

where:

- *T* is the time required to make the complete calculated tour, including the return trip to the ground station;
- *D* is the final distance as calculated by the SA algorithm;
- *S* is the drone's speed as specified in its hardware specifications

If the calculated time *T* is less than the drone's enduring lifetime *L*, then the drone is safely launched. Otherwise, the journey is broken into multiple trips. Figure 7 illustrates the previously mentioned steps.Therefore, the applied predefined threshold is the enduring lifetime of the emitted drone.

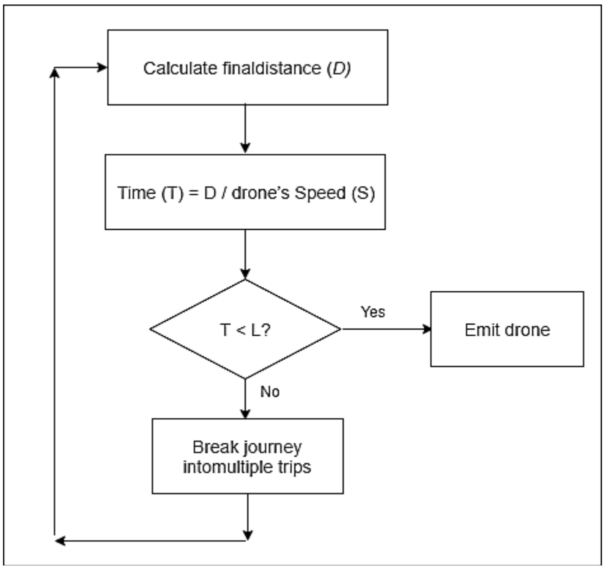

Calculate finaldistance (D)
Time (T) = D / drone's Speed (S)
T < L?
Yes
Emit drone
No
Break journey intomultiple trips

Figure 7. Checking drone's **energy**

4 EXPERIMENTAL RESULTS

This section details the methodology used in the conducted experiments. In addition, the obtained results are discussed, analyzed, and represented graphically. The first subsection reveals the deployed environment in terms of software and hardware specifications. Subsection 4.2 explains the performance metrics used to measure the effectiveness of the developed algorithm. Subsection 4.3 details the general methodology used to collect the figures of the experiments. Finally, subsection 4.4 details the results of both the sequential and parallel CSS algorithm.

4.1 Software and Hardware Specification

This research is implemented on KACT's Saudi supercomputer SANAM. KACST is King Abdel-Aziz City for Science and Technology in Riyadh, Kingdom of Saudi Arabia. SANAMincludes Intel Xeon E5−2650 CPUs, with 12 cores. The access to SANAM is available through a group of interactive login nodes, which are connected to KACST network and Internet [14, 15]. The program is coded under Linux Ubuntu 16, with JAVA using the pj2 library for threads management [7], and NetBeans as programming tool. In addition, Simple Linux Utility for Resource Management (SLURM)is used for Linux clusters management in SANAM. SLURM performs three main tasks: First, it is responsible for nodes allocation, management, execution, and monitoring reserved nodes. Second, it manages waiting work queues and finally, resolves conflicting resource orders [16].

In this research, ten nodes are used. This is the maximum allowed by KACST to the external users.

4.2 Performance Measurements

The main objectives of this research are to minimize the execution time, increase resources utilization, and increase scalability. In addition, the final distance is to be minimized. Therefore, speedup, efficiency, scalability, and final distance are used to evaluate the performance of the parallel program [7].

4.2.1 Speed up:

Speed up is used to measure the extent of the time reduction gained from the parallel implementation as compared to its sequential counterpart. The gained speed up is calculated as the ratio of the execution time of the sequential program *Tseq* to that of the parallel program *Tpar*[7]:

$$\text{Speed up } = \frac{\text{Tseq}}{\text{Epar}} \quad (4)$$

4.2.2 Efficiency

The efficiency (*E*) is used to measure how a program is close to the ideal speed up. In other words, it indicates the effectiveness of the parallel program to use the available resources. The ideal program efficiency is equal to 1. However, the actual efficiency is between 0 and 1. As the efficiency is closer to 1, as the program is making better use of the available hardware resources. Efficiency is computed as the ratio of the actual speed up of the parallel program *Tpar* to the number of processors *K* that are used to run the parallel program. This is expressed in the following formula[7]:

$$\text{Efficiency} = \frac{\text{Tpar}}{\text{K}} \qquad (5)$$

4.2.3 Scalability

Scalability is the ability of a program to adapt to the increasing amount of problem size. In order to measure the scalability of a parallel program, the sequential version is run multiple times; each time the problem size is increased. When the program crashes, and cannot hold any more the given problem size, the last recorded size is taken. This is *Nseq*. The same experiment is repeated with the parallel version to get *Npar*. The scalability is therefore calculated according to the following formula [7]:

$$\text{Scalability} = \frac{\text{Npar}}{\text{Nseq}}(6)$$

It is expected that the problem size increases with the increase of the number of processors.

