INTERNATIONAL JOURNAL

of

COMPUTERS, COMMUNICATIONS & CONTROL

ISSN 1841-9836

ISSN-L 1841-9836

A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2014 Volume: 9 Issue: 2 (April)

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).

Agora University Editing House

http://univagora.ro/jour/index.php/ijccc/

International Journal of Computers, Communications & Control

EDITORIAL ADDRESS:
Agora University/ R&D Agora Ltd. / S.C. Cercetare Dezvoltare Agora S.R.L.
Piaţa Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel./ Fax: +40 359101032
E-mail: ijccc@univagora.ro, rd.agora@univagora.ro, ccc.journal@gmail.com
Journal website: http://univagora.ro/jour/index.php/ijccc/

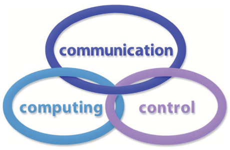
International Journal of Computers, Communications & Control

DATA FOR SUBSCRIBERS
Supplier: Cercetare Dezvoltare Agora Srl (Research & Development Agora Ltd.)
Fiscal code: 24747462
Headquarter: Oradea, Piata Tineretului Nr.8, Bihor, Romania, Zip code 410526
Bank: MILLENNIUM BANK, Bank address: Piata Unirii, str. Primariei, 2, Oradea, Romania
IBAN Account for EURO: RO73MILB0000000000932235
SWIFT CODE (eq.BIC): MILBROBU

International Journal of Computers, Communications & Control

Short Description of IJCCC

Title of journal: International Journal of Computers, Communications & Control
Acronym: IJCCC
Abbreviated Journal Title: INT J COMPUT COMMUN
International Standard Serial Number: ISSN 1841-9836, ISSN-L 1841-9836
Publisher: CCC Publications - Agora University
Starting year of IJCCC: 2006
Founders of IJCCC: Ioan Dzitac, Florin Gheorghe Filip and Mişu-Jan Manolescu
Logo:



Publication frequency: Bimonthly: Issue 1 (February); Issue 2 (April); Issue 3 (June); Issue 4 (August); Issue 5 (October); Issue 6 (December).

Coverage:

- Beginning with Vol. 1 (2006), Supplementary issue: S, IJCCC is covered by Thomson Reuters - SCI Expanded and is indexed in ISI Web of Science.
- Journal Citation Reports(JCR)/Science Edition:
  - Impact factor (IF): JCR2009, IF=0.373; JCR2010, IF=0.650; JCR2011, IF=0.438; JCR2012, IF=0.441.
- Beginning with Vol. 2 (2007), No.1, IJCCC is covered in EBSCO.
- Beginning with Vol. 3 (2008), No.1, IJCCC, is covered in Scopus.

Scope: International Journal of Computers Communications & Control is directed to the international communities of scientific researchers in computer and control from the universities, research units and industry.

To differentiate from other similar journals, the editorial policy of IJCCC encourages the submission of scientific papers that focus on the integration of the 3 "C" (Computing, Communication, Control).

In particular the following topics are expected to be addressed by authors:

- Integrated solutions in computer-based control and communications;
- Computational intelligence methods (with particular emphasis on fuzzy logic-based methods, ANN, evolutionary computing, collective/swarm intelligence);
- Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies).

Contents

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):131-138, April, 2014.

Input Projection Algorithms Influence in Prediction and Optimization of QoS Accuracy

R.D. Albu, I. Dzitac, F. Popentiu-Vladicescu, I.M. Naghiu

Răzvan-Daniel Albu*,
Florin Popentiu-Vlădicescu,**
Iuliana Maria Naghiu
1. University of Oradea,
Universitatii St., 1, 410610 Oradea, Romania
*Corresponding author: ralbu@uoradea.ro
**2. Academy of Scientist in Romania,
54, Splaiul Independentei, 0590094, Bucharest
E-mail: popentiu@imm.dtu.dk.

Ioan Dzitac
1. Aurel Vlaicu University of Arad
Elena Dragoi St., 2, 310330 Arad, Romania
ioan.dzitac@uav.ro
2. Agora University of Oradea
Piata Tineretului 8, 410526 Oradea, Romania
E-mail: idzitac@univagora.ro

Abstract: Regardless of new achievements in the research of prediction models, QoS is still a great issue for high quality web services and remains one of the key subjects that need to be studied. We believe that QoS should not only be measured, but have to be predicted in development and implementation phases. In this paper we assess how different input projection algorithms influence the prediction accuracy of a Multi-Layer Perceptron (MLP) trained with large datasets of web services QoS values.

Keywords: Quality of Service (QoS), adaptive models, web services, large/big data.

1 Introduction

The major web services QoS requirements are: availability, accessibility, integrity, performance, regulatory reliability and security. Models of prediction and/or optimization of QoS in web services are presented in many actual works: [19], [14], [16], [15], [20], [21], etc.

This research work continues the investigation on web services QoS criteria prediction and completes the results obtained in [1]- [5], works (co)authored by first author of this paper. We build in this work an adaptive model that offers good prediction results of QoS in web services. Since more data may lead to more accurate analyses and more precise analyses may lead to more confident decision making, the development of accurate and adaptive prediction models is both challenging and essential. Enhanced conclusions can cause superior operational efficiencies, cost reductions and lower risks.

Scott Zucker, Vice President of Business Services at Family Dollar, said: *"Small data is gone. Data is just going to get bigger and bigger and bigger, and people just have to think differently about how they manage it"* [8]. *Large data* is an expression used to define the exponential growth of data availability and size. Several recent technology improvements, that allow organizations to make the most of big data analytics, are: affordable bigger storage, faster and parallel processors, open source platforms, clustering, virtualization, grid computing, increased throughput and last, but not least, Cloud computing.

Another new term for the quantity of information generated by business, government, and science is: data deluge [10]. For instance, in 2010, the Large Hadron Collider (LHC) facility at CERN delivered 13 petabytes of data. Concurrently to this important growth, data are also becoming strongly interconnected. For example, Facebook is approximately fully connected. Presently, social networks interconnect people or groups who share similar interests, but soon, we expect they will also link software modules such as: Web-based services, or workflows. In recent happenings interrelated with the 2013 Boston Marathon violence, social networks of marathon competitors and high-performance computational systems were combined to group and analyze huge collections of photos and videos, finally leading to the identification of the terrorists. *Big data* is typically characterized by the "three V's": volume, velocity, and variety. In terms of volume, at the end of 2011, Facebook had 721 million individuals and 68.7 billion friendship edges [7]. In terms of velocity, Twitter generates 7 Tbytes of data daily, while Facebook produces 10 Tbytes. On 11 November 2012, a sales event at TaoBao, the largest online shopping marketplace in China, generated 100 million transactions and reached a peak transaction rate of 205,000 per minute [9]. In terms of variety, data today come from various sources, ranging from surveillance videos, to satellite images, to mobile tweets, to sensors and meters in the power grid [18]. A key difference between big data and large data is the rate at which the data can be collected and made accessible for analysis. Large data can also be handled by the traditional reporting and analysis tools. Big data is generally used to describe the massive amount of unstructured data, which costs a lot of time and money for analysis. However, large data may not have such special meaning, they just refer to the volume. We don't think there is any value in defining a threshold for what constitutes "big data." and what means "large data". Therefore, a flexible definition we use is "Big data is data that's an order of magnitude bigger than you're used to". The big data analysis affects the QoS because, if implemented efficiently, the big data infrastructure will permit carriers to preserve constantly optimized services and to set apart in the marketplace by offering QoS reports, diagnostic tools and other decision-support front-ends.

Cloud computing is a technology that exploded in the recent years and seems to be the ideal way to provide big and large data for mainstream uses, because it offers scale-out and on-demand computing resources in a pay-per-use style. For instance, Netflix stores movies and TV shows, while Dropbox saves clients' documents, both in Amazon's Simple Storage Service. In recent years, scale-out data stores, usually mentioned as NoSQL systems, were quickly gaining admiration as a possible solution for applications scaled at Internet level. These stores consist of technologies like: Amazon's DynamoDB, Google's BigTable and Yahoo's PNUTS. To address the "Big and Large data" challenge, NoSQL supporters limit ACID restraints, deliver completely scalable solutions and then gradually add back the Relational Database Management System (RDBMS) features like index or transaction support. A newly appeared paradigm named stream computing facilitates continuous queries over streaming data like social media feeds. Social networking on the cloud could empower sharing based on the social relationship between users. This would possibly make available technologies like volunteer computing. This is a distributed computing model in which associated users donate computing resources to a project. Two examples of volunteer computing are the following projects: Storage@home14 and Boinc15 [18]. In these scenarios, the resources are owned by individuals and they are shared in return for access to other resources. This could hypothetically transform the cloud's economics and raises doubts about the reliability and QoS warranties [11] [13].

2 Input Projection and Optimization Algorithms

Input Projection is the procedure that further reduces input dimensions by automatically mapping multiple pieces of information to single inputs. The main goal of input optimization is to make the network learn faster in the same time offering the same prediction performances or better. Input Optimization automatically determines the most informative inputs through genetic algorithms, greedy search, back-elimination and other techniques [6]. The four input projection algorithms we have investigated in this study are: PCA (Principal Component Analysis), MDS (Multi-dimensional Scaling), SOM (Self Organized Map) and LLE (Locally Linear Embed). Next, we will briefly describe each input projection algorithm.

Principal component analysis (PCA) also named Karhunen-Loeve transform of Singular Value Decomposition (SVD) finds an orthogonal set of directions in the input space and delivers a method of finding the projections into these directions in a well-ordered style. The first principal component is the one that has the largest projection (the shadow of our data cluster in each direction). The orthogonal directions are named the eigenvectors of the correlation matrix of the input vector, and the projections - the equivalent eigenvalues. Because PCA orders the projections, we can decrease the dimensionality by truncating the projections to a given order. The reconstruction error is equal to the sum of the projections left out. The features in the linear projection space become the eigenvalues. PCA networks are typically utilized for data compression, offering the best m linear features, but they can also be used for data reduction in conjunction with multilayer perceptron classifiers [17].

The SOM algorithm is as follows [12]:

a. Initialize the weights with small different random values for symmetry breaking;

b. For each input data find the winning PE using a minimum distance rule;

$$\bar{i}(x) = arg_j min \parallel \bar{x}(n) - w_j \parallel \tag{1}$$

c. For the winning PE, update its weights and those in its neighborhood by:

$$w_j(n+1) = w_j + \eta(n)[x(n) - w_j(n)] \tag{2}$$

Where $\eta(n)$ is the step size. In the beginning, the step size should be large, but decrease progressively to zero, according to:

$$\eta(n) = \frac{1}{a_\eta + b_\eta n} \tag{3}$$

Where a_η and b_η are problem dependent constants.

The purpose of these adaptive constants is to guarantee, in the early stages of learning, malleability and the formation of local neighborhoods as well as, in the later stages of learning, constancy and adjustment of the map. These problems are very difficult to study theoretically, so heuristics have to be involved in the designation of these values.

Multi-Dimensional Scaling (MDS) is a set of related statistical techniques used in information visualization for exploring similarities or dissimilarities in data. The MDS algorithm starts with a matrix of item-item similarities and then assigns a location to each item in a lower dimensional space say m, such that the original similarities or dissimilarities are represented by the relative position in the lower dimensional space. The implemented algorithm measures the similarity or distance between points by correlation coefficient and applies simulated annealing to find the mapping from n dimensional space to m dimensional space, where $n>=m$, such that the distances between points in the m dimensional space approximate the correlation coefficients between points in the n dimensional space [12].

Locally Linear Embedding (LLE) is an unsupervised learning algorithm that computes low dimensional, neighborhood preserving embedding of high dimensional data. LLE attempts to discover nonlinear structure in high dimensional data by exploiting the local symmetries of linear reconstructions. The LLE algorithm includes following three major steps [12]:

1. Based on desired number of inputs say m, get the neighbors of each point in the sense Euclidean distance.

2. Compute the weights W_{ij} that best reconstruct each data point X_i with dimension n from its k neighbors, minimizing the cost defined as:

$$\zeta(W) = \sum_{i=1}^{n} |X_i - \sum_{j=1}^{k} W_{ij} X_j| \tag{4}$$

The solution can be reached by first solving the linear system of equations involving local covariance matrix and then rescaling the weights so that they sum to one.

3. Compute the projected inputs Y_i with dimension m that is best reconstructed by the weights W_{ij} minimizing the quadratic form defined as:

$$\zeta(W) = \sum_{i=1}^{n} |Y_i - \sum_{j=1}^{k} W_{ij} Y_j| \tag{5}$$

This is found by extracting the appropriate bottom m eigenvectors of a Matrix derived from the cost function.

3 Experiments and Results

The adaptive models compared in this study have a name that respects the following syntax:
Topology_ name - number_ of_ hidden_ layers - learning_ rule - input projection_ algorithm

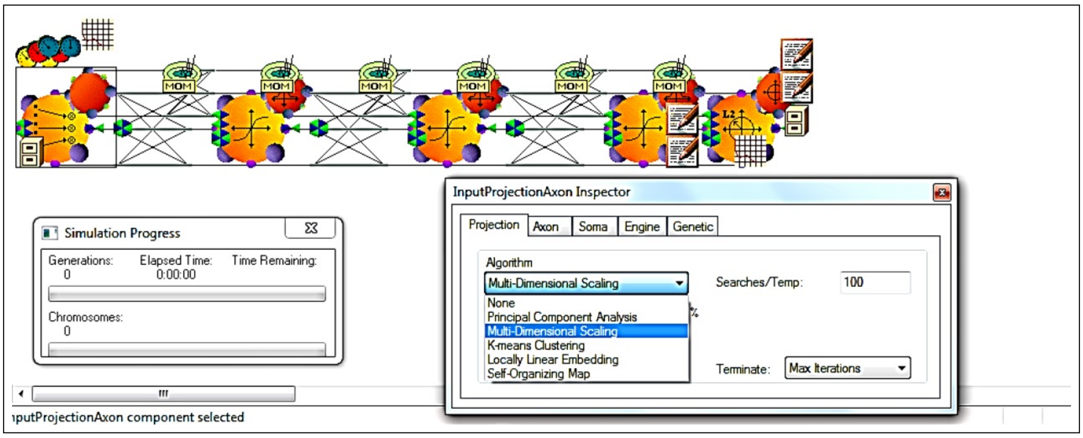
For example, the adaptive model named: MLP-2-CG-SOM is a Multi-Layer Perceptron with two hidden layers, trained using Conjugate Gradient learning rule and Self Organizing Map Input Projection algorithm.

In this study, in order to train the MLP, we made use of two large datasets: RTMATRIX and FPMATRIX. Each matrix has a size of 339 lines x 5825 columns. RTMATRIX consists of web services response time values while FPMATRIX stores similar web services throughput values. In our previous research works [1]- [5] we concluded that MLP-2-M and MLP-2-CG offered the most accurate prediction results for RTMATRIX and FPMATRIX, respectively. FPMATRIX and RTMATRIX were built by Z. Zheng, Y. Zhang and M.R. Lyu in [20] and [21].

To implement, train and test the adaptive models we utilized Neuro Solutions 6.21 development environment. As we can see in Figure 1, the main difference of this Neuro Solution implementation of MLP is the presence of InputProjectionAxon.

The InputProjectionAxon will apply either linear or non-linear transformation to convert n input data set into m input data set, where $n>=m$. The InputProjectionAxon Inspector allows us to select a different input projection algorithm and to set its parameters [12].

In this study, we have tested PCA in conjunction with MLP-2-M, but it works only on a subset of RTMATRIX/FPMATRIX since the separability of the classes is not always guaranteed. Another problem with linear PCA networks is outlying data points. Outliers will distort the estimation of the eigenvectors and create skewed data projections. RTMATRIX and FPMATRIX, having a huge number of values, have also a lot of outliers. Nonlinear networks are better able to handle this case. The importance of PCA analysis is that the number of inputs for the MLP classifier can be reduced a lot, which definitely influences the number of necessary training samples and the training time. And this was achieved when we made use of just a subset of RTMATRIX/FPMATRIX.



Simulation Progress

Generations: 0

Elapsed Time: 0:00:00

Time Remaining:

Chromosomes: 0

InputProjectionAxon Inspector

Algorithm: Multi-Dimensional Scaling

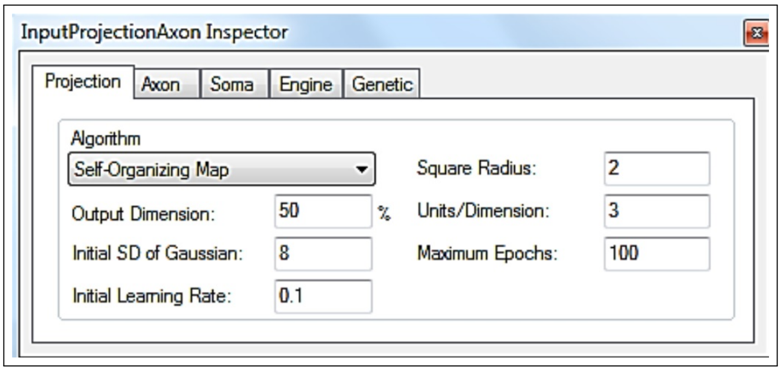
Searches/Temp: 100

Terminate: Max Iterations

ıputProjectionAxon component selected

Figure 1: MLP-2-M with InputProjectionAxon

SOM are an alternative to the PCA concept, and in our experiments, it works on both the entire RTMATRIX/FPMATRIX and a subset of them. By "works" we mean Neuro Solutions offer a results report. The parameters of SOM utilized in our simulation are presented in Figure 2.



InputProjectionAxon Inspector

Algorithm: Self-Organizing Map

Output Dimension: 50 %

Initial SD of Gaussian: 8

Initial Learning Rate: 0.1

Square Radius: 2

Units/Dimension: 3

Maximum Epochs: 100

Figure 2: SOM parameters

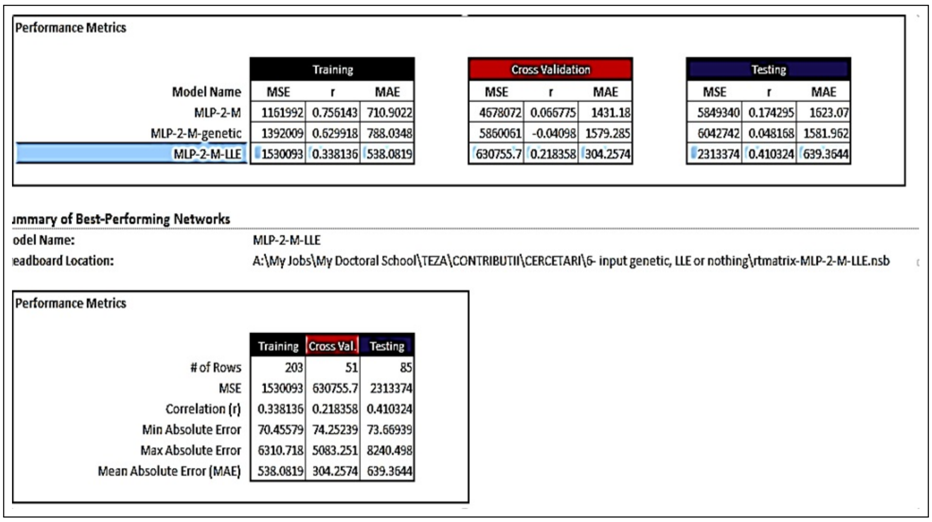
Output Dimension is used to specify the desired number of inputs as a percentage of number of inputs in the original data set. Initial SD of Gaussian is used to set the initial standard deviation for the Gaussian function that is used to define the neighborhood function. Initial Learning Rate is used to specify the starting learning rate and it is gradually reduced until the final learning rate is 0.01. Square Radius is used to define the neighbours. The shape of the neighborhood is defined as a square. Units/Dimension is used to define the smoothness of the projected space or grids. Maximum Epochs is used to specify the maximum number of epochs before termination of the computation.

MDS algorithm applied on the entire RTMATRIX/FPMATRIX determined Neuro Solutions to block indefinitely in a "not responding" state, but when we selected just a subset of them, it worked and offered some results.

LLE offers the best results, in comparison with the other four input projection algorithms, since it works on both FPMATRIX and RTMATRIX and provides the lowest prediction error.

In Neuro Solutions, the best input optimization algorithm is performed by the GeneticControl component. This component implements a genetic algorithm to optimize the inputs. Genetic Algorithms are search procedures based upon the principles of evolution witnessed in nature that combine selection, crossover, and mutation operators. They search for an optimal solution until a termination criterion is met. In Neuro Solutions the criteria used to evaluate the fitness of each potential solution is the lowest cost attained during the training run. The solution to a problem is called a chromosome and consists of a collection of genes, which are simply the inputs to be optimized. The genetic algorithm produces an initial population, evaluates this population by training a neural network for each chromosome and then evolves the population through multiple generations in the search for the best inputs.

In Figure 3 is presented the compared results between MLP-2-M with no input optimization or projection, MLP-2-M with only genetic optimization and MLP-2-M with both genetic and LLE input projection.



Performance Metrics

	Training			Cross Validation			Testing		
Model Name	MSE	r	MAE	MSE	r	MAE	MSE	r	MAE
MLP-2-M	1161992	0.756143	710.9022	4678072	0.066775	1431.18	5849340	0.174295	1623.07
MLP-2-M-genetic	1392009	0.629918	788.0348	5860061	-0.04098	1579.285	6042742	0.048168	1581.962
MLP-2-M-LLE	1530093	0.338136	538.0819	630755.7	0.218358	304.2574	2313374	0.410324	639.3644

ummary of Best-Performing Networks

odel Name: MLP-2-M-LLE

eadboard Location: A:\My Jobs\My Doctoral School\TEZA\CONTRIBUTII\CERCETARI\6- input genetic, LLE or nothing\rtmatrix-MLP-2-M-LLE.nsb

Performance Metrics

	Training	Cross Val.	Testing
# of Rows	203	51	85
MSE	1530093	630755.7	2313374
Correlation (r)	0.338136	0.218358	0.410324
Min Absolute Error	70.45579	74.25239	73.66939
Max Absolute Error	6310.718	5083.251	8240.498
Mean Absolute Error (MAE)	538.0819	304.2574	639.3644

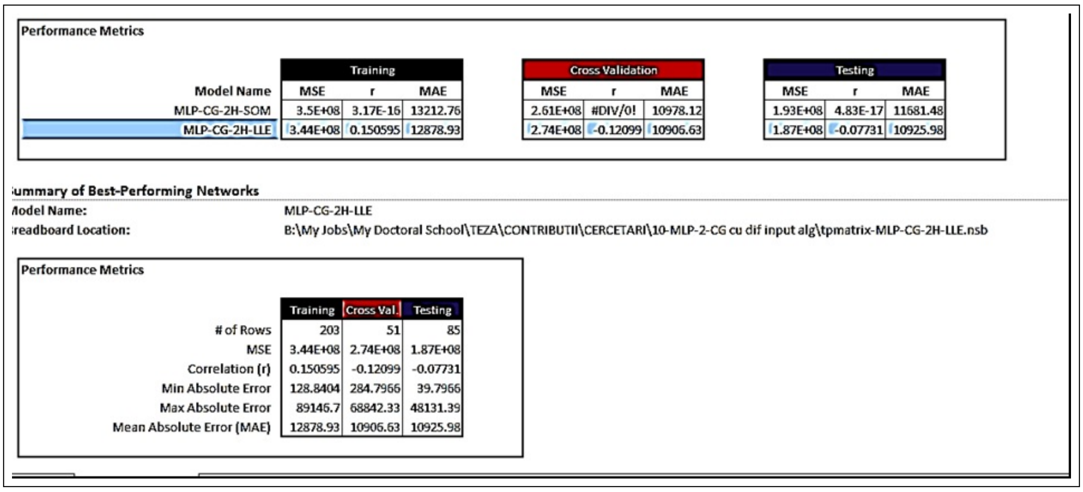
Figure 3: Effect of input optimization and projection on prediction accuracy

The comparison was performed on the entire RTMATRIX and the results show that MLP-2-M-LLE is the most accurate adaptive model for web services response time prediction. As we can observe, the dataset was divided in three subsets: training (50%), cross-validation (30%) and testing (20%). The training data set is obviously used for training, the cross-validation data set tests the model in the training phase and determines when the training is stopped, while the testing data set is utilized to investigate the prediction accuracy of the model when it receives new samples at the input.

Having the experience with the MLP trained on RTMATRIX, the subject of FPMATRIX investigation can be reduced at the comparison between MLP-2-CG-SOM and MLP-2-CG-LLE, both having as input optimization a genetic algorithm. We tested all input projection algorithms on FPMATRIX, each one separately and the only two that could find a solution were LLE and SOM. Consequently, we have labeled MDS and PCA input projection algorithms as not suitable for our prediction problem. The literature also recommends them for other types of problems like classification or clustering. The results of the comparison between LLE and SOM, when training a MLP with two hidden layers, Conjugate Gradient learning rule and Genetic input optimization, are shown in Figure 4.

The results reported in Figure 4 show that LLE is again the best input projection algorithm, since it offers the best prediction accuracy when training the MLP-2-CG on both RTMATRIX and FPMATRIX.

Neuro Solutions offers a fifth input projection algorithm, K-means clustering, but in our researches we have not used it, since it is not appropriate for prediction problems.



Performance Metrics

	Training			Cross Validation			Testing		
Model Name	MSE	r	MAE	MSE	r	MAE	MSE	r	MAE
MLP-CG-2H-SOM	3.5E+08	3.17E-16	13212.76	2.61E+08	#DIV/0!	10978.12	1.93E+08	4.83E-17	11681.48
MLP-CG-2H-LLE	3.44E+08	0.150595	12878.93	2.74E+08	-0.12099	10906.63	1.87E+08	-0.07731	10925.98

ummary of Best-Performing Networks

lodel Name:	MLP-CG-2H-LLE
readboard Location:	B:\My Jobs\My Doctoral School\TEZA\CONTRIBUTII\CERCETARI\10-MLP-2-CG cu dif input alg\tpmatrix-MLP-CG-2H-LLE.nsb

Performance Metrics

	Training	Cross Val.	Testing
# of Rows	203	51	85
MSE	3.44E+08	2.74E+08	1.87E+08
Correlation (r)	0.150595	-0.12099	-0.07731
Min Absolute Error	128.8404	284.7966	39.7966
Max Absolute Error	89146.7	68842.33	48131.39
Mean Absolute Error (MAE)	12878.93	10906.63	10925.98

Figure 4: A comparison between MLP-2-CG-SOM and MLP-2-CG-LLE

4 Conclusions and Future Works

QoS is still essential for high quality web services and remains one of the subjects that raise researchers' interest. More and more authors believe that large data may be as significant to society as the Internet. Social networks can play an important role in large and big data analytics and the technology trends indicate that they soon will interconnect not just people, but software modules like web services.

Consequently, in this research work we have studied four input projections algorithms, in order to determine which one increases the prediction accuracy of a Multi-Layer Perceptron (MLP) with two hidden layers, trained with web services large data. The result reports, for both FPMATRIX and RTMATRIX, show the Locally Linear Embed (LLE) as the most accurate input projection algorithm.

Concluding, MLP with two hidden layer, having as input projection algorithm LLE and a genetic algorithm for input optimization, can provide more accurate prediction results, when it is trained with large datasets of web services QoS criteria values.

In future works we will investigate different prediction adaptive models in order to improve web services QoS criteria prediction accuracy.

Bibliography

[1] R.-D. Albu (2013), *Contributions regarding the quality and reliability of web services*, PhD Thesis, University of Oradea.

[2] R.-D. Albu (2013), Investigating the Effect of Hidden Layers Number on Web Services Response Time Prediction, *Nonconventional Technologies Review*, ISSN 1454-3087, 7(1):4-9.

[3] R.-D. Albu, I. Felea, F. Popentiu-Vlădicescu (2013), On the Best Adaptive Model for Web Services Response Time Prediction, *The 20th Int. Conference on Systems, Signals and Image Processing, IWSSIP 2013*, CD Edition, IEEE Catalog Number : CFP1355E-CDR, ISBN: 978-1-4799-0942-1,39-42.

[4] R.-D. Albu, F. Popentiu-Vlădicescu (2013), On the Best Learning Algorithm for Web Services Response Time Prediction, paper accepted at *ESREL "Annual Conference, Advances in Safety, Reliability and Risk Management.*

[5] R.-D. Albu, F. Popentiu-Vladicescu (2013), A Comparative Study For Web Services Response Time Prediction, *The 9th Int. Scientific Conference eLSE 2013 "eLearning and Software for Education"*, 1: 656-665, Bucharest, , ISSN 2006-026x, CD Edition.

[6] A. Klasnja-Milicevic, M. Ivanovic, A. Nanopoulos (2009), The Use of Nonlinear Manifold Learning in Recommender Systems, *4th Int. Conference On Information Technology*, http://www.zuj.edu.jo/conferences/ICIT09/PaperList/Papers/Aritificial Intelligence/525.pdf.

[7] http://arxiv.org/abs/1111.4503 (available 16.11.2013)

[8] http://blogs.sas.com/content/sascom/2012/04/11/will-big-data-and-high-performance-analytics-flatten-the-world/ (22.10.2013)

[9] http://tech.sina.com.cn/i/ 2012-11-12/00207788375.shtml (available 16.11.2013)

[10] http://www.datadeluge.com/ (available 12.11.2013)

[11] L. Aspirot, P. Belzarena, B. Bazzano, G. Perera (2005), End-to-end quality of service prediction based on functional regression, *Proc. of Third Int. Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs 2005)*, Ilkley, UK, 1-8.

[12] Neuro Solutions help: http://www.aertia.com/docs/nd/neurosolutionshelp.pdf.

[13] P. Belzarena and L. Aspirot (2010), End-to-end quality of service seen by applications: A statistical learning approach, *Int. J. of Computer and Telecommunications Networking*, 54(17):3123-3143.

[14] Hu Y., Mu D., Gao A., Dai G.(2011), The Research of QoS Approach in Web Servers, *Int J Comput Commun*, ISSN 1841-9836, 6(4):636-647.

[15] Navarro M., Donoso Y. (2012), An IMS Architecture and Algorithm Proposal with QoS Parameters for Flexible Convergent Services with Dynamic Requirements, *Int J Comput Commun* ISSN 1841-9836, 7(1):123-134.

[16] Park E.-C. et al. (2011), Quality of Service Control for WLAN-based Converged Personal Network Service, *Int J Comput Commun*, ISSN 1841-9836, 6(4):716-733.

[17] P. Talebi Fard et al. (2013), Semantic Based Networking of Information in Vehicular Clouds Based on Dimensionality Reduction, *Proc. of the third ACM int. symposium on Design and analysis of intelligent vehicular networks and applications*, ACM, 69-76.

[18] Wei Tan, M. Brian Blake, Iman Saleh, Schahram Dustdar (2013), Social-Network-Sourced Big Data Analytics, Web-Scale Workflow, *Internet Computing, IEEE*, 17(5):62-69.

[19] Liang-Jie Zhang, Jia Zhang, Hong Cai (2007), *Services Computing*, Tsinghua University Press, Springer.

[20] Z. Zheng, Y. Zhang, M.R. Lyu (2010), Distributed QoS Evaluation for Real-World Web Services, *Proc. of the 8th Int. Conference on Web Services (ICWS2010)*, Miami, Florida, USA, 83-90.

[21] Z. Zheng, Y. Zhang, M.R. Lyu (2011), Exploring Latent Features for Memory-Based QoS Prediction in Cloud Computing, *Proc. of the 30th IEEE Symposium on Reliable Distributed Systems (SRDS 2011)*, Madrid, Spain, 1-7.

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):139-150, April, 2014.

A New Design for Control Method Based on Hierarchical Deficit Round Robin Scheduler for EPON

Y.-C. Cho, J.-Y. Pan

Ying-Chiang Cho*, Jen-Yi Pan
Department of Electrical Engineering,
National Chung Cheng University
Chia-Yi 62102, Taiwan, R.O.C
*Corresponding author: silvergun@mail2000.com.tw
E-mail: jypan@ccu.edu.tw

Abstract: With the development of ICT (Information and Communication Technology), how to use EPON for ensuring an effective and fair bandwidth allocation as well as the quality of service has become an important issue. Our research is based on Cousin Fair Hierarchical Deficit Round Robin Dynamic Bandwidth Allocation (CFH-DRR DBA), which applies the concepts of hierarchical scheduling to reduce extra actions in information controlling and queue switching and DRR (Deficit Round Robin) to attain the goal of cousin fairness. Our research proposes three additional modules to CFHDDR DBA: (1) an admission control module, which limits the sum weight of QoS-controlled flow; (2) a weight partition module, which isolates the sum weight of other interfering flows from QoS-controlled flows; and (3) the quantum adaptation module, which minimizes the access time of QoS-controlled flows through Quantum distribution. With the help of OMNet++ simulation software, this research presents the improvement of CFHDRR by introducing dynamic DDR Quantum. In addition, it proposes admission control and bounded weight to keep the sum of flows within service capacity. The simulation result shows that, while keeping CFHDRR's fairness, the queuing delay is reduced and the cycle time is effectively controlled so that the packet delay of QoS-controlled flows is minimized and QoS of real-time multimedia in EPON is fairly ensured.

Keywords: Ethernet passive optical network (EPON), Dynamic Bandwidth Allocation, Hierarchical Scheduling, Quality of Service (QoS).

1 Introduction

Network transmission devices have been continuously improving to meet the ever-changing market demands. The original copper wire was replaced by optical fiber, increasing the rate of transmission to Giga-bps today and even to 10 Giga-bps in the future [1] [2]. Nevertheless, digital subscriber line [3] and cable modem [4] are still the only two types of network devices that current network operators provide to their consumers. Cable Modem [4], as well as other copper transmission devices, fails to break the bottleneck of Access Networks (ANs). Optical network's low loss, high capacity, bandwidth flexibility, EMC immunity and high confidentiality have made it the ultimate solution for Last Mile and a possible replacement of copper transmission devises. One of the advantage of using passive optical network is that passive components can process data without electricity, making it preferred by network operators when they improve the existing network architecture [1].

2 Problem statement

The principle guiding dynamic bandwidth allocation for Ethernet passive optical network is the quick allocation of upload bandwidth [8]. The following challenges must be dealt with. The first challenge is the scale expandability in dynamic bandwidth allocation. Based on the present data transmission speed and actual use of Ethernet passive optical network, it is certain that the scale of optimal networks will expand in the future. How to design a scheduling method which is both efficient and cost-effective will be one major task. The second challenge is time urgency. Since online audio and video transmission has been widely used, any slightest delays may result in problems for the users. Yet, not all types of transmission have strict requirements on delays. It poses another task, which is how to identify and protect specific QoS flows. The third challenge is fairness. There are a variety of ways to define fairness. Which definition is most suitable for the target environment and how to allocate the bandwidth so as to achieve fairness will also be highly important issues [11].

3 Problem-Solving Strategies

This research proposes dynamic Quantum allocation method based on Cousin Fair Hierarchical Deficit Round Robin Dynamic Bandwidth Allocation (CFHDRR DBA)) [9] [10] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21], which can keep cycle time stable through Quantum dynamic allocation. This method also makes bandwidth more efficient by modifying the ONU single round translimit of the original cousin-fair hierarchical deficit scheduling method. As shown in figure 1, in cousin-fair hierarchical deficit scheduling, the contribution of ONU (Optical Network Unit) is to report consolidated information that are sent to OLT (Optical Line Terminal). Yet, high control message overhead of OLT makes it difficult to limit the cycle time. This research attempts to solve this problem by analyzing constantly the relation between cycle time and Quantum and adjusting Quantum according to Match-T flows. Match-T is a similar concept to packet feature of Token bucket, which consists of packet Source IP, Source Port, Destination IP, and Destination Port.

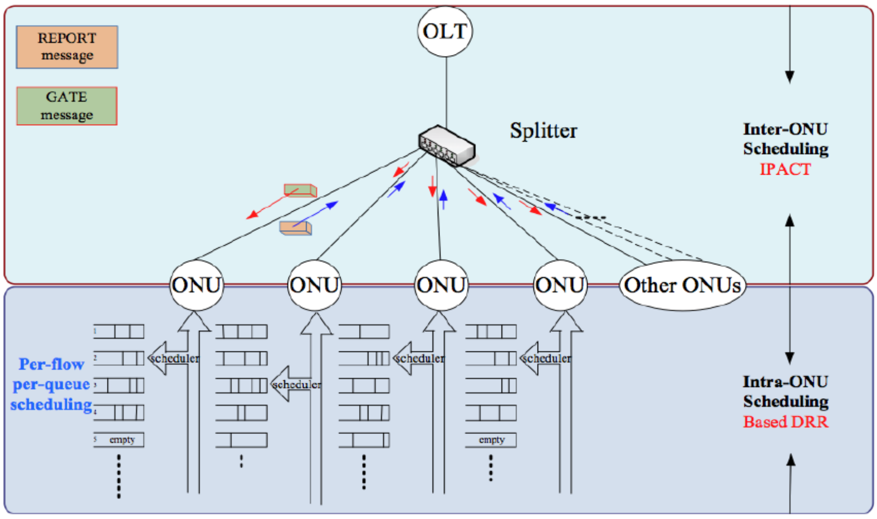


Figure 1: Structure of Adjustable Cycle Control Method based on Hierarchical Deficit Round Robin Scheduling

Our research also proposes an admission control module to Intra ONU scheduler. If a flow is a Match-T flow, when its weight is greater than the weight allocated by the system, its connection will be refused. If a flow is not a Match-T flow, the system will exclude its weight as well as all other non Match-T flows. With a limited sum weight of non Match-T flows, the system is able to allocate enough bandwidth to Match-T flows in every transmission cycle and to send all packets within delay limits. Any changes in the number of flows will cause the system to reconsider the connection requests of new flows according to the sum weight limits: when a new flow is a not a Match-T flow, the system will recalculate the sum weight of non Match-T flows; when a new flow is a Match-T flow, its connection request will be accepted only when the system can allocate enough weight to it.

3.1 Design of an Inter ONU scheduler

This paper proposes a dynamic Quantum allocation method for OLT based on CFHDRR DBA. This method can also be applied to control cycle time. Table 1 describes the parameters used in the above method.

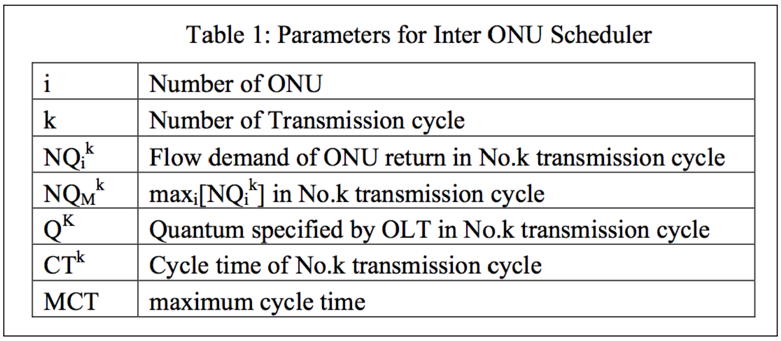


Table 1: Parameters for Inter ONU Scheduler

i	Number of ONU
k	Number of Transmission cycle
NQ_i^k	Flow demand of ONU return in No.k transmission cycle
NQ_M^k	$max_i[NQ_i^k]$ in No.k transmission cycle
Q^K	Quantum specified by OLT in No.k transmission cycle
CT^k	Cycle time of No.k transmission cycle
MCT	maximum cycle time

After ONU scheduling, Intra ONU Scheduler will report the bandwidth of ONU to OLT, so as to reduce packet transmission of MPCP. One disadvantage of ONU scheduling is that OLT cannot calculate the quantities and weights of flows under each ONU. This research proposes to predict the maximum number of upload data after Quantum adjustment according to the ratio of total upload amount of ONU to Quantum, as shown in Formula 1. In the formula, the total upload amount equals the cycle time. Therefore, this formula can be used to calculate the amount of and proportion of quantum adjustment. This formula can also be applied to limit the cycle time.

