International Journal of Research in Advanced Engineering and Technology
ISSN: 2455-0876; Impact Factor: RJIF 5.44
Received: 25-11-2018; Accepted: 29-12-2018
www.newengineeringjournal.in
Volume 5; Issue 1; January 2019; Page No. 49-55

Simulink based RNN models to solve LPM

Belal Ayyoub[1], Ahmad Sharadqh[2], Ziad Alqadi[3], Jamil Al-azzeh[4]
[1, 3, 4] Department of Computer Engineering, Al Balqa'a Applied University, Amman, Jordan
[2] Department of Mechatronics Engineering, Al Balqa'a Applied University, Amman, Jordan

Abstract
Recurrent artificial neural network has shown great success because of its interesting feature, each neuron in this type of artificial neural acts a computational element with memory, providing a neuron with ability to store the calculated output for the next stage of computation.
In this paper we will introduce a methodology base on RNN features, this methodology will be used to build and run simulink based RNN models to solve linear programming problems, the introduced simulink models will be implemented in order to find the optimal values of LPM.

Keywords: ANN, RNN, LPM, constraint, objective function, optimal solution, linear AF, poslin AF

1. Introduction

1.1 Linear programming model

Problems-based on linear programming are defined as the problems of maximizing the profit (or minimizing the cost), which is a linear function subject to linear constraints [1, 2].
Linear programming model (LPM) is usually constructed by:
- The profit (cost) equation, which requires maximization (minimization).
- A set of constraints, each of them is restricted to a constant.

$$Z = c_1\, x_1 + c_2\, x_2 + \ldots\ldots\ldots\ldots\ldots + c_n\, x_n$$

subject to the constraints

$$a_{11}\, x_1 + a_{12}\, x_2 + \ldots\ldots\ldots\ldots + a_{1n}\, x_n \le b_1$$

$$a_{21}\, x_1 + a_{22}\, x_2 + \ldots\ldots\ldots\ldots + a_{2n}\, x_n \le b_2$$

....

$$a_{m1x1} + a_{m2x2} + \ldots\ldots\ldots\ldots + a_{mnxn} \le b_m$$

So the LP problem is to find the values for the variables $x_1, x_2, \ldots\ldots\ldots\ldots, x_n$ which maximize (minimize) the objective function [3, 4].
All the variables must meet the restriction:

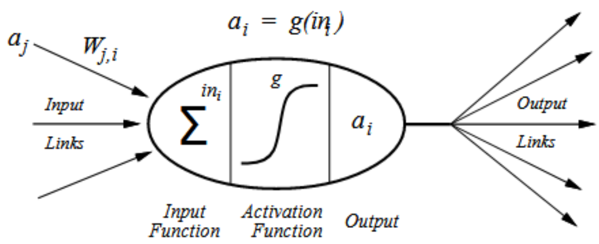
$$x_1,\ x_2,\ \ldots\ldots\ldots\ldots,\ x_n \ge 0$$

Here we have to notice the following:
1. A set of values $x_1, x_2, \ldots\ldots\ldots\ldots, x_n$ which satisfies the constraints is called its solution.
2. Any solution to a problem which satisfies the non-negativity restrictions of the problem is called its feasible solution.
3. Any feasible solution which maximizes (or minimizes) the objective function is called its optimal solution.

For any linear LP problem, we have to follow the following rules in order to build an LPM:
Step 1: Mark the unknowns in the given by x and y.
Step 2: Formulate the objective function.
Step 3: Translate all the constraints in the form of equalities.
Step 4: Solve these equalities simultaneously.
Step 5: Find the values of *x* and *y* for which the objective function $z = ax + by$ has maximum or minimum value (as the case may be).

1.2 Recurrent artificial neural network (RNN)

Artificial neural network (ANN) is a powerful computational model, it contains a set of neurons, these neurons are organized into layers, and the neurons in ANN are fully connected via weights [5, 6, 7].
Each neuron performs the summation of products as shown in figure 1, then according to the selected activation function for the layer computes the output of this neuron [8, 9, 10].



$$a_i = g\left(\sum_j W_{j,i} a_j\right)$$

Fig 1: Neuron operations

Here in these ANN structures the output of each neuron always depends on the inputs, the selected weights, and the selected activation function [11, 12].
In a traditional ANN we always assume that all inputs (and

outputs) are independent of each other [11, 12]. But for many applications such as prediction task and natural language processing task it will be a bad idea to used traditional ANN. RNNs are called *recurrent* because they perform the same task for every element of a sequence [13, 16] with the output being depended on the previous computations and the previous states. On another hand we can assume that RNN is a computational model (like ANN) but it has a "memory" which captures information about what has been calculated so far [14, 15].