4.3 Methodology

The execution time of a parallel program is hardly the same when run multiple times successively. This is because the operating system is conducting its own activities at the same time as the program runs. Since these activities differ from a run to another, the resulting execution time is directly affected. The interference of the operating system always increases the resulting execution time. Therefore, to measure the execution time of a program as accurate as possible, it is run multiple times – from 7 to 10 times – and the execution time is recorded after each run. Then, the minimum of these recordings is taken since this represents the less interference from the part of the operating system.

4.4 Experimental results

The results of both versions of the SA algorithm are reported in the following subsections. The parallel platform is an SMP; with a number of threads ranging from 2 to 8. A set of experiments is conducted according to the previously detailed methodology detailed with the aim of measuring the three main performance metrics: speedup, efficiency, and scalability in addition to the quality of the solution. These are exposed in the following subsections.

4.4.1 Speedup

The first set of experiments aims to explore the impact of the number of waypoints on the execution time. Therefore, the program is run multiple times with various number of waypoints; namely, 50, 100, 150, and 200. The experiment is done only for these four problems sizes as the increments in the execution time is linear.

Table 1 shows the minimum execution time for the sequential, SMP -as performed in [14]-, and CSS. Note that all the parameters of SA; namely, initial temperature, the cooling rate, and the stopping condition, are fixed. Worth to mention, the parameters are chosen after several experiments to ensure the quality of the final solution. Figure 8 shows the impact of the number of waypoints on the execution time.

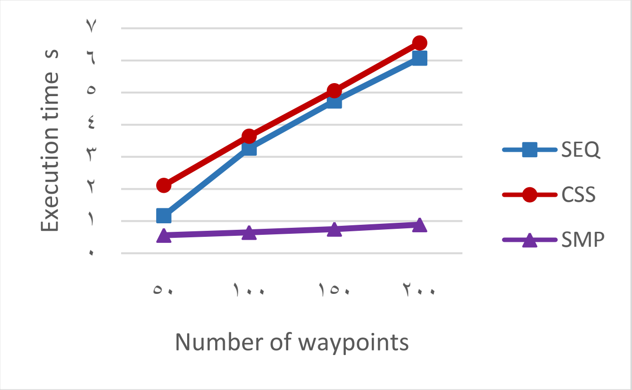


Figure 8. Relationship of Execution Time with Number of Waypoints

As seen in the graph depicted in Figure8, the execution time of sequential and parallel SA versions is proportional to the problem size. It is noted that the execution time of the parallel cluster CSS is the highest when compared to the sequential and SMP. This is due to the fact that in CSS, each node works on the complete search space. In addition,several initial solutions are to be produced and propagated to all nodes at the beginning of the program. Therefore, the communication between the nodes imposes an overhead on the execution time.

On the other hand, the execution time of the parallel SMP is the best. This is explained to the lack of communication overhead, since data is shared in the main memory.

The gained speedup is calculated from the values recorded in Table 1 according to formula 4The results are depicted in Figure 9. The highest speedup is achieved by the SMP version; it is equal to 6.81for 200 waypoints as compared to a speedup of 0.57 for the CSS.

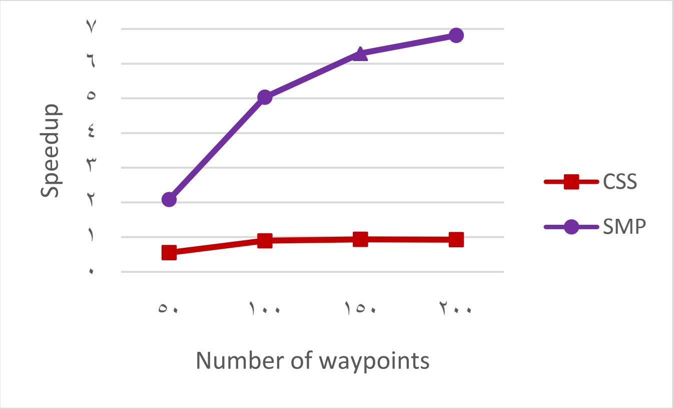


Figure 9. Relationship between Speedupand Number of Waypoints

4.4.2Efficiency Measurement

The efficiency is calculated for both SMP and CSS versions according to formula 5The results are displayed in Table 2. The corresponding graph is depicted in Figure 10. It is noticed that the efficiency is the best with parallel SMP where all threads in the nodes are utilized.

Although the efficiency is equal to 1 for the sequential program, but this does not imply that the program makes full use of the available hardware resources. However, the actual speed of the sequential program and the number of cores are both equal to one.

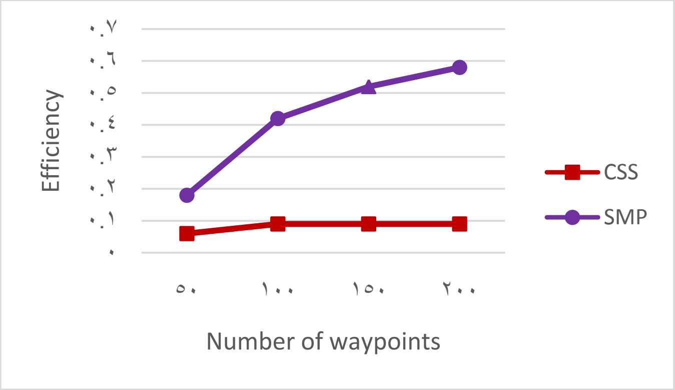


Figure 10. Relationship of Efficiencywith Number of Waypoints for SMP and CSS

4.4.3Scalability Measurement

Here, the largest number of waypoints that can be handled by each parallel SA version is divided by that handled by the sequential program. The results are reported in Table 3, with calculations deduced from formula 6Again, the scalability in parallel SMP is too much better than that of the CSS. This is explained by the fact that all threads in the node are utilized in SMP. Thus, increasing the ability to handle larger problem sizes than sequential and parallel CSS. Moreover, the parallel CSS does not provide any improvement on the sequential SA.