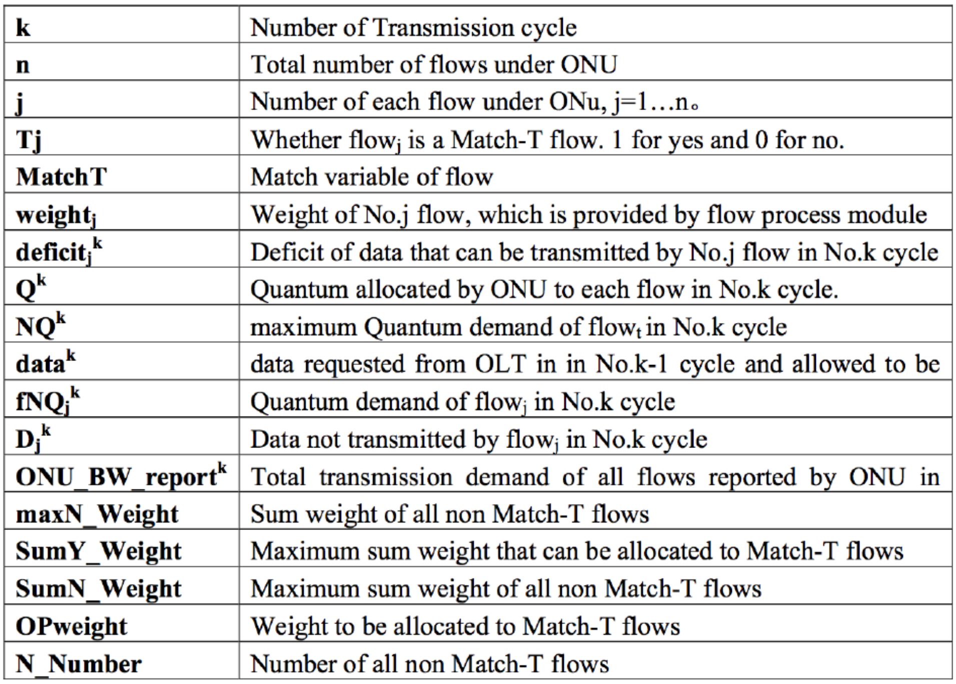
$$\frac{Quantum}{cycletime} = \frac{Quantum\ adjustment}{cycle\ time\ to\ be\ predicted} \tag{1}$$

When receiving a packet, the packet process module of OLT system will first estimate the bandwidth of ONU demand. It will then identify the NQ_i^k in the report message and send it to the dynamic allocation module. After Q^{k+1} is received from the dynamic allocation module, the system will calculate ONU bandwidth and send ONU bandwidth via GATE message. By doing so, ONU is expected to send all packets in the following round and reduce delay time. In the dynamic allocation module, the packet process module reports NQ_i^k in the ONU report message to NQ_i^k record module. If all requirements for starting a new transmission cycle are met, the system will recalculate the Quantum. CT record module is responsible for recording the time spent by every transmission cycle. Data process module identifies NQ_M^k (i.e. the maximum value of NQ_i^k). Quantum calculation module tests what requirements meet the demand of NQ_M^k under MCT restriction. Such requirements should meet the packet demand of the other flows and should be able to reduce delay time. Suppose, in this round, when NQ_M^k is 1300 and CT^k is 0.5ms, by taking NQ_M^{ki} into formula 4, we can calculate the value of CT^{k+1} when the allocated Q^{K+1}=1300. In this case, the value of CT^{k+1} is 1.3ms. If this value is less than MCT, the value of NQ_M^k is given to Q^{K+1}. Q^{K+1} is included in the GATE message to modify the distribution benchmark. If the value of NQ_M=2600 is greater than MCT, then the value of MCT is taken into formula 3 to calculate Q^{K+1}. Q^{K+1} is included in the GATE message to let ONU modifies the distribution benchmark.

3.2 Design of an Intra ONU scheduler

This paper proposes some additional modules based on CFHDRR DBA, including a flow filter module and an admission module. When a new flow is added, the admission module controls the number of Match-T flows and the weight of non Match-T flows to guarantee delay time is minimum. The admission module also calculates how many data does every Match-T flow store in each cycle and divides the number by weight of flow provided by the flow process module. This allows us to acquire the Quantum value in the next round. This design is intended to reduce waiting time of data so as to minimize queue delay. Parameters used in this method are shown below.

Table 2: Parameters for Intra ONU Scheduler



k	Number of Transmission cycle
n	Total number of flows under ONU
j	Number of each flow under ONu, j=1…n。
Tj	Whether flow_j is a Match-T flow. 1 for yes and 0 for no.
MatchT	Match variable of flow
weight_j	Weight of No.j flow, which is provided by flow process module
deficit_j^k	Deficit of data that can be transmitted by No.j flow in No.k cycle
Q^k	Quantum allocated by ONU to each flow in No.k cycle.
NQ^k	maximum Quantum demand of flow_t in No.k cycle
data^k	data requested from OLT in in No.k-1 cycle and allowed to be
fNQ_j^k	Quantum demand of flow_j in No.k cycle
D_j^k	Data not transmitted by flow_j in No.k cycle
ONU_BW_report^k	Total transmission demand of all flows reported by ONU in
maxN_Weight	Sum weight of all non Match-T flows
SumY_Weight	Maximum sum weight that can be allocated to Match-T flows
SumN_Weight	Maximum sum weight of all non Match-T flows
OPweight	Weight to be allocated to Match-T flows
N_Number	Number of all non Match-T flows

When receiving GATE message, to ensure that all requested data from the former round could be transmitted, ONU will send message to packet process model and report the start time and transmission time to data transmission module. Q^K in GATE message will be reported to FLOW bandwidth allocation module. After that, FLOW bandwidth allocation module will allocate $deficit_j^k$ (multiplying Q^K by $weight_j$) of each $flow_j$. Based on the concept of token bucket, this method takes $deficit_j$ as the token to calculate the transmission amount of $flow_j$ (i.e. ftjk) in the next cycle. The transmission amount of all $flow_j$ (i.e. ONU_BW_$report^k$) is summed up with Gather module. After each packet request is transmitted, the method will write a REPORT message. CFHDRR is applied to categorize traffic and FLOW select module is employed to identify Match-T flows. Match-T flows allow factories to send OAM information from OLT to ONU. Source IP, Source Port, Destination IP, and Destination Port are used to describe the categories of selected flows. This method assumes that every flow that is sent to ONU have been categorized and compared. IP utilizes wildcards to express the concept of domain-.192.168.*.*, for example. Flows that meet such requirements are categorized as Match-T flows.

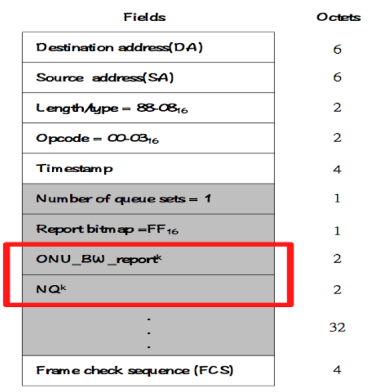
3.3 Design of FlOW admission module

This research applies an admission module to allocate flow weight, which is sent to the flow bandwidth allocation module. However, the method proposed in this paper allocates weight according to preset data provided by network operators. This method may not be able to control delay when there is a large number of flows. We proposes a solution to this problem. Applying the formula below, we can calculate the maximum Q^k (hereafter referred to as Max_Q^k). When there is a large number of flows and the sum weight increases, and Max_Q^k will decrease. If the value of Max_Q^k is less than NQ^k, the method cannot limit delay. Therefore, in order to minimize delay of Match-T flows, we attempts to control the maximum sum weight of non Match-T flows.

$$MaxQ^k = MCT \div \sum_i \sum_j weight_j^i \tag{2}$$

3.4 Revised design of Multi-point control protocol

The method of this research is based on CFHDRR DBA, with similar GATE message. Like CFHDRR DBA, the method also has a 1Byte Quantum field, which will process the 39 Byte Padding of GATE message (not including grant start time and grant length). Using 1 Byte to transmit Quantum won't change the original size of packet and won't impose transmission overhead to data control. There are two Queue report fields, as shown in Figure 2: one is ONU bandwidth demand report (ONU_BW_$report^k$), the other is number of Quantum needed to report the maximum flow (NQ^k). These two fields enable Inter ONU scheduling to function.



Fields	Octets
Destination address(DA)	6
Source address(SA)	6
Length/type = 88-08$_{16}$	2
Opcode = 00-03$_{16}$	2
Timestamp	4
Number of queue sets = 1	1
Report bitmap =FF$_{16}$	1
ONU_BW_report^k	2
NQ^k	2
⋮	32
Frame check sequence (FCS)	4

Figure 2: Report message

4 Simulation techniques and targets

Using OMNet++ [23] [24] [25] [26] [27] as simulation tool, this paper revises modules for Ethernet passive optical network and achieves the expected simulation target. The major revisions are presented below:

1. Revise REPORT message and information contained in GATE message In CFHDRR, REPORT message carries only the demand of bandwidth for ONU. This research proposes to add maximum demand Quantum, which allows OLT to control cycle time.

2. Add an packet analyzer The original module categorizes different flows and transmit upstream packets. In order to identify different flow features, this research will add a packet

analyzer to analyze Match-T flows and to calculate the demanded Quantum, which is helpful for the decision of bandwidth allocation.

5 Simulation and verification

The simulation environment consists of one OLT and N ONUs. Each ONU has Q series, which represents a flow in the ONU. The transmission rate of EPON is Rate_epon Mb/s. The transmission rate of the access line between users and the ONU is Rate_u Mb/s. Each line has the same downstream rate. Network traffic types of EPON are CBR and TCP Session, and our version is TCP-Reno. We utilize network traffic under TCP communication protocol as well as CBR traffic under UDP communication protocol. CBR sends 800 Byte packets each 0.8ms. CBR traffic is distributed in four ONUs and TCP, which serves as background traffic, also to four different ONUs, and each with three flows.

5.1 Simulation lab 1 - Control Cycle Time

To prove that our method can ensure dynamic allocation of Quantum to ONU, control cycle time within a specific length, and observe the influence of burst flows on cycle time. As shown in Figure 3, in this experiment, numbers of flows remain steady at 2ms within every ONU. Two burst CBR flows increases cycle time at the 0.5th second to almost 2.8ms. But in the following cycle, cycle time is brought back to be lower than the maximum cycle time because OLT automatically adjusts the allocated Quantum.

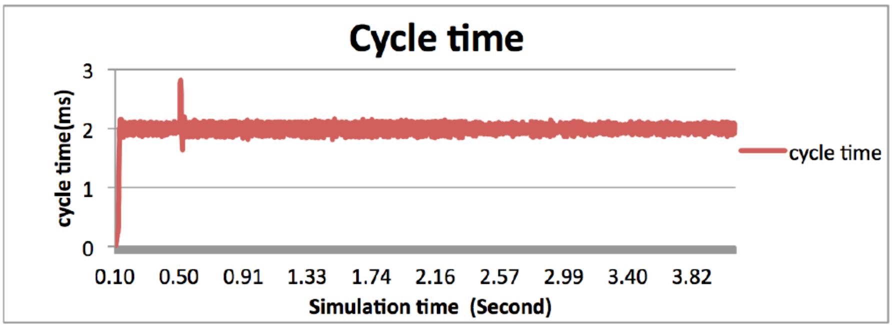


Figure 3: Response of OLT to burst flow and cycle time

5.2 Simulation lab 2 - Different scheduling methods and control of cycle time

Figure 4(a) shows our method can efficiently control cycle time. The simulation experiment begins at the 0.1th second, and flows are added each 0.01 second until the 0.15th second. In Figure 4(b), another method adjusts cycle time to a fixed length, which is inefficient and unstable compared with our method.

Figure 5 shows (a) delay of Match-T flows and (b) packet size of these flows after utilizing our method. Figure 5(a) shows that, in case of full load, our method is able to limit latency of Match-T flows to be less than 1.5ms. Figure 5(b) shows the packet size of Match-T flows in series. Figure 6 shows (a) delay of non Match-T flows and (b) packet size of these flows. In Figure 6, some of the latencies are longer than 1.5ms, which means our method does not reduce latency of non Match-T flows.

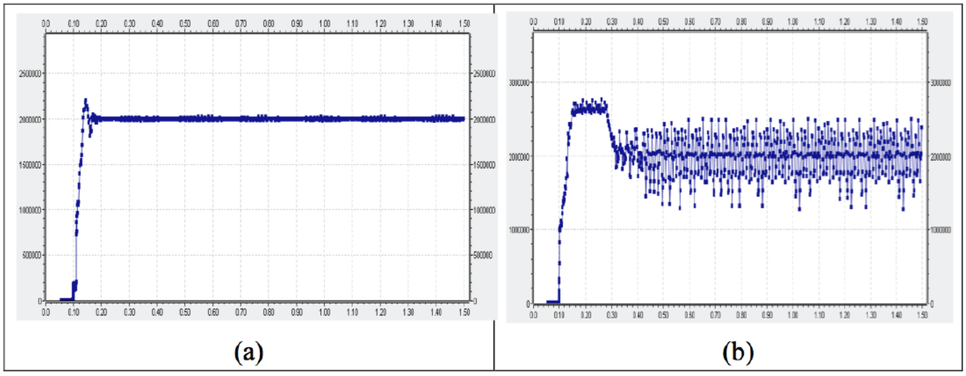


Figure 4: Comparison of cycle time control

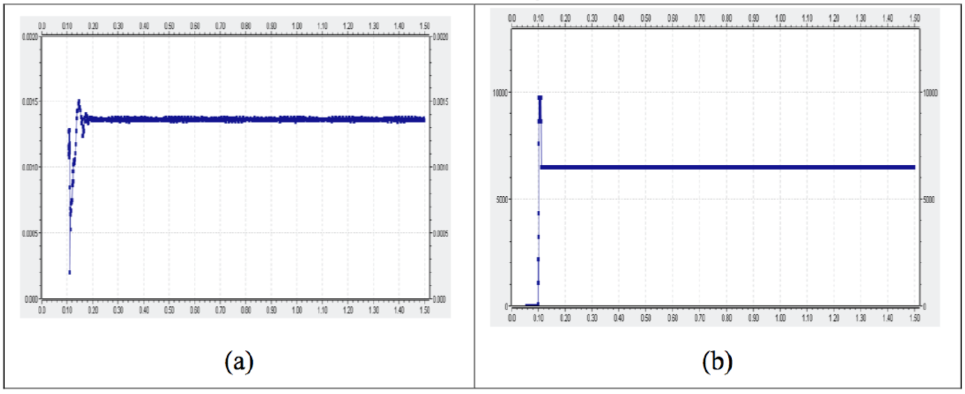


Figure 5: delay of Match-T flows and packet size of these flows

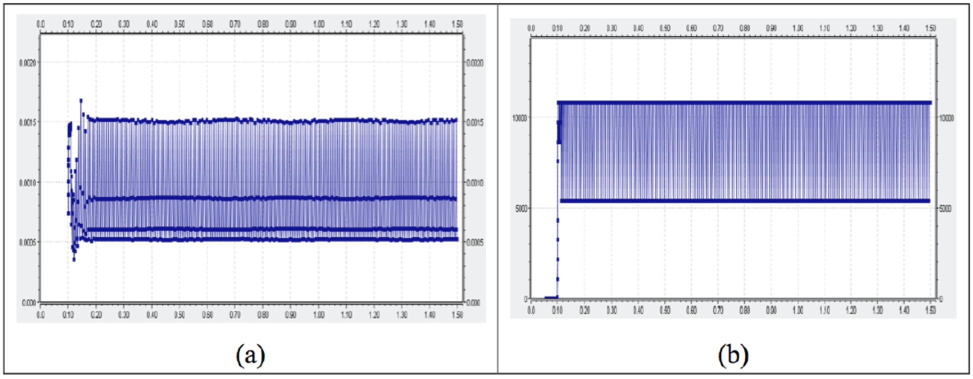


Figure 6: delay of non Match-T flows and packet size of these flows

5.3 Simulation lab 3 - Delay

We continues to examine whether our research can reduce the delay of Match-T flows. Figure 7 shows the delay of each packet in every cycle. The dots represents delays of packets in each cycle. The vertical axis is delay time, using second as its base unit of time. The horizontal axis is experiment time. All flows start at the 0.1th second. In this experiment, packet delays of Match-T flows remain steady within 0.3ms to 0.8ms. As shown in Figure 8, packet delays of non-target packets remain steady within 0.7ms to 1.2ms.

In this simulation experiment, we set the server IPs of voice calls to reduce delays of flows.

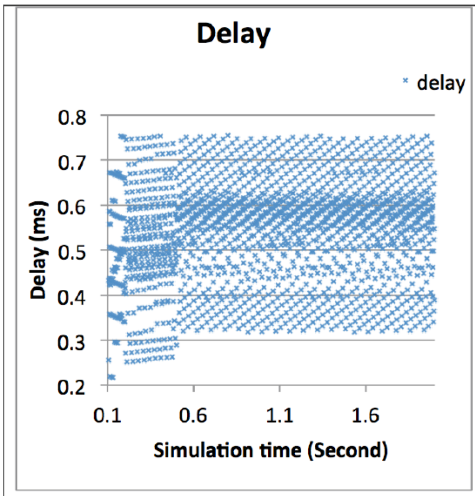


Figure 7: Packet delays of Match-T flow

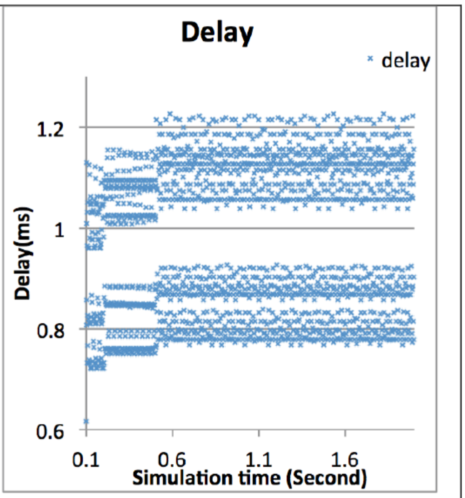


Figure 8: Packet delays of non Match-T flows

Figure 9 shows the delay of each packet in every cycle after some adjustments were made to Match-T and all other factors remain unchanged. As shown in Figure 9, packet delay is reduced and most delays were even less than 0.5ms. Due to reduced delay, cycle time is also decreased to 0.9-1ms.

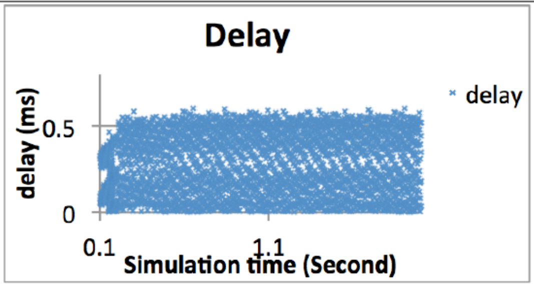


Figure 9: Packet delays of Match-T flows

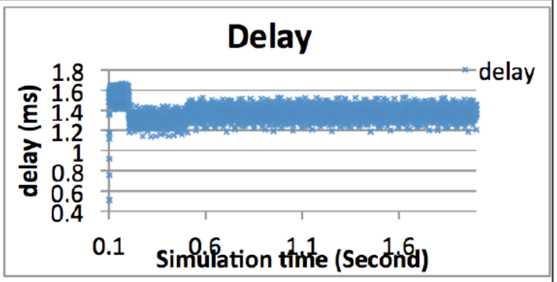


Figure 10: Packet delays of non Match-T flows

Simulation test result proves that categorization of target flows greatly reduce packet delays. This method may increase delay of non Match-T flows at the same time and flow delay may increase in the end. As shown in Figure 10, delays were longer, increasing by 0.3ms at average level. To solve this problem, we proposes Quantum allocation method to ensure that ONU can send all Match-T flows in each cycle. Other non Match-T flows have to wait longer if their requested relatively big Quantum. Increasing the delay of non Match-T flows enables us to better restrict the delay of Match-T flows.

5.4 Simulation lab 4 - Cycle time and average delay time

We also test the performances of our method when numbers of flows are different. As shown in Figure 11, with the same number of flows, changes in packet size and packet frequency don't lead to changes in average delay(1 3). However, as number of flows increases, delay will also increase. This result shows that a larger number of flows requests a larger bandwidth, a longer transmission cycle, and a longer delay.

5.5 Simulation lab 5 - Fairness

Figure 12 is a comparison of throughput between two flows with different weights. New flows added at 0.3th, 0.5th, and 1th second will decrease throughput. The figure below also shows that the ratio of these two flows remains relatively stable during the whole experiment.

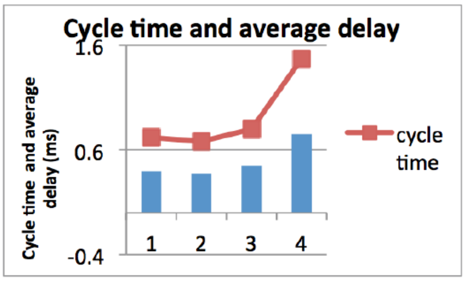


Figure 11: Cycle time and average delay

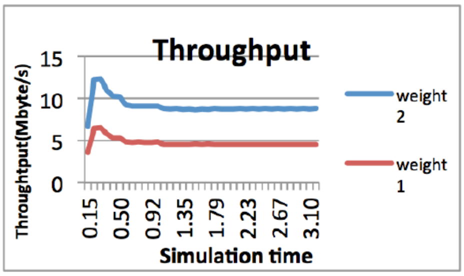
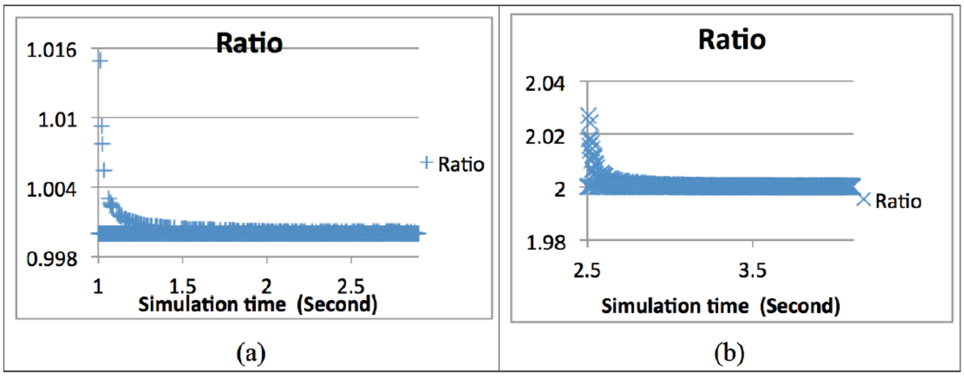


Figure 12: comparison of throughput between two flows with different weights



(a)

(b)

Figure 13: Ratio of queues with different weights in ONU

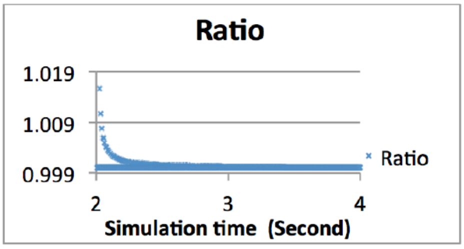


Figure14: Ratio of transmission flows with same weight in different ONU

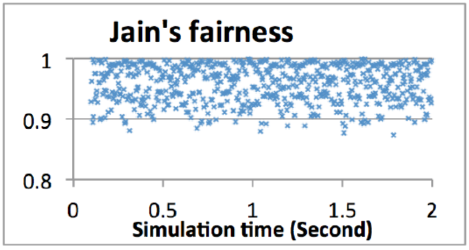


Figure15: Jain's fairness index

In Figure 13(a), the vertical axis is the ratio of data transmitted by two ports in each cycle. The weights of the two ports are all 1. In this experiment, although the starting ratio is relatively large, it gets closer to 1 over time. In Figure 13(a), the vertical axis is the ratio of data transmitted by two other ports in each cycle. The weights of these two ports are different. In this experiment, although the starting ratio is also relatively large, it gets closer to 2 over time.

Figure 14 shows the ratio of transmission flows with same weight in different ONU. The ratio also gets closer to a fixed value over time.The above simulation experiment shows that, with different of same ONU, our method can achieve almost perfect cousin fairness over time. In Figure 15, every dot represents a Jain's fairness index in 10 transmission cycles. Since these fairness index remain steady between 0.9-1, our method is proven to be able to guarantee allocation fairness.

6 Conclusions

Innovations in internet service have pushed up the demand for network bandwidth. The development and application of Ethernet passive optical network technology make it a possible option to meet the demand. Meanwhile, how to ensure an efficient and fair allocation of bandwidth becomes an important issue [13] [16] [30] [31] [32] [33]. This paper proposes a dynamic Quantum allocation method to control cycle time based on Cousin Fair Hierarchical Deficit Round Robin Dynamic Bandwidth Allocation. This method is intended to reduce the overhead of information and to ensure cousin fairness. In addition, with the help of GATE message, it serves to report and calculate the demand of March-T flow Quantum that are sent back by ONU, guarantee fairness of transmission, and reduces packet delay of Match-T flows.

Bibliography

[1] Kramer, Glen, and Gerry Pesavento (2002), Ethernet passive optical network (EPON): building a next-generation optical access network. *Communications magazine IEEE*, 40(2): 66-73.

[2] Udd, Eric (ed.) (1995), *Fiber optic smart structures*, New York: Wiley.

[3] Starr, Thomas, John M. Cioffi, and Peter J. Silverman (1999), *Understanding digital subscriber line technology*, Prentice Hall PTR.

[4] Ginis, George, and John M. Cioffi (2002), Vectored transmission for digital subscriber line systems, *Selected Areas in Communications, IEEE Journal on*, 20(5): 1085-1104.

[5] Sivalingam, Krishna M., and Suresh Subramaniam (eds. (2000), *Optical WDM networks: Principles and practice*, Springer.

[6] Dutta, Rudra, and George N. Rouskas (2000), A survey of virtual topology design algorithms for wavelength routed optical networks, *Optical Networks Magazine*, 1(1): 73-89.

[7] Essiambre, R-J., et al. (2010), Capacity limits of optical fiber networks, *Lightwave Technology, Journal of*, 28(4): 662-701.

[8] Kramer, Glen (2005), *Ethernet passive optical networks*, New York: McGraw-Hill.

[9] Mcgarry, M., Martin Reisslein, and Martin Maier (2008), Ethernet passive optical network architectures and dynamic bandwidth allocation algorithms, *Communications Surveys & Tutorials, IEEE*, 10(3): 46-60.

[10] Kramer, Glen, et al.(2004), Fair queueing with service envelopes (FQSE): a cousin-fair hierarchical scheduler for subscriber access networks, *Selected Areas in Communications, IEEE Journal on*, 22(8): 1497-1513.

[11] Kramer, Glen, and G. Pesavento (2003), EPON: challenges in building a next generation access network, 1st International Workshop on Community Networks and FTTH/P/x, Dallas, Tx.

[12] Dhaini, Ahmad R., et al.(2006), Adaptive fairness through intra-ONU scheduling for ethernet passive optical networks, *Communications ICC'06. IEEE International Conference on*, Vol. 6.

[13] Assi, Chadi M., et al. (2003), Dynamic bandwidth allocation for quality-of-service over Ethernet PONs, *Selected Areas in Communications, IEEE Journal on*, 21(9): 1467-1477.

[14] McGarry, Michael P., Martin Maier, and Martin Reisslein (2004), Ethernet PONs: a survey of dynamic bandwidth allocation (DBA) algorithms, *Communications Magazine, IEEE*, 42(8): 8-15.

[15] Dhaini, Ahmad R., et al. (2007), Dynamic wavelength and bandwidth allocation in hybrid TDM/WDM EPON networks, *Journal of Lightwave Technology*, 25(1): 277-286.

[16] Ghani, Nasir, et al. (2004), Intra-ONU bandwidth scheduling in Ethernet passive optical networks, *Communications Letters*, IEEE, 8(11): 683-685.

[17] Zheng, Jun, and Hussein T. Mouftah (2009), A survey of dynamic bandwidth allocation algorithms for Ethernet Passive Optical Networks, *Optical Switching and Networking*, 6(3): 151-162.

[18] Kim, Chan, Tae-Whan Yoo, and Bong-Tae Kim (2007), A hierarchical weighted round robin EPON DBA scheme and its comparison with cyclic water-filling algorithm, *Communications, 2007. ICC'07. IEEE International Conference on*, IEEE.

[19] Zheng, J. (2006), Efficient bandwidth allocation algorithm for Ethernet passive optical networks, *IEEE Proceedings-Communications*, 153(3): 464-468.

[20] Zheng, Jun, and Hussein T. Mouftah (2005), Media access control for Ethernet passive optical networks: an overview, *Communications Magazine*, IEEE, 43(2): 145-150.

[21] Byun, Hee-Jung, Ji-Myung Nho, and Jong-Tae Lim (2003), Dynamic bandwidth allocation algorithm in Ethernet passive optical networks, *Electronics Letters*, 39(13): 1001-1002.

[22] Kwon, Taeck-Geun, Sook-Hyang Lee, and June-Kyung Rho (1998), Scheduling algorithm for real-time burst traffic using dynamic weighted round robin, *Circuits and Systems, ISCAS'98, Proceedings of the 1998 IEEE International Symposium on*, Vol. 6. IEEE.

[23] OMNeT++ Home Page. [Online]. Available: http://www.omnetpp.org

[24] Varga, András (2001), The OMNeT++ discrete event simulation system, *Proceedings of the European Simulation Multiconference (ESM'2001)*, Vol. 9.

[25] Varga, András, and Rudolf Hornig (2008), An overview of the OMNeT++ simulation environment, *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).*

[26] Varga, András (1999), Using the OMNeT++ discrete event simulation system in education, *Education, IEEE Transactions on*, 42(4): 11p.

[27] Varga, Andras (2010), OMNeT++, *Modeling and Tools for Network Simulation*, Springer Berlin Heidelberg, 35-59.

[28] Varga, Andras (2010), *Omnet++ user manual. OMNeT++ Discrete Event Simulation System*, Available at: http://www. omnetpp. org/doc/manual/usman. html.

[29] B. Andreas. (2010), Epon module for OMNet++ simulator. [Online]. Available: http://sourceforge.net/projects/omneteponmodule/

[30] Luo, Yuanqiu, and Nirwan Ansari (2005), Limited sharing with traffic prediction for dynamic bandwidth allocation and QoS provisioning over Ethernet passive optical networks, *Journal of Optical Networking*, 4(9): 561-572.

[31] Banerjee, Amitabha, et al. (2005), Wavelength-division-multiplexed passive optical network (WDM-PON) technologies for broadband access: a review [Invited], *Journal of optical networking*, 4(11): 737-758.

[32] Luo, Yuanqiu, and Nirwan Ansari (2005), Bandwidth allocation for multiservice access on EPONs, *Communications Magazine*, IEEE, 43(2): S16-S21.

[33] Yang, Kun, et al. (2009), Convergence of ethernet PON and IEEE 802.16 broadband access networks and its QoS-aware dynamic bandwidth allocation scheme, *Selected Areas in Communications, IEEE Journal on*, 27(2): 101-116.

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):151-159, April, 2014.

Algorithm of Maximizing the Set of Common Solutions for Several MCDM Problems and it's Application for Security Personnel Scheduling

S. Dadelo, A. Krylovas, N. Kosareva, E.K. Zavadskas, R. Dadeliene

Stanislav Dadelo, Aleksandras Krylovas,
Natalja Kosareva*, Edmundas Kazimieras Zavadskas
Vilnius Gediminas Technical University
Sauletekio al. 11, LT-10223 Vilnius, Lithuania
E-mail: stanislav.dadelo@vgtu.lt, aleksandras.krylovas@vgtu.lt,
natalja.kosareva@vgtu.lt, edmundas.zavadskas@vgtu.lt
*Corresponding author: natalja.kosareva@vgtu.lt

Ruta Dadeliene
Lithuanian University of Educational Sciences
Studentu st. 39, LT-08106 Vilnius, Lithuania
E-mail: ruta.dadeliene@leu.lt

Abstract: The article deals with the task of elite selection of private security personnel on the basis of objective and subjective criteria. One of the possible solutions of this multiple criteria decision making (MCDM) problem is creation of heuristics allowing to minimize discrepancy of ranks calculated for the objective and subjective criteria on the basis of the best security staff. The proposed heuristic combines interval points re-selection and random points generation methods. Two optimizing algorithms are proposed. It is shown how this method is applied for solving specific task of elite selection from security personnel.

Keywords: preference feature, multiple criteria decision making, heuristics, optimizing algorithm.

1 Introduction

EU private security sector employs millions of people and this figure is constantly growing [1]. Economic trends provide further development of the private security. It is predicted that over the next 10 years, the U.S. private security sector job demand will increase by 15% and this is much higher growth rate compared to the overall job growth around the U.S. economy [2]. Safety business formed specific personnel management challenges. Personnel selection and placement to the necessary positions (ranking) is seen as the most important factor affecting the organizations security, stability and development [3] and for private security this process takes on a deeper meaning. Personnel selection process focuses on measurement and evaluation of specific potential, skills and personal characteristics of candidates. Security personnel evaluation process requires identification of specific characteristics (criteria) for the occupation of the post and their weights determination [4]. It is necessary to develop universal algorithms for personnel selection [5]. Developed selection and evaluation systems are mainly focused on companies operating cost reduction, in order to optimize staffing requirements and layout planning [6], but is not enough in the safety field. Assessing the activities of private security, their complexity and the peculiarities of the hazard, the following complexity – hazard levels are distinguished: 1) protection of civil objects; 2) protection of critical and strategic objects, collection, and personal protection [7]. Evolution of the modern world generated the demand of private security services in military operations. However, armed security services in the "hot" spots of the world providing companies are often confronted with the staff inappropriate behavior in dangerous and emergency situations. Assessing the private security business diversity, personnel evaluation and selection procedure for different tasks becomes necessity. There is a need to look for factors affecting the security staff competence and professional activities as a basis for generation of algorithms for employees evaluation and selection for different tasks procedures [8]. Evaluating of personnel engaged in security features in dangerous environment it is necessary to search for specific solutions [3]. Critical tasks in hazardous environments, which may result in loss of health and even life, must carry only the elite guards. Ryan et al. [9] note that only about 10 – 12 % of the total sample of candidates are able to meet the demands posed by members of special elite forces. Dessler [10] states that only 15% of the total sample of individuals can be evaluated in the maximum. Recent research reveals the elite diagnostic possibilities according to six sigma principle [11]. Private security personnel selection process should point competency assessment in two directions: external (evaluation of subordinate by his immediate superior) and internal (personal competence) [12]. Relevance of the problem occurs in the construction of selection algorithms for private security elite, capable of effectively carry out extremely important and dangerous tasks. This scientific problem is not paid enough attention. The aim of this study is construction approval of private security elite selection algorithm. A similar kind of research has not yet been done.

An overview of possible MCDM methods for this problem solution is provided by Zavadskas and Turskis [13]. Recently fuzzy MCDM methods are becoming increasingly popular [14]. In this article heuristic methods, based on the function, minimizing non-compliance of ranks, calculated for the best security staff in accordance with objective and subjective criteria are proposed.

2 Internal and external evaluation criteria

Participants. One hundred and eighteen security guards were randomly selected from the company G4S Lietuva; twenty two leader managers (experts) of G4S Lietuva with not less than 10 years of service at private security structures involving execution and organization of security ranked the competences chosen by authors of the article.

Security guards internal evaluation (x) (objective testing, measuring, personal competence). Selected security guards were tested and evaluated according to 41 criteria. The data received were classified into six groups of competences (variables) regarding the features analysed [12] (Table 1):

1. Theoretical and practical preparation (x_1): knowledge, skills, abilities, practical experience – acquired throughout life;
2. Professional activity (x_2): carrying out required tasks;
3. Mental qualities (x_3): individual psychical qualities vital for performance of professional activities;
4. Physical development (x_4): morphological indications of a body;
5. Motor abilities (x_5): personal physical conditions allowing carrying out physical tasks at work or home, during leisure, and reflecting the level of physical qualities;
6. Fighting efficiency (x_6): a set of physical and mental qualities influencing the ability to effectively carry out actions fighting an adversary in direct contact.

Table1: Security guards internal evaluation criteria (objective tests and measurements)

Security guards	Criteria (competences)					
	x_1	x_2	x_3	x_4	x_5	x_6
max (highest values)	2.582	2.186	2.651	3.388	2.768	1.696
min (lowest values)	-1.606	-2.567	-2.518	-1.948	-2.596	-1.709
a_1	1.358	-1.503	0.607	0.520	1.381	1.696
a_2	0.259	-0.549	-0.424	-1.487	-0.645	0.068
a_3	-0.984	0.465	0.101	-0.479	0.690	-0.100
⋮	⋮	⋮	⋮	⋮	⋮	⋮
a_{116}	0.225	-0.500	-0.837	-0.599	0.749	0.795
a_{117}	0.561	-1.366	0.438	-0.017	0.013	-0.903
a_{118}	-1.512	-1.391	-2.180	-1.666	-1.520	-0.845

Security guards external evaluation (y) (evaluation of subordinate by his immediate superior). Team leaders evaluated their subordinates based on A. Sakalas [15] modified methodology (Table 2). They have been assessed on nine criteria that could affect security guards professional competences:

1. Specialty knowledge, professionalism (y_1): versatility, knowledge about their and related occupations;
2. Diligence and positive attitude to work (y_2): activeness, responsibility, discipline, zeal, vocation to work;
3. Behaviour with colleagues and supervisors (y_3): the ability to cooperate and work in a team;
4. Reliability at work (y_4): ability and willingness to perform tasks independently;
5. Quality of work (y_5): the ability to avoid mistakes;
6. Workload performance (y_6): the ability to carry out the maximum amount of work;
7. Image (y_7): the self-representational skills (exterior, posture, language culture);
8. Development rate (y_8): the ability to quickly adapt to new requirements and new working conditions;
9. Being promissing (y_9): potential for career.

Table 2: Security guards external evaluation criteria (evaluation of subordinate by his immediate superior)

Security guards	Criteria (competences)								
	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
max (highest values)	5	5	5	5	5	5	5	5	5
min (lowest values)	0	0	0	0	0	0	0	0	0
a_1	3	4	4	4	4	4	4	4	4
a_2	4	4	4	4	5	3	3	4	4
a_3	3	4	4	4	4	4	4	4	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
a_{116}	2	3	3	4	3	3	3	3	2
a_{117}	1	2	2	2	2	3	2	1	1
a_{118}	3	3	4	3	3	3	3	3	2

Security staff competency assessment is depicted in Figure 1.

3 Mathematical model

Definition. Suppose that vector of real components $X = (x_1, x_2, \ldots, x_n), (\forall j) x_j \geqslant 0$, has the meaning of some measurable values and is the set of n criteria values. Any function

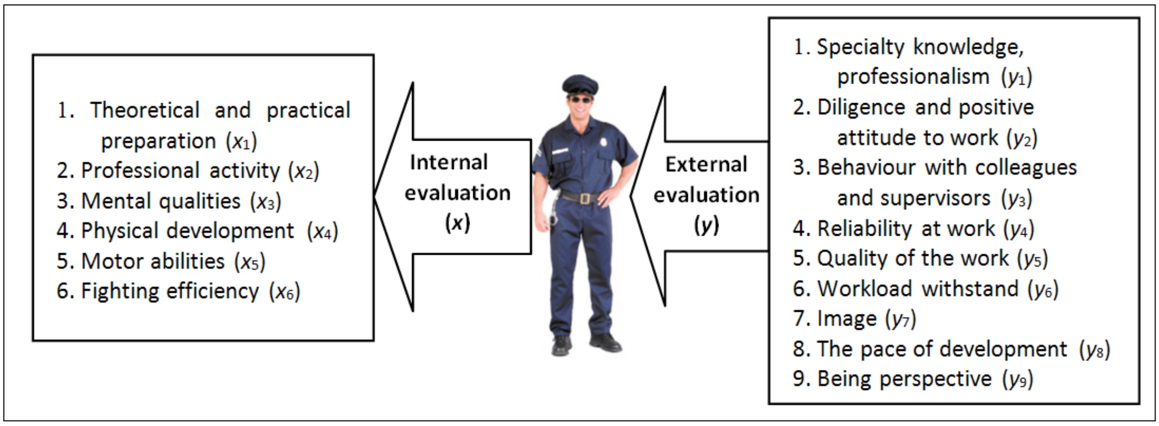

1. Theoretical and practical preparation (x1)
2. Professional activity (x2)
3. Mental qualities (x3)
4. Physical development (x4)
5. Motor abilities (x5)
6. Fighting efficiency (x6)
Internal evaluation (x)
External evaluation (y)
1. Specialty knowledge, professionalism (y1)
2. Diligence and positive attitude to work (y2)
3. Behaviour with colleagues and supervisors (y3)
4. Reliability at work (y4)
5. Quality of the work (y5)
6. Workload withstand (y6)
7. Image (y7)
8. The pace of development (y8)
9. Being perspective (y9)

Figure 1: Research progress

$\varphi: R^n \to R$ will be called **criteria convolution**, if it nondecreasing for each variable, i. e.

$$(\forall j = 1, 2, \dots n, \forall \delta > 0) \;\; \varphi(\dots, x_j + \delta, \dots) \geqslant \varphi(\dots, x_j, \dots). \tag{1}$$

Note, that if for the two alternatives $X^{(1)}$ and $X^{(2)}$ the Pareto dominance is valid $X^{(1)} \succ X^{(2)}$ ($\forall j = 1, 2, \dots n : x_j^{(1)} \geqslant x_j^{(2)}$ and $\exists i : x_i^{(1)} > x_i^{(2)}$), then inequality (1) will be true for the criteria convolution: $\varphi\left(X^{(1)}\right) \geqslant \varphi\left(X^{(2)}\right)$.