A single neuron RNN computes the output and returns this output to the input as shown in figure 2:

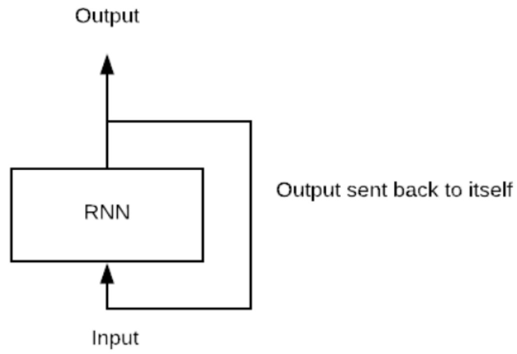


Fig 2: Single neuron RNN

A single neuron RNN can be adapted to simulink model as shown in figure 3:

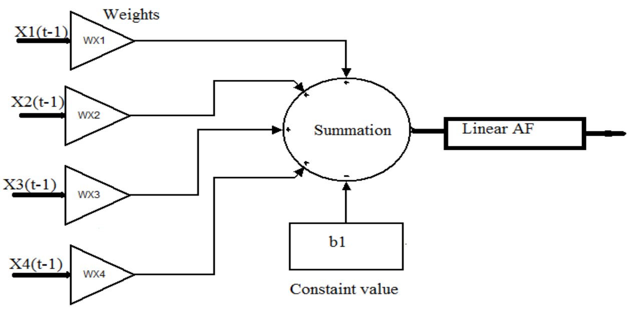


Fig 3: A single neuron representation in simulink

For the constraint: $5x1 + 4x2 + x3 + 0x4 \leqslant 60$ the neuron will be as shown in figure 4:

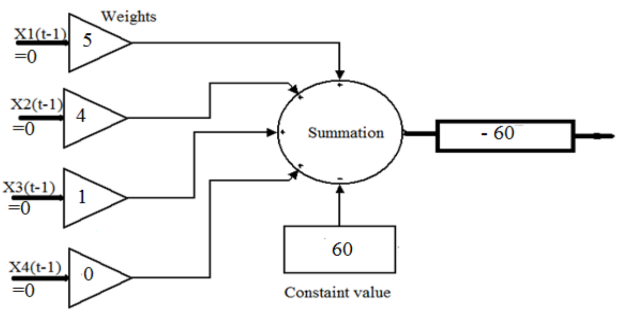


Fig 4: RNN neuron for a selected constraint

2. Proposed methodology to represent LPP using simulink model based on RNN

LPM is constructed by an objective function and a set of equations, one for each constraint, the objective function usually depends on some variables and here we must add some slack variables to solve the un equality in each constraint, one slack variable for each constraint.

The RNN architecture for any LPM will be as shown in figure 5:

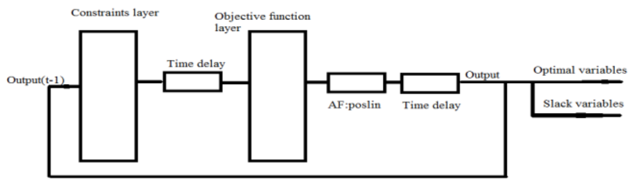


Fig 5: RNN for LPP

To represent LPP using simulink based on RNN we have to follow the following steps:

i) Creating constraints layer

For each constraint we have to create simulink neuron as shown in figure 3, the output of this neuron must be connected to a gain (model parameter (mp1)) with value less than 1, this parameter will control the model output to reach the optimal value, the calculated output must be passed to a delay time to save this value for next calculation. The calculated outputs for time (t) and time (t-1) must be summed to be used as an input to the second layer (objective function layer).

ii) Creating objective function layer

The objective function layer must contain number of neurons equal to the number of variables used in LPM, one neuron for each variable (including slack variables). The weights for each neuron are the parameters of each variable in each constraint and the constant is the parameter of the variable in the objective function.