4.4.4The quality of the route plan

The final distance values are depicted in Table 1 with various SA algorithm versions and waypoints. The corresponding graph is shown in Figure 11. It is noticed that the quality of the route plan is the best with the SMP SA version as compared to the other program versions. However, the CSS SA produces very close distances as compared to the SMP SA. On the other hand, it gives better distance than the sequential. This is because the CSS SA uses more nodes working on different initial solutions; thus, increasing the chance of improving the distance.

5CONCLUSION AND FUTURE WORK

This research targets for generating a minimum route plan distance for a single drone emitted by a military organism for image acquisition. The area in concern may be a sensitive site, a defense

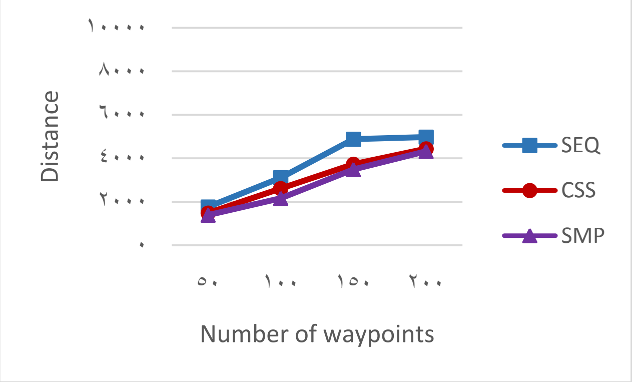


Figure 11. Quality of the final distance on parallel SA

and/or attack war front, or an enemy's territory. Therefore, the route path should be completed in the least amount of time.

The SA algorithm is implemented to solve this problem, which is analogous to the TSP. Since metaheuristics require an extensively long time for execution, parallel computing is deployed to accelerate the SA algorithm.

Therefore, several parallel versions of the SA are developed. First, the parallel SA is previously developed [14]. In this paper, another parallel version is implemented on SANAM supercomputer. Ten nodes are used to implement the program using Java programming language under Linux on SANAM supercomputer.

The two parallel versions are compared in terms of speedup, efficiency, scalability, and the final distance.

The reported results prove that the synchronized parallel SA on SMP outperforms the CSS for all number of waypoints in terms of gained speedup, efficiency, and scalability. On the other hand, the CSS outperforms the SMP SA in terms of quality of solution.

In the future, more methods to parallelize SA are to be investigated.

ACKNOWLEDMENT

The researcherswould like to extend our gratitude to King Abdel-Aziz City for Science and Technology (KACST) in Riyadh for allowing us to use their SANAM supercomputer at all times. Not only this, but the staff was also of great support and helpful; replying to our posed questions promptly albeit their heavy duty load.

REFERENCES

[1] N. Özalp and O. K. Sahingoz, "Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference*, 2013, pp. 308-317.

[2] M. O. UgurCekmez, "A UAV Path planning with parallel ACO algorithm on CUDA platform," presented at the IEEE Unmanned Aircraft Systems (ICUAS), FL, USA, 2014.

[3] M. Coeckelbergh, "Drones, information technology, and distance: mapping the moral epistemology of remote fighting," *Ethics and information technology*, vol. 15, pp. 87-98, Jun 2013.

[4] X.-f. Liu, Z.-w. Guan, Y.-q. Song, and D.-s. Chen, "An optimization model of UAV route planning for road segment surveillance," *Journal of Central South University*, vol. 21, pp. 2501-2510, Jun 2014.

[5] T. Turker, G. Yilmaz, and O. K. Sahingoz, "GPU-Accelerated Flight Route Planning for Multi-UAV Systems Using Simulated Annealing," in *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, 2016, pp. 279-288.

[6] M. Sanjabi, A. Jahanian, S. Amanollahi, and N. Miralaei, "ParSA: parallel simulated annealing placement algorithm for multi-core systems," in *Computer Architecture and Digital Systems (CADS), 2012 16th CSI International Symposium*, 2012, pp. 19-24.

[7] A. Kaminsky, "BIG CPU, BIG DATA: Solving the World's Toughest Computational Problems with Parallel Computing," 2016.

[8] *KACST , The Saudi Supercomputer "SANAM" is the World's 2nd Leader in Energy Efficiency", KACST, 2012. [Online]. Available: https://www.kacst.edu.sa/eng/about/news/Pages/news3841117-3854.aspx. [Accessed: 25- Apr-2017].*

[9] V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 132-141, May. 2013.