Suppose, that permutation $(j_1, j_2, \dots, j_n)$ of natural numbers $\{1, 2, \dots, n\}$ determine the criterion X components x_j preferences $x_{j_1} \succ x_{j_2} \succ \dots \succ x_{j_n}$. For example, permutation $(2, 3, 1)$ means that the most important influence has the second component of criterion $X = (x_1, x_2, x_3)$, and the least influence has the first component.

Definition. Let's say that criteria convolution φ has **preferences feature** $(j_1, j_2, \dots, j_n)$, if all the inequalities are valid:

$$(\forall i < k \;\; \& \;\; \delta > 0) \; \varphi(\dots, x_i + \delta, \dots, x_k, \dots) \geqslant \varphi(\dots, x_i, \dots, x_k + \delta, \dots), \tag{2}$$

when $j_i > j_k$.

Note, that criteria convolutions determined on the basis of weighted averages have preferences feature defined by (2). For example, criteria convolution

$$\varphi(X) = \frac{w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4}{w_1 + w_2 + w_3 + w_4},$$

when $w_2 > w_3 > w_1 > w_4 > 0$ has preferences feature $(2, 3, 1, 4)$.
Convolution

$$\varphi(X) = x_1^{p_1} x_2^{p_2} x_3^{p_3},$$

when $p_2 > p_1 > p_3 > 0$ has preferences feature $(2, 1, 3)$.

3.1 Formulation of optimization problems

Suppose that for the certain alternatives criteria-referenced evaluations are known $X^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}\right)$, $Y^{(k)} = \left(y_1^{(k)}, y_2^{(k)}, \dots, y_m^{(k)}\right)$, $k = 1, 2, \dots, K$ and criteria X, Y preferences are determined: $(j_1, j_2, \dots, j_n)$, $(i_1, i_2, \dots, i_m)$.

Suppose that φ and ψ are convolutions of criteria X and Y respectively, having corresponding preferences features. For each alternative $\left(X^{(k)}, Y^{(k)}\right)$ values of both criteria convolutions

$\varphi\left(X^{(k)}\right)$ and $\psi\left(Y^{(k)}\right)$ are calculated and numbered in ascending order to obtain ranks of alternatives: $r_x^{(k)}$, $r_y^{(k)}$. Note that the lower is the rank, the better is alternative.

Let's denote A_{K_x} and B_{K_y} sets of the best alternatives – those subsets of the set $\{1, 2, \ldots, K\}$ whose items ranks $r_x^{(k)} < K_x$ and $r_y^{(k)} < K_y$ respectively. Here K_x, K_y mean the proportion of the best alternatives to be selected (generally 10–15%, i. e. $K_x, K_y \in [0.10K, 0.15K]$).

Denote $\mathcal{S}_{(j_1,j_2,\ldots,j_n)}$ class of convolutions having preferences feature $(j_1, j_2, \ldots, j_n)$. In this class of functions we'll search for the function φ (analogously ψ). Denote $\left|A_{K_x} \cap B_{K_y}\right|$ number of elements in intersection of sets A_{K_x} and B_{K_y} and formulate the optimization problem:

$$\max_{\substack{\varphi \in \mathcal{S}_{(j_1,j_2,\ldots,j_n)} \\ \psi \in \mathcal{S}_{(i_1,i_2,\ldots,i_m)}}} \left|A_{K_x} \cap B_{K_y}\right|. \tag{3}$$

Notice that if $K_x = K_y = K$ the problem (3) has trivial solution

$$A_{K_x} \cap B_{K_y} = \{1, 2, \ldots, K\}$$

for any convolutions φ, ψ. Decreasing values of parameters K_x, K_y imply the decreasing number of common elements (intersection) of the sets A_{K_x}, B_{K_y}. If this number is less than necessary, we mus increase values of parameters K_x, K_y. This means that criteria X and Y don't match. Choose the numbers $K'_x \leqslant K_x$, $K'_y \leqslant K_y$. The incompatibility of criteria X and Y is quantitatively described by number of all elements of the union of sets $A_{K'_x}$ and $B_{K'_y}$ which don't belong to the intersection in the right side of formula (3), i. e. the alternatives with high rank according to the one and low rank according to another criterion. So the problem (3) could be supplemented with additional one

$$\min_{\substack{\varphi \in \mathcal{S}_{(j_1,j_2,\ldots,j_n)} \\ \psi \in \mathcal{S}_{(i_1,i_2,\ldots,i_m)}}} \left|\left(A_{K'_x} \setminus B_{K_y}\right) \cup \left(B_{K'_y} \setminus A_{K_x}\right)\right|. \tag{4}$$

Notice that the problem (4) also has trivial solution $A_{K'_x} \cup B_{K'_y} = \varnothing$, if $K'_x = K'_y = 1$. In general, this means a certain amount of "fine" for the ignorance of dubious alternatives and numbers $K'_x \leqslant K_x$, $K'_y \leqslant K_y$ must to be the larger.

3.2 Restrictions and general scheme of the research

Formulated optimization problems (3) and (4) aren't easy. Construction of the effective algorithms for their solving is interesting, but poorly investigated issue. Efficiency of these algorithms should strongly depend on the class of criteria convolutions $\mathcal{S}_{(j_1,j_2,\ldots,j_n)}$. In this article we restrict our investigation with the functions from the class of the weighted averages:

$$\varphi_{(j_1,j_2,\ldots,j_n)}\left(x_1, x_2, \ldots, x_n\right) = \frac{\sum\limits_{i=1}^{n} w_{j_i} x_{j_i}}{\sum\limits_{i=1}^{n} w_i}. \tag{5}$$

here $w_{j_1} \geqslant w_{j_2} \geqslant \cdots \geqslant w_{j_n} \geqslant 0$.

Preferences feature $(j_1, j_2, \ldots, j_n)$ is determined by experts estimates of the form $x_{j_1} \succ x_{j_2} \succ \cdots \succ x_{j_n}$. In this article we limit ourselves to calculation of the average estimates for criteria ranks and numbering of x_j accordingly. There are many various ranking methods and their comparison will be our further investigation.

3.3 Solving the personnel selection problem (case study)

The particular managerial task is solved in the paper. It requires finding an acceptable alternative sets having a clear and easily verifiable interpretation of the subject matter. Therefore, the comparison of solutions received by various methods could be their interesting practical suitability criteria and have a long-term value.

Task parameters are as follows:
the number of alternatives (security personnel compared with each other) – $K = 118$;
the number of vector criterion X components (physical measurements of emploees) $n = 6$;
the number of vector criterion Y components (managers assessment) $m = 9$.

Criteria X and Y components priority was determined from expert estimates. Each of 22 experts had to specify priority permutations $x_{j_1} \succ x_{j_2} \succ \cdots \succ x_{j_n}$ and $y_{i_1} \succ y_{i_2} \succ \cdots \succ y_{i_m}$. These evaluations are shown in the Table 3:

Table 3: Criteria X and Y components ranks

Expert	x_1	x_2	x_3	x_4	x_5	x_6	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
1	6	1	5	3	2	4	7	9	4	8	5	3	6	2	1
2	5	2	3	1	6	4	7	9	5	6	8	2	3	4	1
3	5	1	3	2	6	4	5	9	4	6	8	1	7	3	2
4	6	1	5	3	4	2	5	9	7	8	6	3	4	2	1
5	6	1	5	2	3	4	7	9	6	8	5	3	4	2	1
6	5	1	4	2	6	3	5	9	6	8	7	3	4	2	1
7	4	2	3	1	6	5	4	8	5	9	7	2	6	3	1
8	6	2	5	3	4	1	6	8	5	9	7	3	4	2	1
9	5	4	6	3	2	1	7	9	4	8	6	3	5	1	2
10	5	4	6	1	2	3	6	7	4	9	8	3	5	2	1
11	4	1	3	2	6	5	5	8	6	9	7	2	4	3	1
12	6	2	4	1	5	3	6	9	5	7	8	4	3	2	1
13	6	4	5	1	3	2	5	9	6	8	7	3	4	2	1
14	6	3	5	1	4	2	7	8	6	9	5	4	3	2	1
15	4	1	3	2	6	5	5	9	7	8	6	3	2	4	1
16	4	1	3	2	6	5	7	9	8	6	5	4	2	3	1
17	6	1	4	2	3	5	6	8	7	9	4	5	2	3	1
18	6	3	4	1	5	2	6	9	8	7	4	5	2	3	1
19	4	1	5	2	6	3	4	9	8	6	7	1	5	3	2
20	6	2	4	1	5	3	5	9	6	7	8	3	4	2	1
21	5	2	6	3	4	1	6	8	7	9	5	4	3	2	1
22	4	1	6	3	2	5	6	9	7	8	5	4	2	3	1
Σ	114	41	97	42	96	72	127	190	131	172	138	68	84	55	25
$\overline{r_i}$	5.18	1.86	4.41	1.91	4.36	3.27	5.77	8.64	5.96	7.82	6.27	3.09	3.82	2.50	1.14
rank	1	6	2	5	3	4	5	1	4	2	3	7	6	8	9
prefs	$X: \ (1,3,5,6,4,2)$						$Y: \ (2,4,5,3,1,7,6,8,9)$								

Component's X and Y ranks preferences were calculated by method of sum (average) of ranks (i. e. the average place of each component was calculated and preferences drawn up.

3.4 Optimization algorithm

The *heuristic* is used for solving problem (3). Functions $\varphi(X)$ and $\psi(Y)$ will be selected from the class of functions (5) as follows:

$$\varphi(X) = \alpha_1 x_1 + \alpha_3 x_3 + \alpha_5 x_5 + \alpha_6 x_6 + \alpha_4 x_4 + \alpha_2 x_2, \ \alpha_1 \geqslant \alpha_3 \geqslant \alpha_5 \geqslant \alpha_6 \geqslant \alpha_4 \geqslant \alpha_2 > 0,$$

by analogy accordingly to the preference feature given in the last row of Table 3:

$$\psi(Y) = \beta_2 y_2 + \beta_4 y_4 + \cdots + \beta_9 y_9,\ \beta_2 \geqslant \beta_4 \geqslant \beta_5 \geqslant \cdots \geqslant \beta_8 \geqslant \beta_9 > 0.$$

Weighted coefficients α_j and β_i are determined so as to satisfy the normalizing condition

$$\alpha_1 + \cdots + \alpha_6 = \beta_1 + \cdots + \beta_9 = 1. \tag{6}$$

Designate $R^{(k)}(\alpha)$ and $R^{(k)}(\beta), k = 1, 2, \ldots, K$ ranks of numbers $\left\{\varphi\left(X^{(1)}\right), \varphi\left(X^{(2)}\right), \ldots, \varphi\left(X^{(K)}\right)\right\}$ and $\left\{\psi\left(Y^{(1)}\right), \psi\left(Y^{(2)}\right), \ldots, \psi\left(Y^{(K)}\right)\right\}$ $(K = 118)$.

Determine function of ranks discrepancies, i. e. sum of squares of highest ranks differences according to criteria X and Y:

$$CR_{K_x,K_y}(\alpha,\beta) = \sum_{\substack{k \in \{1,2,\ldots,118\}: \\ R^{(k)}(\alpha) \leqslant K_x \ \&\ R^{(k)}(\beta) \leqslant K_y}} \left(R^{(k)}(\alpha) - R^{(k)}(\beta)\right)^2 \tag{7}$$

and minimize this function

$$\min_{\substack{\varphi \in \mathcal{S}_{(j_1,j_2,\ldots,j_n)} \\ \psi \in \mathcal{S}_{(i_1,i_2,\ldots,i_m)}}} CR_{K_x,K_y}(\alpha,\beta). \tag{8}$$

Here K_x and K_y are chosen so as $\left|A_{K_x} \cap B_{K_y}\right| = 12$, since the goal is to select the top 12 security guards.

Minimization problem (8) is solved by re-selecting of values α_j, β_i. For the search of values α_1, α_3, α_5 and β_2, β_4, β_5 evenly divide intervals:

$$\alpha_1 \in \left[\frac{1}{6}, 1\right),\ \alpha_3 \in \left[\frac{1-\alpha_1}{5}, \min\{\alpha_1, 1-\alpha_1\}\right),\ \alpha_5 \in \left[\frac{1-\alpha_1-\alpha_3}{4}, \min\{\alpha_3, 1-\alpha_1-\alpha_3\}\right),$$

$$\beta_2 \in \left[\frac{1}{9}, 1\right),\ \beta_4 \in \left[\frac{1-\beta_2}{8}, \min\{\beta_2, 1-\beta_2\}\right),\ \beta_5 \in \left[\frac{1-\beta_2-\beta_4}{7}, \min\{\beta_4, 1-\beta_2-\beta_4\}\right).$$

Values α_6, α_4, β_3, β_1, β_7, β_6, β_8 are chosen randomly in the intervals constructed analogously. For example, α_4 is uniformly distributed in the interval

$$\alpha_4 \in \left[\frac{1-\alpha_1-\alpha_3-\alpha_5-\alpha_6}{2}, \min\{\alpha_6, 1-\alpha_1-\alpha_3-\alpha_5-\alpha_6\}\right).$$

The remaining values α_2 and β_9 are determined from the normalizing condition (6).

General scheme of the research is represented in Figure 2.

4 Results and Discussion

This heuristic enabled to achieve the criterion (8) value $CR_{12,12}(\alpha,\beta) = 429$ with the following values of weights:

α_1	α_3	α_5	α_6	α_4	α_2
0.31	0.19	0.17	0.17	0.12	0.04

β_2	β_4	β_5	β_3	β_1	β_7	β_6	β_8	β_9
0.41	0.24	0.08	0.08	0.06	0.05	0.03	0.03	0.02

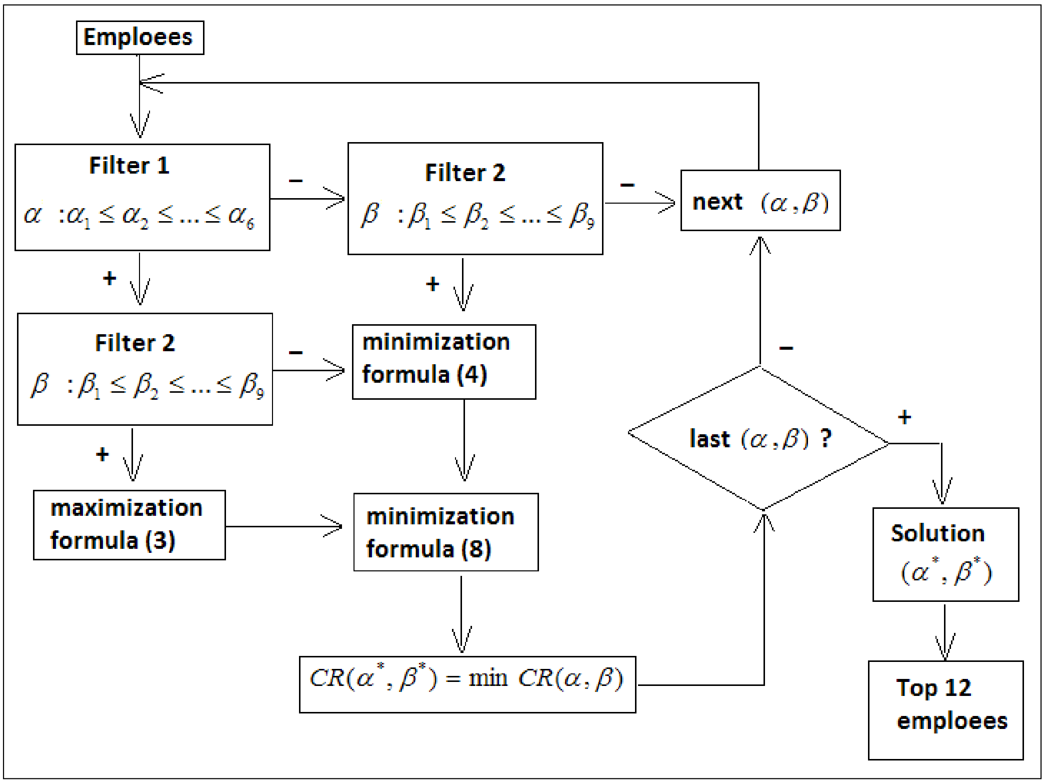

Emploees
Filter 1
$\alpha\ :\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_6$
–
Filter 2
$\beta\ :\beta_1 \leq \beta_2 \leq \ldots \leq \beta_9$
–
next (α,β)
+
+
Filter 2
$\beta\ :\beta_1 \leq \beta_2 \leq \ldots \leq \beta_9$
–
minimization formula (4)
–
last (α,β) ?
+
+
maximization formula (3)
minimization formula (8)
Solution (α^*,β^*)
$CR(\alpha^*,\beta^*) = \min CR(\alpha,\beta)$
Top 12 emploees

Figure 2: Research progress

Top 12 security guards are those belonging to the intersection $A_{K_x} \cap B_{K_y}$: a_{21}, a_{34}, a_{36}, a_{47}, a_{56}, a_{76}, a_{77}, a_{81}, a_{91}, a_{102}, a_{106}, a_{112}.
According to another heuristics we are looking for such convolutions among all $\varphi \in \mathcal{S}_{(j_1,j_2,\ldots,j_n)}$ and $\psi \in \mathcal{S}_{(i_1,i_2,\ldots,i_m)}$ which maximize (3) and minimize (4). Since there are many solutions of problem (3)–(4), among the entire sample of solutions we are looking for one that minimize the sum (7). Let's choose $K'_x = K'_y = 4$. Then we reached even lower criterion (8) value $CR_{12,12}(\alpha,\beta) = 406$ and the following X and Y weights were obtained:

α_1	α_3	α_5	α_6	α_4	α_2
0.26	0.18	0.18	0.17	0.13	0.08

β_2	β_4	β_5	β_3	β_1	β_7	β_6	β_8	β_9
0.21	0.12	0.12	0.12	0.10	0.10	0.09	0.07	0.07

Top 12 security guards were selected:
a_{21}, a_{34}, a_{36}, a_{47}, a_{56}, a_{76}, a_{77}, a_{81}, a_{91}, a_{102}, a_{106}, a_{111}. This solution differs from the first solution by only one employee – in place of a_{112} is an employee a_{111}. Another 4 security guards were distinguished having high rank ($\geqslant 4$) by one and low rank by another criterion: a_6, a_{26}, a_{46}, a_{54}.

Attention is drawn to the fact that future studies are interesting for the aspect of development of general design of the study for the similar problems, since a very different assessments have similar structure. Briefly this scheme can be described as follows. The same objects are independently characterized by several vector criteria (in this paper there are two, but the methodology is easily generalized for greater number of vector criteria). Vector components significance is not the same and significance hierarchy $(j_1, j_2, \ldots, j_n)$ is known (usually from the expert estimates). The proposed methodology allows the construction of the alternatives evaluation criteria by application of general design of the study and using all the existing information.

Bibliography

[1] van Dijk, J. (2008); *The world of crime. Los Angeles*, CA: Sage Publications.

[2] Bureau of Justice Statistics. (2008); Census of state and federal correctional federal correctional facilities, 2005. Washington, DC: *Office of Justice Programs, U.S. Department of Justice.*

[3] Garland B.; Hogan N.L.; Kelley T.; Kim B.; Lambert E.G. (2013); To Be or Not to Be Committed: The Effects of Continuance and Affective Commitment on Absenteeism and Turnover Intent among Private Prison Personnel. *Journal of Applied Security Research*, 8(1). 65-88.

[4] Dadelo, S.; Turskis, Z.; Zavadskas, E. K.; Dadeliene, R. (2012); Multiple criteria assessment of elite security personal on the basis of aras and expert methods. *Journal of economic computation and economic cybernetics studies and research*, 46(4). 65–88.

[5] Brucker P.; Qu R.; Burke E. (2011); Personnel scheduling: Models and complexity. *European Journal of Operational Research*, 210. 467–473.

[6] van den Bergh J.; Belien J.; De Bruecker P.; Demeulemeester E.; De Boeck L. (2013); Personnel scheduling: A literature review. *European Journal of Operational Research*, 226. 367–385.

[7] Carlos M. (2011); A survey of the European security market. *Economics of Security Working Paper* 43, Berlin: Economics of Security.

[8] Dadelo, S.; Turskis, Z.; Zavadskas, E. K.; Dadeliene, R. (2013); Integrated multi-criteria decision making model based on wisdom-of-crowds principle for selection of the group of elite security guards. *Archives of Budo*, 9(2). 135–147.

[9] Ryan, M.; Mann, C.; Stilwell, A. (2003); *The Encyclopedia of the World's Special Forces: Tactics, history, strategy, weapons.* Amber Books Ltd., London.

[10] Dessler, G. (1999); *Essentials of Human Resource Management*, Upper Saddle River, NJ: Prentice Hall, 298.

[11] Suresh S.; Antony J.; Kumar M.; Douglas A. (2012); Six Sigma and leadership: some observations and agenda for future research, *The TQM Journal*, 24(3). 231–247.

[12] Dadelo, S. (2005); Czynniki determinujce kompetencjae pracownikow ochrony na Litwie. AWF Warszawa–Vilnius, 2005 [in Polish, abstract in English, in Lithuanian]

[13] Zavadskas, E. K.; Turskis, Z. (2011); Multiple criteria decision making (MCDM) methods in economics: an overview, *Technological and Economic Development of Economy*, 17(2). 397–427.

[14] Kosareva, N.; Krylovas, A. (2013); Comparison of accuracy in ranking alternatives performing generalized fuzzy average functions, *Technological and Economic Development of Economy*, 19(1). 162–187.

[15] Sakalas, A. (2003); *Personnel management.* Vilnius: Margi rastai. [in Lithuanian]

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):160-171, April, 2014.

A Multi-objective Optimization Algorithm of Task Scheduling in WSN

L. Dai, H.K. Xu, T. Chen, C. Qian, L.J. Xie

Liang Dai, Hongke Xu, Qian Chao
School of Electronic and Control Engineering
Chang'an University
Xi'an 710064, China
E-mail: ldai1981@gmail.com, xuhongke@chd.edu.cn

Ting Chen
School of Information Engineering
Chang'an University
Xi'an 710064, China
E-mail: tchen@chd.edu.cn

Lijing Xie
Information and Navigation College
Air Force Engineering University
Xi'an 710077, China

Abstract: Sensing tasks should be allocated and processed among sensor nodes in minimum times so that users can draw useful conclusions through analyzing sensed data. Furthermore, finishing sensing task faster will benefit energy saving. The above needs form a contrast to the lower efficiency of task-performing caused by the failure-prone sensor. To solve this problem, a multi-objective optimization algorithm of task scheduling is proposed for wireless sensor networks (MTWSN). This algorithm tries its best to make less makespan, but meanwhile, it also pay much more attention to the probability of task-performing and the lifetime of network. MTWSN avoids the task assigned to the failure-prone sensor, which effectively reducing the effect of failed nodes on task-performing. Simulation results show that the proposed algorithm can trade off these three objectives well. Compared with the traditional task scheduling algorithms, simulation experiments obtain better results.

Keywords: Wireless Sensor Networks (WSN); task scheduling; multi-objective optimization; improved NSGA-II .

1 Introduction

Sensing tasks should be allocated and processed among sensor nodes in minimum times so that users can draw useful conclusions through analyzing sensed data. Furthermore, finishing sensing task faster will benefit energy saving, which is critical in system design of wireless sensor networks. The primary objective of task scheduling in wireless sensor networks is to find an optimal strategy of splitting the original tasks received by SINK into a number of sub-tasks as well as distributing these sub-tasks to the sensors in the right order. The directed acyclic graph [1], independent task sets[2] and divisible load theory [3] are usually used as modeling tools for task scheduling in wireless sensor networks, but these models only take the makespan as the main objective, and assign the task to sensors. However, wireless sensor networks are widely applied to both abominable and military environments. Meanwhile, the complexity of networks, the limited energy of sensors, and potential physical or logical faults, bring challenge to task scheduling of wireless sensor networks.

Wireless sensor network is one kind of the widely used distributed real-time systems. How to assign tasks of system to sensors in unstable and unreliable network environment, and guarantee their deadlines is one of the key techniques in wireless sensor networks. In wireless sensor network environments, QoS guided task scheduling problem is complex and challenging, especially when the tasks have multiple needs.

In this paper, a multi-objective optimization algorithm of task scheduling is proposed for wireless sensor networks. It is the first time that the NSGA-II [4] algorithm is used to analyze the task scheduling for wireless sensor networks. Based on the characteristics of wireless sensor networks, makespan optimization, the energy-consuming balance optimization and task-performing probability optimization are included. A mathematical model used to optimize the task scheduling problem by NSGA-II was built and the solution was presented, and a detailed process to solute the multi-objective programming model is put forwards. The problem is solved with a multi-objective genetic algorithm (GA) optimization method combined with linear programming (LP) and a group of pareto solutions are provided.

2 Related work and motivation

Wireless sensor networks have restrictions due to energy, memory, and communication ability. We should realize the goal of the improvement and enhancement of sensor networks performance in real time, economy, power aware and harmony.

Some studies over the past decade have been conducted to reduce the overall energy consumption for task scheduling in wireless sensor networks by using diverse techniques [5-14]. Heemin presents an energy-efficient task assignment and migration framework for sensor networks. With the proposed framework, optimal task transformation and assignment is sought so as to minimize given cost function[6]. Younis M presented an optimization scheme for task allocation to gateways [7]. The task allocation problem is modeled as a zero-one nonlinear program. He study system partitioning of computation to improve the energy efficiency of a wireless sensor networking application. Wang explored system partitioning between the sensor cluster and the base station, employing computation-communication tradeoffs to reduce energy dissipation[8]. Also he showed that system partitioning of computation within the cluster can also improve the energy efficiency by using dynamic voltage scaling. Tian presented a task mapping and scheduling solution for real-time applications (RT-MapS) in WSNs[9]. RT-MapS incorporates wireless channel modeling, hyper-DAG extension, concurrent task mapping, communication and computation scheduling, and dynamic voltage scaling (DVS) methods. Tian also presented a task mapping and scheduling solution for energy-constrained applications in WSNs, energy-constrained task mapping and scheduling (EcoMapS)[10]. EcoMapS incorporates channel modeling, concurrent task mapping, communication and computation scheduling, and sensor failure handling algorithm. The performance of EcoMapS is evaluated through simulations with randomly generated directed acyclic graphs (DAG).Yu proposed an energy-balanced allocation of a real-time application onto a single-hop cluster of homogeneous sensor nodes connected with multiple wireless channels[11]. An epoch-based application consisting of a set of communicating tasks is considered. Each sensor node is equipped with discrete dynamic voltage scaling (DVS). The time and energy costs of both computation and communication activities are considered. He proposed both an Integer Linear Programming (ILP) formulation and a polynomial time 3-phase heuristic. Gu presented an application independent task mapping and scheduling solution in multi-hop Video sensor networks (VSNs) that provides real-time guarantees to process video feeds[12]. The processed data is smaller in volume which further releases the burden on the end-to-end communication. Using a novel multi-hop channel model and a communication scheduling algorithm, computation tasks and associated communication events are scheduled simultaneously with a dynamic critical-path scheduling algorithm. Dynamic voltage scaling (DVS) mechanism is implemented to further optimize energy consumption. Xie developed a novel task allocation strategy called BEATA (Balanced Energy-Aware Task Allocation) for collaborative applications running on heterogeneous networked embedded systems[13]. The BEATA algorithm aims at blending an energy-delay efficiency scheme with task allocations, thereby making the best tradeoffs between energy savings and schedule lengths. Besides, He introduced the concept of an energy-adaptive window, which is a critical parameter in the BEATA strategy. By fine-tuning the size of the energy-adaptive window, users can readily customize BEATA to meet their specific energy-delay trade-off needs imposed by applications. Further, he built a mathematical model to approximate energy consumption caused by both computation and communication activities. A task allocation framework for Underwater Acoustic Sensor Networks (UW-ASNs) that participate as a team to accomplish critical missions is developed by Kulkarni[14]. The team formed as a result of this task allocation framework is the subset of all deployed AUVs that is best suited to accomplish the mission while adhering to the mission constraints.

Most of the existing task scheduling algorithms is normal ones without consideration of harmony and reliability in our research. This problem will need to be further gone into for improving the wireless sensor networks characteristics on real time, economy, power aware and harmony.

The above discussed task scheduling algorithms are mainly for the problem of assignment tasks with the objective of minimizing the makespan and energy consumption, effective algorithms is discussed in these papers, and these algorithms will be able to distribute tasks to nodes with the expected value of the minimum earliest-finish-time and make them executed parallel and efficiently. However, tasks distributed to remote sensors may be unable to complete because of physical failures or attack, and the complexity, dynamics and open deployment of the wireless environment increases the possibility of that happening. Existing task scheduling policies are not consider this question, so tasks will be assigned to the sensors with lower reliability to perform, what make measurement tasks automatically cease to be in force due to sensor failure, and then lowered the efficiency and QoS of wireless sensor networks.

Compared with the previous research, the main contributions of this paper are presenting the concept of " task-performing probability ", and proposing a multi-objective optimization task scheduling algorithm for wireless sensor networks considering makespan optimization, the energy-consuming optimization and task-performing probability optimization simultaneously. Consideration shall also be given to ensuring a reasonable distribution of tasks to reliable sensors, taking into account the makespan and energy consumption. This algorithm schedules tasks avoiding allocation some tasks to unreliable sensors, thus effectively lowering the influence of sensor failure on task performing.

3 The Task Scheduling Model for WSNs

In this section, we describe mathematical models which were built to represent a task scheduling framework.

3.1 Task Model

Generally, a wireless sensor network consists of a set of heterogeneous sensors in abominable or military environments. Sensor nodes always break down due to hardware failure, software error, energy exhaustion and disturb from outer environment, so the precondition of the tasks accomplished successfully is the sensors to provide a stable hardware and software needed to perform the given tasks. Each sensor is in a "active" or "inactive" state at a given time. The task is impossibly accomplished on a inactive sensor, and the status information also will be missed or ineffective.

Let λ_i be the failure probability of sensor S_i , then the task-performing probability of the tasks accomplished successfully on sensor S_i is $1 - \lambda_i$. We assumed that the failure process of each sensor is independent and yields to poisson distribution.

We defined the task set as $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$. According to the previous assumption, the tasks in the task set is independent each other.

3.2 Makespan Model

Suppose that a wireless sensor network consists of m sensors, $S = \{S_1, S_2, \cdots, S_m\}$. There are n independent tasks competing the m sensors. We aim to scheduling the n tasks to the m sensors reasonably, to make the minimum makespan of the tasks. And we characterize a $n \times m$ matrix X satisfying the scheduling results. When $x_{i,j} = 1$ ($x_{i,j} \in X$), it meant that we schedule the task τ_i on sensor S_j to process, otherwise, $x_{i,j} = 0$.

The task-processing time of each sensor can be estimated by forecasting techniques and history status based on task type. The task-processing time can be represented by a $n \times m$ matrix Y, in which the matrix element $y_{i,j}$ represents the estimated task-processing time that task τ_i runs on sensor S_j.The task-processing time of sensor S_j is the sum of all the tasks processing time that the tasks run on S_j.

It can be expressed as follow:

$$T_j = \sum_{i=1}^{n} y_{i,j} \tag{1}$$

Then, the makespan that the n tasks scheduled to the m sensors according to the scheduling result X is expressed as:

$$T(\Gamma, S, X) = Max_{S_i \in S}[R(S_j)] \tag{2}$$

3.3 Energy Consumption Model

Because of the limited energy in wireless sensor networks, the research on low power technology and long lifetime is pivotal in the architecture of wireless sensor networks. The energy consumption of wireless sensor networks mainly composed of communication and task-performing energy consumption.

Communication energy consumption is relevant to the minimum energy-communication for a given standard distance P_0 and the distance $d_{i,j}$ between sensor S_i and sensor S_j.

$$P_{i,j} = (4\pi)^2 d_{i,j}^2 \beta P_0 / d_0^2 G_t G_r \lambda^2 \tag{3}$$

where G_t and G_r are the antenna gains (with respect to an isotropic radiator) of the transmitting and receiving antennas respectively, λ is the wavelength, and β is the energy consumption factor. Since $(4\pi)^2 \beta P_0 / G_t G_r \lambda^2$ is constant, one unit of communication energy consumption is indicated as $d_{i,j}^2 / d_0^2$.

Task-performing energy consumption of sensor S_j for finishing all its tasks is:

$$E_j = C_j T_j \tag{4}$$

where C_j is the task-performing energy consumption of one certain task in one unit time.

The relative task-performing energy consumption of sensor S_j can be represented as follow:

$$R_j = E_j / Max_{S_i \in S}(E_j) \tag{5}$$

Excluding energy consumption, the lifetime of wireless sensor networks is the critical estimated part in task scheduling too. To extend network lifespan, we should balance the energy consumption of every sensor during the period of ordinary operation. We can measure the uniformity of sensors' residual energy based on entropy theory.

The uniformity of sensors' residual energy after data transmission in t can be represented as follow:

$$H_{it} = -\sum p(E_{it}) \log p(E_{it}) \tag{6}$$

where E_{it} is the residual energy of sensor S_j in t, and H_{it} is the value of residual energy entropy in the network. The higher H_{it}, the more residual energy in the network, then the longer lifetime of the network.

Thus, we define the evaluation function of energy consumption for sensor S_j is that:

$$C(S_i) = -d_{i,j}^2 R_j H / d_0^2 \tag{7}$$

The less $C(S_i)$ means that the energy consumption of data transmission and executing tasks has less effect on the residual energy in the network.

The evaluation function of energy consumption for the network is defined as:

$$C(L, S, X) = \sum_{i=1}^{m} C(S_i) \tag{8}$$

The smaller the value of $C(L, S, X)$, the longer network lifetime.

3.4 Task-performing Probability

The concept of task-performing probability comes from the survivability of distributed system. Survivability represents that tasks are capable of being performed steadily and regularly in the networks.

Definition. The task-performing probability means the probability that task could be accomplished successfully on specified sensor.

Let $P(\tau_i, S_j)$ be the task-performing probability that task τ_i accomplished on sensor S_j. In accordance with the illustrations in the previous section, the task-performing probability is $\exp(-\lambda_j t)$ that sensor be in the normal (ACTIVE) state within t. Because the tasks can be performed normally only when the given sensor is in a "normal" state, so we can get:

$$P(\tau_i, S_j) = \exp(-\lambda_j R(S_j)) \tag{9}$$

Let $P(\Gamma, S, X)$ is the task-performing probability the n tasks to the m sensors according to the scheduling result X, then we can get:

$$P(\Gamma, S, X) = \exp[-\sum_{i=1}^{i=n} \sum_{j=1}^{j=m} \lambda_j x_{i,j} R(S_j)] \tag{10}$$

Taking the task-performing probability as goal is to avoid the tasks scheduled to the sensors with low reliability.

Lowering the influence of sensor failure on task performing, then to maximize the task-performing probability of task set Γ, that is, we will maximize $P(\Gamma, S, X)$ by our task scheduling algorithm to obtain the most suitable X. Thus, the quality of service for wireless sensor networks will be improved.

In eq. 9, let $L(\tau_i, S_j) = -\lambda_j R(S_j)$, we can see that if we want to increase the value of $P(\tau_i, S_j)$, then we must lower the $L(\tau_i, S_j)$.

Let $L(\Gamma, S, X) = \sum\limits_{i=1}^{i=n} \sum\limits_{j=1}^{j=m} \lambda_j x_{i,j} R(S_j)$, then we can see that maximizing $P(\Gamma, S, X)$ means to Minimize $L(\Gamma, S, X)$.

The task scheduling algorithm taking the task-performing probability as goal makes the value of $L(\Gamma, S, X)$ as minimal as possible.

4 Optimal Task Scheduling Based on Improved NSGA-II

4.1 Multi-Objective Optimization Problem

In the case of a Multi-Objective Optimization (MOO) problem, there is usually no single solution that is optimum with respect to all objectives. There are a set of optimal solutions known as pareto optimal solutions. Without additional information, all these solutions are equally satisfactory. The goal of MOO is to find as many of these solutions as possible.

In the research of pareto front many methods have been proposed. David Schaffer first implemented a multi-objective evolutionary algorithm called the vector-evaluated genetic algorithm or VEGA in 1984. His algorithm started off well but tended to converge to a single solution. To prevent the convergence to a single solution, Goldberg and Richardson suggested using a non-dominated sorting procedure coupled with a niching strategy called sharing. Sharing takes into account that individuals in the same niche must share the available resources. This concept is integrated into the pareto genetic algorithm by increasing the cost of chromosomes as a function of their distance from each other. Closely grouped chromosomes will find their costs increased more than chromosomes that are spaced far apart.

The multi-objective genetic algorithm (MOGA)[15] starts by finding all non-dominated chromosomes of a population and gives them a rank of one. These chromosomes are removed from the population. Next, all the non-dominated chromosomes of this smaller population are found and assigned a rank of two. This process continues until all the chromosomes are assigned a rank. The largest rank will be less than or equal to the size of the population. Usually, there are many solutions that have the same rank. The selection procedure uses the chromosome ranking to determine the mating pool. MOGA also uses niching on the cost to distribute the population over the pareto optimal region [16].

Non-dominated Sorting Genetic Algorithm (NSGA)[17] ranks chromosomes in the same manner as MOGA. The NSGA algorithm then calculates a unique value. This unique value is related to the distance between each solution and its two closest neighbors. Distance may be calculated from the variable values or the associated costs. The resulting values are scaled between 0 and 1 and subtracted from the cost. Further information about the evolution of this method can be found in [4,16,17].

As discussed above, the goal of MTWSN is to find the solutions giving the best trade-off between the three conflict objectives, known as Pareto optimal. MOGAs[4,17] are recognized to be well qualified to tackle multi-objective optimization problems. NSGA-II[4] is one of the most popular MOGAs. Some concepts of multi-objective optimization problem are defined as follows.

Definition 1 (Multi-objective optimization problem). Given an n-dimensional decision vector $x = \{x_1, x_2, \cdots, x_n\}$ in the solution space X, find a vector x^* that maximizes a given set of k objective functions $f(x^*) = \{f_1(x^*), f_2(x^*), \cdots, f_k(x^*)\}$. The solution space X is generally restricted by a series of constraints, such as $g_j(x^*) = b_j$ for $j = 1, 2, \cdots, m$.

Definition 2 (Dominance). A vector $u = \{u_1, u_2, \cdots, u_n\}$ is said to dominate a vector $v = \{v_1, v_2, \cdots, v_n\}$ if and only if u is partially less than v, i.e., $\forall\ i = 1, 2, \cdots, n,\ u_i \leq v_i \wedge \exists i =$

$1, 2, \cdots, n,\ u_i \leq v_i$.

Definition 3 (Pareto optimal solution). A solution $x_u \in X$ is said to be pareto optimal if and only if there is no $x_v \in X$ for which $v = f(x_v) = \{v_1, v_2, \cdots, v_n\}$ dominates $u = f(x_u) = \{u_1, u_2, \cdots, u_n\}$.

Definition 4 (Pareto optimal set and front). Let $A \subseteq X$. The nondominated set regarding A, represented by X_p, is defined as $X_p = \{z \in A | z\ is\ nondominated\ regardingX\}$. The corresponding objective function values in the objective space are defined as $Y_p = F(X_p) = \{f(z) | z \in X_p\}$, where X_p is called the pareto optimal set and Y_p is called the cohere pareto optimal front.

However, the solutions found by original NSGA-II are likely to be inferior or only comparable to that by classical heuristic search algorithms because of premature convergence. To find perfect solutions, a delete operator for NSGA-II is proposed to enhance the search capability. When selecting the elitist, if neither of the two individuals in a population wins out and their genes are the same, then delete one of them. Furthermore, a circulation selection is presented to preserve excellent genes of the parent population. Suppose there are K individuals in a population $(ind_1, ind_2, \cdots, ind_K)$ when the crossover operations are carried out. The first time the operation is carried out with (ind_1, ind_2) as parents, the second time (ind_2, ind_3) are taken as parents, and so on. Similarly, the last child is done by (ind_K, ind_1). By this way, K offspring individuals are generated. The genes of each parent are inherited by two offspring individuals, thus avoiding the loss of excellent solutions.