The output of each neuron in this layer must be passed to poslin AF in order to omit the negative results and make them equal zeros. The poslined outputs must be passed to a gain(model parameter mp2) to control the outputs, then the calculated value must be passed to a time delay in order to save the previous outputs for summation, the outputs of this layer must give the optimal values for LPM variable and they must feed the constraint layer to enhance the values of the variables(when they are not optimal).

To make this methodology clear let us take the following LPM:

$$max y = 5x1$$

Subject to:

$$2x1 \leq 25$$

The optimal solution of this model is x1=12.5

This model contains 1 constraint:

$$2x1 + x2 - 25 = 0$$

Applying the methodology, we can reach the following simulink model (see figure 6):

Here we have 2 layers: the constraints layer contains 1 neuron and the objective function layer contains 2 neurons:

Let us select the model parameters (n1=0.1, m2=1, m2=1)

Running this model will give us the optimal solution for x1 after 31 time iterations as shown in figure 7, we have to notice that the value of x1 after using extra iterations time will remain the same (see figure 8)

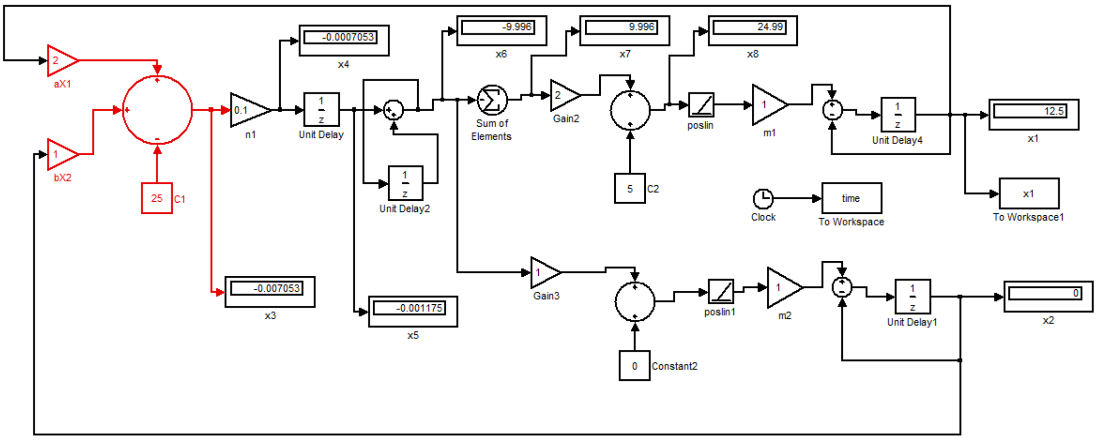


Fig 6: Model example 1

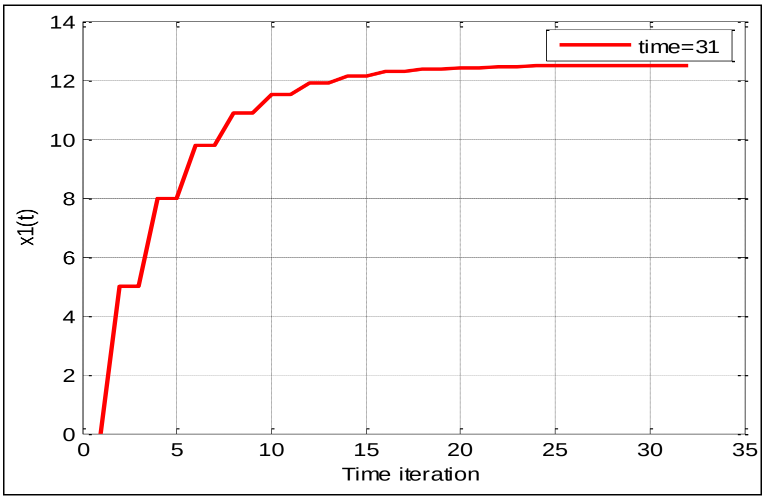


Fig 7: x1 after 31 time iterations

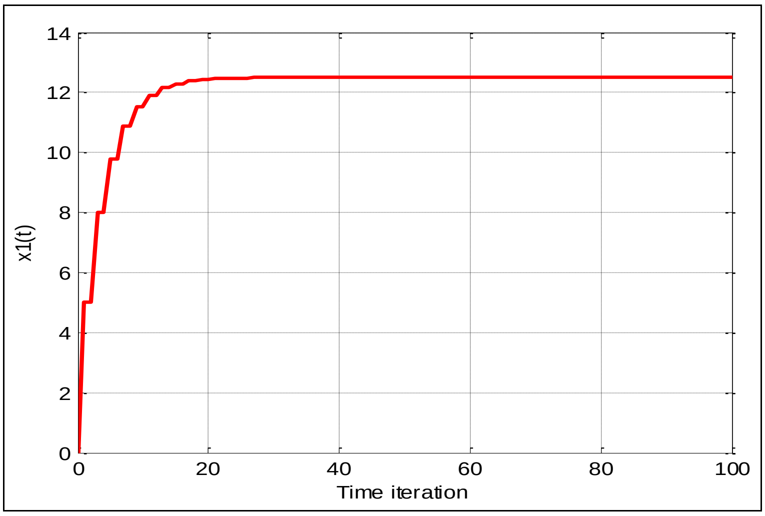


Fig 8: x1 after 100 time iterations