[10] V. Roberge, M. Tarbouchi, and F. ALLAIRE, "Parallel hybrid metaheuristic on shared memory system for real-time UAV path planning," *International Journal of Computational Intelligence and Applications*, vol. 13, p. 1450008, Jun. 2014.

[11] E.-G. Talbi, *Metaheuristics: from design to implementation* vol. 74 ,pp 126-133: John Wiley & Sons, 2009.

[12] A. Ferreiro, J. García, J. G. López-Salas, and C. Vázquez.:An efficient implementation of parallel simulated annealing algorithm in GPUs, *Journal of Global Optimization*, vol. 57, pp. 863-890, Nov. 2013.

[13]S. Zaghloul and E. Alsafi, "Drone route planning for military image acquisition using parallel simulated annealing", International Journal of New Computer Architectures and their Applications (IJNCAA), vol. 7, no. 3, Sep, 2017.

[14] D. Rohr, S. Kalcher, M. Bach, A. A. Alaqeeliy, H. M. Alzaidy, D. Eschweiler, V. Lindenstruth, S. B. Alkhereyfy, A. Alharthiy, and A. Almubaraky, "An energy-efficient multi-GPU supercomputer," in *High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC, CSS, ICESS), 2014 IEEE Intl Conf on*, 2014, pp. 42-45.

[15] "Intel® ARK (Product Specs). (2017). Intel® Xeon® Processor E5-2650 v4 (30M Cache, 2.20 GHz) Product Specifications. [online] Available at: https://ark.intel.com/products/91767/Intel-Xeon-Processor-E5-2650-v4-30M-Cache-2_20-GHz [Accessed 13 Dec. 2017].".

[16] Slurm.schedmd.com. (2017). Slurm Workload Manager. [online] Available at: https://slurm.schedmd.com/quickstart.html [Accessed 14 Dec. 2017]."

[17] T. Rauber and G. Rünger, *Parallel programming: For multicore and cluster systems*: Springer Science & Business Media, 2013.

[18] A. Kaminsky, "Building Parallel Programs: SMPs, Clusters, and Java. Cengage Course Technology (2010)," ISBN 1-4239-0198-3.

Table 1 Relationship between the execution time and number of waypoints with the corresponding output distance

#waypoints	Sa algorithm version	Execution time (s)	Distance
50	**Sequential**	1.1711	1762
	Parallel cluster CSS	2.113	1483
	Parallel SMP	0.561	1389
100	**Sequential**	3.276	3110
	Parallel cluster CSS	3.644	2610
	Parallel SMP	0.651	2162
150	**Sequential**	4.741	4880
	Parallel cluster CSS	5.059	3733
	Parallel SMP	0.753	3487
200	**Sequential**	6.072	4976
	Parallel cluster CSS	6.544	4439
	Parallel SMP	0.891	4320

Table 2 Efficiency of sequential, SMP, and CSS versions

#waypoints	Sa algorithm version	Execution time (s)	Speed up	Efficiency
50	**Sequential**	1.1711	1	1
	Parallel cluster CSS	2.113	0.55	0.055
	Parallel SMP	0.561	2.09	0.17
100	**Sequential**	3.276	1	1
	Parallel cluster CSS	3.644	0.9	0.09
	Parallel SMP	0.651	5.03	0.42
150	**Sequential**	4.741	1	1
	Parallel cluster CSS	5.059	0.94	0.09
	Parallel SMP	0.753	6.3	0.52
200	**Sequential**	6.072	1	1
	Parallel cluster CSS	6.544	0.93	0.093
	Parallel SMP	0.891	6.81	0.57

Table 3 Measure scalability of parallel SA

SA algorithm version	Size up	Scalability
Sequential	3100	--
Parallel cluster CSS	3000	0.97
Parallel SMP	23000	7.4

A Comprehensive Survey of Security Related Challenges in Internet of Things

E. Ezema, A.Abdullah, NFM Sani
Faculty of Computer Science and Info. Technology, Universiti Putra Malaysia
Serdang 43400, Selangor D.E Malaysia
ernestezema@gmail.com, *azizol@upm.edu.my*, *fazlida@upm.edu.my*

ABSTRACT

The IoT network is a decentralized type of network which can sense information and transmit the information to a base station. Due to small size of the sensor nodes, energy consumption and security is seen as the major challenge to the IoT network. The LEACH is the energy efficient protocol which can divide the whole network into fixed size clusters. In each cluster, the cluster heads are selected, to transmit data to base station. The cluster heads are selected in the network based on the energy of each node and distance from sensor node to base station. The energy of the sensor node is dissipated when each node receives or transmits data to the base station. The energy is also consumed when the sensor nodes aggregate data to cluster head. In this paper, we analyze existing research works in IoT system and proposed security approaches. In the literature, we have studied that many related challenges have been discovered and work on, but they are still open challenges to these different approaches and techniques. The paper highlights various security challenges of IoT and contributions from researcher.