4.2 An Efficient Task Scheduling Algorithm

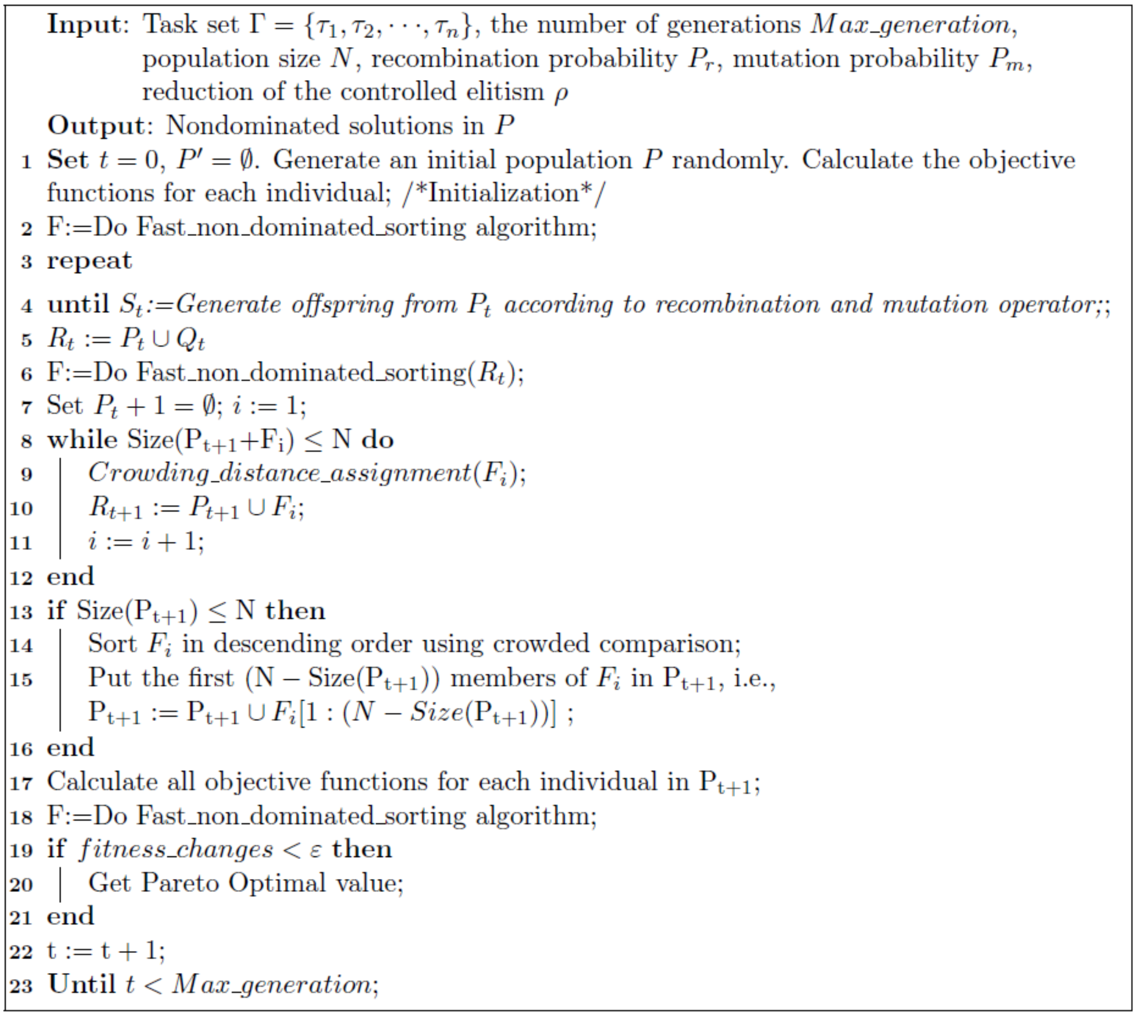
The motivation here is to provide the user with a set of pareto optimal solutions by the NSGA-II algorithm and give it the flexibility to choose the best possible solution from this set, depending on the specific application requirements.

Now, we can construct the cover set using the optimal pareto solutions generated by the improved NSGA-II algorithm. The chromosomes of a genetic algorithm contain all the building blocks to a solution for the genetic operators and the fitness function. In our implementation, each individual node is represented by a one-bit binary number called gene. This one-bit gene defines the status of the sensors as follows:

$$x_{i,j} = \begin{cases} 1,\ if\ task\ \tau_i\ runs\ on\ S_j \\ 0,\ otherwise \end{cases} \tag{11}$$

We assign each individual with three fitness functions, makespan, energy consumption and task-performing probability. By introducing the non-dominated sorting approach and the crowded distance operator the replacement scheme is executed. First, a combined population $R_t = P_t \cup Q_t$ is formed with the parent population P_t and the offspring population Q_t, where t is the number of generation. Therefore the population R_t will be of size 2N. And it is sorted according to the non-domination and crowded comparison. By adding solutions from the first front till the size exceeds N, the new parent population P_{t+1} is formed. After that, the solutions of the last accepted front are sorted according to the crowed comparison and the first $(N - Size(P_{t+1}))$ points are picked. In this way, the population P_{t+1} of size N is constructed. Subsequently, it is used for the circulated selection, crossover, and mutation to create a new population Q_{t+1} of size N. The recombination operator used in this paper is K-point crossover. After recombination, the mutation operator is applied to complement some genes in the chromosomes of the child randomly.

This entire process is repeated until the difference of fitness values among the current pareto optimal set and the previous one is less than a chosen precision (ε). The main procedure of



```
Input: Task set $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$, the number of generations $Max\_generation$,
       population size $N$, recombination probability $P_r$, mutation probability $P_m$,
       reduction of the controlled elitism $\rho$
Output: Nondominated solutions in $P$
Set $t = 0$, $P' = \emptyset$. Generate an initial population $P$ randomly. Calculate the objective
functions for each individual; /*Initialization*/
F:=Do Fast_non_dominated_sorting algorithm;
repeat
until $S_t$:=Generate offspring from $P_t$ according to recombination and mutation operator;;
$R_t := P_t \cup Q_t$
F:=Do Fast_non_dominated_sorting($R_t$);
Set $P_t + 1 = \emptyset$; $i := 1$;
while Size($\mathrm{P_{t+1}}$+$\mathrm{F_i}$) $\leq$ N do
    $Crowding\_distance\_assignment(F_i)$;
    $R_{t+1} := P_{t+1} \cup F_i$;
    $i := i + 1$;
end
if Size($\mathrm{P_{t+1}}$) $\leq$ N then
    Sort $F_i$ in descending order using crowded comparison;
    Put the first (N $-$ Size($\mathrm{P_{t+1}}$)) members of $F_i$ in $\mathrm{P_{t+1}}$, i.e.,
    $\mathrm{P_{t+1}} := \mathrm{P_{t+1}} \cup F_i[1 : (N - Size(\mathrm{P_{t+1}}))]$ ;
end
Calculate all objective functions for each individual in $\mathrm{P_{t+1}}$;
F:=Do Fast_non_dominated_sorting algorithm;
if $fitness\_changes < \varepsilon$ then
    Get Pareto Optimal value;
end
t := t + 1;
Until $t < Max\_generation$;
```

Algorithm 1: A Genetic Approach to Task Scheduling Algorithm

algorithm is described as algorithm 1:

5 Performance Evaluation

In this section, we will present the simulation results as the performance evaluation of our proposed task scheduling algorithm. The performance of our proposed algorithm is compared with RT-Maps [9], EcoMaps [10], and EBTA [11] task scheduling algorithms in wireless sensor networks in terms of the makespan, and energy consumption of task scheduling. For our experiments, we use 50 sensor nodes and a sink, which is not power constrained. Nodes are distributed in a 100X100 meter square area yielded to random poisson distribution. To evaluate the performances of the task scheduling algorithms properly, experimental parameters of the four algorithms were set as the same ones.

The performance of the genetic algorithm is greatly affected by a number of factors, such as the population size, the probability of mutation and crossover, and the method of scheduling. We have run a number of experiments with different values of these parameters to determine the optimal set for our network size. Finally, the GA parameters used in our simulations are listed in table 1. Although we allow the GA to run for a maximum of 100 generations, we have observed

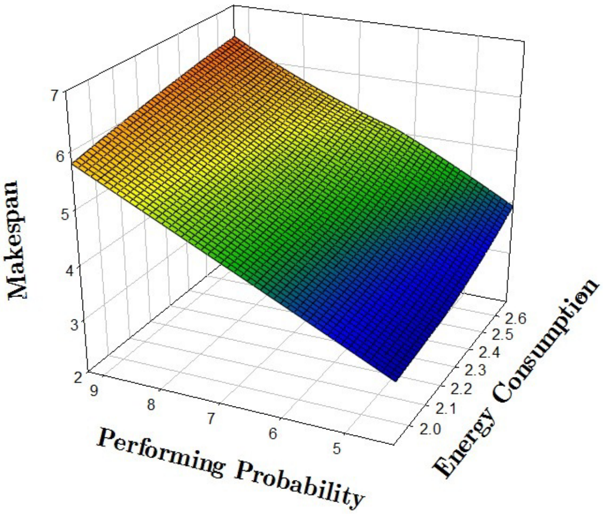


Figure 1: Schematic diagram of statistic information for scheduling

that the best solution is typically found within 40 generations.

Table 1: GA parameters used in Simulation

Parameter	Value
Population size	200
Recombination rate	0.9
Mutation rate	0.005
Reduction rate	0.5

The simulation results are shown in Fig. 1 to Fig. 3.

Figure 1 delineates the statistical information of the proposed task scheduling algorithm. We investigates the space of optimum configuration parameters (makespan, energy consumption and task-performing probability), using unconstrained optimization, to highlight the trade-offs faced by the WSNs considered. In order to solve the optimization problem, we chose a pareto-compliant ranking method based on evolutionary techniques, namely the Non-dominated Sorting Genetic Algorithm-II (NSGA-II). Fig. 1 shows the result of the joint optimization of the three objective functions. These results show that the higher the balanced energy consumption or the higher the probability of task performing, the shorter the makespan. The important outcome of the results in Fig. 1 is that it provides details about the optimal network configuration, since tuning the network with the parameters derived from the execution of the NSGA-II algorithm guarantees that the network performance is not biased toward any one of the performance indicators.

The makespan metric is compared between proposed task scheduling algorithms for wireless sensor networks in Figure 2. It explains that the makespan for wireless sensor networks is lower for proposed algorithms than RT-Maps [9], EcoMaps [10], and EBTA [11]. This is because that in proposed algorithm the tasks have been finished on minimum makespan as the target function.

Figure 3 shows the comparative performance in terms of energy consumption of sensor nodes.With the increase of the number of tasks in the network, the energy consumption of task scheduling and data transmission in the algorithms increases. However, the energy consumption in the proposed algorithm is lower than RT-Maps [9], EcoMaps [10], at fixed number of tasks.

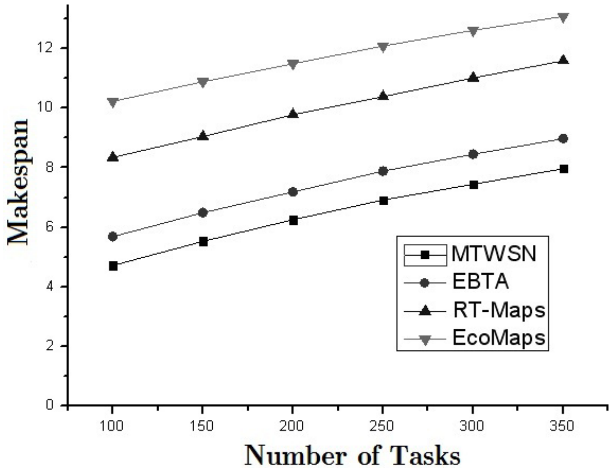


Figure 2: Makespan of different scheduling algorithms

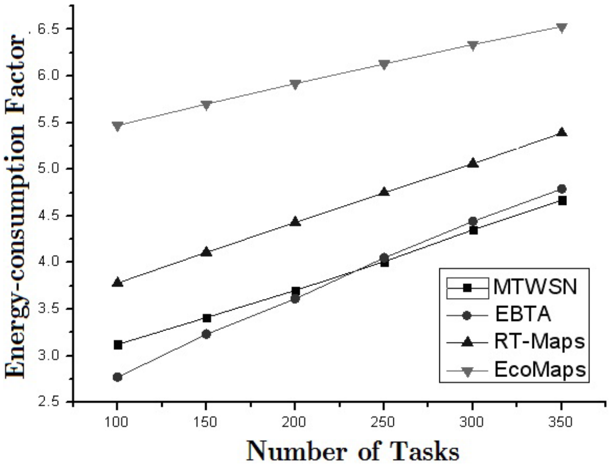


Figure 3: Energy-cosumption factor of different scheduling algorithms

Our proposed task scheduling algorithm results in lowest energy consumption for higher number of tasks. This shows the efficiency of the proposed algorithm.

6 Conclusions

The optimization of task scheduling is studied to reduce the energy consumption and ensure the effective information acquisition in wireless sensor network. A multi-objective optimization algorithm of task scheduling is proposed for wireless sensor networks in this paper. It is the first time that the NSGA-II algorithm is used to analyze the task scheduling for wireless sensor networks. Based on the characteristics of wireless sensor networks, makespan optimization, the energy-consuming balance optimization and task-performing probability optimization were included. A mathematical model used to optimize the task scheduling problem by NSGA-II was built and the solution was presented, and a detailed process to solute the multi-objective programming model was put forwards.

Bibliography

[1] Z. Zeng, A. Liu, D. Li (2008), A Highly Efficient DAG Task Scheduling Algorithm for Wireless Sensor Networks, *In Proc. of ICYCS 2008*,Zhang Jia Jie , Hunan , China, 570-575.

[2] J. Lin, W. Xiao, F. L. Lewis (2009), Energy-Efficient Distributed Adaptive Multisensor Scheduling for Target Tracking in Wireless Sensor Networks. *IEEE Transactions on Instrumentation and Measurement*, 58(6):1886 - 1896.

[3] L. Dai, Y. Chang, Z. Shen (2011), An Optimal Task Scheduling Algorithm in Wireless Sensor Networks, *Int J Comput Commun*, ISSN 1841-9836, 6(1):101-112.

[4] K. Deb, A. Pratap, S. Agarwal (2002), A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182-197.

[5] J. Mao, C. G. Cassandras, Q. Zhao (2007), Optimal Dynamic Voltage Scaling in Energy-Limited Nonpreemptive Systems with Real-Time Constraints. *IEEE Transactions on Mobile Computing*, 6(6):678 - 688.

[6] P. Heemin, B. S. Mani (2003), *Energy-efficient Task Assignment Framework for Wireless Sensor Netwoks*, UC Los Angeles: The Berkeley Electronic Press.

[7] M. Younis, K. Akkaya, A. Kunjithapatham (2003), Optimization of Task Allocation in a Cluster-based Sensor Network. *In Proc. of the Eighth IEEE International Symposium on Computers and Communication*, Netherlands: IEEE Computer Press, 329-334.

[8] A. Wang, A. Chandrakasan (2002), Energy-efficient DSPs for Wireless Sensor Networks. *IEEE Signal Processing Magazine*, 19(4):68-78.

[9] Y. Tian, J. Boangoat, E. Ekici (2006), Real-time Task Mapping and Scheduling for Collaborative In-network. *In Proc. of the 20th International Parallel and Distributed Processing Symposium.* Rhodes Island: IEEE Computer Press, 1-10.

[10] Y. Tian, E. Ekici (2005), Energy-constrained Task Mapping and Scheduling in Wireless Sensor Networks. *In Proc. of the Workshop on Resource Provisioning and Management in Sensor Networks.* Washington, D C: IEEE Computer Society, 211-218, 2005.

[11] Y. Yu, V. K. Prasanna (2005), Energy-balanced Task Allocation for Collaborative Processing in Wireless Sensor Networks. *ACM/Kluwer Mobile Networks and Applications Journal*, 10(1):115-131.

[12] Y. Gu (2007), Real-time Multimedia Processing in Video Sensor Networks, *Signal Processing : Image Communication*, 22(3):237-251.

[13] T. Xie, X. Qin (2008), An Energy-Delay Tunable Task Allocation Strategy for Collaborative Applications in Networked Embedded Systems. *IEEE Transactions on Computers*, 57(3):329 - 343.

[14] I. S. Kulkarni, D. Pompili (2010), Task Allocation for Networked Autonomous Underwater Vehicles in Critical Missions. *IEEE Journal on Selected Areas in Communications*, 28(5):716-727.

[15] C. M. Fonseca, P. J. Fleming (1993), Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, *In Genetic Algorithms: Proceedings of the Fifth International Conference*, San Mateo, CA: Morgan Kaufmann, 416-423.

[16] R. L. Haupt, S. E. Haupt (2004), Practical Genetic Algorithms. John Wiley & Sons.

[17] N. Srinivas, K. Deb (1995), Multiobjective optimization using nondominated sorting in genetic algorithms. *Journal of Evolutionary Computation*, 2(3):221-248.

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):172-186, April, 2014.

A New Information Filling Technique Based On Generalized Information Entropy

S. Han, L. Chen, Z. Zhang, J.-X. Li

Shan Han*, Lin Chen, Zhi Zhang, Jianxun Li
Department of Automation, Shanghai Jiao Tong University,
Key Laboratory of System Control and Information Processing,
Ministry of Education of China, Shanghai 200240, China
E-mail: hanshan@sjtu.edu.cn, clcyk2002@aliyun.com,
mark1896bc@gmail.com, lijx@sjtu.edu.cn
*Corresponding author: hanshan@sjtu.edu.cn

Abstract: Multi-sensor decision fusion used for discovering important facts hidden in a mass of data has become a widespread topic in recent years, and has been gradually applied in failure analysis, system evaluation and other fields of big data process. The solution to incompleteness is a key problem of decision fusion during the experiment and has been basically solved by proposed technique in this paper. Firstly, as a generalization of classical rough set, interval similarity relation is employed to classify not only single-valued data but also interval-valued data in the information systems. Then, a new kind of generalized information entropy called "H'-Information Entropy" is suggested based on interval similarity relation to measure the uncertainty and the classification ability in the information systems. Thus, the innovated information filling technique using the properties of H'-Information Entropy can be applied to replace the missing data by some smaller estimation intervals. Finally, the feasibility and advantage of this technique are testified by two actual applications of decision fusion, whose performance is evaluated by the quantification of E-Condition Entropy.

Keywords: Multi-Sensor Decision Fusion, Rough Set Theory, Generalized Information Entropy, Information Classification, Information Filling.

1 Introduction

Decision fusion is the general name of data-based decision making methods. The purposes of these methods are automatically discovering important facts hidden in a mass of data collected from varieties of sensors or other sources, and expressing them by the natural language of decision rules. Then, people can effectively use the simplified decision rules to aid decision making in future applications. Obviously, this target puts forward higher requirements for both sensors capability and the information processing method. Besides basic statistical method, neural networks and Bayesian networks, rough set theory as a favorable mathematical tool with high performance of information acquisition and classification takes into account consequently [1] [2].

The theory of rough set was firstly proposed by Pawlak in 1982 [3]. It is an extension of classical set theory for the study of information systems characterized by inexact, uncertain and vague, and has been widely used in the fields of knowledge discovery, decision fusion, data mining, pattern recognition, and so on. With the rapid development of rough set, some new frameworks were proposed to extend its application range. The tolerance relation [4], valued tolerance relation [5], similarity relation [6], together with the limited tolerance relation [7], laid a solid foundation for the progress of generalized rough set. Until now, the improvements of binary relations are still attracting the scholars attentions [8]- [11].

As to decision fusion of multi-sensor, data acquired from different sensors is real-valued which characterizes objects of interest. Because of the imprecision of acquisition, the fuzziness of cognition and the limitation of knowledge [12] [13], these attributes with real-valued data usually expressed as Single-valued Information System (SIS), Interval-valued Information System (IIS) or the combination of SIS and IIS cannot be classified by any binary relation mentioned above absolutely. The discretization of a long interval by dividing the range into a certain number of partitioning small intervals and using symbolic values to replace these small intervals before further calculation is the popular handling method [14] [15], but existing dispute about the cut-off points [16]. Additionally, the classical rough set may generate an unacceptably large number of classifications from the discretized data resulting in too many classification rules to make final decisions. More important, data missing and uncertain called incompleteness may occur unavoidably because of loss of documents, difficulties of measurement or malfunction of sensors, and increases the difficulties of information processing [17]. The direct information filling methods have not been paid enough attention to, and existed methods [12] [13] are lack of precision to make optimal decisions. Thereupon, the researches of solutions to the classifications and the incompleteness in information systems, together with the following calculations of rough set become significant [18].

In this paper, the discussions are paying attention to solve above issues based on rough set theory emphatically. The notion, properties and measurement of generalized rough set are reviewed firstly in Section 2. In Section 3, a brand new “H’-Information Entropy" is defined for the measurement of uncertainty and classification ability in the information systems according to the interval similarity relation. In Section 4, an innovated information filling technique is proposed by the use of H’-Information Entropy to solve the incompleteness in both SIS and IIS, whose performance is evaluated by the quantification of E-Condition Entropy. In Section 5, the advantage and practicability of this new information filling technique are testified by two integrated examples. The missing data in the decision table is filled by smaller estimation intervals and the certain decision rules are extracted by the calculation of rough set finally.

2 The Theory of Generalized Rough Set

Information systems with the form of decision tables provide a convenient basis for the representation of objects in terms of their attributes. But the uncertainty and incoordination take place in most of decision tables. Thereupon, rough set as a powerful method is used to deal with these issues.

2.1 Basic Concept

Let $S = (U, A)$ be an information system, where $U = \{u_1, u_2, \ldots, u_{|U|}\}$ ($|U|$ means the cardinality of U) is a non-empty finite set of objects, denoting the whole research objects of the information system. $A = \{a_1, a_2, \ldots, a_n\}$ is a non-empty finite set of attributes, denoting the whole attributes of the information system. $f : U \to V_a$ is a mapping for $a \in A$,where V_a is called the domain of attribute a. This constitutes the basic research contents of rough set.

Let $P \subseteq A$ be a subset of attributes. T is a binary relation (including but not limited to equivalence relation). $S_p(u)$ denotes the object set $\{v \in U | (u, v) \in T\}$, called a class or an information granule. If $(u_i, u_j) \in T$, then $S_P(u_i) \neq S_P(u_j)$ in general conditions.

The whole classes of U are obtained according to T [19]:

$$K(P) = (S_P(u_1), S_P(u_2), \ldots, S_P(u_{|U|})) \tag{1}$$

At this time, $K(P)$ is not restricted to “partition", but may be the “covering" of U, namely:

$$\cup S_P(u_i) = U \qquad i = 1, 2, \ldots, |U| \tag{2}$$

Then, a rough set can be described by the definitions of lower and upper approximations.

Definition 1. Let $S = (U, A)$ be an information system, $X \subseteq U$. With an arbitrary binary equivalence relation on U, the lower and upper approximations of X are [3]:

$$\underline{R}(X) = \cup\{S_P(u_i)|S_P(u_i) \subseteq X\} \qquad i = 1, 2, \ldots, |U| \tag{3}$$

$$\overline{R}(X) = \cup\{S_P(u_i)|S_P(u_i) \cap X \neq \emptyset\} \qquad i = 1, 2, \ldots, |U| \tag{4}$$

The covering-based rough set is the great generalization of classical rough set. So, the application range of rough set is extended significantly.

2.2 The Measurement of Rough Set

The uncertainty in rough set refers to the size of classes called information granules generally determined by specified attributes in the information system. It can be measured by information entropy [20]. Small information granules imply a precise description of rough set.

Definition 2. Let $S = (U, A)$ be an information system, and $P \subseteq A$ be a subset of attributes.$K(P) = (S_P(u_1), S_P(u_2), \ldots, S_P(u_{|U|}))$ are the classes of U created by some binary relation according to P. Information entropy of P is defined as [19]:

$$H(P) = -\sum_{i=1}^{|U|} \frac{1}{|U|} \log_2 \frac{|S_P(u_i)|}{|U|} \tag{5}$$

This definition of entropy is called "H-Information Entropy". In other words, H-Information Entropy quantifies the classification ability of objects set U according to the attributes subset P.

The relationship between different classes can be compared by partial relation " $\preceq$ ". Let $P, Q \subseteq A$ be subsets of attributes. If $K(P) = (S_P(u_1), S_P(u_2), \ldots, S_P(u_{|U|}))$ and $K(Q) = (S_Q(u_1), S_Q(u_2), \ldots, S_Q(u_{|U|}))$. The partial relation " $\preceq$ " is defined as:

$$K(P) \preceq K(Q) \Leftrightarrow \forall i = 1, 2, \ldots, |U|, S_P(u_i) \subseteq S_Q(u_i) \tag{6}$$

Theorem 3. *If $K(P) \preceq K(Q)$, then $H(Q) \leq H(P)$ [19].*

This theorem explains that H-Information Entropy is monotone increasing with the decrease of elements in each class. The thinner classes are obtained by stricter binary relations. The bigger the H-Information Entropy is, the smaller the information granules will be.

In order to mine the internal relations of subsets P and Q, the research of coordination of these two attributes becomes significant. The result embodies the support degree of one subset of attributes to another. It is just the major application of rough set in decision fusion.

The condition entropy is introduced to measure this support degree [21]. It is proved that E-Condition Entropy is suitable for both partition-based and covering-based rough set [19] [22].

Definition 4. Let $P, Q \subseteq A$ be two subsets of attributes. If $K(P) = (S_P(u_1), S_P(u_2), \ldots, S_P(u_{|U|}))$ and $K(Q) = (S_Q(u_1), S_Q(u_2), \ldots, S_Q(u_{|U|}))$, then E-Condition Entropy is:

$$E(Q/P) = \sum_{i=1}^{|U|} \frac{|S_P(u_i)| - |S_P(u_i) \cap S_Q(u_i)|}{|U|^2} \tag{7}$$

In particular, if $K(P) \preceq K(Q)$, which means the classes obtained by P are totally included in the classes obtained by Q, then $E(Q/P) = 0$ [23]. Thus, some attributes "entirely" support the some others. Rough set degenerates to the classical set. It is considered that if E-Condition Entropy is small, the support degree is high. Then, the coordination in rough set is better.

3 Generalized Information Entropy

In the decision tables, the values of attributes may be some interval ranges instead of simple single values. In order to measure the uncertainty in such kind of system, the new information entropy is defined in this section, called H' information entropy.

3.1 Interval Similarity Relation

The information system with interval-valued data is now called Interval-valued Information System (IIS).

Definition 5. Let $S = (U, A)$ be an IIS. If $x \in U$ and $a \in A$, then the interval range of $f(x, a)$ is denoted as $f(x, a) = [f(x, a)^L, f(x, a)^U]$.

As mentioned in Introduction, it is clear that the classical binary relations are not suitable for the classifications of IIS now. Additionally, it is considered that the length of interval implies the uncertainty of cognition. Because the intersection degrees among objects are highly different, the negligence will enlarge the possibility of intersection factitiously. Thus, the classification ability and the cognition of the system will be reduced. So, the interval similarity relation is employed as the binary relation to classify the objects of IIS in this paper.

Definition 6. Let $S = (U, A)$ be an IIS. $x, y \in U$ and $a \in A$. The interval similarity degree of $f(x, a)$ and $f(y, a)$ is defined as:

$$P^a_{xy} = \frac{|f(x,a) \cap f(y,a)|}{|f(x,a) \cup f(y,a)|} \tag{8}$$

where || means the absolute length of interval.

In particular, there may exist some single-valued data in IIS in practical applications. That is $f(x, a) = f(x, a)^L = f(x, a)^U = Constant$. In order to generalize the definition of interval similarity degree, if $f(y, a) = [f(y, a)^L, f(y, a)^U]$ and $Constant \in [f(y, a)^L, f(y, a)^U]$, then we define $P^a_{xy} = 1$, and vice versa. Thus, the definition of similarity degree can be improved.

Definition 7. Let $S = (U, A)$ be an IIS. $x, y \in U$ and $a \in A$. the interval similarity relation is defined as:

$$S(A) = \{(x, y) \in U \times U | P^a_{xy} \geq \alpha, \forall a \in A\} \tag{9}$$

where α is the similarity threshold value.

Obviously, if $f(x, a)$ and $f(y, a)$ are both single-valued data, then the interval similarity relation will degenerate to simple equivalence relation by this definition.

Deduction 1. The interval similarity relation satisfies reflexivity and symmetry.

It is significant that if an information system is classified by interval similarity relation, the interval length has direct effect on the classification ability. That is interval similarity may affect information entropy of the system.

3.2 Definition of Generalized Information Entropy

As the improvement, interval similarity degree defined in Equation (8) will be added to classical information entropy. A new kind of generalized information entropy is defined to measure the uncertainty and classification ability in the information systems.

Definition 8. Let $S = (U, A)$ be an information system and $B \subseteq A$ be a subset of attributes. $K(B) = (S_B(u_1), S_B(u_2), \ldots, S_B(u_{|U|}))$ are the classes of U created by interval similarity relation according to B. P_{xy} is the minimum interval similarity degree between two objects (x and y).

$$P_{xy} = \begin{cases} min\{P_{xy}^k\} & P_{xy}^k > 0, \forall k \in B \\ 0 & else \end{cases} \tag{10}$$

The generalized information entropy called "H'-Information Entropy" is defined as:

$$H\prime(B) = -\sum_{i=1}^{|U|} \frac{1}{|U|} \log_2 \frac{\sum\limits_{j=1}^{|U|} P_{u_i u_j}}{|U|} \tag{11}$$

Deduction 2. H'-Information Entropy satisfies non-negativity, symmetry, continuity, monotonicity and extremum property.

Proof.

1) By the definition of interval similarity degree, it is obvious that $0 \le P_{u_i u_j} \le 1$. Then,

$$0 < \sum_{j=1}^{|U|} P_{u_i u_j} \le |U| \Rightarrow 0 < \frac{\sum\limits_{j=1}^{|U|} P_{u_i u_j}}{|U|} \le 1 \Rightarrow -\log_2 \frac{\sum\limits_{j=1}^{|U|} P_{u_i u_j}}{|U|} \ge 0 \tag{12}$$

That is H'-Information Entropy is nonnegative.

2) Moreover, because of the symmetry of interval similarity degree ($P_{u_i u_j} = P_{u_j u_i}$), H'-Information Entropy is also symmetric.

3) The numerical ranges of $P_{u_i u_j}$ are continuous and $P_{u_i u_j}$ constitute the whole variables of H'-Information Entropy. The operations of addition, multiplication, and logarithm are continuous totally. Hence, H'-Information Entropy keeps the continuity.

4) Let $S = (U, A)$ be an information system and $B_1, B_2 \subseteq A$ be subsets of attributes. The condition that $B_1 \subseteq B_2$ is assumed, then there is $K(B_2) \preceq K(B_1)$ [24].

Because $P_{xy}^{B_1} = min\{P_{xy}^k\}$ ($P_{xy}^k > 0, \forall k \in B_1$) and $P_{xy}^{B_2} = min\{P_{xy}^k\}$ ($P_{xy}^k > 0, \forall k \in B_2$), with the assumption $B_1 \subseteq B_2$, the consequent that $P_{xy}^{B_1} \ge P_{xy}^{B_2}$ is obtained. Hence, $H\prime(B_1) \le H\prime(B_2)$. As a result,

$$K(B_2) \preceq K(B_1) \Rightarrow H\prime(B_1) \le H\prime(B_2) \tag{13}$$

It should be noticed that this monotonicity is under the constraint ($B_1 \subseteq B_2$) in this paper. If there is no inclusion relation between two subsets of condition attributes, this monotonicity may be broken down.

5) If $\sum\limits_{j=1}^{|U|} P_{u_i u_j} = 1$, then $H\prime(B) = \log_2 |U|$. If $\sum\limits_{j=1}^{|U|} P_{u_i u_j} = |U|$, then $H\prime(B) = 0$. Hence, there exist two extreme values.

4 The Solution to Incompleteness

Data missing and uncertain called incompleteness occur commonly in data acquisition by multi-sensor. In this section, the research focus will turn to this incomplete information system and try to find the solution to incompleteness.

4.1 The New Information Filling Technique

Now, a new information filling method is proposed based on the H'-Information Entropy defined above.

Theorem 9. *In an IIS with one attribute, it will assuredly increase the H'-Information Entropy when a new object of minimal interval is added into the system.*

Proof. Let $S = (U, A)$ be an IIS. $\forall a \in A$, $S = (U, a)$ is an IIS with one attribute. $u_{|U+1|}$ is the newly added object. There are:

$$f(u_{|U+1|}, a) = [f(u_{|U+1|}, a)^L, f(u_{|U+1|}, a)^U] \tag{14}$$

$$|f(u_{|U+1|}, a)| = f(u_{|U+1|}, a)^U - f(u_{|U+1|}, a)^L = \delta \tag{15}$$

When $u_{|U+1|}$ is added into original IIS, the new H'-Information Entropy is:

$$\begin{aligned} H\prime(a)_1 &= -\sum_{i=1}^{|U+1|} \frac{1}{|U+1|} \log_2 \frac{\sum_{j=1}^{|U+1|} P_{u_i u_j}}{|U+1|} \\ &= -\frac{1}{|U+1|}(\log_2 \frac{\sum_{j=1}^{|U|} P_{u_1 u_j} + P_{u_1 u_{|U+1|}}}{|U+1|} + \log_2 \frac{\sum_{j=1}^{|U|} P_{u_2 u_j} + P_{u_2 u_{|U+1|}}}{|U+1|} + \cdots \\ &\quad + \log_2 \frac{\sum_{j=1}^{|U|} P_{u_{|U|} u_j} + P_{u_{|U|} u_{|U+1|}}}{|U+1|} + \log_2 \frac{\sum_{j=1}^{|U|} P_{u_{|U+1|} u_j} + P_{u_{|U+1|} u_{|U+1|}}}{|U+1|}) \end{aligned} \tag{16}$$

Obviously, $P_{u_{|U+1|} u_{|U+1|}} = 1$. $\sum_{j=1}^{|U|} P_{u_i u_j} (i = 1, 2, \ldots, |U|)$ is unchanged since the newly added object does not affect the intersection relations in original system. When δ is a minimal interval, there is $\delta \ll |f(u_i, a)| (i = 1, 2, \ldots, |U|)$. Therefore, whether δ has an intersection with other interval or not, it is obtained:

$$P_{u_i u_{|U+1|}} = P_{u_{|U+1|} u_i} = 0 \qquad i = 1, 2, \ldots, |U| \tag{17}$$

As a result:

$$\begin{aligned} H\prime(a)_1 &= -\frac{1}{|U+1|}(\log_2 \frac{\sum_{j=1}^{|U|} P_{u_1 u_j}}{|U+1|} + \log_2 \frac{\sum_{j=1}^{|U|} P_{u_2 u_j}}{|U+1|} + \cdots + \log_2 \frac{\sum_{j=1}^{|U|} P_{u_{|U|} u_j}}{|U+1|} + \log_2 \frac{1}{|U+1|}) \\ &\overset{\text{def}}{=} H\prime(a)_{10} \end{aligned} \tag{18}$$

Let $\sum_{j=1}^{|U|} P_{u_i u_j} = P_i$ for convenience, then:

$$\begin{aligned} H\prime(a)_{10} &= -\frac{1}{|U+1|}(\log_2 \frac{P_1}{|U+1|} + \log_2 \frac{P_2}{|U+1|} + \cdots + \log_2 \frac{P_{|U|}}{|U+1|} + \log_2 \frac{1}{|U+1|}) \\ &= -\frac{1}{|U+1|} \log_2 \frac{P_1 P_2 \cdots P_{|U|}}{|U+1|^{|U+1|}} \\ &= \frac{1}{|U+1|} \log_2 \frac{|U+1|^{|U+1|}}{P_1 P_2 \cdots P_{|U|}} \end{aligned} \tag{19}$$

The H'-Information Entropy of original system is:

$$\begin{aligned} H\prime(a)_0 &= -\sum_{i=1}^{|U|} \frac{1}{|U|} \log_2 \frac{\sum\limits_{j=1}^{|U|} P_{u_i u_j}}{|U|} \\ &= \frac{1}{|U|} \log_2 \frac{|U|^{|U|}}{P_1 P_2 \cdots P_{|U|}} \end{aligned} \tag{20}$$

Because $|U| \geq 1$ and $P_i \geq 1 (i = 1, 2, \ldots, |U|)$, Hence,

$$\begin{aligned} H\prime(a)_{10} &= \frac{1}{|U+1|} \log_2 \frac{|U+1|^{|U+1|}}{P_1 P_2 \cdots P_{|U|}} \\ &= \log_2 |U+1| - \frac{1}{|U+1|} \log_2 (P_1 P_2 \cdots P_{|U|}) \\ &> \log_2 |U| - \frac{1}{|U+1|} \log_2 (P_1 P_2 \cdots P_{|U|}) \\ &> \log_2 |U| - \frac{1}{|U|} \log_2 (P_1 P_2 \cdots P_{|U|}) \\ &= H\prime(a)_0 \end{aligned} \tag{21}$$

It is clear that, Theorem 9 can be expanded to a SIS. That is in a SIS with one attribute, when a new object of minimal interval is added into the system, it will assuredly increase the H'-Information Entropy if the interval range of newly added object doesn't include any existed single-valued data in the attribute. As long as this promise exists, $P_{u_i u_{|U+1|}} = P_{u_{|U+1|} u_i} = 0$ as well according to Equation (8). Consequently, the simplifying process from Equation (16) to (18) stays the same.

In order to make the research more intuitive, the following discussions are based on the assumption that the increase of interval is unidirectional. Hence, the lower limit or upper limit is fixed.

Theorem 10. *In an information system with one attribute, the H'-Information Entropy will be monotone decreasing with the interval increase of newly added object within a limit.*

Proof. Followed by previous proof, δ is increased gradually.

1) If the original system is an IIS, obviously, only the terms of $p_i \overset{\text{def}}{=} P_{u_i u_{|U+1|}} = P_{u_{|U+1|} u_i} = \frac{|f(u_i,a) \cap \delta|}{|f(u_i,a) \cup \delta|}$ are changing during the increase of δ. Firstly, $H\prime(a)_1$ will be monotone decreasing because of the continuous increase of p_i. The increase trend of p_i will be sustained until the intersection of $u_i (i = 1, 2, \ldots, |U|)$ and $u_{|U+1|}$ is steady, which means the upper /lower limit of δ reaches the upper /lower limit of arbitrary u_i (as showed in Fig.1). After that, $|f(u_i, a) \cap \delta|$ is a constant while $|f(u_i, a) \cup \delta|$ is still increased. Then, p_i will turn to decrease. Once this happens to any $u_i \in U$, the trend of $H\prime(a)_1$ may be changed.

2) If the original system is a SIS, there should exist growing amount of $p_i = 1 (i = 1, 2, \ldots, |U|)$ with the increase of δ. Thus, $H\prime(a)_1$ will be monotone decreasing until the interval range of δ is long enough to cover the largest number of original single-valued data in the domain of the attribute a.

Theorem 10 shows that the newly added object will have more opportunity to be classified with other objects with interval increase under the interval similarity relation. This phenomenon will cause the weakness of classification ability and the reduction of information entropy.

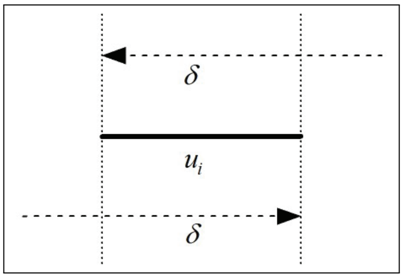


Figure 1: Intersection of Two Intervals.

Deduction 3. In an information system with one attribute, there is a minimum value of H'-Information Entropy with the interval increase of newly added object.

$H\prime(a)_1$ is continuous according to Deduction 2 and it has been proved that $H\prime(a)_1$ is monotone decreasing within a limit. Consequently, there is a minimum value in the limit.

Deduction 4. In an IIS with one attribute, after the upper /lower limit of newly added object reaches the maximal /minimal value in the domain of the attribute a, the H'-Information Entropy will be monotone increasing with the interval increase.

Deduction 5. In an SIS with one attribute, after the upper /lower limit of newly added object reaches the maximal /minimal value in the domain of the attribute a, the H'-Information Entropy will be a constant with the interval increase.

In an IIS, once the upper /lower limit of newly added object reaches the upper /lower limit of interval-valued data of entire original objects, p_i will turn to decrease with the increase of δ for all $u_i \in U$. Then, $H\prime(a)_1$ is changed to increase assuredly. Meanwhile, in a SIS, once the upper /lower limit of newly added object reaches the maximal /minimal value of single-valued data of entire original objects, p_i will be the number of covered single-valued data and remains unchanged regardless of the interval extension. Then, $H\prime(a)_1$ is a constant assuredly.

The deductions above explain that when the interval of the newly added object is increased from zero to the maximal /minimal value in the domain of the attribute a, there exists at least one global minimum value of H'-Information Entropy.

Now, the missing data in the information systems can be dealt with above-mentioned results. The new information filling technique follows the assumption that the objects with missing data have no effect with the classifications of information. In other words, regardless of the existing of the existence of these incomplete objects, the information entropy of corresponding condition attribute should be unchanged.

The process of the new information filling technique is considered as a new object of minimal interval adds into the complete information system attribute by attribute. Then, the H'-Information Entropy is surely increased implying the rapid improvement of classification ability. According to Deduction 3, 4 and 5, the trend of H'-Information Entropy is decreased firstly and increased or flat finally. The filling interval is expected to make the H'-Information Entropy close to the original one as much as possible before the upper /lower limit reaches the maximal /minimal value in the domain of corresponding attribute. Thus, a smaller filling interval is obtained. Because of the expanded definition of interval similarity relation, this new information filling technique can be applied for both IIS and SIS.