Here we have to professional in selecting model parameters (n and m) table 1 shows how the selected values of these parameters affect the value of x1 and number of time iteration:

Table 1: Model 1 implementation results

Model parameters			Number of time iterations	Calculated x1 value	Remarks
n1	m1	m2			
0.1	0.1	1	3900	12.49	Optimal not reached
0.1	0.1	1	3950	12.50	Optimal reached
0.1	0.1	1	3950	12.50	Optimal
0.5	0.5	1	2	12.50	Optimal reached
0.5	0.5	1	20	12.50	optimal
0.5	0.5	0.5	2	12.50	Optimal reached
0.01	0.01	0.5	35000	12.49	Optimal with error
0.01	0.01	0.5	36000	12.49	Optimal with error
0.6	0.1	1	55	12.49	Optimal with error
0.6	0.1	1	58	12.50	Optimal reached
0.6	0.1	1	100	12.50	optimal
1	1	1	2	15	No optimal
1	1	1	10	15	No optimal
0.1	0.01	1	3950	12.5	Optimal reached

3. Using the proposed methodology to implement real LPM

Now let us use the proposed methodology to build and run some RNN simulink models to solve linear programming problems:

Let us take the following LPM:

$$max y = 40x1 + 50x2$$

Subject to:

$$2x1 + x2 \leq 32$$

$$2x1 + 3x2 \leq 48$$

Here we have to write this model in a standard form by adding slack variables:

$$2x1 + x2 + x3 - 32 = 0$$

$$2x1 + 3x2 + 0x3 + x4 - 48 = 0$$

The RNN simulink model as shown in figure 9 will contain 2 layers:

- The constraint layer, which contains 2 neurons, 1 for each constraint.
- The objective function layer, which contains 4 neurons, one for each variable (including the slack variables).