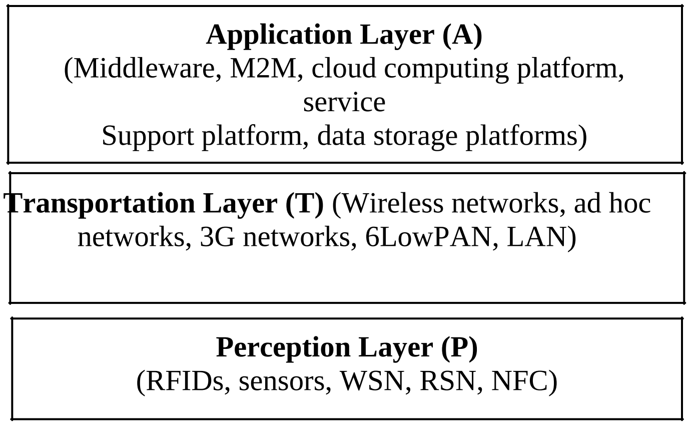
KEYWORDS

Internet of Things (IoT); LEACH Protocol; IoT Revolution; Challenges; IoT direction

1 INTRODUCTION

The application of the IoT is widely used in many applications due to large growth of mobile devices, embedded and wireless network communication, cloud computing and data analytics. Large numbers of devices are connected over public or private Internet Protocol Networks with the help of billions of objects that can sense, communicate and share information [1]. The combination of communication capabilities which are given by the data transmission is given by these lines present. RFID is known to be the main object within the IoT. The building of global infrastructure for RFID tags which is known to be a wireless layer seen on the top of Internet. The communication is made amongst network of interconnected objects and the interconnected computers. There are different Internet Protocol (IP) location for the objects in some instance. These objects are embedded within the complex systems. The IoT devices perform various functions, devices with sensors are used for example to gather information related to temperature, and other aspects present in the surroundings [2]. The sensors can interact to each other to transfer the gathered information and provide further processing as per the requirements of the current applications. There is no standard IoT architecture suggested by any researcher. Everyone has their own perception and have suggested their own architecture for IoT, It has three layers: perception layer, transportation layer and application layer, as shown in figure 1. There is communication between each layer as shown in the figure below:

Figure 1: Layers in IoT Architecture



Application Layer (A)
(Middleware, M2M, cloud computing platform, service
Support platform, data storage platforms)

Transportation Layer (T) (Wireless networks, ad hoc networks, 3G networks, 6LowPAN, LAN)

Perception Layer (P)
(RFIDs, sensors, WSN, RSN, NFC)

1.1 Perception layer

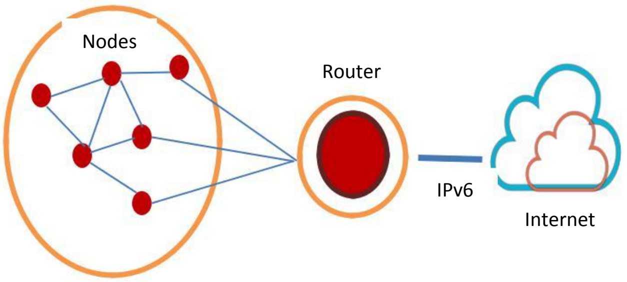
The perception layer is the layer which deals with the physical aspects which is mainly consists of sensors, controllers or RFIDs. Nodes have been utilized for data acquisition and data control [3]. With the help of Wireless Sensor Networks (WSN), Radio Sensor Networks (RSN) or Near Field Communications (NFC), all the nodes are connected internally with the sensor network.

Figure 2. Perception layer description

1.2 Transportation layer

The transportation layer is the layer by which remote and heterogeneous sensor networks relate to the global world [4]. The main function of this layer is to ensure the information transmission and its storage. It contains the wireless networks, ad-hoc networks for data access and Ipv4- based Internet, 6LowPAN technology for data transmission.

Figure 3. 6LoWPAN Network



1.3 Application layer

The application layer supports all kinds of business services and interacts with end user directly. It also includes middleware, M2M, cloud computing platform, service support platform [5]. All the collected data from the transportation layer are processed and analyzed by this layer. Various IoT applications are supported by this layer such as intelligent transportation, smart home, smart healthcare etc.

In the recent year very much, importance is given to the Security and privacy as it protects the data from any theft. Protection of data is very much necessary with the increase in the growth of the data nowadays, hence various mechanism is invented to minimize the major limitation of IoT. Security within these systems is always major concerns as there are numerous systems involved during the communication being held [6]. Thus, the data involved within these systems is to be made secure. Various data isolation techniques are provided here which can help in providing encryption measures within the systems. With the application of these systems it can be made sure that the data being transmitted to the destination reaches there without any modifications or stealing of important information by the unauthorized access [7]. One of other major concerns within these systems is the violation of privacy of data present in them. To ensure that only the authorized users are given access to the private information, various algorithms are proposed here which can ensure that no unauthorized users have access to this information.