In general, the process is listed below.

1) Calculate the H'-Information Entropy ($H\prime(a)_0$) of original information system for each condition attribute without incomplete objects.

2) Add a new object with minimal interval ($\delta = 10^{-3}$ is recommended) to one attribute. If

the original system is SIS, the lower /upper limit of the interval should be set at the left /right side of minimal /maximal value in the domain of corresponding attribute of the objects with the same decision. If the original system is IIS, the lower /upper limit of the interval should be known and fixed in advance just like the forms of $f(x,a) = [f(x,a)^L, *]$ or $f(x,a) = [*, f(x,a)^U]$. Thus, the initial position marked as A is established. The H'-Information Entropy is $H\prime(a)_1|_{start}$ at the beginning.

3) Forward /backward increase the length of the new interval by a setting step size until its upper /lower limit reaches the maximal /minimal value in the domain of the attribute. Fig.2 is just showing the process of forward increasing for instance. Calculate the new H'-Information Entropy for each step. Record the following data:

a. Mark the position of interval's upper /lower limit as B when the corresponding new H'-Information Entropy reaches global minimum value $H\prime(a)_1|_{min}$ for the first time.

b. The position of maximal /minimal value in the domain of the attribute is marked as C, and the corresponding H'-Information Entropy is $H\prime(a)_1|_{end}$.

c. Mark the position of interval's upper (lower) limit as D when the corresponding new H'-Information Entropy is equal to the original H'-Information Entropy ($H\prime(a)_1 = H\prime(a)_0$) for the first time. D may not exist in some cases.

4) If $H\prime(a)_1|_{min} \leq H\prime(a)_0$, then the estimation interval is $|A - D|$ (as showed in Fig. 2(Left)).If $H\prime(a)_1|_{min} > H\prime(a)_0$,then the estimation interval is $|A - B|$ (as showed in Fig. 4(Right)). Thus, there are $|A - D| \leq |A - C|$ and $|A - B| \leq |A - C|$, so the estimation interval is shortened.

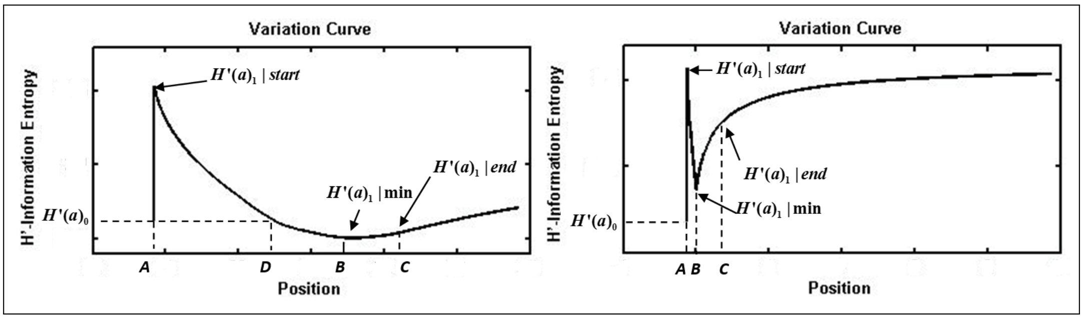


Figure 2: Variation Curve of H'-Information Entropy.

4.2 Performance Analysis

In order to check the performance of proposed information filling technique, a deduction is introduced firstly.

Deduction 6. In an information system, the shortening of estimation intervals will improve the coordination of the system.

Proof. Let $S = (U, A)$ be an information system. $A = C \cup D$, where C is the set of condition attributes and D is set of the decision attributes. If $B \subseteq C$, then the classes obtained according to B and D are:

$$K(B) = (S_B(u_1), S_B(u_2), \ldots, S_B(u_{|U|})) \tag{22}$$

$$K(D) = (S_D(u_1), S_D(u_2), \ldots, S_D(u_{|U|})) \tag{23}$$

With the shortening of estimation intervals in B, there exists a new classification:

$$K(B)\prime = (S_B(u_1)\prime, S_B(u_2)\prime, \ldots, S_B(u_{|U|})\prime) \tag{24}$$

Obviously, short estimation interval will decrease the intersection degree and reduce the objects in the classes made by interval similarity relation. That is $S_B(u_i)\prime \subseteq S_B(u_i)$, $|S_B(u_i)\prime| \leq |S_B(u_i)|$ $(i = 1, 2, \ldots, |U|)$. Hence, $S_B(u_i)\prime \cap S_D(u_i) \subseteq S_B(u_i) \cap S_D(u_i)$ and $|S_B(u_i)\prime \cap S_D(u_i)| \leq |S_B(u_i) \cap S_D(u_i)|$. E-Condition Entropy studied in Section 2.2 is used to measure the coordination in rough set. According to its definition, there is:

$$\begin{aligned} E(D/B) - E(D/B)\prime &= \sum_{i=1}^{|U|} \frac{|S_B(u_i)| - |S_B(u_i) \cap S_D(u_i)|}{|U|^2} - \sum_{i=1}^{|U|} \frac{|S_B(u_i)\prime| - |S_B(u_i)\prime \cap S_D(u_i)|}{|U|^2} \\ &= \sum_{i=1}^{|U|} \frac{|S_B(u_i)| - |S_B(u_i)\prime| - (|S_B(u_i) \cap S_D(u_i)| - |S_B(u_i)\prime \cap S_D(u_i)|)}{|U|^2} \\ &= \sum_{i=1}^{|U|} \frac{|S_B(u_i)| - |S_B(u_i)\prime| - |(S_B(u_i) - S_B(u_i)\prime) \cap S_D(u_i)|}{|U|^2} \\ &\geq 0 \end{aligned} \tag{25}$$

The equality holds if and only if $(S_B(u_i) - S_B(u_i)\prime) \subseteq S_D(u_i)$. So, $E(D/B) \geq E(D/B)\prime$.

It is mentioned that small E-Condition Entropy implies great support degree, which means the shortening of estimation intervals improves the coordination between corresponding condition attributes and decision attributes. Because of the generality of this result to the whole condition attributes, the coordination of the information system is improved consequently.

Now, the benefits of smaller estimation intervals obtained in this paper are reflected in two aspects. On the one hand, the coordination between condition attributes and decision attributes is improved according to Deduction 6. That is the positive region in the rough set is expanded, while the uncertainty is decreased. The reliability of decision will be increased as a result. On the other hand, the shortened estimation intervals are beneficial for decision making. The new decision rules will be fine and precise with the extending of covering range.

5 Experimental Studies

The new information filling technique can be applied to process the incompleteness in large amounts of data acquired by different sensors for decision fusion in a variety of fields. Here, two integrated experiments will be studied below to validate the feasibility and superiority of proposed method.

5.1 Single-valued Data Estimation and Filling

In the study of single-valued data evaluation and filling, a part of ICU data of one patient is acquired by different sensors and listed in Table 1. In this decision table, the Column "U" refers to the times of examinations. Columns "$c1$" to "$c5$" denote the data acquired by sensors of temperature, heart rate, systolic blood pressure, spo2 and respiration rate, and compose five condition attributes. The Column "d" denotes four degrees of ICU decisions and composes the decision attribute. Furthermore, the missing data is denoted as "$*$" in the table.

By applying the new information filling technique, a small interval can be estimated to fill the missing data. Figure 3 shows this process of information filling and we can get $f(u17, c1) = [36.3, 36.4]$ (as showed in Figure 3(left)) and $f(u3, c3) = [103.0, 115.9]$ (as showed in Figure 3(right)) as a result.

Then, E-Condition Entropy discussed before can be used to evaluate the performance of above result. We get $E(D/C) = 0$ without the objects with missing data, meaning the condition

U	$c1$	$c2$	$c3$	$c4$	$c5$	d
1	37.8	62.0	103.0	94.0	18.0	2
2	37.7	61.0	120.0	97.0	16.0	2
3	37.7	61.0	*	97.0	16.0	2
...	...	...	...	...	...	...
14	37.3	62.0	116.0	98.0	18.0	3
15	37.3	62.0	116.0	98.0	18.0	3
16	37.3	62.0	116.0	98.0	18.0	3
17	*	111.0	142.0	95.0	20.0	1
18	36.3	111.0	142.0	95.0	20.0	1
19	36.4	111.0	142.0	95.0	20.0	1
...	...	...	...	...	...	...
49	36.7	98.0	133.0	92.0	20.0	4
50	36.7	98.0	133.0	92.0	20.0	4

Table 1: Incomplete ICU Data.

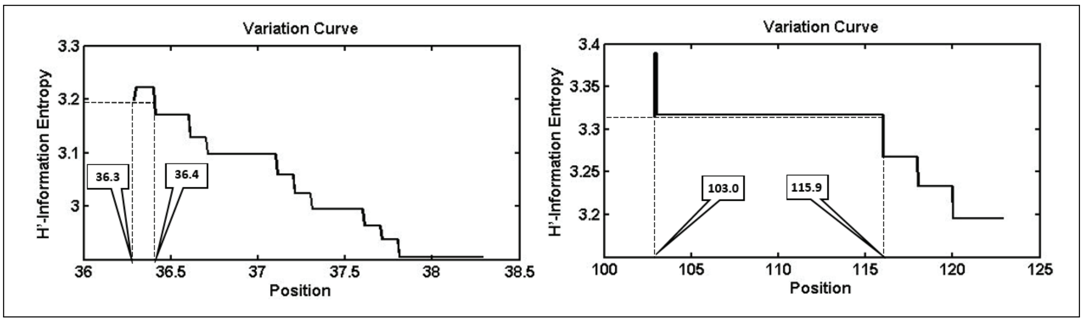


Figure 3: Information Filling of ICU data.

attributes entirely support each decision in the original system. Then, this index is still 0 when u3 and u17 with filling intervals are added to the system. That is the new information filling technique doesn't change the coordination between condition attributes and decision attribute in this example and the results make sense.

5.2 Interval-valued Data Estimation and Filling

In the study of interval-valued data estimation and filling, a part of thruster experimental data is acquired by different sensors. In order to simplify the table, only the objects with missing data denoted as "$*$" are listed in Table 2. The Column "U" refers to the times of experiment. Columns "$c1$" to "$c4$" denote the data acquired by different pressure and temperature sensors, and compose four condition attributes. These acquired data is formed as interval values including the minimum and maximum from the each sensor without data preprocessing. The Column "d" denotes three types of failure simplified as symbolic values and composes the decision attribute. We need to fill the missing data effectively and discover the failure rules hidden in the data as accurately as possible.

1) Firstly, the decision table is completed by the method in Ref. [13] and showed in Table 3. Briefly, "$*$" is replaced by the minimal /maximal value in the domain of each condition attribute.

2) Then, the decision table is completed by the new information filling technique proposed in this paper and showed in Table 4.

In order to reveal the difference between above two methods, the classifications of condition attributes are made by interval similarity relation. The cardinality of each class obtained by the two methods can be compared in Figure 4.

U	$c1$	$c2$	$c3$	$c4$	d
3	[4.344, 4.604]	[0.029, 0.037]	[4.600, *]	[23.865, 26.819]	1
6	[4.283, *]	[0.024, 0.033]	[4.613, 4.889]	[23.872, 25.826]	1
21	[4.801, 4.997]	[0.027, 0.035]	[*, 4.516]	[28.700, 30.848]	2
25	[*, 4.993]	[0.023, 0.030]	[4.627, 4.904]	[28.663, 30.806]	2
28	[4.445, 4.711]	[0.025, 0.037]	[4.236, 4.490]	[28.826, *]	2
43	[2.629, *]	[0.024, 0.037]	[4.600, 4.876]	[30.881, 32.116]	3
46	[2.653, 3.812]	[0.025, 0.033]	[4.619, 4.896]	[30.016, *]	3
55	[1.470, 2.551]	[0.027, 0.030]	[4.507, *]	[30.497, 31.717]	3
...	...	...	...	...	...

Table 2: Incomplete Thruster Experimental Data.

U	$c1$	$c2$	$c3$	$c4$	d
3	[4.344, 4.604]	[0.029, 0.037]	[4.600, 4.999]	[23.865, 26.819]	1
6	[4.283, 4.999]	[0.024, 0.033]	[4.613, 4.889]	[23.872, 25.826]	1
21	[4.801, 4.997]	[0.027, 0.035]	[4.062, 4.516]	[28.700, 30.848]	2
25	[1.015, 4.993]	[0.023, 0.030]	[4.627, 4.904]	[28.663, 30.806]	2
28	[4.445, 4.711]	[0.025, 0.037]	[4.236, 4.490]	[28.826, 32.150]	2
43	[2.629, 4.999]	[0.024, 0.037]	[4.600, 4.876]	[30.881, 32.116]	3
46	[2.653, 3.812]	[0.025, 0.033]	[4.619, 4.896]	[30.016, 32.150]	3
55	[1.470, 2.551]	[0.027, 0.030]	[4.507, 4.999]	[30.497, 31.717]	3
...	...	...	...	...	...

Table 3: Completed Thruster Experimental Data 1.

The histogram indicates that, with the same binary relation, more than 30% classes obtained from the completed decision table by the new information filling technique are “thinner" than before. That is, the new information filling technique can achieve a classification ability no less than the traditional one. The classes are more precise and the uncertainty is lower now. Additionally, E-Condition Entropy is calculated as $E(D/C) = 0.0033$ in Table 4, while this index is 0.0106 in Table 3. The result also implies a better coordination obtained by the new information filling technique between condition attributes and decision attributes.

Then, the reduct of the system is $c1, c3, c4$ obtained by heuristic algorithm of attribute reduction based on E-Condition Entropy [22] [25]. The certain decision rules (failure rules) can be achieved explicitly (as showed in Table 5) since all the missing data has been filled.

At present, the estimation intervals in each attribute are relatively small in this group of certain failure rules. It has a better significance for the practical decision making. The result is agreed with the performance analysis in Section 4.2. Consequently, the problems of incompleteness in thruster experimental data are solved and the failure rules are achieved favorably.

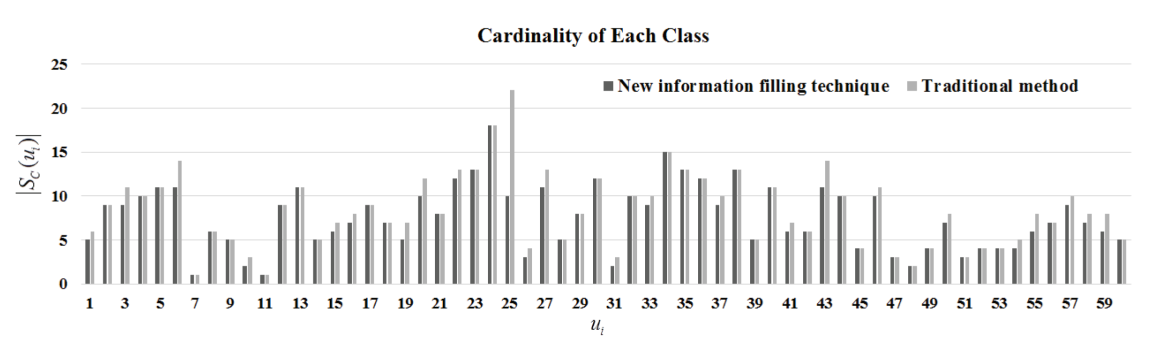


Figure 4: Cardinality of Each Class.

U	$c1$	$c2$	$c3$	$c4$	d
3	[4.344, 4.604]	[0.029, 0.037]	[4.600, 4.753]	[23.865, 26.819]	1
6	[4.283, 4.601]	[0.024, 0.033]	[4.613, 4.889]	[23.872, 25.826]	1
21	[4.801, 4.997]	[0.027, 0.035]	[4.235, 4.516]	[28.700, 30.848]	2
25	[4.877, 4.993]	[0.023, 0.030]	[4.627, 4.904]	[28.663, 30.806]	2
28	[4.445, 4.711]	[0.025, 0.037]	[4.236, 4.490]	[28.826, 30.471]	2
43	[2.629, 3.810]	[0.024, 0.037]	[4.600, 4.876]	[30.881, 32.116]	3
46	[2.653, 3.812]	[0.025, 0.033]	[4.619, 4.896]	[30.016, 31.240]	3
55	[1.470, 2.551]	[0.027, 0.030]	[4.507, 4.837]	[30.497, 31.717]	3
. . .	. . .	. . .	. . .	. . .	. . .

Table 4: Completed Thruster Experimental Data 2.

Rules	Descriptions
$R1$	If $c1 \in [4.458, 4.725] \wedge c3 \in [4.867, 4.997] \wedge c4 \in [23.779, 25.730]$, then $d = 1$
$R2$	If $c1 \in [4.456, 4.732] \wedge c3 \in [4.358, 4.619] \wedge c4 \in [23.717, 25.658]$, then $d = 1$
. . .	. . .
$R19$	If $c1 \in [4.801, 4.997] \wedge c3 \in [4.235, 4.516] \wedge c4 \in [28.700, 30.848]$, then $d = 2$
$R20$	If $c1 \in [4.817, 4.989] \wedge c3 \in [4.350, 4.611] \wedge c4 \in [29.200, 30.328]$, then $d = 2$
. . .	. . .
$R35$	If $c1 \in [2.947, 4.215] \wedge c3 \in [4.868, 4.999] \wedge c4 \in [30.391, 31.607]$, then $d = 3$
$R36$	If $c1 \in [3.678, 4.047] \wedge c3 \in [4.358, 4.619] \wedge c4 \in [29.950, 31.148]$, then $d = 3$
. . .	. . .

Table 5: Certain Decision Rules.

6 Conclusions

The solution to incompleteness in decision fusion is presented with the help of a new information filling technique based on generalized information entropy in this paper. Interval similarity relation is employed as the binary relation to classify both single-valued and interval-valued data acquired by different sensors. As the improvement of classical rough set, a new kind of generalized information entropy called H'-Information Entropy is provided to solve the incompleteness occurring unavoidably in information systems. So, the missing data in the decision tables can be filled evidently and favorably. The calculation procedure of this innovated information filling technique is listed in detail. Finally, two integrated examples are given to evaluate the performance of proposed method. The results turn out the superiority and practicality in real applications.

Acknowledgements

This work is jointly supported by National Natural Science Foundation (61175008) and State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System (CEMEE2014K0301A).

Bibliography

[1] Pawlak, Z.; Skowron, A. (2007); Rudiments of rough sets. *Information Science*, ISSN 0020-0255, 177(1): 3-27.

[2] Tay, F.E.H.; Shen, L.-X.(2003); Fault diagnosis based on rough set theory. *Engineering Applications of Artificial Intelligence*, ISSN 0952-1976, 16(1): 39-43.

[3] Pawlak, Z.(1982); Rough sets. *International Journal of Computer & Information Sciences* , ISSN 0885-7458, 11(5): 341-356.

[4] Kryszkiewicz, M.(1998); Rough Set Approach to Incomplete Information Systems. *Information Sciences*, ISSN 0020-0255, 112(1-4): 39-49.

[5] Stefanowski, J.; Tsoukiàs, A.(1999); On the Extension of Rough Sets under Incomplete Information. *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture Notes in Computer Science*, ISSN 0302-9743, 1711: 73-81.

[6] Stefanowski, J.; Tsoukiàs, A.(2001); Incomplete Information Tables and Rough Classification. *Computational Intelligence*, ISSN 0824-7935, 17(3): 545-566.

[7] Wang, G.-Y.(2002); Extension of Rough Set Under Incomplete Information Systems. *Journal of Computer Research and Development*, ISSN 1000-1239, 39(10): 1238-1243.

[8] Greco, S.; Matarazzo, B.; Slowinski, R.(1999); Rough Approximation of A Preference Relation By Dominance Relations. *European Journal of Operational Research*, ISSN 0377-2217, 117(1): 63-83.

[9] Guan, Y.-Y.; Wang, H.-K.(2006); Set-valued information systems. *Information Sciences*, ISSN 0020-0255, 176(17): 2507-2525.

[10] Leung, Y.; Wu, W.-Z.; Zhang, W.-X.(2006); Knowledge acquisition in incomplete information systems: A rough set approach. *European Journal of Operational Research*, ISSN 0377-2217, 168(1): 164-180.

[11] Yang, X.-B.; Yu, D.-J.; Yang, J.-Y.; Song, X.-N.(2009); Difference Relation-Based Rough Set And Negative Rules In Incomplete Information System. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, ISSN 0218-4885, 17(5): 649-665.

[12] Yang, X.-B.; Yu, D.-J.; Yang, J-Y.; Wei, L.-H.(2009); Dominance-based rough set approach to incomplete interval-valued information system. *Data & Knowledge Engineering*, ISSN 0169-023X, 68(11): 1331-1347.

[13] Zhao, L.; Zhang, X.; Xue, Z.(2011); Security Assessment for Incomplete Interval-valued Information System. *Computer Engineering*, ISSN 1000-3428, 37(11): 146-148.

[14] Grzymala-Busse, J.W.; Stefanowski, J.(2001); Three Discretization Methods for Rule Induction. *International Journal of Intelligent Systems*, ISSN 1098-111X, 16(1): 29-38.

[15] Beynon, M.J.(2004); Stability of continuous value discretisation: An application within rough set theory. *International Journal of Approximate Reasoning*, ISSN 0888-613X, 35(1): 29-53.

[16] Leung, Y.; Fischer, M.-M.; Wu, W.-Z.; Mi, J.-S.(2008) A rough set approach for the discovery of classification rules in interval-valued information systems. *International Journal of Approximate Reasoning*, ISSN 0888-613X, 47(2): 233-246.

[17] Yang, X.-B.; Yang, J.-Y.(2012); Incomplete Information System and Rough Set Theory-Models and Attribute Reductions, Springer- Verlag: Berlin, Heidelberg, ISBN 978-3-642-25934-0.

[18] Zhang, N.; Miao, D.-Q.; Yue, X.-D.(2010); Approaches to Knowledge Reduction in Interval-Valued Information Systems. *Journal of Computer Research and Development*, ISSN 1000-1239, 47(8): 1362-1371.

[19] Liang, J.-Y.; Qian, Y.-H.(2008); Information granules and entropy theory in information systems. *Science in China Series F: Information Sciences*, ISSN 1674-733X, 51(10): 1427-1444.

[20] Düntsch, I.; Gediga, G.(1998); Uncertainty Measures of Rough Set Prediction. *Artificial Intelligence*, ISSN 0004-3702, 106(1): 109-137.

[21] Wang, J.; Miao, D.-Q.(1998); Analysis on Attribute Reduction Strategies of Rough Set. *Journal of Computer Science and Technology*, ISSN 1000-9000, 13(2): 189-193.

[22] Teng, S.-H.; Zhou, S.-L.; Sun, J.-X.; Li, Z.-Y.(2010); Attribute Reduction Algorithm Based on Conditional Entropy under Incomplete Information System. *Journal of National University of Defense Technology*, ISSN 1001-2486, 32(1): 90-94.

[23] Liang, J.-Y.; Chin, K.S.; Dang, C.-Y.; Yam, R.C.M.(2002); A new method for measuring uncertainty and fuzziness in rough set theory. *International Journal of General Systems*, ISSN 0308-1079, 31(4): 331-342.

[24] Xu, J.-C.; Sun, L.(2010); A New Knowledge Reduction Algorithm Based on Decision Power in Rough Set. *Transactions on Rough Sets XII, Lecture Notes in Computer Science*, ISSN 0302-9743, 6109: 76-89.

[25] Li, F.; Yin, Y.-Q.(2009); Approaches to knowledge reduction of covering decision systems based on information theory. *Information Sciences*, ISSN 0020-0255, 179(11): 1694-1704.

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):187-200, April, 2014.

VLSI Architecture for High Performance 3GPP (De)Interleaver for Turbo Codes

J.M. Mathana, S. Badrinarayanan, R. Rani Hemamalini

J. Magdalene Mathana
Electronics and Communication Dept,
S.A Engineering College, Chennai, India
E-mail: jm.mathana@gmail.com

S. Badrinarayanan
Vinayaka Missions University, Salem, India
E-mail: badri22.jm@gmail.com,

R. Rani Hemamalini
Elect. and Comm. Dept,
St.Peter's College of Engg and Tech, Chennai, India
E-mail: ranihema@yahoo.com

Abstract: Interleaving along with error correction coding is an effective way to deal with different types of error in digital data communication. Error burst due to multipath fading and from other sources in a digital channel may be effectively combated by interleaving. Normally the interleaver / deinterleaver pair is often designed as reconfigurable architectures able to deal with requirements of large data length variability found in the newest communication standards. In this work reconfigurable interleaver architecture for the turbo decoder in 3rd Generation Partnership Project (3GPP) standard is presented. The interleaver is a key component of radio communication systems. Using conventional design methods, it consumes a large part of silicon area in the design of turbo encoder and decoder. The proposed interleaver utilizes the algorithmic level hardware simplifications and generates 100 manage the ow of data streams to achieve very low cost solution. The proposed technique reduces consumption of FPGA resources to a large extent compared with existing state-of-the-art interleaver for turbo codes. The proposed architecture con- sumes only 4856 logic elements by hardware optimization.

Keywords: 3GPP, interleaver, reconfigurable, turbo codes.

1 Introduction

Turbo codes [1] have been shown to provide superior error performance amidst hostile transmission media. 3rd generation systems based on the 3GPP air interface standard [2] must support turbo coding at data rates of between 384 kbps to 2Mbps, while maintaining low mobile station power. A turbo code is formed from the parallel concatenation of two codes separated by an interleaver [3] .The generic design of a turbo code is depicted in Figure 1. The two encoders used are normally identical. The code is in a systematic form, i.e. the input bits also occur in the output. The Turbo decoder consists of two elementary decoders in a serial concatenation scheme. Since soft decoding performs better than hard decoding, the first decoder provides a weighted soft decision in the form of A Posteriori Probabilities (APPs) to the second decoder. The decoding proceeds in an iterative fashion as illustrated in Figure.2 [4]. The interleaver reads the bits in a pseudo-random order. The choice of the interleaver is a crucial part in the turbo code design. The task of the interleaver is to "scramble" bits in a pseudo-random, albeit predetermined fashion. This serves two purposes. Firstly, if the input to the second encoder is interleaved, its output is usually quite different from the output of the first encoder. This means that even if one of the output code words has low weight, the other usually does not, and there is a smaller chance of producing an output with very low weight. Higher weight is beneficial for the performance of the decoder. Secondly, since the code is a parallel concatenation of two codes, the divide-and-conquer strategy can be employed for decoding. If the input to the second decoder is scrambled, also its output will be different or "uncorrelated" from the output of the first encoder. This means that the corresponding two decoders will gain more from information exchange [5].

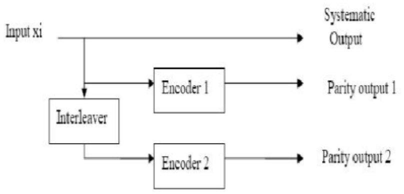


Figure 1: Turbo Encoder

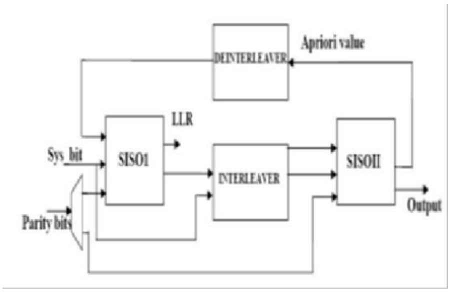


Figure 2: Turbo Decoder

A flexible and reconfigurable interleaver architecture for multimode communication environment is presented in [6]. The presented hardware in [6] enables the mapping of vital types of interleavers including multiple block interleavers and convolutional interleaver onto a single architecture. A parallel interleaver design that support for high throughput is presented in [7].In the proposed work, circuit level VLSI optimization approach is used to reduce the overall computation complexity and provides 100 % interleaved paths.

The paper is organized as follows. In section 2, a review of 3GPP algorithm is discussed. Section 3 describes the proposed reconfigurable architecture of interleaver/deinterleaver. The controller for the proposed reconfigurable interleaver architecture operated in two phases is explained in section 4. Section 5 describes about the address generation and section 6 describes about exception handling. In section 7, simulation and synthesis results are reported and section 8 concludes the paper.

2 Interleaver algorithm

The 3GPP-defined turbo code interleaver algorithm [8] comprises a series of mathematically complex processes that map an input sequence of length K (40 - 5114) to a scrambled sequence. The algorithm can be summarized as follows.

Block size 'K'denotes the number of bits input to turbo code internal interleaver and takes one of the value from 40 to 5114.

Determine the number of rows 'R' of the rectangular matrix such that,

R= 5; if (40<K<159)
R=10; if((160<K<200)or(481<K<530))
R=20 ;if (K= any other value)
where rows are numbered form 0 to R-1.

Determine the prime number 'p' to be used in intra-row permutations and the number of columns 'C' of the rectangular matrix as
if (481< K <530)
then p= 53 and C = p

else

Find the minimum prime number p from table 2 in [6] such that K < (Rx(p-1)) and determine C Such that
C = p-1; if K< (R (p-1))
C = p; if R (p-1) $< K <$ R p)
C=p+1; if (R p) < K

For every prime number p there exists a primitive root v.

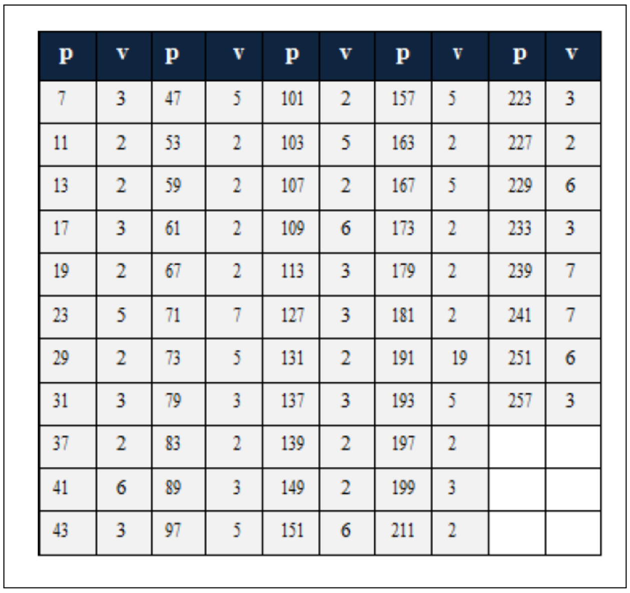
Construct the base sequence S(j) for intra row permutation such that
S(j) = (v*S(j-1))mod (p) and S(0) = 1 where j=1,2,...,(p-2).

Determine the prime integers in the sequence q(i) such that
g.c.d(q,p-1) = 1, q(i)> 6 and q(i)> q(i-1) for i=1,2,.....R-1.

Permute the sequence q(i)to make the sequence r(i) such that
r(i) = T(q(i)), i = 0,1,...,R-1. where T(i) is a simple indexing transform.

It is defined by the standard and shown in Table 1.

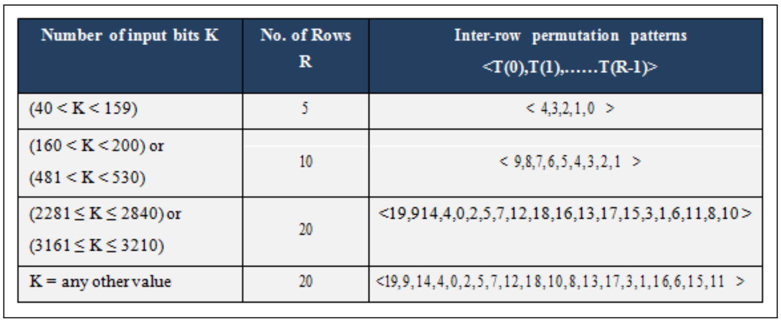
Table 1 List of prime number p and associated primitive root 'v'



p	v	p	v	p	v	p	v	p	v
7	3	47	5	101	2	157	5	223	3
11	2	53	2	103	5	163	2	227	2
13	2	59	2	107	2	167	5	229	6
17	3	61	2	109	6	173	2	233	3
19	2	67	2	113	3	179	2	239	7
23	5	71	7	127	3	181	2	241	7
29	2	73	5	131	2	191	19	251	6
31	3	79	3	137	3	193	5	257	3
37	2	83	2	139	2	197	2		
41	6	89	3	149	2	199	3		
43	3	97	5	151	6	211	2		

Perform the intra row permutation
if C = p , U(i,j)=S[(j*r(i))mod(p-1)] where U(i,p-1) = 0
if C= p+1,U(i,j)=S[(j*r(i)mod(p-1)] where U(i,p-1) = 0; and U(i,p) = p
and if K= R C, then exchange U(R-1,0) with u(R-1,p)
if C = p-1, U(i,j)=S[(j*r(i))mod(p-1)]-1 where i = 0,1,...,R-1
and j = 0,1,..,p-2.

Table 2 Inter-Row Permutation Patterns



Number of input bits K	No. of Rows R	Inter-row permutation patterns <T(0),T(1),......T(R-1)>
(40 < K < 159)	5	< 4,3,2,1,0 >
(160 < K < 200) or (481 < K < 530)	10	< 9,8,7,6,5,4,3,2,1 >
(2281 ≤ K ≤ 2840) or (3161 ≤ K ≤ 3210)	20	<19,914,4,0,2,5,7,12,18,16,13,17,15,3,1,6,11,8,10>
K = any other value	20	<19,9,14,4,0,2,5,7,12,18,10,8,13,17,3,1,16,6,15,11 >

Perform the inter row permutation for the rectangular matrix based on the pattern T(i).It is denoted as U(i)
U(i) = U(T(i)), where i=0,1,..,R-1.

Read out the addresses column wise.

Looking at the interleaver algorithm in 3GPP standard, the computationally intensive functions found are Modulo Computation, Intra-row and Inter-row Permutations, Multipliers, Finding least prime integers, and Computing greatest common divisor. Some of these complex functions are implemented using the support from ROM, while the others are simplified to reduce the hardware usage.

The interleaving process can be easily separated into two main phases. The two phases are Pre-computation phase and Execution phase. As different block sizes has different interleaving patterns, thus every time some pre-computation is needed while changing the block size 'K'. The only parameter applied to interleaver is the block size (K) and rest of the parameters is computed by the hardware itself.

2.1 Pre-computation Phase

Pre-computation phase involves pre-computing the various parameters required for the interleaving process. It has to be performed each time the change in the block size occurs, that means for a fixed block size K, these operations have to be performed only once.

2.2 Execution phase

Execution phase calculates the intra row permutations Ui,j, and the interleaved address i_addr .Write or Read the data bits (in/ from a data RAM) can be taken place depending upon whether interleaving or deinterleaving is performed of addresses could also be computed and saved in the memory only.

For each data block, the execution phase has to be performed. The set of addresses could also be computed and saved in the memory only once.This approach permits to reduce the implementation size of Interleaving / Deinterleaving algorithm as compared to the memory or look up table (LUT) based approach, where the interleaving addresses are saved in memory. Thus execution phase performs basic permutations which form the interleaving process. The basic permutations performed in this phase are illustrated in fig.4.

3 Proposed Architecture of Reconfigurable Interleaver

The main objective of the proposed work is to device an architecture capable of managing every one of the 5074 different block sizes of data ranging from 40 to 5114 defined in the 3GPP

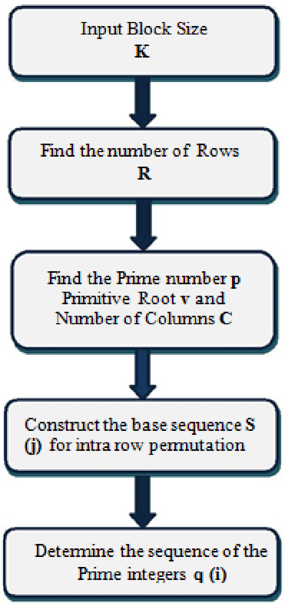


Figure 3: Flow graph for Pre-computation Phase

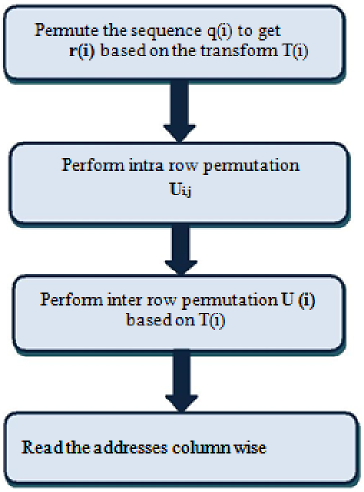

Permute the sequence q(i) to get
r(i) based on the transform T(i)
Perform intra row permutation
Uij
Perform inter row permutation U (i)
based on T(i)
Read the addresses column wise

Figure 4: Flow graph for Execution Phase

standard, while maintaining low hardware complexity. The proposed interleaver is said to be reconfigurable, since it adapts or varies its interleaving architecture depending upon the block size 'K' of the input rectangular matrix. The reconfigurable interleaver accepts only the block size of the rectangular matrix as the input. The rest of the parameters such as the number of rows, columns and the base sequence for permutation (both intra row and inter row permutation) are calculated by the reconfigurable architecture for interleaver address generation. The block diagram of proposed reconfigurable interleaver/deinterleaver is shown in Figure 5. It consist of modulo-computation block, Multiplication addition and comparison hardware, Controller circuit , I/O RAM, Data RAM, Exceptional logic block, S(j) RAM and LUTs etc.

3.1 Precomputation of Parameters

The computation of parameters in pre-computation for 3GPP is discussed in detail by Blakley [9]. Some parameters are computed using look up tables while the others need some close loop or recursive computations. First, the row value 'R' with logical functions is calculated. Parameters like prime number 'p' and primitive root 'v' are stored as a pair in a look up table just like in the standard and LUT is read via a counter generated by the controller based on equation (1).

$$K \leq R*(p+1) \rightarrow (K-R) \leq (R*p) \tag{1}$$

In order to perform the operation in equation (1) the architecture given by Blakley is used in the proposed work. It is shown in Figure 6.The same hardware architecture is used in execution phase also. For computation of 'p' and 'C' operations such as the multiplication, addition and comparison are required. It is controlled by the flags generated by the controller. Once 'p' and 'v' are obtained then numbers of columns have to be determined.