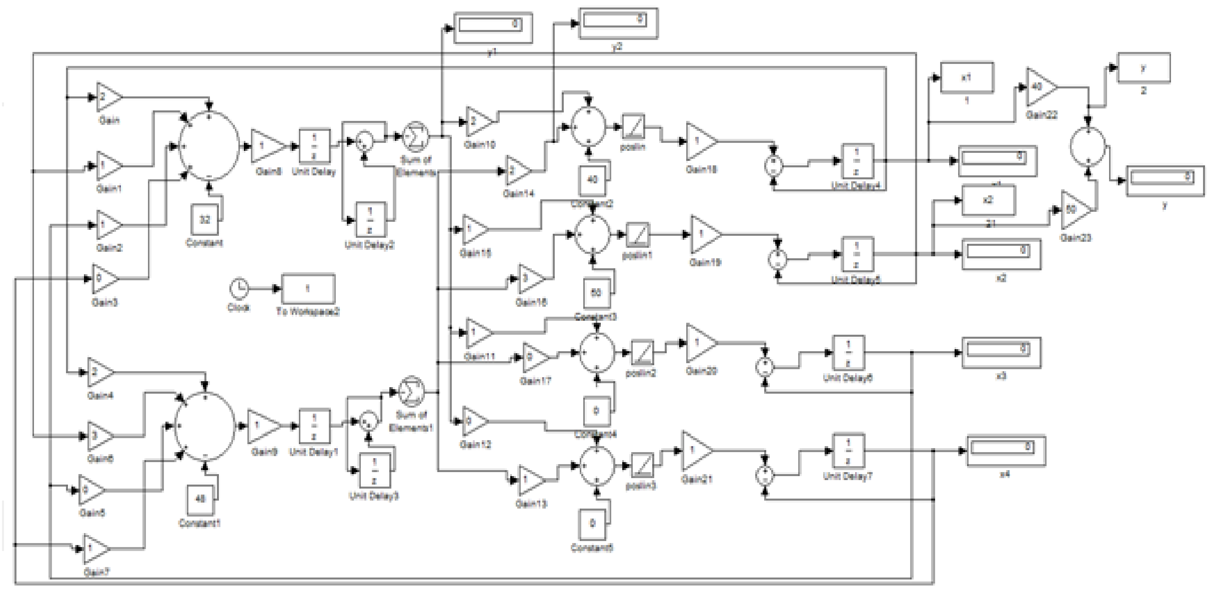


Fig 9: RNN simulink model for real LPM

This model was implemented for the model parameters: Gain8=0.015; Gain18=-0.07; Gain19=0.035; Gain20=0.02; Gain21=0.02 and with 6000 time iterations the output was very closed to optimal solution as shown in figure 10:

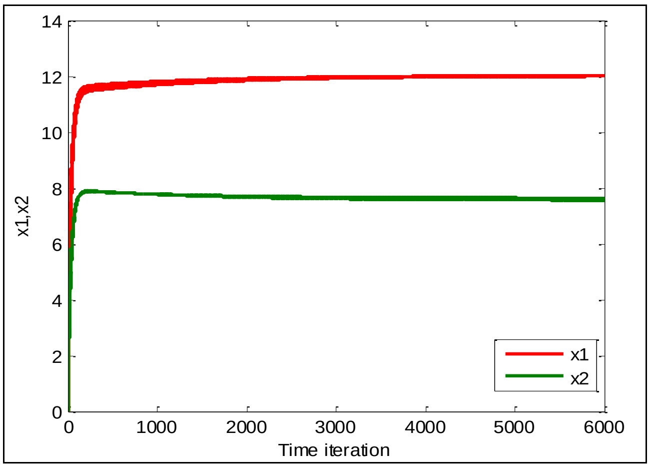


Fig 10: Simulation results of LPM1

Adjusting the model parameters for: Gain 8=0.015; Gain9=0.08; Gain18=-0.0825; Gain19=0.09325; Gain20=0.015; Gain21=0.05 and with 4000 time iterations the output was very closed to optimal solution as shown in figure 11:

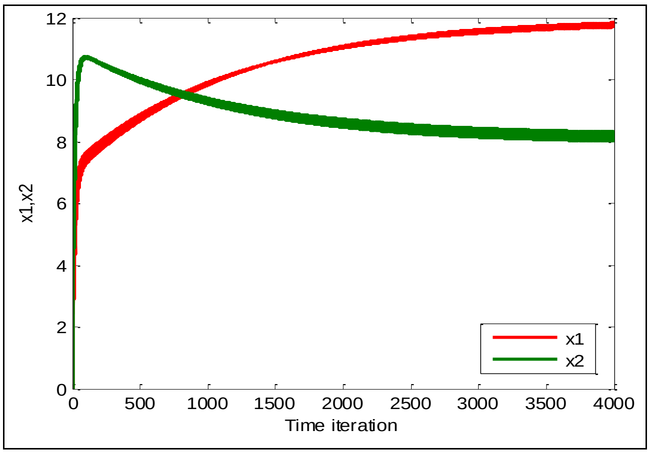


Fig 10: Simulation results of LPM1 after updating model parameters

The RNN simulink model shown in figure 9 suits any LPM with 2 constraints, the only thing needed is to adjust the weights in the constraint and objective function layers and to update the base constants connected to each neuron.

For LPM with 3 and constraint, the model shown in figure 9 must be updated as follows:

- Add an extra neuron in the constraint layer for each extra constraint.
- Add one neuron in the objective function layer for each extra variable (including the slack variables).

The above explained model was adjusted and implemented for various LPMs with 2 and 3 constraints; the obtained solutions were equal the optimal solution or very closed to them (with error less than 1.4%, but selecting the appropriate values for the model parameter will lead the error to zero).

4. Conclusion

The feature of RNN as a computing element with memory was used to build and run simulink models.

A methodology of creating simulink based RNN models to be used for LPM solving was introduced. Some LPM were selected, simulink models were built and tested, the experimental results showed that constructed simulink model using the proposed methodology was accurate and always gave the optimal solution of LPM.