1.4 Misdirection Attack in WSN

It is the attack in which packets are routed by the attacker to its children to other distant nodes but do not transfer to its legitimate parent. The main purpose of the intruder is to increase the latency by misdirecting the incoming messages due to which few packets are prevented from reaching the base station. This layer is mostly faced with DoS attacks. The most popular Denial of Service Attack is the Misdirection attack. It changes the path of the packets to create confusion among nodes. Misdirection attacks are of different types and can be performed in two ways:

Figure 4. Application layer description

i. Packets forwarded to a node large away from the destination: This type of misdirection attack is very dangerous as forwarded packets are transferred to a sensor node which is far away and prevents packets to reach the destination on time. It decreases the throughput and increases the delay infinitely [8].

ii. Packets forwarded to a node close to actual destination: This attack is less intense as compared to previous one because it took long route to transfer packets to its destination node. Due to which there is increase in delay and decrease in throughput.

2 LITERATURE SURVEY

Yogeesh Seralathan, et.al (2018) Stated that all the devices in the internet of things are controlled and connected with the help of internet. Proper security is required on the IoT devices to protect the data from any malware attacks [9]. Large number of sensitive data is processed by the devices due to which the use of IoT devices increases widely. To large number of botnets, Malware like Mirai is widely used nowadays. This malware has been utilized in DDoS attacks as well in which every second up to 1.2 Terabytes of networks traffic is generated. They performed various experiments, to determine compromise done by an IoT device's in case of threat for the security and privacy of the data and they provide a case study of an IP camera. They also presented the importance of securing IoT and provide essential security practices for mitigating device exploitation.

Chalee Vorakulpipat, et.al (2018) presented the critical issue currently faced by the device's due large utilization of these devices [10]. The major issue faced currently is the issue of the network security in the devices. The use of devices nowadays increased drastically to access the corporate networks due to which they are prone to the major security risks. It is very necessary to use more services as most of the people are shifted from personal computers to mobile devices that lead to widely utilization of the IoT devices. Due to these devices it is easy to access more channels for the corporate information. The need of the IoT security changes according to market needs as services of the IoT devices changes from time to time. They presented a concern related to IoT security, reviews, and challenges faced by the devices as well as discussed, the three generations of the IoT security.

Jesus Pacheco, et.al (2017) presented a framework for the security of IoT for the integration of a Smart Water Systems in the IoT, in a secure way. There are four layers in this used framework such as devices, communication, service, and application layers. They also showed the procedure to use the threat model to protect or secure gateway which is the necessary part of the communication gateway [11]. The functionality of this method is based on the concept that it utilizes a profile that is developed to accurately and characterizes the normal operations of gateway. As per analysis, it is demonstrated that proposed approach of ABAIDS can detect both known and unknown attacks with high detection rates and low false positive alarms. They also have insignificant overhead in terms of memory and CPU usage. Proposed method protects the normal operation of the gateway to provide the availability.

Se-Ra Oh, et.al (2017) presented a connected, intelligent and context-aware device that works collectively known as internet of things (IoT). The IoT devices are growing quickly in the recent years as it provides the common functions of IoT devices that are helpful to all. Due to all these advantages, IoT platforms are diversified into various platforms such as oneM2M, FIWARE and IoTivity [12]. Security is the main consideration in the IoT devices as they are more vulnerable to attacks and directly affect the IoT device in the IoT platform. In the interworking process, they are more prone to critical influence in all connected IoT platforms. The security architecture of the oneM2M was discussed in this paper. Therefore, they developed an OAuth 2.0-basedoneM2M security component to provide authentication and authorization which is necessary for the security of IoT and for the protection of interworking between IoT platforms.

U. M. Mbanaso, et.al(2017) presented a novel configurable policy-based pacification and the threats and vulnerabilities faced by an IoT system were analyzed [13]. This specification has been utilized to scale proportionately in solving trust, confidentiality and privacy issues in Distributed environments. To solve all the issues in multiple domains, these devices work collectively, and smart entities have to more trusted, reliable and secure for the security and safety of end-to-end connectivity. A mechanism was proposed by author in this paper by which all the IoT entities can express their capabilities and requirements. For the negotiation of provable attributes and resources they constructed a fine-grained policy mutually. To solve the dispute resolution and auditable, they provide a mechanism which solve the issues such as trust, privacy and confidentiality in a unified manner. This method provides a greats success in the IoT environments.

Yiqun Zhang, et.al (2018) presented a major challenge for the IoT devices to support different cryptographic algorithms and standards within the physical constraints. Author proposed a Recryptor in this paper which is are configurable cryptographic processor which utilizes its computational capabilities to enhance the existing memory of a commercial general-purpose processor [14]. A 10-transistorbitcell supports, in-memory bit line computing for the support of different bitwise operations up to 512-bits wide. The high-throughput computing capabilities near memory are provided by the custom-designed shifter, rotator, and S-box modules as they are located near the memory. The programmability of the Recryptor's was demonstrated by implementing the cryptographic primitives of various public/secret key cryptographies and hash functions. 6.8% average speedup and 12.8% average energy was achieved by Recryptor running at 28.8 MHz in 0.7 V as compared to software- and hardware.

Ibrahim R. Waz, et.al (2017) showed various IoT security items and methods to achieve requirements of these items in a constrained environment, they provide security at every stage such as device, transmission of data, data on reset, service, and user [15]. To deal with different IoT platforms all the items are combined in one stack due to which integrity between the items establish. This integration assures the continuity of security from one stage to another stage. With the help of middleware to the user and vice versa, full data from the IoT device are traced.