The three possible column values are p-1, p or p+1 and always the following condition shown by the equation (5.2) should be satisfied.

$$(R*C) \geq K \tag{2}$$

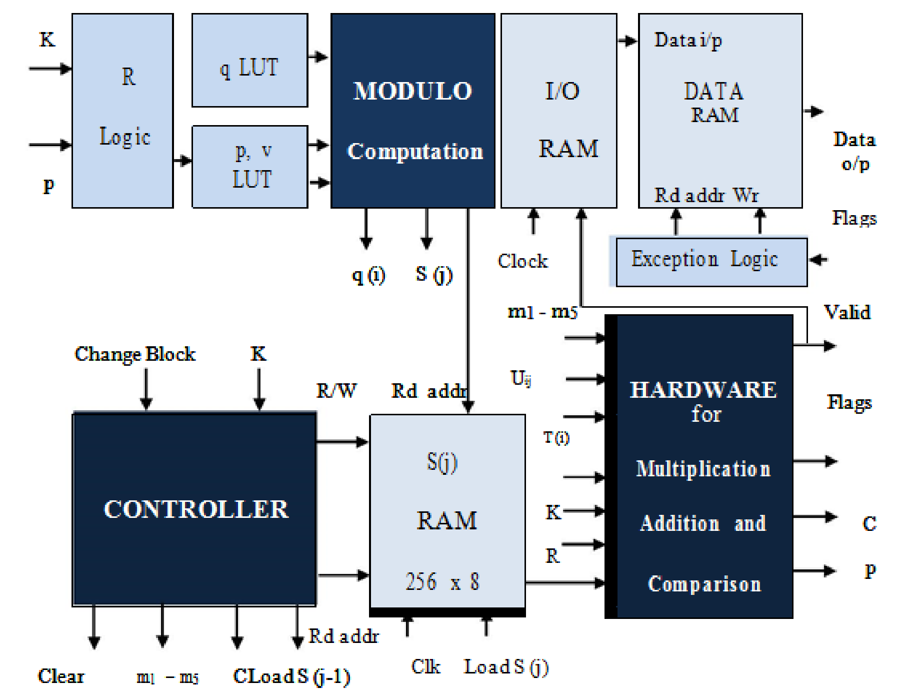


Figure 5: Block Diagram of Proposed Interleaver/Deinterleaver

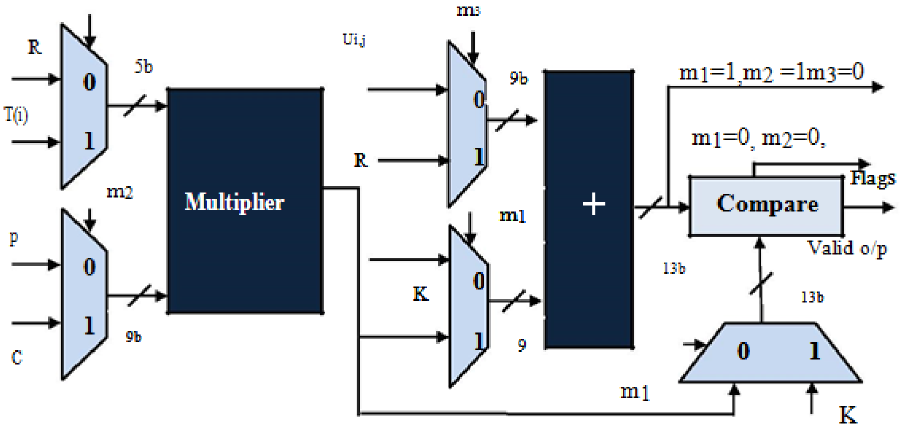


Figure 6: Hardware for Multiplication, Addition and Comparison

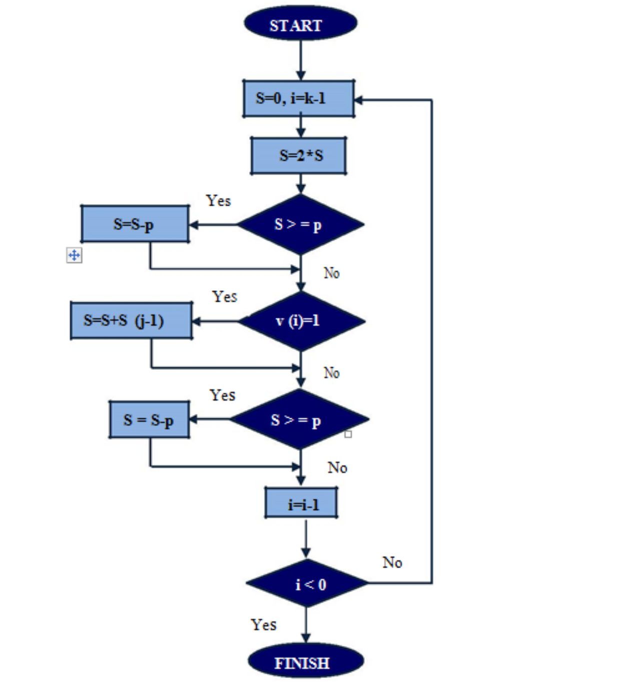


Figure 7: Flow Diagram for the Calculation of S(j)= (v ×S (j-1)) Mod p

The computation of intra-row permutation pattern S(j) required modulo computation. Modulo function is computed iteratively using the Interleaved Modulo Multiplication Algorithm [9].It is an iterative numerical algorithm based on additions, shifts, comparisons and bit retrieval. In order to calculate the S(j) sequence, it is mandatory to perform the modulo operation. A flow diagram depicting this algorithm is shown in Figure 7.

The hardware for the computation of S (j) is shown in Figure 8. This architecture uses two adders, a multiplier and a subtractor to compute the S (j) values. The S array is computed as follows (3) with S (0) = 1.

$$S(j) = (v * S(j-1))mod(p) \tag{3}$$

Where j =1, 2... (p-2)

From the above equation it is known that, primitive root 'v' is required to compute S(j), which is 5 bits. So maximum of 5 iterations are needed to compute one modulo multiplication. With this architecture every S (j) is determined and writes in a RAM of 256 x 10 bits. The size of the RAM depends upon the value of 'p'. The number of cycles needed to calculate every S (j) depends on v, which can take only six different values 2, 3,5,6,7 and 19. For v = 2 and v = 3 only one iteration is required for v = 5, v = 6 and v =7 two iterations and for v =19 four iterations are required.

According to the 3GPP algorithm, the next step is calculation of the prime sequences q (i). It can be noticed that the qi sequences [10] are almost the same in most cases and they only differ by one or two elements from other sequences. Based on this observation, sub groups of q (i) sequence are placed into a ROM and then choose one according to the 'p' value [11]. In the proposed work, instead of finding the least prime number sequence q (i), q(i) mod (p-1) is determined. This gives the benefit of computing the RAM address recursively and avoiding computation of modulo function. This idea is introduced by Shin and Park [12] and later used by Wang and Li [13]. According to the equation (4) q (i) is calculated.

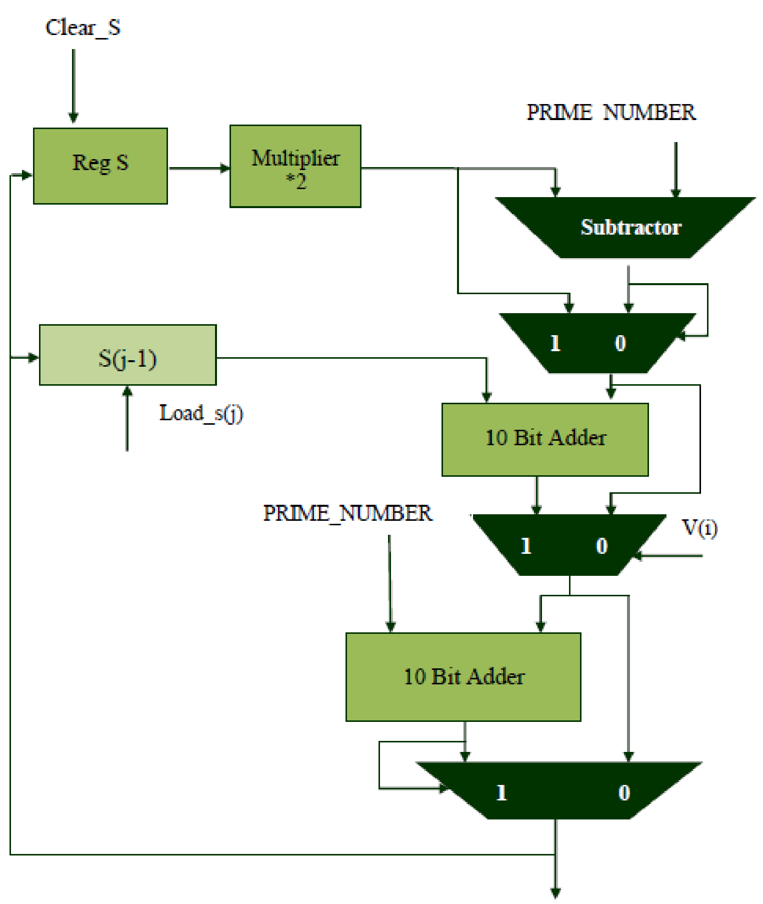


Figure 8: Architecture for Performing [v ×S (j-1)] Mod p

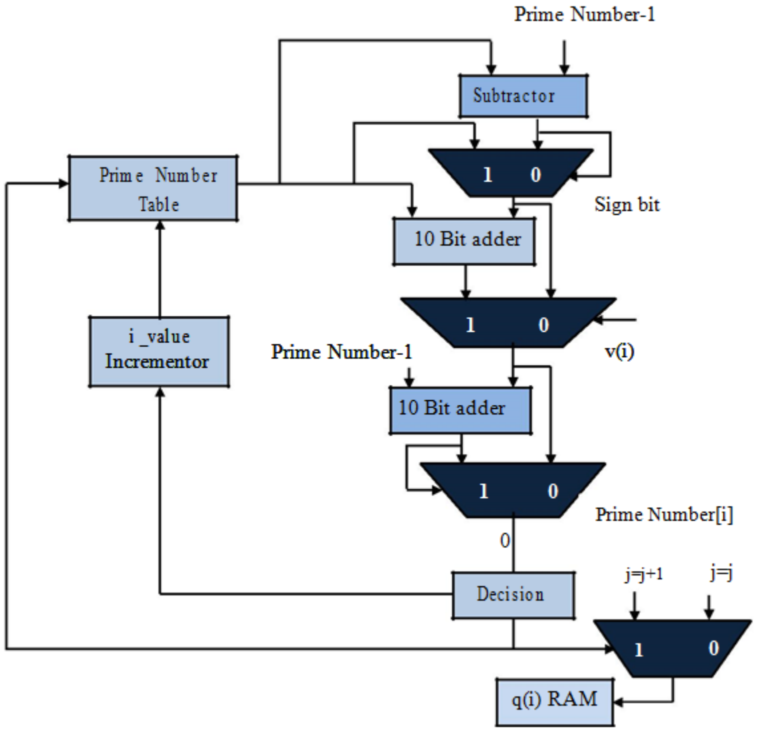


Figure 9: Architecture for Performing q(i)

$$q(i) = q(i) mod p - 1 \tag{4}$$

where i = 0,1,2,3,...p-1

The architecture used to perform q(i) is shown in Figure 9. In the proposed work, the architecture of S (j) is used with some alterations to determine q mod (p-1). It is calculated in such a way that the following conditions have to be satisfied.

g c d(q, p-1) = 1, q(i) > 6 and q(i) > q(i-1) for i = 1,2,.....R-1.

g c d is the Greatest Common Divisor. The selected prime number from the prime number table should be 6 and g c d (q, p-1) = 1.If the above condition is satisfied then the value of i = i+1, j = j+1 and the next prime number is selected. The selected prime number becomes one of the member of q(i) and the value of 'j' is also incremented by1.If the condition is not satisfied then i=i+1,j=j.

3.2 Execution Phase Computation

After completing the pre-computation phase, the controller is set in execution phase and the hardware is configured to perform run time computations for the generation of the interleaved addresses.RAM address are computed using the hardware shown in Figure 6. The recursive function used to compute the RAM address and multiply?add function to compute the final interleaved address are as follows:

$$Ram_a dr(i, j) = [Ram_a dr(i, j - 1)\ qmod\ (i)] mod(p - 1) \tag{5}$$

It can be seen that computing the RAM address using qmod (p-1) instead of q helps to avoid the full computation of modulo multiplication. After computing the RAM Address, the final interleaved address is computed by the multiply?add function. The interleaved addresses are obtained by performing the inter row T (i) and intra row Ui,j permutations in executing phase, where T(i) is stored in a LUT while the Ui,j is stored in a RAM. With these parameters and the using the equation (6), the interleaving addresses i_addr are finally generated.

$$i_a ddr\ ((i \times C) + j) = (C \times T(i)) + Ui, j \tag{6}$$

with i = 0,1,...,R-1; j = 0,1,...,C-1

[C×T(i)] is computed by the architecture shown in Figure 6.The result of this computation points to the first element of each of the 'R' rows in the rectangular matrix, and then by adding Ui,j with the same architecture, a displacement along every row is obtained. Ui,j is obtained by calculating modulo operation (j × r(i)) mod (p-1).It has the same form as (v × S(j-1))mod p in the pre-computation phase. Since the computation of S (j) and Ui,j are performed at different phases, the same architecture is used with some modifications. Since the same architecture is used in both the phases, hardware complexity is reduced.

3.3 Modulo Computation

One of the important blocks in the proposed reconfigurable architecture is "Modulo Computation". This block plays a crucial role both in execution phase and in pre-computation phase.

Figure 10 shows the architecture of modulo computation introduced by Rizwan Asghar and Dake [10]. The main block is depicted in Figure 10 by blue colour. It is separated and shown in Figure 11a. In pre-computation phase, this architecture works 100% efficiently. But in execution

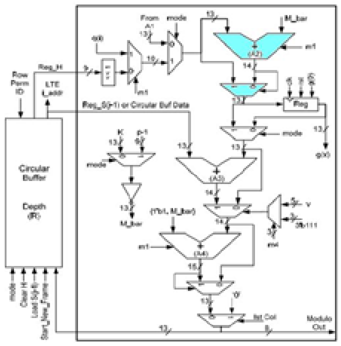


Figure 10: Modulo Computation

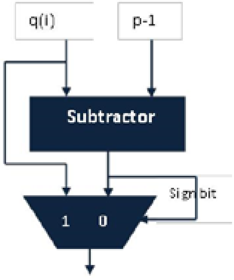


Figure 11a: q(i) Mod p-1

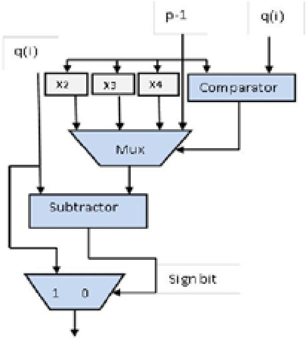


Figure 11b: q(i) mod p-1 with Modification

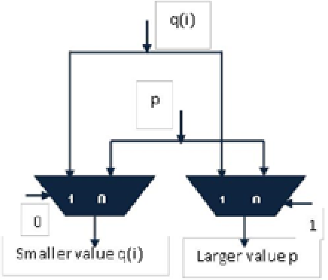


Figure 12: Modified Architecture

phase, its efficiency is only 88%. The problem will occur when q (i) > 2(p-1)[10]. This problem is nullified by performing subtractions like q(i) - (p-1), q(i) - 2(p-1), q(i) - 3(p-1) and q(i) - 4(p-1) instead of q(i) - (p-1) [11]. It is illustrated in Figure 11b. In the proposed work, this structure is further modified as shown in Figure 12.

In order to nullify the problem, always q(i) must be less than p. In the proposed work q (i) is compared with p. After comparison the small value is assigned to q (i) and the large value is assigned to p .Hence in the proposed work, during the execution phase also modulo computation block works 100% efficiently. There are three multipliers are used.[11] to generate 2(p-1), 3(p-1) and 4(p-1). But the proposed modified architecture does not contain any multipliers; hence hardware complexity is reduced. This architecture performs all modulo operations required by the 3GPP standard. With this modification the obtained hardware architecture works properly in both the pre-computing and executing phase.

4 Control Finite State Machine

In order to synchronize every block in the architecture, a "Controller" block generates all control signals; its state machine diagram is shown in Figure 13.

The controller mainly works for the pre-computation phase of 3GPP. It configures different functions like multiply, add, compare and modulo computation hardware to find all the vital parameters R, C, p, v and S (j). Usually interleavers in the turbo coded system work with the same block size 'K' several times before changing it. When an interleaving operation is required the pre-computation phase is performed, where S (j) is calculated and placed in RAM. Then the execution phase starts where interleaved addresses are calculated continuously meanwhile 'K'

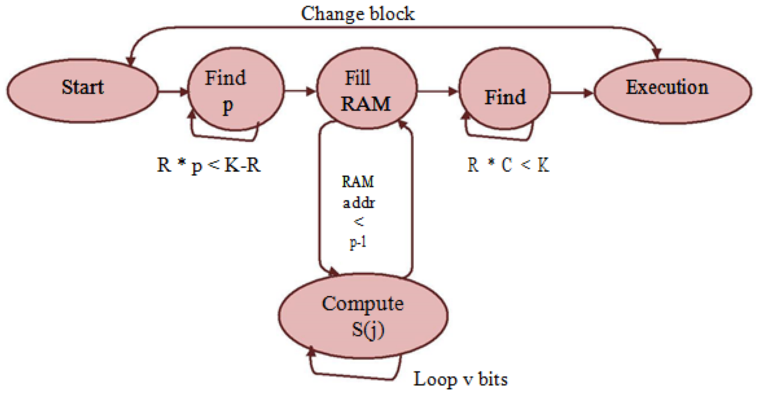


Figure 13: Control State Machine for Interleaver/Deinterleaver

does not change. The new pre-computation phase can be initiated by changing the block size 'K'.

5 Exception Handling

In the 3GPP interleaving algorithm, there are some exceptions that are reflected in the architecture. The final address is tagged valid or invalid using the comparator. This is called pruning of the interleaver and is needed for the case when interleaver block size is not exactly equal to R*C. For example, for K = 43 the rectangular matrix size is 5x10, and then there would be seven invalid addresses. Because of that, a valid address cannot be generated per clock cycle.

Interleavers presented by Shin and Park [12] Wang and Li[13], [11] and Carlos Sánchez [14] provides the interleaved addresses with valid or invalid tag. But in the proposed architecture, if the input data stream of any size between 40 and 5114, then the interleaver rearranges it according to the corresponding 3GPP interleaved path and provides the data stream even the exception is present. In order to manage data streams, a data RAM is used in the proposed design. When the proposed architecture is operated in interleaver mode, this RAM is used to hold data input when invalid addresses are generated and waiting for a valid address in order to write this data to DATA RAM. In deinterleaver mode, when invalid address is read from DATA RAM, the I/O RAM ignores this data, waiting for a valid data and then output this data.

6 Performance Analysis Of Reconfigurable Interleaver

The RTL code for the hardware blocks is written in Verilog and downloaded it to the FPGA Cyclone III from ALTERA. The RTL diagram of reconfigurable interleaver / deinterleaver is shown in Figure 14.

The number of clock cycles spent for pre-computation through is mentioned in Table 3. These cycles mainly depend on the value of p and v for a particular block size.

The maximum clock frequency of 69.91 MHz is shown in Figure 15.Table 4 summarizes comparison of the performance with existing state-of-the-art interleavers. As it can be seen from the Table 4, the proposed design occupies about 4586 out of 5136 logic elements.

It is worth mentioning that although the proposed design is bigger than others, it includes the hardware for exceptions handling, an I/O RAM and a DATA RAM as well as extra hardware to control them. In this way, the proposed architecture is capable of working both as an interleaver and as a de-interleaver.In this proposed interleaver/deinterleaver latency and throughput can

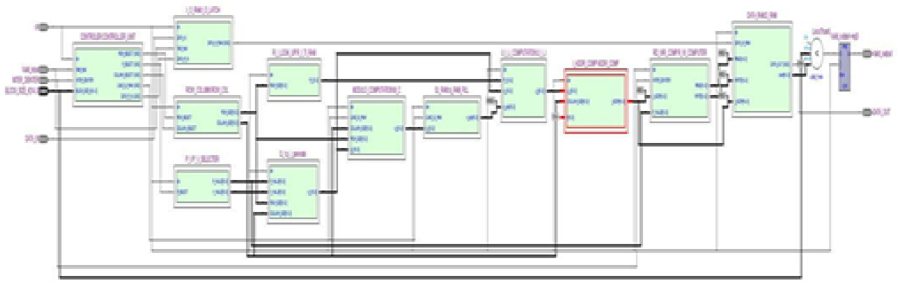
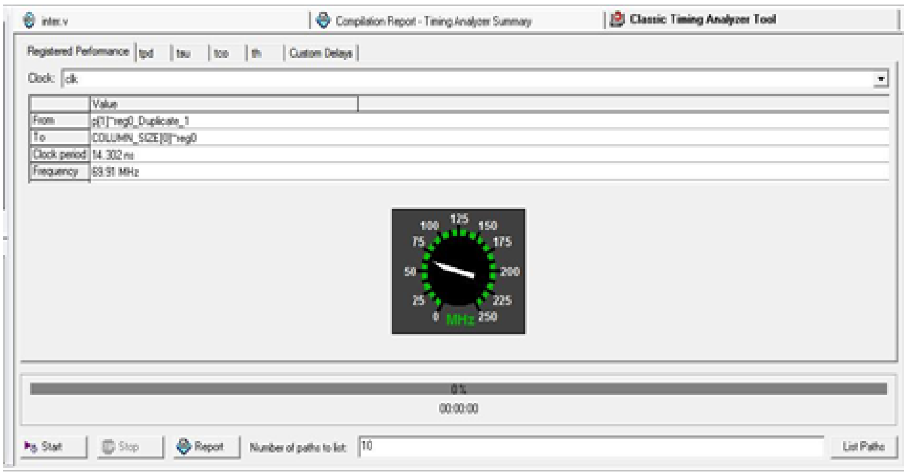


Figure 14 RTL Diagram of reconfigurable interleaver / deinterleaver

Table 3 Average Interleaved Address Rate for Different Block Size

Block Size	Pre-computation Cycle	Execution Phase Cycles/ Frame	Average valid address /cycle
40	27	40	1.000
41	27	42	0.970
2041	132	2050	0.995
4241	493	4410	0.961
4840	557	4840	1.000
5114	569	5120	0.998



Clock: clk

	Value
From	p[1]~reg0_Duplicate_1
To	COLUMN_SIZE[0]~reg0
Clock period	14.302 ns
Frequency	69.91 MHz

Number of paths to list: 10

Figure 15: Timing Analyzer Summary

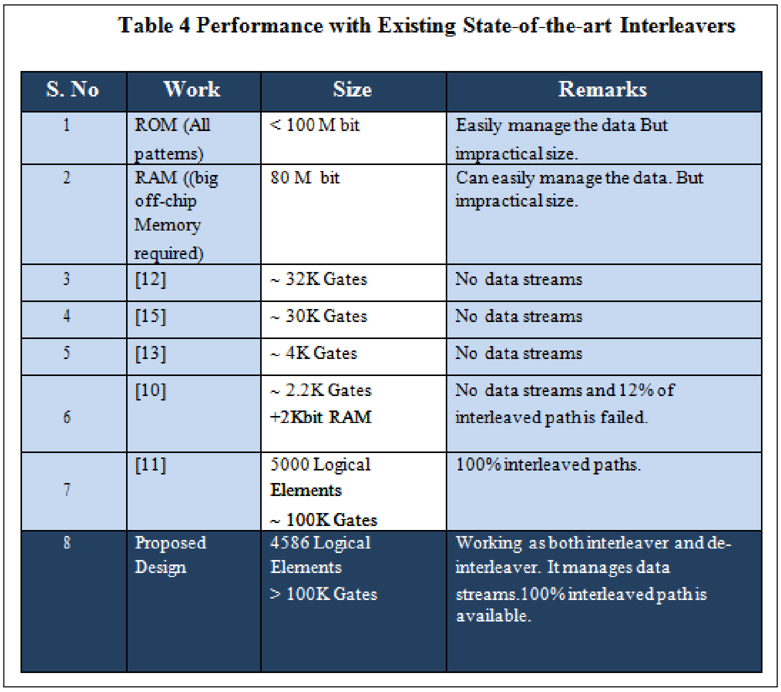


Table 4 Performance with Existing State-of-the-art Interleavers

S. No	Work	Size	Remarks
1	ROM (All patterns)	< 100 M bit	Easily manage the data But impractical size.
2	RAM ((big off-chip Memory required)	80 M bit	Can easily manage the data. But impractical size.
3	[12]	~ 32K Gates	No data streams
4	[15]	~ 30K Gates	No data streams
5	[13]	~ 4K Gates	No data streams
6	[10]	~ 2.2K Gates +2Kbit RAM	No data streams and 12% of interleaved path is failed.
7	[11]	5000 Logical Elements ~ 100K Gates	100% interleaved paths.
8	Proposed Design	4586 Logical Elements > 100K Gates	Working as both interleaver and de-interleaver. It manages data streams.100% interleaved path is available.

vary depending on the block size. The maximum block size is K=5114.The proposed architecture requires 5120 clock cycles which represents a realistic insignific ant overhead.

7 Conclusions and Future Work

In this paper a fully functional 3GPP Turbo code interleaver/ deinterleaver architecture that receives an input data stream of any size established by the 3GPP standard and delivers this stream interleaved or deinterleaved depending on the user requirements is presented. In this proposed design modulo computation is done by computer algorithm to calculate and took advantage of using the same hardware in pre-computation and execution phases by multiplexing it. Novel VLSI architectures have been introduced in this design. By adding RAMs for data handling and achieved a complete architecture that can perform interleaving/deinterleaving operations as required by the 3GPP standard for Turbo codes.One of the important parameter is pre-computation cycle cost. If the pre-computation cycle cost is less then, the design supports fast switching among different standards. Hence future work is to design reconfigurable interleaver which is suitable for a multimode environment.

Bibliography

[1] 3rd Generation Partnership Project (3GPP) TSG-RAN, "*Multiplexing and Channel coding,*" Release 4, Version 4.2.0, Sept. 2001.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima (1993), Near-Shannon Limit Error- Correcting Coding and Decoding : Turbo Codes, *Proceedings of ICC, Geneva,Switzerland*, 1064-1070.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima (2003), Near Shannon limit error correcting coding and decoding:Turbo codes. *Proceedings of the IEEE International Conference on Commun*, Geneva, Switzerland, May 2003.

[4] M Valenti (1999), *Iterative Detection and Decoding of Wireless Communications*, Ph.D thesis, Virginia Polytechnic and State University, July 1999.

[5] University of South Australia, Institute for Telecommunications Research, Turbo coding Research group, http://www.itr.unisa.edu.

[6] Rizwan Asghar and Dake Liu (2010), Multimode Flex- Interleaver Core for Baseband Processor Platform, *Journal of Computer Systems, Networks, and Communications*, 1-16.

[7] Rizwan Asghar and Dake Liu (2010), Towards radix4,parallel interleaver design to support high throughput turbo decoding for reconfigurability, *33rd IEEE SARNOFF symposium*, Princeton, New Jersey, USA, 1-5.

[8] *3rd Generation Partnership Project, Technical Specification Group Radio Access Network; Multiplexing and Channel Coding*,Release 6, 3GPP TS 25.212 v6.0.0 (2003-12).

[9] G.R.Blakley (1983), A Computer Algorithm for Calculating the Product A*B mod M, *IEEE Trans. On Comp*, 32(5):497-500.

[10] Rizwan Asghar and Dake Liu (2008), Very low cost Configurable Hardware Interleaver for 3G turbo decoding, *3rd International Conference on Information and Communications Technologies Theory to Applications*, ICTTA 2008, Damascus, Syria, 1-5.

[11] Hector Borrayo - Sandoval , R. Parra-Michel, Luis F.González-Pérez, Fernando Landeros Printzen Claudia Feregrino-Uribe (2009), Design and Implementation of a Configurable interleaver/deinterleaver for Turbo Codes in 3GPP Standard , In *Proc. of IEEE International Conference on Reconfigurable Computing and FPGAs*, Cancun, Mexico, 320- 325.

[12] M. Shin and I.-C. Park (2003), Processor - based turbo interleaver for multiple third generation wireless standard, *IEEE Commun. Lett.*, 7(5):210 - 212.

[13] Z. Wang and Q. Li (2007), Very low - complexity hardware interleaver for Turbo decoding, *IEEE Trans. on Circuits and Sys. - II*, 54(7):636 - 640, July 2007.

[14] Carlos R.Sanchez, R. Parra-Michel and M.E Guzmán- Renteria (2008), Design and implementation of a multi-standard interleaver for 802.11a, 802.11n, 802.16e & DV standards, *International Conference on Reconfigurabl Computing and FPGAs, ReConFig 2008*, Cancun, Mexico, 379 - 384.

[15] P.Ampadu and K. Kornegay (2003), An efficient hardware interleaver for 3G turbo decoding, *Proc. of Radio and Wireless Conference, RAWCON'03*, Boston, 199-201.

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):201-208, April, 2014.

Robust PID Stabilization of Linear Neutral Time-Delay Systems

H. Farokhi Moghaddam, N. Vasegh

Hassan Farokhi Moghadam*
Faculty of Electrical and Computer Engineering
Shahid Rajaee Teacher Training University
Lavizan, Tehran, Iran.
*Corresponding author: hassan.farokhi89@gmail.com

Nastaran Vasegh
Faculty of Electrical and Computer Engineering
Shahid Rajaee Teacher Training University
Lavizan, Tehran, Iran.
E-mail: n.vasegh@srttu.edu

Abstract: This paper deals with the problem of robust stabilization of neutral time-delay systems using Proportional-Integral-Derivative (PID) controller. A graphical approach which will obtain the set of robust stabilizing PID controllers is presented. The main point in this approach is the frequency response of the neutral system, which can effectively reduce the mathematical complexities of system modeling. It is shown that our the proposed method is effective and useful for many of real control systems deal with time-delays and parametric uncertainties. It is illustrated by an example at the end.

Keywords: Robust stabilization, PID controller, Time-delay, Neutral systems.

1 Introduction

Proportional-Integral-Derivative (PID) controllers have become the best known controllers since their entrance to industrial applications and are used frequently due to their simplicity and good performance [1]. Despite their wide application, determining of all stabilizing regions of PID controllers has been studied just only in the last decade [2]. In [3, 4], a mathematical generalization of the Hermite–Biehler theorem to find all stabilizing PID controllers for systems with time–delay has been used. In [5], the entire region of PID controller has been obtained for linear time–delay systems. In recent years, the robust stabilization of neutral time–delay systems has received considerable attention and has been has been the focus of much research (see [6, 7] and references within).

In this paper, a graphical PID controller design approach for control systems with neutral time–delay is proposed which guarantees the required stabilizing margins of the closed loop system for the uncertain system. Basic knowledge of robust control theory seems sufficient in this proposed approach and the method is easy to understand. Therefore, simplicity is the main characteristic of the approach. The results of the simulations show the effectiveness of proposed method.

The rest of the paper is organized as follows. Section 2 states the problem and preliminary. In section 3 the nominal and robust stability criterion are discussed. All different stabilizing regions which will ensure nominal and robust stability are determined in section 4. Simulation results which will show the correctness of the design approach are shown for an illustrative example in section 5. Finally, results of this paper are summarized in section 6.

2 Problem Statement

The most general form of PID controller is the second order system in the s-domain defined by

$$K(s) = k_p + \frac{k_i}{s} + k_d s. \tag{1}$$

In this paper, this controller is implemented for a linear neutral delayed system with the following general transfer function form:

$$G(s) = \frac{r(s)}{p(s) + q(s)e^{-Ts}} \tag{2}$$

where T is time–delay $p(s)$, $q(s)$ and $r(s)$ are real polynomials. According to [8], the system is of neutral type if degrees of $p(s)$ and $q(s)$ are the same.

PID Robustness of this type of system is the main consideration in this paper. Although there are different forms of model uncertainty [9], but an additive uncertainty modeling structure which is shown in Figure 1, is chosen in this paper as this uncertainty description is closely related to the transfer function representation of system.

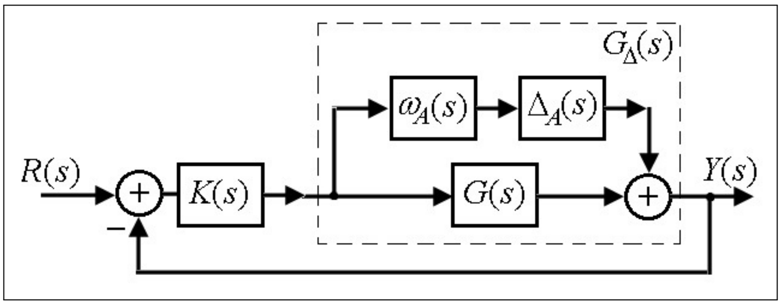


Figure 1: Feedback system with additive uncertainty [9]

Here $K(s)$ represents the PID controller defined by transfer function in (1), the input signal is $R(s)$ and the output signal is $Y(s)$, $G_\triangle(s)$ is the uncertain system, $G(s)$ is the nominal system without uncertainty, $\omega_A(s)$ means the additive weight and is a rational transfer function in this study and $\Delta_A(s)$ is a stable transfer function so that $\forall s\ |\Delta_A(s)| \leq 1$.This kind of uncertainty can be represented by following equation in frequency domain approach.

$$G_\triangle(j\omega) = G(j\omega) + \omega_A(j\omega)\Delta_A(j\omega), \quad |\Delta_A(j\omega)| \leq 1\,\forall\omega. \tag{3}$$

3 Stabilization of Linear Uncertain Neutral Delayed systems

In this section, we study the mathematical formulations for the nominal and robust stability problem in details in following sub–sections.

3.1 Nominal Stability of Linear Neutral Delayed Systems

In the nominal case, based on standard unity feedback like Figure 1 (without uncertainty), the characteristic equation in frequency domain ($s = j\omega$), can be written as

$$\Delta(j\omega) = 1 + K(j\omega)G(j\omega). \tag{4}$$

PID controller and transfer function form at (1-2) can be represented as

$$\begin{aligned} K(j\omega) &= k_p + j(k_d\omega - \frac{k_i}{\omega}), & (5) \\ G(j\omega) &= \frac{N((j\omega))}{D(j\omega)} = \frac{r(j\omega)}{p(j\omega) + q(j\omega)(\cos(T\omega) + j\sin(T\omega))}. & (6) \end{aligned}$$

Let $M(j\omega) = N(j\omega)D(-j\omega)$, so

$$G(j\omega) = \frac{\mathrm{Re}(M(j\omega)) + j\mathrm{Im}M(j\omega)}{|D(j\omega)|^2} = \frac{R_M(j\omega) + jI_M(j\omega)}{|D(j\omega)|^2}. \tag{7}$$

Therefore (4) can be written as

$$\Delta(j\omega) = 1 + (k_p + j(k_d\omega - \frac{k_i}{\omega}))(\frac{R_M(j\omega) + jI_M(j\omega)}{|D(j\omega)|^2}). \tag{8}$$

The numerator becomes

$$\begin{aligned} &\omega k_p R_M(j\omega) + k_i I_M(j\omega) - k_d\omega^2 I_M(j\omega) + \omega|D(j\omega)|^2 + \\ &\qquad j(k_p\omega I_M(j\omega) - k_i R_M(j\omega) + k_d\omega^2 R_M(j\omega)) \end{aligned} \tag{9}$$

Now all stabilizing regions can be determined like what is shown in Figure 2 (Right). Note that we complete the calculations here just for a constant value of k_d in the (k_p, k_i) plane, for brevity. The (k_p, k_d) plane for a constant value of k_i is obtained similarly, but for the (k_i, k_d) plane at a constant value of k_p, nominal and robust stability cannot be obtained directly by (15) and (21) in the next section. For more accuracy, based on [10], for constant value of k_i and k_d stability boundaries are defined in the (k_p, k_i) plane and (k_p, k_d) plane, respectively and then the lines that define the boundary of stability are extracted. The illustrative example in section 5 shows the effectiveness of the approach.

3.2 Robust Stability of Linear Neutral Delayed Systems

In order to analyze the stability in presence of uncertainty nominal system should be stable. For getting robust stability in this condition, following theorem is stated from [9].

Theorem 1. *Small gain theorem: If $L(s)$ is a system with a stable loop transfer function, then the closed-loop system is stable if*

$$||L(j\omega)||_\infty < 1, \quad \forall\omega, \tag{10}$$

where $L(s) = K(s)G(s)$.

As a result from this theorem, if $||L(j\omega)||_\infty < 1$ the closed loop system is stable. Using this theorem and uncertainty structure in Figure 1, it is usually said that if system is robustly stable, the following constraint is met for additive uncertainty:

$$\Big|\Big|\frac{\omega_A(j\omega)K(j\omega)}{1 + K(j\omega)G(j\omega)}\Big|\Big|_\infty \leq \gamma, \quad \forall\omega. \tag{11}$$

For robust case, we can rewrite (6) as:

$$\Delta_R(j\omega) = 1 + K(j\omega)G(j\omega) - \frac{1}{\lambda}(\omega_A(j\omega)K(j\omega)). \tag{12}$$

The rational function can be written as:

$$\omega_A(j\omega) = R_\omega(\omega) + jI_\omega(\omega). \tag{13}$$

Then by continuing calculation in (9), the numerator of (12) can be simplified as

$$\Delta_R(j\omega) = \mathrm{Re}(\Delta_R(j\omega)) + j\mathrm{Im}(\Delta_R(j\omega)). \tag{14}$$

4 All Stabilizing PID Controller Regions

The following steps are proposed to find the boundaries of all stabilizing PID controller.

1. Decompose the frequency form of the system without delay and uncertainty into real and imaginary parts, and substitute them into (5)-(9) to obtain the parameters.
2. Analyze the neutral system in presence of time-delay and additive uncertainty, and do the previous step again. (Eqs. (5)-(14) should be considered)
3. Determine the PID nominal and robust stability boundaries of the system using their related equations in this section, respectively.
4. Finally, plot all the different stabilizing regions.

Here based on the step 3, we'll complete equations obtained in the previous section. For the nominal case, by putting (10) equal to zero, we have

$$\begin{aligned} &k_p\omega R_M(j\omega)+k_iI_M(j\omega)-k_d\omega^2I_M(j\omega)+\omega|D(j\omega)|^2=0,\\ &k_p\omega I_M(j\omega)-k_iR_M(j\omega)+k_d\omega^2R_M(j\omega)=0. \end{aligned} \tag{15}$$

which results in

$$k_p(\omega))=-\frac{|D(j\omega)|^2R_M(j\omega)}{|M(j\omega)|^2}, \tag{16}$$

and

$$k_i(\omega))=k_d\omega^2-\frac{\omega|D(j\omega)|^2I_M(j\omega)}{|M(j\omega)|^2}. \tag{17}$$

Since

$$R_M(j\omega)=|M(j\omega)|\cos(\theta(\omega))\quad I_M(j\omega)=|M(j\omega)|\sin(\theta(\omega)) \tag{18}$$

where $\theta(\omega)=\angle G(j\omega)$, (16) and (17) can be simplified as

$$k_p(\omega))=-\frac{|D(j\omega)|^2\cos(\theta(\omega))}{|M(j\omega)|}, \tag{19}$$

and

$$k_i(\omega))=k_d\omega^2-\frac{\omega|D(j\omega)|^2\sin(\theta(\omega))}{|M(j\omega)|}. \tag{20}$$

Robust consideration yields

$$\begin{aligned} &\Big(\omega R_M(j\omega)-\frac{\omega|D(\omega)|^2}{\lambda}R_\omega(\omega)\Big)k_p+\Big(I_M(j\omega)-\frac{|D(j\omega)|^2}{\lambda}I_\omega(\omega)\Big)k_i=\\ &\Big(I_M(j\omega)-\frac{|D(\omega)|^2}{\lambda}I_\omega(\omega)\Big)\omega^2k_d-\omega|D(j\omega)|^2,\\ &\Big(\omega I_M(j\omega)-\omega\frac{|D(\omega)|^2}{\lambda}I_\omega(\omega)\Big)k_p+\Big(-R_M(j\omega)+|\frac{D(j\omega)|^2}{\lambda}R_\omega(\omega)\Big)k_i=\\ &\Big(\frac{D(j\omega)|^2}{\lambda}R_\omega(\omega)-R_M(j\omega)\Big)\omega^2k_d. \end{aligned} \tag{21}$$

which implies

$$k_p(\omega))=-\frac{|D(j\omega)|^2\Big(R_M(j\omega)-\frac{D(j\omega)|^2}{\lambda}R_\omega(\omega)\Big)}{Q(j\omega)}, \tag{22}$$

and

$$k_i(\omega)) = \omega^2 k_d - \frac{\omega|D(j\omega)|^2\Big(I_M(j\omega) - \frac{D(j\omega)|^2}{\lambda} I_\omega(\omega)\Big)}{Q(j\omega)}, \tag{23}$$

where

$$Q(j\omega) = |M(j\omega)|^2 + |\omega_A(j\omega)|^2 - 2\frac{D(j\omega)|^2}{\lambda}(R_M(j\omega)R_\omega(\omega) + I_M(j\omega)I_\omega(\omega)). \tag{24}$$

which can be simplified using (18) if necessary. Using (21)–(24), the nominal stability boundary and robust stability region are obtained in the (k_p, k_i) plane. The approach is explained in the following illustrative example.

5 Illustrative Example

In this example we consider the neutral time-delay system with following transfer function considered in [8].

$$G(s) = \frac{1}{(s+1)^4(s+1+se^{-s})}$$

The range of the unknown time–delay is considered as $T \in [0.5, 1.5]$. So the uncertain model of the system is $G_\Delta(s) = \frac{1}{(s+1)^4(s+1+se^{-Ts})}$. For this system, we can write (3) as,

$$\Big|\frac{G_\Delta(s) - G(s)}{\omega_A(s)}\Big| \leq 1, \quad \forall\omega. \tag{25}$$

Therefore, it is required to find

$$|\omega_A(j\omega)| \geq |G_\Delta(j\omega) - G(j\omega)| \tag{26}$$

where

$$|G_\Delta(j\omega) - G(j\omega)| = \Big|\frac{1}{j\omega+1)^4}\Big|\Big|\frac{1}{j\omega+1+j\omega e^{-j\omega T}} - \frac{1}{j\omega+1+j\omega e^{-j\omega\overline{T}}}\Big|. \tag{27}$$

For determining the $\omega_A(s)$, at first we plot the additive uncertainty which is the right part of inequality and is shown in Figure 2 and then based on approximate estimation, transfer function form is determined. MATLAB/SISOTOOL is used to facilitate this choice. The MATLAB/cftool can also be used by fitting a good rational function. Here it is obtained as

$$\omega_A(s) = \frac{1.6s^2}{(s+1)^4(s+2)}. \tag{28}$$

This is specified in Figure 2 (Left) which clearly covers the entire uncertainty region. Then the all PID stabilizing regions for this system with $T = 0.5$ are shown in Figure 2 (Right) which is obtained as a result of analysis done in previous sections. Figure 4 (Right side) shows the (k_i, k_d) plane for $k_p = 0.2$. The intersection of all regions inside the nominal stability boundary of the (k_p, k_i) plane is the robust stability region.