The simulink models are very simple to construct, and the constructed model can be easily updated to suit any LPM with any number of constrains, and number of variables and any objective function.

5. References

1. Ziad A, Alqadi, Majed O, Al-Dwairi, Amjad A, Abu Jazar, Rushdi Abu Zneit. Optimized True-RGB color Image Processing, World Applied Sciences Journal, 8(10):1175-1182, ISSN 1818-4952, 2010.
2. Moustafa AA, Alqadi ZA. Color Image Reconstruction Using A New R'G'I Model, Journal of Computer Science. 2009; 5(4):250-254.
3. Matrouk K, Al-Hasanat A, Alasha'ary H, Al-Qadi Z, Al-Shalabi H. Speech fingerprint to identify isolated word person, World Applied Sciences Journal. 2014; 31(10):1767-1771.
4. Al-Azzeh J, Zahran B, Alqadi Z, Ayyoub B, Abu-Zaher M. A Novel zero-error method to create a secret tag for an image, Journal of Theoretical and Applied Information Technology. 2018; 96(13):4081-4091.
5. Ziad Alqadi AA, Mohammed Abu Zalata K, Ghazi Qaryouti M. Comparative Analysis of Color Image Steganography, JCSMC. 2016; 5(11):37-43.
6. Jose M. Hiding Image in Image Using LSB Insertion Method with Improved Security and Quality, International Journal of Science and Research. 2014; 3(9):2281-2284.
7. Patel RM, Shah DJ. Conceal gram: Digital image in image using LSB insertion method, International Journal of Electronics and Communication Engineering & Technology. 2013; 4(1):230-2035.
8. Akhtar N, Johri P, Khan S. Enhancing the security and quality of LSB based image steganography, 5th International Conference on Computational Intelligence and Communication Networks, Mathura, India, 2013.
9. Juneja M, Sandhu PS. An improved LSB based Steganography with enhanced Security and Embedding/Extraction, 3rd International Conference on Intelligent Computational Systems, Hong Kong China, 2013.
10. Alasha'ary H, Matrouk K, Al-Hasanat A, A lqadi Z, Al-Shalabi H. Improving Matrix Multiplication Using Parallel Computing, International Journal on Information Technology (I.RE.I.T.), 2013, 1(6). ISSN 2281-2911
11. Bilal Zahran, Ziad Alqadi, Jihad Nader, Ashraf Abu Ein A Comparison Between Parallel and Segmentation Methods Used for Image Encryption-Decryption, International Journal of Computer Science & Information Technology (IJCSIT), 8(5):201.
12. Alqadi ZA, Abu-Jazzar A. Analysis of Program Methods Used For Optimizing Matrix Multiplication, Journal of Engineering. 2005; 15(1):73-7.
13. Alqadi ZA, Aqel M, El Emary IM. Performance Analysis and Evaluation of Parallel Matrix Multiplication Algorithms. World Applied Sciences Journal. 5(2), ISSN 1818-4952, 2008
14. Alasha'ary H, Matrouk K, Al-Hasanat A, Alqadi Z, Al-Shalabi H. Improving Matrix Multiplication Using