Israr Ahmed1, et.al (2017) elaborated on the significant effects of Internet of Things in our day to day life. They are widely used in the homes, hospitals and installed outside to control and report all the changes occur in the environment. It is considered as the system which is widely used in the physical and, virtual world and with their associated things. The Internet of Things is widely utilized in almost every field such as billions of devices, individuals and administrations used this technology for the protection and extraction of the useful information. For the identification of each object, an ID is assigned as a proof [16]. In the upcoming years, all the devices will relate to a smart device which will be controlled by a handheld device for the processing. The major issue in the IoT is the protection and security as most of the IoT devices are battery operated and requires less power consumption. In the security and protection real concern of the IoT is the Identification, Authentication and device diversity. Therefore, all the critical issue, related to privacy and safety was discussed in this paper.

Zhen Ling, et.al (2017) showed an end-to-end view of IoT security and privacy in this paper which is the fundamental requirement of the any device. At first, they presented the end-to-end view of an IoT system to guide the risk assessment and design of an IoT system. In related to security and privacy, they evaluated the 10 basic IoT functionalities [17]. Based on this they presented, the requirement of the security and privacy in terms of IoT system, software, and networking and big data analytics in the cloud. Second, they presented a vulnerability analysis of the Edimax IP camera system, by utilizing end-to-end view of IoT security and privacy. This system is firstly represented in this paper which identifies various attacks by which all the cameras are controlled by the manufacturer. As per preformed experiments, it is demonstrated that it is necessary to raise alarms against the IoT manufacturers due to discovered attacks.

3 TABLE OF COMPARISON

Table 1: Comparison of contributions and techniques used by different authors to solve IoT security challenges.

Authors' Names	Year	Description	Outcomes
Yogeesh Seralathan,Tae (Tom) Oh, Suyash Jadhav, Jonathan Myers, Jaehoon	2018	To large number of botnets, Malware like Mirai is widely used nowadays. This malware has been utilized in DDoS attacks as well in which every second up to 1.2 Terabytes of	They performed various experiments, to determine compromise done by an IoT device's in case of threat for the security and privacy of the data

Jeong, Young Ho Kim, and Jeong Neyo Kim		networks traffic is generated.	and they provide a case study of an IP camera.
Chalee Vorakulpipat, Ekkachan Rattanalerdnusorn, Phithak Thaenkaew, Hoang Dang Hai	2018	The major issue faced currently is the issue of the network security in the devices. The use of devices nowadays increased drastically to access the corporate networks due to which they are prone to the major security risks.	They presented a concern related to IoT security, reviews, and challenges faced by the devices as well as discussed, the three generations of the IoT security.
Jesus Pacheco, Daniela Ibarra, Ashamsa Vijay, Salim Hariri	2017	They presented a methodology for the development of a threat model and this model has been utilized for the identification of the potential attacks against each layer, their effects of the devices and methods to mitigate and recover from these attacks.	Proposed method protects the normal operation of the gateway to provide the availability.
Se-Ra Oh, Young-Gab Kim	2017	In the interworking process, they are more prone to critical influence in all connected IoT platforms. The security architecture of the oneM2M was discussed in this paper.	They developed an OAuth 2.0-basedoneM2M security component to provide authentication and authorization which is necessary for the security of IoT and for the protection of interworking between IoT platforms.
U. M. Mbanaso, G. A. Chukwudebe	2017	A mechanism was proposed by author in this paper by which all the IoT entities can express their capabilities and requirements. For the negotiation of provable attributes and resources they constructed a fine-grained policy mutually.	To solve the dispute resolution and auditable, they provide a mechanism which solve the issues such as trust, privacy and confidentiality in a unified manner. This method provides a greats success in the IoT environments.
Yiqun Zhang, Li Xu, Qing Dong, Jingcheng Wang, David Blaauw, and Dennis Sylvester	2018	Author proposed a Recryptor in this paper which is are configurable cryptographic processor which utilizes its computational capabilities to enhance the existing memory of a commercial general-purpose processor	The programmability of the Recryptor's was demonstrated by implementing the cryptographic primitives of various public/secret key cryptographies and hash functions. 6.8% average speedup and 12.8% average energy was achieved by Recryptor running at 28.8 MHz in 0.7 V as compared to software- and hardware.

Ibrahim R. Waz,Mohamed Ali Sobh, Ayman M. Bahaa-Eldin	2017	To deal with different IoT platforms all the items are combined in one stack due to which integrity between the items establish.	This integration assures the continuity of security from one stage to another stage. With the help of middleware to the user and vice versa, full data from the IoT device are traced.
Israr Ahmed, Saleel A.P, Babak Beheshti, Zahoor Ali Khan, Imtiaz Ahmad	2017	In the security and protection real concern of the IoT is the Identification, Authentication and device diversity.	Therefore, all the critical issue, related to privacy and safety was discussed in this paper.
Zhen Ling, Kaizheng Liu, Yiling Xu, Yier Jin, Xinwen Fu	2017	At first, they presented the end-to-end view of an IoT system to guide the risk assessment and design of an IoT system. Second, they presented a vulnerability analysis of the Edimax IP camera system, by utilizing end-to-end view of IoT security and privacy.	As per preformed experiments, it is demonstrated that it is necessary to raise alarms against the IoT manufacturers due to discovered attacks.