To check the correctness of stabilizing regions in Figures 6 and 7, two points from each plane are chosen: one is selected from the robust stability region and another one is selected from nominal region (outside this region). Then the transfer functions of PID controllers are extracted and the constraint criterion in (11) is checked by Bode plot. For example in the (k_p, k_i) plane in Figure 5 (Left), the robust chosen point is $(1.02, 0.301)$ and another one is $(2.47, 0.298)$. Their related PID controllers are:

$$K_1(s) = 1.02 + \frac{0.301}{s} + 0.3s, \quad K_2(s) = 2.47 + \frac{0.298}{s} + 0.3s$$

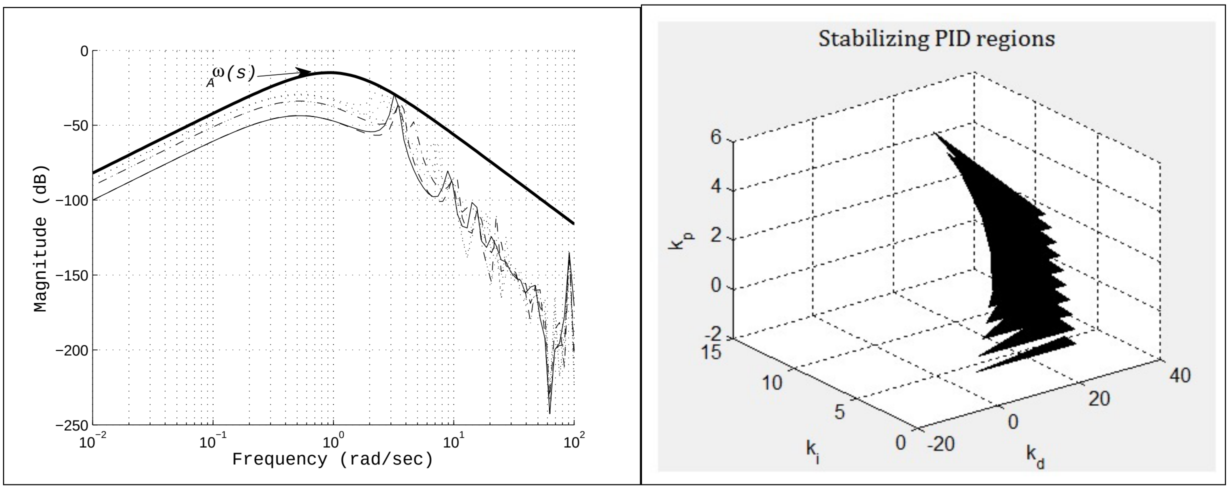


Figure 2: Left; Uncertainty region and the additive uncertainty weight. Right; All nominal stabilizing PID regions in the case $T = 0.5$.

Now it can be seen that for robust $K_1(s)$, we have $\left\|\frac{\omega_A(j\omega)K_1(j\omega)}{1+K_1(j\omega)G(j\omega)}\right\|_\infty = 0.2476$ but for K_2, it results in $\left\|\frac{\omega_A(j\omega)K_2(j\omega)}{1+K_2(j\omega)G(j\omega)}\right\|_\infty = 1.4 > 1$. It is clear that our expectation has been satisfied. In Figure 5 (left side), their magnitude is shown for different frequencies which is verified these calculations. Repeating this procedure in MATLAB shows that any controller selected from inside the robust stability region is capable of robust stabilization of the system.

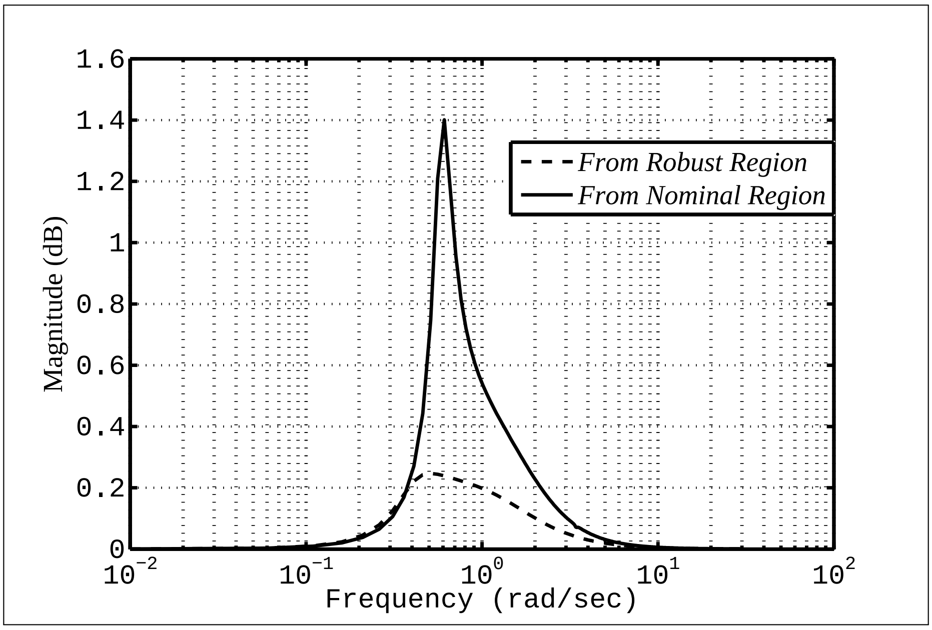


Figure 3: Magnitude of two different nominal and robust points.

Step response of the closed loop system for the entire unknown time-delay range is also plotted in the left side of Figure 4 to show the stability and good performance of robust system. The nominal and the robust stability region in the (k_p, k_d) plane for $k_i = 0.5$ and in the (k_p, k_i) plane for $k_d = 0.3$ are also shown in Figure 5 (right and left), respectively.

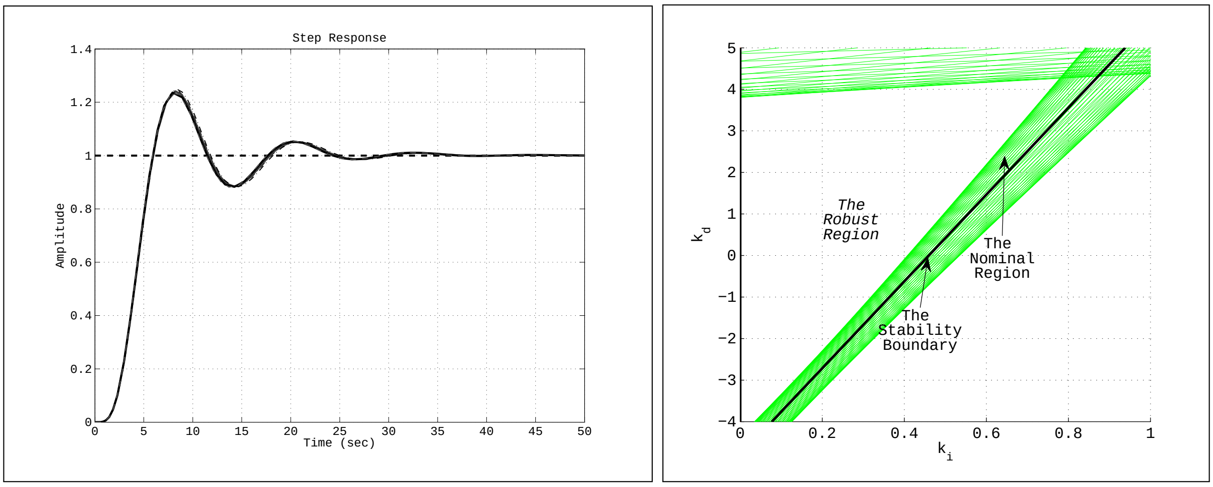


Figure 4: Left; Closed loop step response for entire uncertainty range. Right; The stabilizing region in the (k_i, k_d) plane for $k_p = 0.2$.

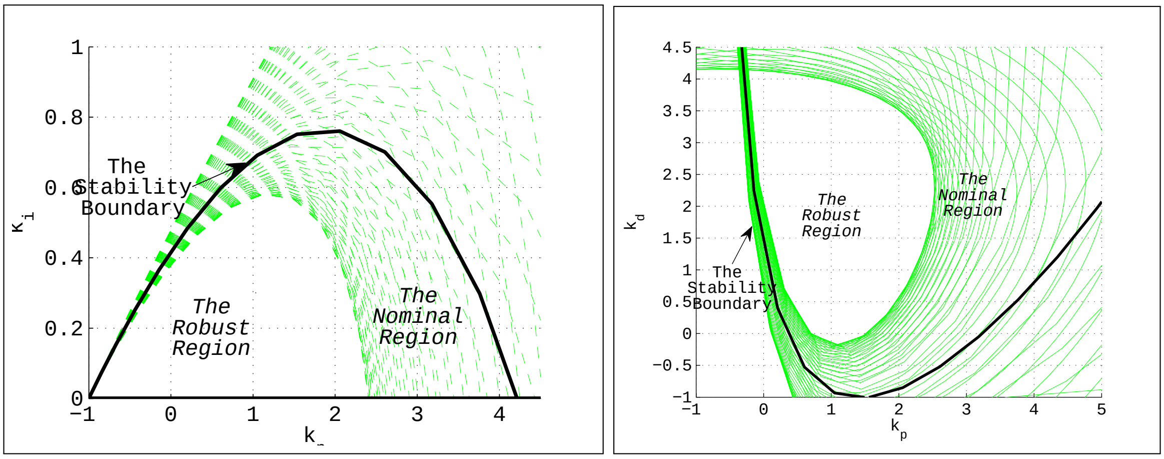


Figure 5: The nominal and robust stability region: Left; (k_p, k_i) plane for $k_d = 0.3$. Right; (k_p, k_d) plane for $k_i = 0.5$.

6 Conclusion

In this paper, a graphical design approach was introduced for finding all PID stabilizing regions of uncertain neutral time-delay system. This proposed design method was based on the frequency response of the system and it can reduce the complexities involved in system modeling. An example for illustrating this approach was studied. The results were satisfactory and the nominal and robust stability regions were determined.

Bibliography

[1] O.T. Altinoz, A.E. Yilmaz, G.-W. Weber (2012), Application of Chaos Embedded PSO for PID Parameter Tuning, *Int J Comput Commun*, ISSN 1841-9836, 7(2):204-217.

[2] B. Fang (2010), Computation of stabilizing PID gain regions based on the inverse Nyquist plot, *J Proc Cont*, 20:1183-1187.

[3] K. W. Ho, A. Datta, S.P. Bhattacharya (1999), Generalizations of the Hermite-Biehler theorem, *Lin Alg its App*, Vol. 302-303, pp. 135-153.

[4] K. W. Ho, A. Datta, S.P. Bhattacharya (2003), PID stabilization of LTI plants with time-delay, *Proceeding 42nd IEEE Conference on Decision and Control*, 4038-4043.

[5] N. Hohenbichler (2009), All stabilizing PID controllers for time-delay systems, *Automatica*, 45:2678-2684.

[6] L. Dugard, E. I. Verriest (1998), *Stability and Control of Time-delay Systems*, London: Springer-Verlag.

[7] O.M. Kwon, J.H. Park, S.M. Lee (2008), On stability criteria for uncertain delay-differential systems of neutral type with time-varying delays, *AppL Math Compu*, 197:864-873.

[8] J. R. Partingtona, C. Bonnet (2004), H∞ and BIBO stabilization of delay systems of neutral type, *Syst & Cont Lett*, 52:283-288.

[9] S. Skogestad, I. Postlethwaite (2001), *Multivariable Feedback Control*, John Wiley & Sons.

[10] G.J. Silva, A. Datta, S.P. Battacharrya (2005), *PID Controllers for Time-Delay Systems*, Birkhauser, Boston, 2005.

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):209-216, April, 2014.

Comprehensive Energy Efficient Algorithm for WSN

C. Tang

Chengpei Tang
School of engineering, Sun Yat-sen University
Guangzhou 510006 P. R. China
E-mail: tchengp@mail.sysu.edu.cn

Abstract: Wireless sensor networks has been widely used. Energy problem is one of the important problems influencing the complete application. Sensor nodes use batteries as power source and have quite limit lifetime. So, efficiency of energy management becomes a key requirement in wireless sensor network design. Based on particle swarm optimization and ant colony optimization, a comprehensive algorithm with weight analysis has been proposed in the paper. In the algorithm, optimization method would be firstly used to determine the nodes number; then, particle swarm optimization would be used to divide the networks into some clusters; finally, ant colony optimization is used to require the best transmission path and select the cluster head. The simulation results show that the new algorithm has higher energy efficiency and balanced energy consumption. It can extend the network lifetime.

Keywords: Wireless Sensor Network (WSN), particle swarm optimization, ant colony optimization, energy efficiency.

1 Introduction

Wireless Sensor Networks (WSNs) offer a new way of real-time monitor systems that can be used in ample of real life applications, such as temperature, sound, pressure, and so on [1]- [4]. It is a wireless network and often composed of many nodes or sensors to monitor vast kinds of conditions. Generally speaking, WSN consists of vast number of sensors which should be cheaply enough to use in low cost. Then, there would be many limitations on the nodes, such as cheap power sources, cheap circuit, and so on. Sensors play critical role in the collection of information, and the power source is quite important. Usually, the sensor use battery as power source and due to the wireless network, the battery can not be charged or replaced, so, it is quite important to develop proper algorithms to reduce the energy consumption.

LEACH is one of the most popular clustering mechanism in WSN. Many communication protocols based on LEACH for WSN have been developed. The main research fields are energy efficiency, wireless link reliability [5], real-time capabilities [6], or quality-of-service. For the reason of sensor nodes just can use battery without power input. So, energy efficiency research is always a key issue in WSN design with high reliability. Many methods and algorithms based on LEACH have been proposed, such as improved LEACH [7]- [10], NDEA [11], TB-LEACH [12], LEACH-SM [13], energy balanced based routing protocol [14], V-LEACH [15], TL-LEACH and DD-LEACH [16], LEACH-HPR [17], etc. Research of energy efficiency is an endless research, and more and more methods to extend lifetime of nodes or networks would be proposed.

In this paper, a comprehensive algorithm based on particle swarm optimization and ant colony optimization with weight analysis has been proposed. The algorithm is mainly used in multi-hop conditions, which can not be set many base stations. In the algorithm, firstly, particle swarm optimization would be used to divide the networks into more than one clusters; secondly, cluster head would be elected with existing energy and distance weighted; thirdly, ant colony optimization would be used to require the best information or data transmission path.

The remainder of the paper is organized as follows: LEACH algorithm would be revisited in section 2. Comprehensive LEACH algorithm and simulation and verification would be expressed in section 3. Section 4 would give the conclusion.

2 LEACH

LEACH protocol is a kind of algorithm. In the network, the nodes work in cluster. In the cluster, a node would be selected as the cluster head [18]- [20]. The process is organized in periodical manner, and each round would be divided into two steps: cluster building step and stable data communication step. In the first step, close nodes would make a cluster dynamically, and one node would be chosen to be the cluster head, which collects, processes and send information to a sink node. In the second step, each node in the cluster would sent message to the cluster head, and then the head would deal with and send it. In the process, head node would collect and fuse information and send it to sink node, so it would consume more energy than other nodes. Leach algorithm could meet the demand that each node in one cluster would have equal possibility to be the head node to balance the exist energy.

The election algorithm of head node in LEACH would be described as following: (1) every node would produce a random number between 0 and 1, and if the number is less than a predefined value $T(n)$, then it would be elected as cluster head and send declaration to other nodes, and if the node has been the cluster node before, and the $T(n)$ would be set to 0. It means that the node cannot be head again. If a node has not been the head node before, the probability of being selected is $T(n)$. $T(n)$ would increase with the number of being head node increases. So the nodes, which have not been head node before, would have a bigger probability. When just one node of not being elected as cluster head node has left, $T(n)$ of the node would be set as 1. That means the node would be cluster head.

$$T(n) = \begin{pmatrix} \frac{p}{1-p(r \bmod (1/p))}, if(n \in G) \\ \\ 0 \qquad otherwise \end{pmatrix} \tag{1}$$

Where, p means the probability of the number of cluster head in all nodes of the cluster; r is the number of the current round, G is the nodes which have not been the head node before. After the election of the head node, the head node would send information to other nodes. And other nodes would join different clusters dynamically based on the distance and some other indexes. When all the nodes join the clusters, they would send message to the cluster head and the cluster head would produce the time message TDMA to inform all the nodes in the clusters. Due to keep the nodes in one cluster to join another, the head node would also send CDMA code at the same time. Each node would send message to the head node in time-interval after they receive the TDMA and CDMA code. When the transmission is over, cluster head would collect and process the data and results would be sent to sink node and the round would be going into next.

3 C-LEACH

Comprehensive LEACH (C-LEACH) is aiming at low cost, lifetime extended WSN.

3.1 Main idea of C-LEACH

The flowchart of C-LEACH is shown in figure 1. It includes some main steps: (1) particle swarm optimization would be used to divide the network into more than one clusters with similar number of nodes; (2) cluster head would elected with existing energy and distance weighted; (3) ant colony optimization would be used to require the best information or data transmission path.

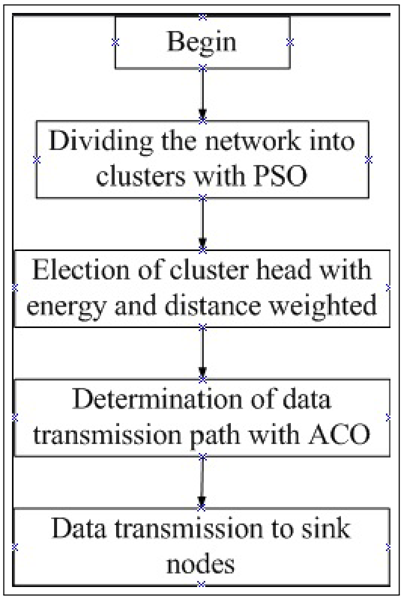

Begin
Dividing the network into clusters with PSO
Election of cluster head with energy and distance weighted
Determination of data transmission path with ACO
Data transmission to sink nodes

Figure 1: Flowchart of C-LEACH

3.2 The detail steps of algorithm

Step 1: Division of the network into some clusters

(1) It is assumed that there are N nodes in the network, and would be going to split into M clusters, this means there are N/M nodes in every cluster. Firstly, draw a splitter to make the whole network into two domains with same nodes number, and the split line would be expressed as:

$$U = (x, y, \theta).(7)$$

Where, (x, y) is location on the splitter of nodes, θ is the angular between the splitter and x-axle.

Define the function $fitness$ as the following:

$$fitness = (c_1 - f_1 N)^2 + (c_2 - f_2\mathrm{N})^2.(8)$$

Where, $c_i(i = 1, 2)$ is the number of nodes in domain i. f_i would be determined as the following:

$$f_i = \frac{M_i}{M}, i = (1, 2).(9)$$

Where, M_i is the expectation of number of cluster nodes in domain i.

Then, we would complete the first division.

The algorithm of cluster division is composed of some steps:

(1) All nodes in the network send data (including all kinds of information) to sink node. Sink node would split the network into many clusters after received the message and define Q particles;

(2) Set random number to parameters of x, y, θ, and construct the split line. The whole network would be divided into $Q \times 2$ clusters. Due to the location of nodes in the network are known, $c_i(i = 1, 2)$ of each node would be determined to calculate the values $fitness$.

(3) Compare the value of $fitness$ with minimum $fitness$ value in the last round, the less one would be elected as a general extrum P_{gd}; Compare the $fitness$ value of individual node, the least would be elected as individual extrum P_{id}, and update the value of x, y, θ:

$$X_{xid} = X_{xid} + v_{xid}.(10)$$

$$X_{yid} = X_{yid} + v_{yid}.(11)$$

$$X_{\theta id} = X_{\theta id} + v_{\theta id}.(12)$$

Where, X_{xid}, X_{yid} represent the location of particle; $X_{\theta id}$ is the angular of the splitter; $v_{xid}, v_{yid}, v_{\theta id}$ are the search speed in three dimensions, and they would be determined as following:

$$v_{xid} = \omega v_{xid} + c_1 \times rand()(P_{id} - X_{xid}) + c_2 \times rand()(P_{gd} - X_{xid}).(13)$$

$$v_{yid} = \omega v_{yid} + c_1 \times rand()(P_{id} - X_{yid}) + c_2 \times rand()(P_{gd} - X_{yid}).(14)$$

$$v_{\theta id} = \omega v_{\theta id} + c_1 \times rand()(P_{id} - X_{\theta id}) + c_2 \times rand()(P_{gd} - X_{\theta id}).(15)$$

Where, c_1 and c_2 are the study factor, and $c_1 = c_2 = 2$; rand () is random number between 0 and 1; ω is the weighted factor.

(4) After the update of x, y, θ, go to step 2 to continue the research process. When the value of $fitness$ is 0 or equal to the maximum times of search, the process would finished. Ideally, if the value of $fitness$ is approximately 0, and the whole network would be divided into two clusters with equal nodes.

(5) Use the method of above to divide the clusters, until get the demanded number of clusters.

Step 2: Search the nearest multi-hop path by ACO

In this step, first of all is electing cluster head. High existing energy of nodes would be elected as cluster head followed by LEACH.

This search is based on single cluster, so the path in each cluster would be different. And one search path has no influence to that of others. That is to say path search in each cluster is independent. Cluster nodes send data or information to sink nodes, if need multi-hop would adopt this algorithm to find the nearest path.

According to the definition described before, forward neighbor nodes would be expressed as:

$$N_f(i) = \{v_j | v_j \in V, d_{i,j} \le r, \alpha_{i,j} \le \beta_i\}.(16)$$

Each node would have a storage model to reserve the information between self and neighbor nodes, including location, existing energy and neighbor node location and existing energy, and so on. In order to express the pheromone, each node would have a record τ_{ij} to store the density of pheromone. Ant in the network is a data package with memory and storage capability, which just require minimal space. The ant would have characteristics as following:

(1) The ant can remember the information of nodes it passed;

(2) The ant has capability of recording the nodes it passed with a proper order and forming a path, the ant would return after it get to the cluster node and update the pheromone.

(3) Ant just can jump to the forward neighbor nodes;

(4) Ant can read and modify the information it locates. Ants in the network would begin with source node, and goes to the cluster head with a way of multi-hop to find a nearest way between source node and cluster head node. Ant locates at v_i would calculate the jump probability $P_{i,j}^k$ according to the information of pheromone and existing energy of the neighbor nodes.$P_{i,j}^k$represents the probability of ant k jump to forward neighbor nodes. It is calculated as following:

$$P_{i,j}^k = \begin{cases} \frac{[\tau_{i,j}]^\mu \cdot [\eta_{i,j}]^\lambda}{\sum\limits_{v_H \in N_f(v_f)} [\tau_{i,H}]^\mu \cdot [\eta_{i,H}]^\lambda}, v_j \in N_f(v_i) \\ \\ 0 \quad otherwise \end{cases} \quad (17)$$

Where, $\tau_{i,j}$ is value of path $l_{i,j}$ from node v_i to its neighbor node v_j; μ is a factor about distance and pheromone and it means the effect of information accumulated to the travel of ants. Bigger the value of μ is, shorter path the ant would choose. λ is a factor of energy, which means the effect of energy in the path selection. If value of λ is bigger, the ant would choose a higher existing energy node to jump. At first time t_0, pheromone $\tau_{i,j}(t_0)$ on the path $l_{i,j}$ of nearby nodes. It can be calculated as:

$$\tau_{i,j}(t_0) = \frac{d_{i,D}}{d_{i,j} + d_{j,D}} \times (1 - \frac{d_{i,j}}{\sum\limits_{v_H \in N_f(v_i)} d_{i,H}}).(18)$$

$\eta_{i,j}$ is a function of energy, which can be calculated as following:

$$\eta_{i,j} = \frac{e_j}{\sum\limits_{v_H \in N_f(v_i)} e_H}.(19)$$

Where, e_H is existing energy of forward neighbor node $v_H \in N_f(H)$ of node i.

Forward neighbor nodes set limits the jumped node. After the v_j is elected to be the next jump node of v_i, next jump node of v_j would be calculated according to the location of cluster head. Following the same method, the ant would jump to the cluster head. In the searching path, if the current sensor j has no forward neighbor nodes, it would be marked as invalid node. The ant would return to node i, and node j with no forward neighbor node would be deleted. Then, the ant would select the next jump node again. If all nodes in forward neighbor nodes set are deleted, this means there is no available path between source node and cluster head. If a source node has no available path to the cluster head, it has to transmit the information to its nearest node outside the forward neighbor nodes set, and choose the best path with the method described before. An ant travel from source node to cluster head and return back to the source node with the pheromone information updated is defined to be a process. All ants finish a process, we can say a loop is completed and loop index t would plus 1. In the loop , the pheromone density in the path l_{ij} would be adjusted according to equation as following:

$$\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t) + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k}(t).(20)$$

$$\Delta\tau_{i,j}^{k}(t) = \begin{pmatrix} \frac{Q}{L^k(t)}, & \text{If ant k pass} \\ & \\ 0, & \text{otherwise} \end{pmatrix} \quad (21)$$

Where, m is the number of ants; $\Delta\tau_{i,j}^{k}(t)$ is pheromone left on path l_{ij} at loop t; $L^k(t)$ is the length of ant k moves at loop t; Q is a constant about pheromone. ρ is also a constant which describe the volatilization of pheromone. Value of ρ is often between 0 and 1 to keep the convergence of the algorithm. In addition of judgment, nearest path of each node would be found.

3.3 Simulation and verification

The simulation result of C-LEACH would compared with the result of ACO with the whole network as a cluster and all information of each node transfer directly to sink node without to cluster node. Same model has been used to verify the advance of two algorithms. Figure 2 shows the energy consumption of the two algorithm in sending 100 data packages. It is easy to see the energy saving with C-LEACH.

Figure 3 shows the efficiency of node energy with different algorithm in sending 100 data packages. It can be seen that energy efficiency is higher under C-LEACH than that under ACO algorithm.

Figure 4 shows the success rate in path search with different algorithm. It can be seen that success rate of less nodes in the network with C-LEACH is lower than that under ACO and when the network with more nodes, the success rate under C-LEACH is higher. The reason may be the division of the network, and each cluster has less nodes. We can make a decision that the method used in network with more nodes would have a better effect.

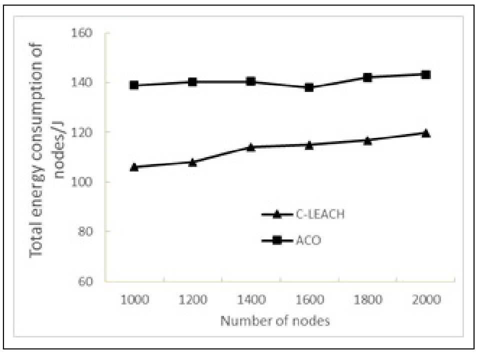


Figure 2: Node energy consumption with different algorithm

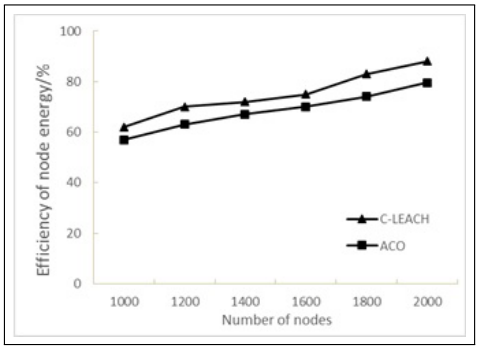


Figure 3: Efficiency of node energy with different algorithm

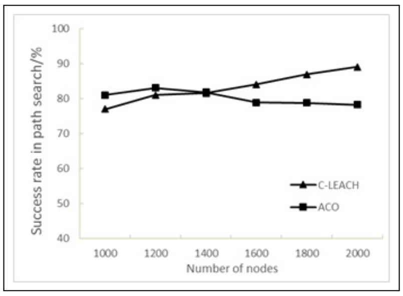


Figure 4: Success rate in path search with different algorithm

Figure 5 shows the lifetime of the network with two different algorithms. From the figure, we can see the lifetime of network under C-LEACH is longer than that under ACO.

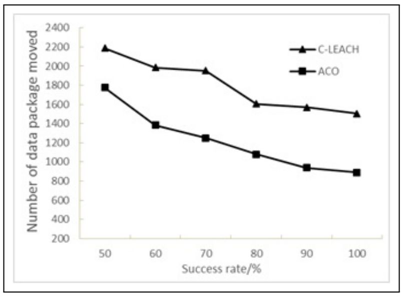


Figure 5: Lifetime of network

4 Conclusion

The paper proposed a comprehensive method to save energy of WSN. It uses PSO dividing the whole network into clusters and then ACO has been used to find the nearest path to transfer the data package. Results of the simulation show that, C-LEACH is an effective method used in the WSN with multi-hop conditions, especially used in vast area with more nodes. The method can extend the lifetime of the network and increase the energy efficiency.

Bibliography

[1] Hart, J. K., Martinez, K. (2006). Environmental Sensor Networks: A revolution in the earth system science. *Earth-Science Reviews*, 78: 177-191.

[2] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, J. Lees (2006). Deploying a Wireless Sensor Network on an Active Volcano. *IEEE Internet Computing*, 10(2):18-25.

[3] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, P. Corke (2005). Data collection, storage, and retrieval with an underwater sensor network. In *Proc.of the 3rd international conference on Embedded networked sensor systems*, 154-165.

[4] Martinez, K.; Hart, J. K.; Ong, R. (2009). Deploying a Wireless Sensor Network in Iceland. Lecture Notes in Computer Science, *Proc. Geosensor Networks*, 5659, 131-137.

[5] Anastasi. (2010). A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4, *Wireless Sensor Networks*, 7(1):52-65.

[6] Pruter S., Moritz G., Zeeb E., Golatowski F., Timmermann D (2008). Applicability of Web Service Technologies to Reach Real Time Capabilities. *11th IEEE Int. Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, 229-233.

[7] Yuhua Liu, Yongfeng Zhao, JingjuGao (2009). A New Clustering Mechanism Based On LEACH Protocol. *2009 Int. Joint Conference on Artificial Intelligence*, 715-718.

[8] Fuzhe Zhao, You Xu, Ru Li, Wei Zhang (2012). Improved Leach Communication Protocol for WSN. *2012 Int. Conf. on Control Engineering and Communication Technology*, 700-702.

[9] Jia Xu, Ning Jin, Xizhong Lou, Ting Peng, Qian Zhou, Yanmin Chen (2012). Improvement of LEACH protocol for WSN. *2012 9th Int. Conf. on Fuzzy Systems and Knowledge Discovery (FSKD 2012)*, 2174-2177.

[10] Wei Wei, Peiyi Shen, Liang Zhang, Hu Xu, Juan Song, Wenzeng Zhang, Wei Wang (2012). LEACH-Based Energy-Conserved Improved Protocol for WSNs. *International Journal of Digital Content Technology and its Applications (JDCTA)*, 6:163-171.

[11] Weiping Luan, Changhua Zhu, Bo Su, Changxing Pei.(2012). An Improved Routing Algorithm on LEACH by Combining Node Degree and Residual Energy for WSNs. *IOT Workshop 2012, CCIS*, 312, 104 C109.

[12] Hu Junping, Jin Yuhui, Dou Liang (2008). A Time-based Cluster-Head Selection Algorithm for LEACH. *2008 IEEE*, 1172-1176.

[13] Bilal Abu Bakr, LeszekLilien (2011). A Quantitative Comparison of Energy Consumption and WSN Lifetime for LEACH and LEACH-SM. *2011 31st Int. Conf. on Distributed Computing Systems Workshops*, 182-191.

[14] Pan Xue-feng, LI La-yuan (2011). Design of an Energy Balanced Based Routing Protocol for WSN. *2011 IEEE*, 366-369.

[15] Mrs. Asha Ahlawat, MsVineeta Malik (2013). An EXTENED VICE-CLUSTER SELECTION APPROACH TO IMPROVE V LEACH PROTOCOL IN WSN. *2012 Third Int. Conf. on Advanced Computing and Communication Technologies*, 236-240.

[16] Ravi Kishore Kodali, NarasimhaSarma, NVS. (2013). Energy Efficient Routing Protocols for WSN's. *2013 Int. Conf. on Computer Communication and Informatics (ICCCI -2013).*

[17] Li Han (2010). LEACH-HPR: An Energy Efficient Routing Algorithm for Heterogeneous WSN. *2010 IEEE*, 507-511.

[18] Xu Long-long, Zhang Jian-jun (2010). Improved LEACH Cluster Head Multi-hops Algorithm in Wireless Sensor Networks. *Ninth Int. Symposium on Distributed Computing and Applications to Business, Engineering and Science*, 10-12.

[19] Zhuang Jun, Qiang Chun-Xia, Feng Wan-Li (2012). Research of cross-layer and multi-hops algorithm based on energy and location. *Proc. of the 2012 International Conference on Industrial Control and Electronics Engineering, ICICEE 2012*, 1781-1784.

[20] Yang Yong-Jian, Jia Bing, Wang Jie (2013). An improved algorithm for LEACH protocol in wireless sensor network. *Journal of Beijing University of Posts and Telecommunications*, 36(1): 105-109.

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):217-226, April, 2014.

A Metamodel for an Adaptive Control System

F. Valles-Barajas

Fernando Valles-Barajas
Universidad Regiomontana, Information Technology Department
15 de Mayo 567 pte., C.P. 64000 colonia centro, Monterrey,
Nuevo León, México, Tel. +52 81 8220-4733
E-mail: fernando.valles@acm.org, fernando.valles@ieee.org

Abstract: In this paper a metamodel for an Adaptive Control System (ACS) is described. This metamodel was built employing USE, which is a UML-based specification environment. The main goal of the metamodel is to complement other models describing different views of an ACS. As the reader will notice, the metamodel is composed of a graphical and a mathematical model. Weak constraints are specified in the graphical model using a Unified Modeling Language (UML) class diagram, while strong constraints are defined in the mathematical model using the Object Constraint Language (OCL).

Keywords: Adaptive Control, metamodels, OCL, UML

1 Introduction

A software process describes who is doing what, how, and when [9]. One of the phases of a software process is design. In this phase a model of a system, that will later be implemented, is constructed [15]. This model is useful to detect flaws but also for documenting and to establish a communication channel with system's user. Depending on the process being constructed designers can use text or mathematics or a combination of both to build models. An advantage of using text, which is an informal technique, is that the resulting design is easy to understand and can be rapidly constructed. Its disadvantage is that sometimes a model made using an informal technique can lead to a misunderstanding. Formal models, which rely on mathematics, do not have this disadvantage.

On the other hand, control systems are used in industry to assist control engineers in maintaining processes in a desired state (see [3], [25], [10]). Control algorithms are embedded in control systems. Sometimes control systems are applied to control critical systems, whichrequire free errors designs and because of one of the part of control systems is software, recently control engineers are applying software processes in the building of control algorithms [18].

Proposal of the paper: Usually a design covers one view of the system being modeled. According with [7], the design views of system are: structural, procedural and behavior. While making the design's structural view, the designer should document constraints affecting system entities and the relations between them [7]. This documentation can be made using text or a formal language to get a more precise specification. In this paper, the author explains how a semi formal modeling language and a formal language can be used to specify constraints affecting the parts of an adaptive control system as well as constraints affecting the relations between them. The Unified Modeling Language (UML) [5], [14] and the Object Constraint Language (OCL) will be used to specify these constraints.

Previous works: The first step in a software process is to gather requirements system. In [17] a requirements process for control system software is presented. This process is based mainly in the Rational Unified Process (RUP).

As is recommended in Personal Software Process (PSP) [7], once the requirements are captured by software engineers, the next step is to make a pre-design for the purpose of prediction and planning. In [19] a proposal for the building of pre-design for control systems is presented.

Further details, not covered in the pre-design, can be specified in the design phase. PSP proposes that design can be analyzed from several perspectives. In [20] a proposal, based on PSP, for modeling the structural view of control systems is introduced; in this view the entities of the system, its attributes and relations among the system entities are modeled. Once the entities of system are specified, software designer should detect entities having several states and model them using a state machine. In [21] the authors model an adaptive control system using a state machine.

The design views proposed in PSP are a subset of the UML diagrams, which model systems in more detail. In [23] a survey of the application of UML to model mechatronics systems is presented.

Once software design is made, code must be built. In [18] and in [22] the authors explain best practices in using programming languages at the moment code for control system software is made.

Related works: The Z language is a formal modeling language based on first-order logic and set theory. This modeling language has been applied to the specification of critical systems. In [8], this language is used to specify a control program for a radiation therapy machine.

Another application of the Z modeling language in critical systems is described in [15], where a system that monitors the blood glucose level of diabetics and automatically injects insulin when it is required is specified in the Z language.

The disadvantage of the Z language in comparison with OCL, which is the modeling language used in this paper, is that OCL complements UML and it does not use mathematical symbols to make specifications. The latter characteristic could be attractive to control engineers not familiar with mathematical logic.

Control systems are usually modeled using differential equations and analyzed using, for example, Lyapunov stability theory. In [2] a novel approach based on Hoare logic for reasoning about control systems is presented.

In [12] the authors present intent-specifications model for a robotic software control system. According to the authors, an intent-specification is composed of seven levels that as a whole model the entire software control system. Each level models the system from a different perspective and, in particular, in level 4 a design representation of the system is included.

The Unified Modeling Language (UML) has also been used to model software for control systems. In [24] the authors propose a methodology for generating code for Programmable Logic Controllers (PLCs) from UML diagrams. The application of UML for process control was evaluated from a usability and cognitive science point of view. The authors performed an experiment to evaluate the acceptation of UML for control engineers.

Some researchers are studying mapping between traditional tools used by control engineers for software modeling and UML. For example, in [16] the author analyzed mapping between function blocks, which are defined by the International Electro-technical Commission as the basic construct for distributed control applications and UML. When compared with UML, function blocks do not consider all the benefits of object oriented theory. Another paper including functional blocks and UML is [13]; in that paper the authors apply UML activity diagrams to model function blocks.

Theorem provers have also been used to specify and verify the correctness of control software.

For example, in [6] the authors apply the interactive theorem prover PVS to model the Light Control System, which is a benchmark in formal methods.

Outline of the paper: In this section, the motivation of this work has been explained. Section 2 contains a brief description of an adaptive control system. Section 3 describes the tools used in this paper to specify the constraints of an adaptive control system. In section 4 the adaptive control system design is included. The last section contains concluding remarks.

2 Adaptive Control Systems

Fig. 1 shows the configuration of an indirect adaptive control system. In this kind of adaptive control, a model of the process ($Gp(z^{-1})$) is obtained based on a set of input-output measurements ($(u(k), y(k))$) and then the controller ($Gc(z^{-1})$) is designed using this model [10]. The

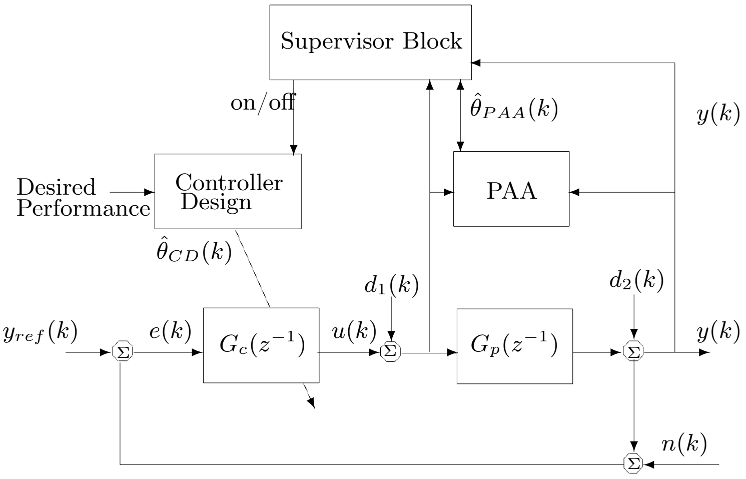


Figure 1: Block diagram for an Adaptive Control System

parameter adaptation algorithm (PAA) block is the responsible for obtaining the parameter vector of the process (see $\hat{\theta}_{PAA}(k)$) in fig. 1). The controller design block specifies the parameters of the controller ($\hat{\theta}_{CD}(k)$) based on the model obtained by the PAA and on the desired performance specified by the system operator.

An adaptive control system must control a process in spite of disturbances $d_1(k)$, $d_2(k)$, noise $n(k)$ and the parametric variations of the process. The supervisor block is in charge of detecting any event that may provoke a decreasing in the performance of system; in these cases the supervisor will turn off the controller G_c

In fig. 1, $y_{ref}(k)$ is the reference, $e(k)$ is the control error, $u(k)$ is the manipulated variable and k is the k^{th} sampling time.