Parallel Computing, International Journal on Information Technology (I.RE. I.T.). 2013; 1:6. ISSN 2281-2911
15. Ziad Al-Qadi, Musbah Aqel. Performance analysis of parallel matrix multiplication algorithms used in image processing, World Applied Sciences Journal. 2009; 6(1):45-52.
16. Jamil Al-Azzeh, Bilal Zahran, Ziad Alqad, Salt Pepper Noise. Effects and Removal International Journal on Electrical Engineering and Informatics, 2(4).
17. Vimala T. Salt and Pepper Noise Reduction Using Mdbutm Filter With Fuzzy Based Refinement. 2012; 2:5.
18. Jassim FA. Image Denoising Using Interquartile Range Filter with Local Averaging, International Journal of Soft Computing and Engineering (IJSCE). 2013; 2(6):424-428.
19. Jamil S, AL-Azzeh: Distributed Mutual Inter-Unit Test Method for D-Dimensional Mesh-Connected Multiprocessors with Round-Robin Collision Resolution: Jordanian Journal of Computers and Information Technology, 2019.
20. Jamil AL-Azzeh, Bilal Zahran, Ziad Alqadi, Belal Ayyoub, Muhammed Mesleh: A Novel Based On Image Blocking Method To Encrypt-Decrypt Color JOIV: International Journal on Informatics Visualization, 2019.
21. Jamil AL-Azzeh, Evaluation Method for SDN Network Effectiveness in Next Generation Cellular Networks: International Journal of Communication Networks and Information Security, 2018.
22. Jamil AL-Azzeh: Improved testability method for mesh-connected VLSI multiprocessors: Jordanian Journal of Computers and Information Technology, 2018.
23. Jamil AL-Azzeh: A Novel Zero-Error Method to Create a Secret Tag for an Image; Journal of Theoretical and Applied Information Technology, 2018.
24. Jamil AL-Azzeh, Qualitative risk analysis of software development; Asian Journal of Information Technology, 2018.
25. Bilal Zahran, Jamil AL-Azzeh, A Modified LBP Method To Extract Features From Color Images: Journal of Theoretical and Applied Information Technology, 2018.
26. Jamil AL-Azzeh, Information Technologies for Supporting Administrative Activities of Large Organizations; DESIDOC Journal of Library & Information Technology, 2018, 38(3).
27. Jamil AL-Azzeh: A Distributed Multiplexed Mutual Inter-Unit in-Operation Test Method for Mesh-Connected VLSI Multiprocessors; Jordan Journal of Electrical Engineering, 2017 10:5.
28. Jamil AL-Azzeh: Fault-Tolerant Routing in Mesh-Connected Multicomputer based on Majority-Operator-Produced Transfer Direction Identifiers; Jordan Journal of Electrical Engineering, 2017, 3(2).
29. Jamil AL-Azzeh, Analysis of Self-Similar Traffic Models in Computer Networks; International Review on Modelling and Simulations. 2017: 10:5.
30. Jamil AL-Azzeh. Statistical Analysis of Methods Used to enhanced Color Image Histogram; XX International Scientific and Technical Conference; Russia, 2017.
31. Mazen Abuzaher, Jamil AL-Azzeh. JPEG Based Compression Algorithm; International Journal of Engineering and Applied Sciences, 2017, 4(4).
32. Jamil AL-Azzeh, ZigBee, Bluetooth and Wi-Fi Complex Wireless Networks Performance Increasing; International Journal on communications antenna and propagation. 2017; 7:1.
33. jamil al-azzeh, implementing built-in test in analog and mixed-signal embedded-core-based system-on-chips; asian journal of Information Technology, Medwell Journals, 2016.
34. Jamil AL-Azzeh, Creating a Color Map to be used to Convert a Gray Image to Color Image; International Journal of Computer Applications (0975–8887). 2016; 153:2.
35. Jamil AL-Azzeh: Analysis of Second Order Differential Equation Coefficients Effects on PID Parameters International Journal on Numerical and Analytical Methods in Engineering (IRENA). 2016; 4:2.
36. Jamil AL-Azzeh; Automated Demodulation of Amplitude Modulated Multichannel Signals with Unknown Parameters Using 3D Spectrum Representation Research Journal of Applied Sciences, Engineering and Technology, Maxwell Scientific Publication, 2016, 05.
37. Jamil S Al-Azzeh, Adaptive Regulation of Radiated Power Radio Transmitting Devices in Modern Cellular Network Depending on Climatic Conditions. Contemporary Engineering Sciences, 2016, 9.
38. Mazin Al Hadidi, Jamil AL-Azzeh, Methods of Risk Assessment for Information Security Management, International Review on Computers and Software (I.RE.CO.S.), 11:2. ISSN 1828-6003 (impact factor = 6.14)**.** February 2016.
39. Jamil AL-Azzeh, Bidirectional Virtual Bit-slice Synchronizer: A Scalable Solution for Hardware-level Barrier Synchronization. Research Journal of Applied Sciences, Engineering and Technology, Maxwell Scientific Publication Corp. 2015; 11(8):902-909.
40. Jamil AL-Azzeh. The Organization of Built-in Hardware-Level Mutual Self-Test in Mesh-Connected VLSI Multiprocessors; International Journal on Information Technology (I.RE.I.T.), Praise Worthy Prize, 2015, 3.
41. Jamil AL-Azzeh; an approach to achieving increased fault-tolerance and availability of multiprocessor-based computer systems. Australian Journal of Basic and Applied Sciences, 2014.
42. Jamil AL-Azzeh, Computer simulation of vibration robot created for the wall movement; Research Journal of Applied Sciences. 2014; 9:597-602,
43. Jamil AL-Azzeh. Review of Methods of Distributed Barrier Synchronization of Parallel Processes in Matrix VLSI Systems, International Review on Computers and Software (IRECOS), Praise Worthy Prize, Part A. 2013; 8(4):42- 46.
44. Jamil AL-Azzeh, Fastest Color Model for Image Processing Using Embedded Systems Australian Journal Of Basic And Applied Sciences. 2013; 7(14):83-89. 7.
45. Jamil AL-Azzeh, Using Virtual Network to Solve Freight Company Problems; World Applied Sciences Journal. 2013; 27(6):754-758. (SJR indicator = 0.17)
46. Jamil AL-Azzeh, Detection of eyes using FCM, International Review on Computers and Software