4 CONCLUSIONS

In this paper, we have showed that IoT is a decentralized type of network were sensor nodes aggregate data to base the station. From the above study, the self-configuring nature of the network security in IoT, it is also shown that energy consumption is a major challenge. The various issues related to IoT is highlighted and the solutions from proposed techniques are described in the contributions by various researchers as reviewed. We have compared the contributions to give a better understanding of the present and the future of the IoT system.

REFERENCES

[1] Xinlie Wang, Jianqing Zhang, Eve. M. Schooler, “Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT”, Communications (ICC), 2014 IEEE International Conference, vol. 19, issue 3, pp. 56-88, 2014.

[2] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, “Internet of things (IoT): A vision, architectural elements, and future directions,” Elsevier Future Generation Computer System, Vol. 29, issue 4, pp. 23-66, 2013.

[3] Mohamed Abomhara and Geir M. Koien, “Security and Privacy in the Internet of Things : Current Status and Open Issues”, In Privacy and Security in Mobile Systems (PRISMS), pages 1–8. IEEE, vol. 7, issue 6, pp. 18-3, 2014.

[4] Ahmad W Atamli and Andrew Martin, “Threat-Based Security Analysis for the Internet of Things”, In Secure Internet of Things (SIoT), vol. 4, issue 1, pages 35–43, 2014.

[5] Luigi Atzori, Antonio Iera, and Giacomo Morabito, “The Internet of Things: A survey”, Computer Networks, vol. 8, issue 6, pp. 18-30, 2010.

[6] Sachin Babar, Parikshit Mahalle, Antonietta Stango, Neeli Prasad, and Ramjee Prasad, “Proposed security model and threat taxonomy for the Internet of Things (IoT)”, In International Conference on Network Security & Applications (CNSA), volume 89, pages 420–429. Springer Berlin Heidelberg, vol. 4, issue 1, pp. 25-30, 2010.

[7] Riccardo Bonetto, Nicola Bui, Vishwas Lakkundi, Alexis Olivereau, Alexandru Serbanati, and Michele Rossi, “Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples”, 2012

IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2012 - Digital Proceedings, vol. 11, issue 6, pp. 13-30, 2012.

[8] Jan Camenisch and Els Van Herreweghen, "Design and implementation of the idemix anonymous credential system

", In Vijayalakshmi Atluri, editor, Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002, pages 21–30. ACM, 2002.

[9] Yogeesh Seralathan, Tae (Tom) Oh , Suyash Jadhav, Jonathan Myers, Jaehoon (Paul) Jeong+, Young Ho Kim, and Jeong Neyo Kim, "IoT Security Vulnerability: A Case Study of a Web Camera", International Conference on Advanced Communications Technology(ICACT), IEEE, vol. 13, issue 9, pp. 16-30, 2018.

[10] Chalee Vorakulpipat, Ekkachan Rattanalerdnusorn, Phithak Thaenkaew, Hoang Dang Hai, "Recent Challenges, Trends, and Concerns Related to IoT Security: An Evolutionary Study", International Conference on Advanced CommunicationsTechnology(ICACT), vol. 7, issue 4, pp. 14-33, 2018.

[11] Jesus Pacheco, Daniela Ibarra, Ashamsa Vijay, Salim Hariri, "IoT Security Framework for Smart Water System", 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications, IEEE,vol. 9, issue 3, pp. 11-30, 2017.

[12] Se-Ra Oh, Young-Gab Kim, "Development of IoT Security Component for Interoperability", IEEE, vol. 12, issue 4, pp. 67-89, 2017.

[13] U. M. Mbanaso, G. A. Chukwudebe, "Requirement Analysis of IoT Security in Distributed Systems", 2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON), IEEE, vol. 5, issue 7, pp. 20-30, 2017.

[14] Yiqun Zhang, Li Xu, Qing Dong, Jingcheng Wang, David Blaauw, and Dennis Sylvester, "Recryptor: A Reconfigurable Cryptographic Cortex-M0 Processor With In-Memory and Near-Memory Computing for IoT Security", IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 9, issue 3, pp. 25-56, 2018.

[15] Ibrahim R. Waz, Mohamed Ali Sobh, Ayman M. Bahaa-Eldin, "Internet of Things (IoT) Security Platforms", IEEE, vol. 6, issue 4, pp. 5-19, 2017

[16] Israr Ahmed1, Saleel A.P2, Babak Beheshti3, Zahoor Ali Khan4, Imtiaz Ahmad, "Security in the Internet of Things (IoT)", The Fourth HCT INFORMATION TECHNOLOGY TRENDS (ITT 2017), Dubai, UAE, vol. 9, issue 5, pp. 9-30, 2017

[17] Zhen Ling, Kaizheng Liu, Yiling Xu, Yier Jin, Xinwen Fu, "An End-to-End View of IoT Security and Privacy", IEEE, vol. 7, issue 4, pp. 22-30, 2017