(IRECOS), Praise Worthy Prize, Part A. 2012; 7(4):1428-1434. (impact factor = 6.14).
47. Jamil AL-Azzeh. An optical character recognition, Contemporary Engineering Sciences. 2012; 5(11):521-529.
48. Jamil AL-Azzeh, Analysis of Matrix Multiplication Computational Methods European Journal of Scientific Research. 2014; 121(3):258-266.
49. Ziad A. Al-Qadi, Musbah J. Aqel Performance analysis of parallel matrix multiplication algorithms used in image processing: World Applied Sciences Journal. 2009; 6(1):45-52.
50. Mohammed Abuzalata. Ziad Alqadi, Jamil Al-Azzeh; Qazem Jaber Modified Inverse LSB Method for Highly Secure Message Hiding. International Journal of Computer Science and Mobile Computing. 2019; 8(2):93-103
51. Qazem Jaber, Rashad J. Rasras, Mohammed Abuzalata, Ziad Alqadi, Jamil Al-Azzeh; Comparative Analysis of Color Image Encryption-Decryption Methods Based on Matrix Manipulation: International Journal of Computer Science and Mobile Computing. 2019/3; 8:2.
52. Jamil Al-Azzeh, Ziad Alqadi, Mohammed Abuzalata Performance Analysis of Artificial Neural Networks used for Color Image Recognition and Retrieving: International Journal of Computer Science and Mobile Computing. 2019; 8:2.
53. Rashad J. Rasras, Mohammed Abuzalata, Ziad Alqadi, Jamil Al-Azzeh, Qazem Jaber, Comparative Analysis of Color Image Encryption-Decryption Methods Based on Matrix Manipulation International Journal of Computer Science and Mobile Computing. 2019; 8(3):14-26.
54. Jamil Al-Azzeh, Bilal Zahran, Ziad Alqadi, Belal Ayyoub, Muhammed Mesleh, A Novel Based on Image Blocking Method to Encrypt-Decrypt Color; International Journal on informatics visualization. 2019; 3:1.
55. Jamil Al-Azzeh, Zahran B, Alqadi Z, Ayyoub B, Mesleh M. A Novel Based On Image Blocking Method To Encrypt-Decrypt Color, International Journal on Informatics Visualization. 2019; 3:1.
56. Zahran B, AL-Azzeh J, Al Qadi Z, Al Zoghoul M, Khawatreh, A Modified LBP Method to Extract Features from Color Images, Journal of Theoretical and Applied Information Technology(JATIT). 2018; 96:10.
57. AL-Azzeh J, Zahran B, Alqadi Z, Ayyoub B, ABU-ZAHER M. A novel Zero-error Method to Create a Secret Tag for an Image, Journal of Theoretical and Applied Information Technology (JATIT). 2018; 96(13):4081-4091.
58. Jamil AL-Azzeh, Zahran B, Alqadi Z. Salt and Pepper Noise: Effects and Removal", International Journal on Informatics Visualization. 2018; 2(4):252-256.
59. Jihad Nader, Ziad Alqadi, Bilal Zahran. Analysis of Color Image Filtering Methods, International Journal of Computer Applications (IJCA). 2017; 174(8):12-17.
60. Ziad Alqadi, Bilal Zahran, Jihad Nader, Estimation and Tuning of FIR Low pass Digital Filter Parameters, International Journal of Advanced Research in Computer Science and Software Engineering. 2017; 7(2):18-23.