3 UML & OCL: tools to model software systems

UML is a modeling language, based on object oriented theory, developed by the three amigos with the aim of modeling complex software [14]. Every of the UML diagrams models a different view of a system; for example class diagrams specify system entities and the relations between

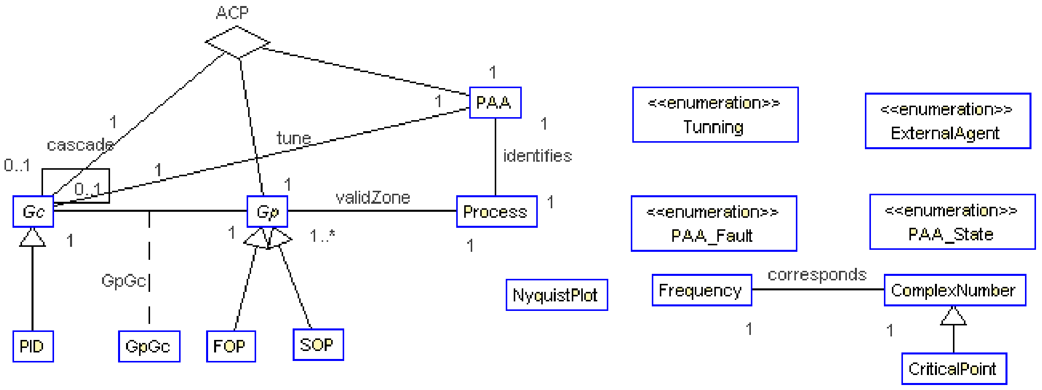


Figure 2: Class diagram for an Adaptive Control System

them. UML has acquired a good acceptation in the software community, though sometimes models created in UML lead to miscommunication.
In these cases it is recommended to complement UML models with an OCL specification, which is a formal modeling language based on logic and set theory. One of the advantages of OCL is that it does not use mathematical symbols to build models; instead it uses a textual representation. For designers without strong background in logic, this characteristic of OCL may result attractive.

4 UML/OCL specification for an Adaptive Control System

In this section the Adaptive Control System metamodel is presented. First of all, a graphical model is described using UML and as will be shown this model contains weak constraints on the elements composing an Adaptive Control System. With the introduction of the UML model the disadvantages of only using graphical models to specify an Adaptive Control System are demonstrated.
A model complementing the graphical model is then constructed using OCL. As the reader will see the weak constraints established in the UML model will be strengthened by using OCL.

4.1 UML specification

Fig. 2 contains a UML class diagram for the ACS of fig. 1. For simplicity, this diagram does not contain all of the details of an ACS; for example, the supervision block and the controller design block are not considered in this diagram. Fig. 3 and 4 contain the USE specification of the class diagram of fig. 2.

As the reader may notice a composition relation between classes G_p (process model), G_c (controller) and PAA (Parameter Adaptation Algorithm) was specified (see the composition ACS in the class diagram of fig. 2 and its definition in the USE specification of fig. 4).
Also, an association class between classes G_p and G_c was defined. As can be seen in the USE specification of fig. 4, this class was defined as an association class because it defines attributes that do not belong to a particular class; for example T_s (time sampling) is associated to the entire system and not to a particular class.
A reflexive relation, named cascade, was defined in the class G_c. Master and slave roles of this relation are specified in fig. 4.

```
model AdaptiveControlSystem

enum ExternalAgent {d1, d2, n}
enum PAA_State {on, off}
enum PAA_Fault {poorExcitation, perturbation, lowM}
enum Tuning {quarterDecayRatio, stepResponse,
  IAE, ITAE, other}

abstract class Gc
attributes
  id  : Integer
  isInAutomatic: Boolean
  R: Sequence(Real)
  S: Sequence(Real)
  T: Sequence(Real)
  tuningType: Tuning
  indicator : String
constraints
  inv updateDisplay:
    isInAutomatic implies indicator = 'automatic'
  inv ValidId:
    id >= 1
end

class PAA
attributes
  theta: Set(Real)
  phi: Set(Real)
  lambda: Real
  state : PAA_State
  faults : Set(PAA_Fault)
operations
  updateGp( B: Sequence(Real), A: Sequence(Real) )
constraints
 inv:
  if  faults->notEmpty then state = #off
  else state  = #on
  endif
end
```

```
class PID < Gc
attributes
  tao_d: Real
  tao_i: Real
  Kc: Real
operations
  assignGp(gp: Gp)
end

abstract class Gp
attributes
  B: Sequence(Real)
  A: Sequence(Real)
  d: Integer
  na: Integer
  nb: Integer
  np: NyquistPlot
operations
  isMonicHurwitzPolynomial (): Boolean
constraints
  inv sructure3:
     d>0 and na>0 and nb>0
  inv structure:
    na >= nb
  inv structure2:
    A->size() = na and B->size() = nb
end

class FOP < Gp
attributes
  K: Real
  theta: Real
  tao: Real
operations
  initGpFOP()
  fromContinousToDiscrete()
constraints
  inv structure:
    nb = 1 and na = 2
end

class SOP < Gp
constraints
  inv structure:
    nb = 2 and na = 3
end
```

Figure 3: USE specification for an Adaptive Control System (part I)

```
class Process
attributes
  identifiedGp: Gp
  affected : Set(ExternalAgent)
end

class Frequency
attributes
  value: Real
end

associationclass GpGc between
  Gp [1] role model2;
  Gc [1] role controller ;
attributes
  Ts: Integer
  affectedBy: Set(ExternalAgent)
  faultHistory : Bag(ExternalAgent)
  gainMargin: Real
  phaseMargin: Real
  modulusMargin: Real
  delayMargin: Real
operations
  registerAFault(f : ExternalAgent)
end

association tune between
  PAA [1] role tunner;
  Gc [1] role controller2 ;
end

association corresponds between
  Frequency [1]
  ComplexNumber [1]
end

class ComplexNumber
end

class CriticalPoint < ComplexNumber
end

class NyquistPlot
attributes
  w: Sequence(Frequency)
  c: Set(ComplexNumber)
end

association cascade between
  Gc [0..1] role master;
  Gc [0..1] role slave;
end

association validZone between
  Process [1] role plant;
  Gp [1..*] role Model;
end

association identifies between
  PAA [1] role identifier ;
  Process [1] role process;
end

composition ACP between
  PAA [1]
  Gp [1]
  Gc [1]
end
```

Figure 4: USE specification for an Adaptive Control System (part II)

The model defined in fig. 2, 3 and 4 contain some flaws; for example this model allows that a controller G_{c_1} plays at the same time master and slave role in the cascade association.
Another design flaw occurs between classes G_c and G_p as is explained as follows. PID controllers have demonstrated to have a good performance when the process being controlled can be modeled as a first or second order process (see [11]), however the class diagram of fig. 2 allows that any kind of process may participate in the relation between G_c and G_p.
These design flaws justify the addition of OCL constraints to the ACS class diagram model shown in fig. 2 and defined in fig. 3 and 4.

4.2 OCL specification

In this section, invariants on the class diagram defined in fig. 2 are defined. Some invariants affect more than two elements of the class diagram. Invariants are explained in the same order they appear in fig. 5 and 6.

Fig. 5 defines three constraints on the class controller, G_c. Invariant *onlyOneRole* assures that a controller G_{c_1} cannot be at the same time, both a master controller and a slave controller. Invariant *differentIDs* constraints that in a cascade relation the controller playing the master role has a different *id* that the controller playing the slave role.
The last invariant of the class G_c specifies that if the type of controller is a PID controller, then an appropriate tuning type must be selected (quarterDecayRatio, stepResponse, IAE or ITAE

```
context Gc
  inv onlyOneRole:
    master−>notEmpty implies slave−>isEmpty

 inv differentIDs:
 (master−>notEmpty implies self.id <> master.id) and (slave−>notEmpty implies self.id <> slave.id)

  inv validGp:
    self.oclIsTypeOf( PID ) implies
      tuningType <> #other and (model2.oclIsTypeOf( FOP ) or model2.oclIsTypeOf( SOP ))
          and model2.oclIsTypeOf( FOP ) implies model2.oclAsType( FOP ).theta < gpGc.Ts

context PID::assignGp( gp: Gp )
  pre: FOP.allInstances−>union( SOP.allInstances )−>includes( gp )
  post: model2.plant.identifiedGp = gp

context NyquistPlot
  inv orderedFreq1:
    let integers = Sequence{1 .. w−>size()} in
      integers−>forAll(e1, e2: Integer | w−>at(e2).value > w−>at(e1).value implies e2 > e1)

  inv orderedFreq2: −− another way to specify invariant orderedFreq1
    w−>forAll(f1, f2: Frequency | w−>indexOf(f2) > w−>indexOf(f1) implies f2.value > f1.value)

  inv stability :
    w−>forAll(f: Frequency | CriticalPoint.allInstances−>excludes(f.complexNumber))

  inv completeness:
    w−>forAll(f: Frequency | ComplexNumber.allInstances−>exists(cn | f.complexNumber=(cn)))
```

Figure 5: USE constraints for an Adaptive Control System (part I)

among others, see the enumeration Tuning defined in fig. 3). Besides this, the process being controlled should be modeled as a first order process or as a second order process. If the process is a first order process then the dead time, θ, should be smaller than T_s.

A method to assign a process model to a PID controller is defined in fig. 5. This method forces that the model, G_p, should be either a first order process or a second order process.

Four invariants are defined on the class NyquistPlot. Invariants *orderedFreq1* and *orderedFreq2* declares that the frequencies specified in a Nyquist plot should be ordered from low to high. Invariant *stability* defines that a Nyquist plot should not contain the critical point.
The last invariant of class NyquistPlot specifies that every frequency in a Nyquist plot has a complex number assigned to it.

Fig. 6 declares a method of the class G_p, which constraints that the polynomial A of G_p should be monic; in other words the leading coefficient should be 1.

The method *updateGp* defined on the class PAA, updates the model G_p as long as the model identified by the PAA is inside the valid zone of the process being controlled and there is not fault affecting the entire system.

Invariant *stabilityCriteria* defines the standard values to get a robust controller recommended in [3]. Method *registerAFault* defined on the association class *GpGc* updates the logging of system faults.

```
context Gp :: isMonicHurwitzPolynomial(): Boolean
  pre: A−>first() = 1.0
  post: result = true

context PAA::updateGp( B: Sequence(Real), A: Sequence(Real) )
  pre: state = #on
  pre: process.Model−>exists(gp: Gp | gp.A = A and gp.B = B)
  pre: Set{faults, process.Model.gpGc.affectedBy}−>isEmpty

  post: process.identifiedGp.A = A and process.identifiedGp.B = B

context GpGc
  inv stabilityCriteria :
    gainMargin >= 6 and modulusMargin >= −6
      and Sequence{ 30, 31 .. 59, 60 }−>includes(phaseMargin) and delayMargin = 0.1*Ts

context GpGc::registerAFault(f : ExternalAgent)
  post: faultHistory = faultHistory@pre−>including(f)

context Process
  inv not_a_PID:
   let ip = identifiedGp in let fop = ip.oclAsType( FOP ) in
     ip.oclIsTypeOf( FOP ) and (fop.theta > fop.tao*0.25 and fop.theta < ip.gpGc.Ts) implies
         PID.allInstances−>excludes( ip.controller )
```

Figure 6: USE constraints for an Adaptive Control System (part II)

The last invariant of fig. 6 constrains the class Process. If the process is modeled as a first order process then θ should be less than 0.25τ (τ is the constant time of a first order process) and less than T_s.

5 Conclusions

In this paper an informal modeling language (UML) and a formal modeling language (OCL) have been applied to model an adaptive control system. UML was used to specify weak constraints whereas OCL was applied to strength the model made in UML.
The author of this paper believes that the model included in this work, will be useful for control engineers that want to have a better understanding of how to apply UML and OCL in the modeling of control systems.
A simple example (an Adaptive Control System) was used in this paper to illustrate how control system software can be specified using an informal notation (UML) together with a formal modeling language (OCL). The technique proposed in this paper should be used to model software for complex control systems, e.g. nuclear plants, robots, planes among others.

The future lines of research based on this paper are:

- Due to the fact that UML is a general purpose modeling language, in some cases a more specialized language gives a more precise specification than that obtained using UML. This drawback of UML has been studied and analyzed by the UML community and to overcome this problem UML extension mechanisms have been proposed; one of these are profiles. A profile based on UML and OCL will be studied in future papers.

- There are some tools to prove if some constraints are fulfilled by one specification (see for example [1] and [4], among others). Building one tool with this characteristic would be useful to check the models created by the technique proposed in this paper.

- Every modeling technique specifies a system from a particular perspective. In this paper, a UML class diagram was used to model the entities composing a control system along with the constraints among them; strong constraints were specified using OCL. Other perspectives not considered in this paper could model the interaction between the entities composing the system and the interaction between the user of a system and the system itself. A study of other models complementing the perspectives of control system software will be made in a future research.

Bibliography

[1] R. Arthan, P. Caseley, C. O'Halloran and A. Smith (2000). *ClawZ: Control laws in Z*, Third IEEE International Conference on Formal Engineering Methods (ICFEM 2000), 2000.

[2] R. Arthan, U. Martin, E. A. Mathiesen and P. Oliva (2007). Reasoning about linear systems, *Proceedings of 5th IEEE International Conference on Software Engineering and Formal Methods SEFM*, 2007.

[3] K. J. Åström and R. M. Murray (2012). *Feedback Systems: An Introduction for Scientists and Engineers, Version v2.11b*, Princeton University Press.

[4] S. Bensalem, P.Caspi, C.Parent-Vigouroux and C.Dumas (1999). A methodology for proving control systems with Lustre and PVS, 7th *Working Conference on Dependable Computing for Critical Applications (DCCA7)*, San Jose, January 1999.

[5] M. Fowler (2003). *UML Distilled*, Addison-Wesley, 3rd edition, 2003.

[6] A. de Groot, J. Hooman (20000), Analyzing the Light Control System with PV, *Journal of Universal Computer Science*, vol. 6, no. 7.

[7] W. S. Humphrey (2005). *PSP : A Self-Improvement Process for Software Engineers*, Addison-Wesley.

[8] J. Jacky, J. Unger, M. Patrick, D. Reid and R. Risler (1997). Experience with Z developing a control program for a radiation therapy machine, *Lecture Notes in Computer Science*, 1212:317-328.

[9] P. Kruchten (2003). *The Rational Unified Process: An Introduction*, Addison-Wesley Professional, 3rd edition.

[10] I. D. Landau and G. Zito (2006). *Digital Control Systems: Design, Identification and Implementation*, Springer.

[11] A. Leva, C. Cox, and A. Ruano (2002). *Hands-on PID autotuning: a guide to better utilization*, IFAC Professional Briefs, 2002.

[12] I. Navarro, K. Lundqvist, and N. Leveson (2001) An intent-specifications model for a robotic software control system, 20th *Conference Digital Avionics Systems*, 2001. DASC.

[13] S. Panjaitan, G. Frey (2005), Functional Design for IEC 61499 Distributed Control Systems using UML Activity Diagrams. *Proceedings ICICI 2005*, Bandung, Indonesia, 64-70.

[14] J. Rumbaugh, I. Jacobson, and G. Booch (2004). *The Unified Modeling Language Reference Manual*, Addison-Wesley Professional, 2nd edition.

[15] I. Sommerville (2010). *Software Engineering*, Addison-Wesley, 9^{th} edition.

[16] K. C. Thramboulidis 92004), Using UML in Control and Automation: A Model Driven Approach, 2^{nd} *IEEE International Conference on Industrial Informatics INDIN'04*, 24th -26th June, 2004, Berlin, Germany.

[17] F. Valles-Barajas (2007), A requirements engineering process for control engineering software, *Innovations in Systems and Software Engineering: A NASA Journal*, 3(4):217-227.

[18] F. Valles-Barajas and W. Schaufelberger (20080, Use of object-oriented languages in control engineering, *Journal of Research in Computing Science*, vol.36. 2008.

[19] F. Valles-Barajas (2009), A SysML requirements model for the 1992 ACC robust control benchmark, *Information Technology and Control*, 38(3):245-251. 2009.

[20] F. Valles-Barajas and W. Schaufelberger (2010), A proposal for the software design of control systems based on the personal software process, *International Journal of Innovative Computing, Information and Control*, 6(8):3451-3466. 2010.

[21] F. Valles-Barajas (2010), A Novel Model for Adaptive Control Systems; A State Machine Approach, *Int J Comput Commun*,ISSN 1841-9836, 5(3):292-300. 2010.

[22] F. Valles-Barajas (2011), A survey of high-level programming languages in control systems, *The International Arab Journal of Information Technology*, 8(2):178-187.

[23] F. Valles-Barajas (2011), A survey of UML applications in mechatronic systems, *Innovations in Systems and Software Engineering: A NASA Journal*, 7(1):43-51.

[24] B. Vogel-Heuser, D. Friedrich, U. Katzke and D. Witsch (2005), Usability and benefits of UML for plant automation some research results *Engineering*, vol. 3, no. 1.

[25] B. Wittenmark, K. J. Åström, and K. E. Årzén. Computer control: An overview, *IFAC Professional Briefs*, 2002.

INT J COMPUT COMMUN, ISSN 1841-9836
9(2):227-242, April, 2014.

Dynamic Behavior Analysis of Membrane-Inspired Evolutionary Algorithms

G. Zhang, J. Cheng, M. Gheorghe

Gexiang Zhang*, Jixiang Cheng
School of Electrical Engineering,
Southwest Jiaotong University
Chengdu, 610031, P.R. China
*Corresponding author: zhgxdylan@126.com

Marian Gheorghe
Department of Computer Science,
The University of Sheffield,
Regent Court, Portobello Street, Sheffield, S1 4DP, UK

Abstract: A membrane-inspired evolutionary algorithm (MIEA) is a successful instance of a model linking membrane computing and evolutionary algorithms. This paper proposes the analysis of dynamic behaviors of MIEAs by introducing a set of population diversity and convergence measures. This is the first attempt to obtain additional insights into the search capabilities of MIEAs. The analysis is performed on the MIEA, QEPS (a quantum-inspired evolutionary algorithm based on membrane computing), and its counterpart algorithm, QIEA (a quantum-inspired evolutionary algorithm), using a comparative approach in an experimental context to better understand their characteristics and performances. Also the relationship between these measures and fitness is analyzed by presenting a tendency correlation coefficient to evaluate the importance of various population and convergence measures, which is beneficial to further improvements of MIEAs. Results show that QEPS can achieve better balance between convergence and diversity than QIEA, which indicates QEPS has a stronger capacity of balancing exploration and exploitation than QIEA in order to prevent premature convergence that might occur. Experiments utilizing knapsack problems support the above made statement.

Keywords: Membrane computing, membrane-inspired evolutionary algorithm, dynamic behavior, quantum-inspired evolutionary algorithm; knapsack problem.

1 Introduction

Membrane computing, initiated by Păun in 1998 [1], focuses on the investigation of the models, called membrane systems or P systems, abstracted from the structure and the functioning of the living cell as well as from the cooperation of cells in tissues, organs, and other populations of cells. Thompson Institute for Scientific Information, ISI, listed the seminal paper as a fast breaking record and this area as an emerging research front in computer science in 2003, and thereby membrane computing becomes a branch of natural computing and has developed very fast into a vigorous scientific discipline [2–6].

Aiming at investigating the interactions between membrane computing and evolutionary computation, membrane-inspired evolutionary algorithms (MIEAs) are considered as a class of hybrid optimization algorithms, which use the concepts and principles of meta-heuristic search methodologies and the hierarchical or network structures of P systems, and to some extent, some of the rules of P systems [7,8]. A MIEA is regarded as a successful paradigm extending P system models with capabilities that make them amenable to real-world applications [9]. Due to a very wide range of applications of meta-heuristic search methodologies, such as genetic algorithms and tabu search, there is a very promising perspective of applying P systems to solve various complex and difficult engineering problems.

In recent years, many investigations referred to MIEAs. The first version of membrane algorithms was designed with a nested membrane structure (NMS) and a local search heuristic for solving travelling salesman problems, which are well-known NP-hard optimization problems [10]. An approach combining NMS and genetic algorithms was presented and tested by using six benchmark functions [11]. In the preceding work, we proposed a MIEA, called a quantum-inspired evolutionary algorithm based on P systems (QEPS), which incorporates a one-level membrane structure (OLMS) and a quantum-inspired evolutionary algorithm (QIEA) [12]. A well-known NP-complete optimization problem, knapsack problem, was used to carry out extensive experiments, which show that QEPS achieves better solutions than its counterpart QIEA and OLMS has an advantage over NMS. In [7, 13–15], QEPS and its modified versions were presented to solve various problems, such as radar emitter signal analysis and image processing. In [16] and [17], DNA sequences design was optimized by designing a MIEA based on crossover and mutation rules and a dynamic MIEA combining the fusion and division rules of P systems with active membranes and search strategies of differential evolution (DE) and particle swarm optimization (PSO), respectively. In [18], a memory mechanism was considered in the design of MIEAs. In [19], a hybrid MIEA was presented by combining OLMS with PSO to solve constrained optimization problems. In [8], a MIEA was proposed by using the network membrane structure of a tissue P system with five cells to organize five representative DE variants.

However, since MIEAs were initiated in 2004, a question has been asked many times by researchers from the areas of membrane computing and evolutionary computation. The question refers to the role played by P systems in MIEAs, that is, what advantages do P systems bring to MIEAs? This is also a critical and tough question that has been haunting many researchers in the field of membrane computing. So the motivation of this work is to try to some extent to find an appropriate answer for this question.

In this study, we propose the analysis of the dynamic characteristics of MIEAs by using a set of population diversity and convergence measures to comparatively investigate the evolving processes of QEPS and its counterpart algorithm, QIEA. Due to the difficulty in theoretically reasoning about MIEAs, we mainly focus on the experimental analysis. Also we present a tendency correlation coefficient to analyze the relationship between the population diversity and convergence measures and fitness to understand the importance of each measure and to provide suggestions on how to improve the performance of MIEAs with respect to population diversity and convergence. Furthermore, experiments conducted on knapsack problems are presented.

The rest of this study is organized as follows. Section 2 gives a brief description of QIEA and QEPS, which is helpful to understand the dynamic behavior analysis of MIEAs expounded in Section 3. Specific examples follow in Section 4 to verify the analysis presented in the preceding section. Section 5 concludes this work.

2 QIEA and QEPS

2.1 QIEA

Inspired by quantum computing, Han and Kim [20] proposed a novel evolutionary algorithm, called QIEA, for a classical computer. QIEA consists of three main components: quantum-inspired bit (Q-bit) representation, a probabilistic observation and a quantum-inspired gate (Q-gate) [21]. In QIEA, a genotypic gene is represented by using a Q-bit defined by a pair of numbers $(\alpha,\ \beta)$ denoted as $[\alpha\ \beta]^T$, where $|\alpha|^2$ and $|\beta|^2$ are probabilities that the observation of the Q-bit will render a '0' or '1' state. A string of Q-bits is applied to represent a Q-bit individual. The connection between genotypic representation (Q-bit representation) and phenotypic individuals (binary solutions) is established by the probabilistic observation. The Q-gate is used to produce offspring. Generally speaking, QIEA is composed of the following steps:

(i) Initialization: a population $Q(t)$ with n Q-bit individuals is generated, $Q(t)$=$\{\boldsymbol{q}_1^t$, $\boldsymbol{q}_2^t$, $\cdots$, $\boldsymbol{q}_n^t\}$ at generation t (here $t = 0$), where $\boldsymbol{q}_i^t$ $(i = 1, 2, \cdots, n)$ is an arbitrary individual in $Q(t)$, which is represented as

$$\boldsymbol{q}_i^t = \begin{bmatrix} \alpha_{i1}^t | \alpha_{i2}^t | \cdots | \alpha_{il}^t \\ \beta_{i1}^t | \beta_{i2}^t | \cdots | \beta_{il}^t \end{bmatrix}, \tag{1}$$

where l is the number of Q-bits, i.e., the string length of the Q-bit individual.

(ii) Observation: a probabilistic observation is used to produce binary solutions $P(t)$, $P(t)$=$\{\boldsymbol{x}_1^t$, $\boldsymbol{x}_2^t$, $\cdots, \boldsymbol{x}_n^t\}$, by observing the states of $Q(t)$, to be specific, a binary bit 0 or 1 is obtained in terms of the probability, either $|\alpha_{ij}^t|^2$ or $|\beta_{ij}^t|^2$ of $\boldsymbol{q}_i^t$, $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, l$. Thus a Q-bit individual with l Q-bits results in a binary solution $\boldsymbol{x}_i^t$ $(i = 1, 2, \cdots, n)$ with l binary bits.

(iii) Evaluation: the binary solution $\boldsymbol{x}_i^t$ $(i = 1, 2, \cdots, n)$ in $P(t)$ is evaluated thus obtaining its fitness. Additionally the best solution among $P(t)$ is stored.

(iv) Offspring generation: Q-gates are performed on Q-bit individuals in $Q(t)$ to produce their corresponding individuals at the next generation. For example, the j-th Q-bit in the i-th Q-bit individual $\boldsymbol{q}_i^t$, $j = 1, 2, \cdots, l$, $i = 1, 2, \cdots, n$ is updated by applying the current Q-gate $\boldsymbol{G}_{ij}^t(\theta)$. QIEA uses a quantum rotation gate as a Q-gate; this is given by

$$\boldsymbol{G}_{ij}^t(\theta) = \begin{bmatrix} \cos\theta_{ij}^t & -\sin\theta_{ij}^t \\ \sin\theta_{ij}^t & \cos\theta_{ij}^t \end{bmatrix}, \tag{2}$$

where θ_{ij}^t is an adjustable Q-gate rotation angle.

(v) Termination condition: the maximal number of evolutionary generations or the maximal number of function evaluations could be utilized to stop the algorithm. If the termination condition is satisfied, the algorithm will stop and output the final results, otherwise, the generation number increases by 1, i.e., $t = t + 1$, and the algorithm goes back to Step (ii).

2.2 QEPS

In the process of investigating the interactions between P systems and evolutionary algorithms, we presented a MIEA, QEPS [12], which was designed with the hierarchical framework of a cell-like P system, the objects consisting of Q-bits and classical bits, the rules made up of Q-gate evolutionary rules in QIEA and evolution rules in P systems. QEPS uses OLMS, where the skin membrane contains m elementary membranes defining m regions. Q-bits, organized as a Q-bit individual in a proper way, are treated as multisets of objects. Classical bits, obtained from their corresponding Q-bits by using a probabilistic observation, are arranged as a binary string and are dealt with also as multisets of objects. In QEPS, a binary string corresponds to a solution of a problem. The set of rules are responsible for evolving the system and selecting the best fit Q-bit individuals. All the objects and rules are appropriately placed in the membrane structure.

More precisely the P system-like framework consists of

(i) a membrane structure $[[\,]_1, [\,]_2, \cdots, [\,]_m]_0$ with m+1 regions contained in the skin membrane, denoted by 0;

(ii) an alphabet that consists of all possible Q-bits and classical bits;

(iii) a set of terminal symbols, $T = \{0, 1\}$;

(iv) initial multisets $w_0 = \lambda$,

$w_1 = \boldsymbol{q}_1\boldsymbol{q}_2 \cdots \boldsymbol{q}_{n_1}$,

$w_2 = \boldsymbol{q}_{n_1+1}\boldsymbol{q}_{n_1+2} \cdots \boldsymbol{q}_{n_2}$,

$\cdots$

$w_m = \boldsymbol{q}_{n_{(m-1)}+1}\boldsymbol{q}_{n_{(m-1)}+2} \cdots \boldsymbol{q}_{n_m}$,

where $\boldsymbol{q}_i$, $1 \leq i \leq n$, is a Q-bit individual; n_j, $1 \leq j \leq m$, is the number of individuals in w_j; $\sum_{j=1}^{m} n_j = n$, where n is the total number of individuals in this computation;

(v) rules which are classified as

a evolution rules in each of the compartments 1 to m which are transformation-like rules updating a Q-bit individual according to the current Q-gate;

b observation rules which make binary solutions from Q-bit individuals;

c communication rules which send the best fit individual binary representation from each of the m elementary membranes into the skin membrane and then the overall best binary representation from the skin membrane to each elementary membrane.

3 Dynamic Behavior Analysis

This section analyzes the dynamic behaviors of MIEAs in the process of evolution from two perspectives, the population diversity and convergence. Six diversity and four convergence measures are introduced to comparatively exhibit the evolutionary behaviors of QEPS and QIEA. We start from population diversity analysis and then turn to convergence analysis. Finally, a tendency correlation coefficient is proposed to evaluate the relationship between diversity and convergence measures and the quality of solutions.

3.1 Population Diversity Analysis

Population diversity is crucial for a population-based search method to prevent premature convergence toward local optima. Diversity measures are used to evaluate the levels and types of varieties of individuals in a population [22]. In this subsection, six diversity measures are considered for QEPS and QIEA, and they are respectively

(1) D_{qbw}: Q-bit distance between the best and worst Q-bit individuals corresponding to the best and worst fitness values in a population, respectively. D_{qbw} is described as

$$D_{qbw} = \frac{1}{m}\sum_{j=1}^{m}\left||a_{bj}|^2 - |a_{wj}|^2\right| \tag{3}$$

where $|a_{bj}|^2$ and $|a_{wj}|^2$ are probabilities of the j-th Q-bit in the best and worst Q-bit individuals, respectively; m is the number of Q-bits in a Q-bit individual. $0 \leq D_{qbw} \leq 1$. A larger value of D_{qbw} gives a hint of larger distance between the best and worst Q-bit individuals.

(2) D_{qa}: average Q-bit distance of all Q-bit individuals in a population. D_{qa} is defined as

$$D_{qa} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left\{ \frac{1}{m} \sum_{k=1}^{m} \left| |\alpha_{ik}|^2 - |\alpha_{jk}|^2 \right| \right\} \tag{4}$$

where $|a_{ik}|^2$ and $|a_{jk}|^2$ are probabilities of the k-th Q-bit in the i-th and j-th Q-bit individuals, respectively; m is the number of Q-bits in a Q-bit individual; n is the number of individuals in a population. D_{qa} is the average value of the Q-bit distance between $n(n-1)$ pairs of Q-bit individuals. $0 \leq D_{qa} \leq 1$. A larger value of D_{qa} suggests a larger distance between each pair of Q-bit individuals in a population.

The two diversity measures above are obtained in Q-bit space, so they can be regarded as genotypic diversity measures for QEPS and QIEA. In what follows we will introduce four phenotypic diversity measures: Hamming distance between the best and worst binary individuals (D_{hbw}) in a population, mean Hamming distance of all binary individuals (D_{hm}) in a population, and two diversity measures based on dispersion statistical measures including the diversity between chromosomes (D_{bc}) and the diversity between the alleles (D_{ba}) [22].

(3) D_{hbw} and D_{hm} are depicted as

$$D_{hbw} = \frac{1}{m} \sum_{i=1}^{m} (x_{bi} \oplus x_{wi}) \tag{5}$$

$$D_{hm} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left\{ \frac{1}{m} \sum_{k=1}^{m} (x_{ik} \oplus x_{jk}) \right\} \tag{6}$$

where x_{bi} and x_{wi} are the i-th bits in the best and worst binary solutions, respectively; m is the number of bits in a binary solution; n is the number of individuals in a population; the symbol $\oplus$ is exclusive OR operator; x_{ik} and x_{jk} are the k-th bits in the i-th and j-th binary solutions, respectively. D_{hbw} and D_{hm} vary between 0 and m. Larger values of D_{hbw} and D_{hm} indicate more varieties between the best and worst binary individuals, and each pair of binary individuals in a population, respectively.

(4) D_{bc} and D_{ba} are defined as

$$D_{bc} = \frac{1}{n-1} \left(\frac{\sum_i S_i^2}{L} - \frac{S^2}{L*n} \right) \tag{7}$$

$$D_{ba} = \frac{1}{L-1} \left(\frac{\sum_j S_j^2}{n} - \frac{S^2}{L*n} \right) \tag{8}$$

where n is the population size; L is the length of a chromosome; S is the sum of genes ‘1’; S_i and S_j are the sum over a row i and the sum over a column j, respectively. D_{bc} and D_{ba} have the following properties [23]:

(i) If the population is homogeneous in either 0 or 1, D_{bc} and D_{ba} equals zero;

(ii) When all the chromosomes in the population are identical, D_{bc} is zero and D_{ba} holds a constant value that is dependent on how many genes ‘1’ are in a chromosome.

In what follows we use knapsack problems, which are described in Section 4.1, to show the changes of the six population diversities in the evolution. Figures 1–6 illustrate comparisons of

QEPS and QIEA for three knapsack problems with 400, 600 and 800 items. Each subfigure in Figs.1–6 provides results of 30 independent random runs (green solid lines for QEPS and cyan solid lines for QIEA) and the mean values over 30 runs (black bold solid lines for QEPS and black bold dash-dot lines for QIEA). The values of D_{qbw}, D_{qa}, D_{hbw}, D_{hm}, D_{bc} and D_{ba} in Figs.1–6 are obtained by setting population size to 20 and the maximal number 20000 of function evaluations (NoFE) as the stopping condition.

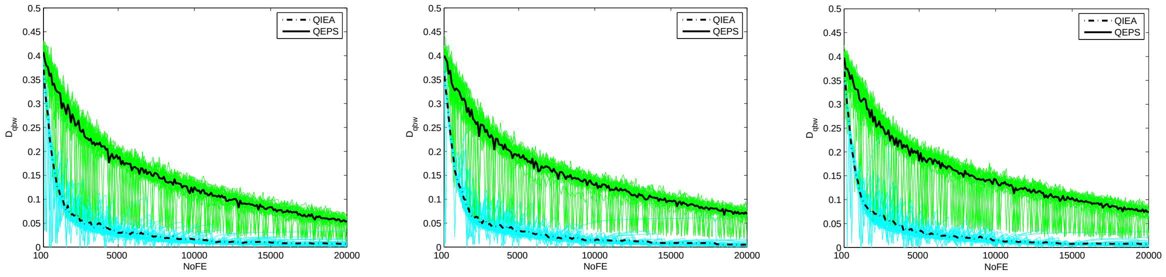
From the results, shown in Figs.1–6, about the comparisons of population diversity between QEPS and QIEA, we can draw the following conclusions:

(i) The three subfigures, which correspond to the respective knapsack problems with 400, 600 and 800 items, in each of Figs.1–6, show respectively consistent trends for QEPS and QIEA, which indicates the reasonableness of the six diversity measures to a certain degree.

(ii) Figures 1–2 show that QEPS can maintain better population diversity than QIEA in Q-bit space. The two algorithms have approximate values, of about 0.4, for D_{hbw} and D_{hm} in the initial states. As NoFE mounts up, the values of D_{hbw} and D_{hm} of Q-bit individuals in QEPS decrease gradually to around 0.5 at NoFE 20000, whereas the values in QIEA rapidly fall below 0.5 at NoFE 3000 and approximate 0 at NoFE 20000.

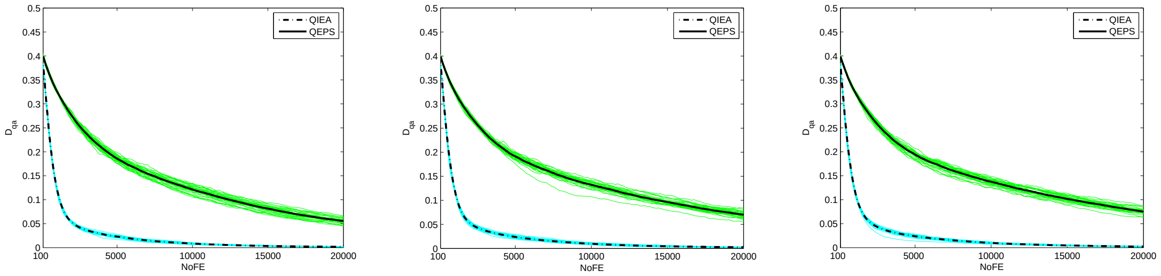
(iii) The Hamming distance D_{hbw} between the best and worst binary individuals and the mean Hamming distance D_{hm} of all binary individuals in Figs.3–4 show a clear picture of QEPS having a greater potential to preserve the population diversity than QIEA. More specifically, QEPS and QIEA have almost the same initial values and decreasing values for changes of D_{hbw} and D_{hm} with respect to NoFE, but QEPS maintains a much higher level of population diversity than QIEA throughout the evolutionary process. The six subfigures in Figs.3–4 also demonstrate that QIEA loses population diversity too fast and very quickly goes down to a small value close to 0 when NoFE reaches values around 10000, which implies that the individuals in QIEA become nearly identical and therefore lose exploration capability. When NoFE increases to 20000, QEPS still has one-fourth of initial values of D_{hbw} and D_{hm}.

(iv) D_{bc} and D_{ba}, two diversity measures based on dispersion statistical measures in Figs.5–6, also clearly illustrate that QEPS has better population diversity than QIEA through the whole process. D_{bc} values of QIEA rapidly fall down to 0 and D_{ba} values of QIEA rapidly rise to and stay at a higher steady level, while QEPS maintains more varieties in the population, even when the algorithm stops at values of NoFE close to 20000.



(a) 400 items (b) 600 items (c) 800 items

Figure 1: D_{qbw} of QEPS and QIEA with items 400, 600 and 800.

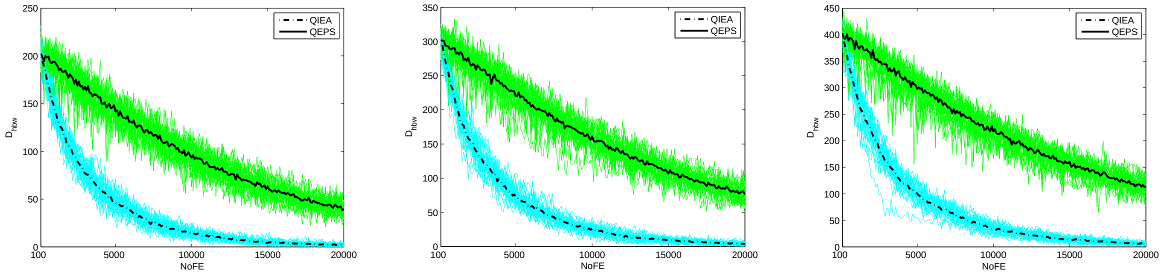


(a) 400 items

(b) 600 items

(c) 800 items

Figure 2: D_{qa} of QEPS and QIEA with items 400, 600 and 800.

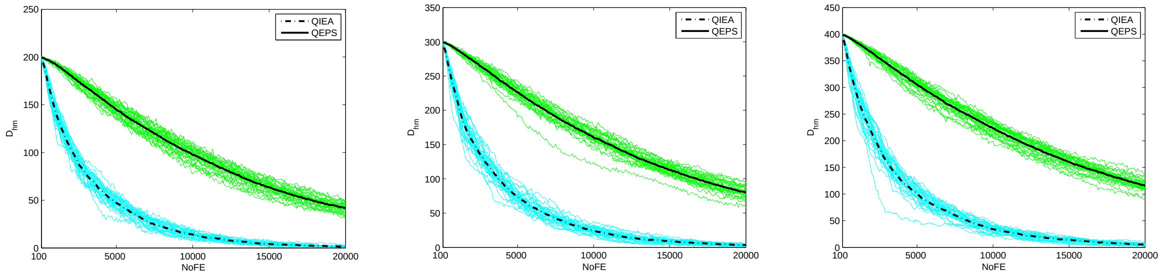


(a) 400 items

(b) 600 items

(c) 800 items

Figure 3: D_{hbw} of QEPS and QIEA with items 400, 600 and 800.



(a) 400 items

(b) 600 items

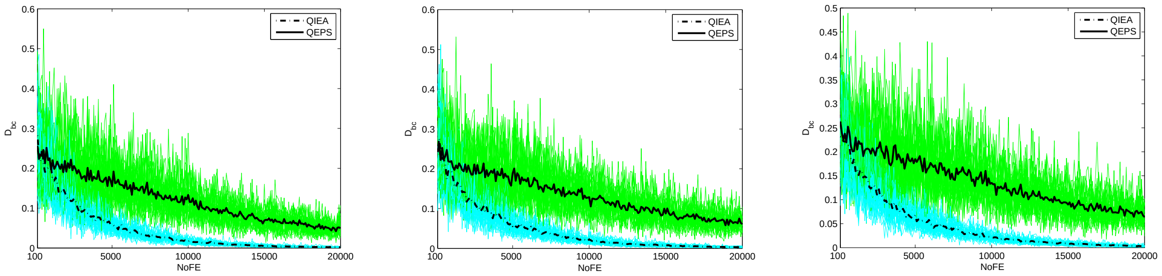
(c) 800 items

Figure 4: D_{hm} of QEPS and QIEA with items 400, 600 and 800.

The loss of population diversity means that the algorithm will fail to further explore the solution space. In the following description, we will go further to analyze the convergence performance of membrane algorithms.

3.2 Convergence Analysis

Convergence is very important for a meta-heuristic search method as it shows the speed of the method in finding a satisfactory solution to an optimization problem. In this subsection, the convergence behavior of MIEAs is observed by presenting four measures: best Q-bit individual convergence (C_{qb}), average Q-bit individual convergence (C_{qa}), the best fitness convergence (C_{fb})

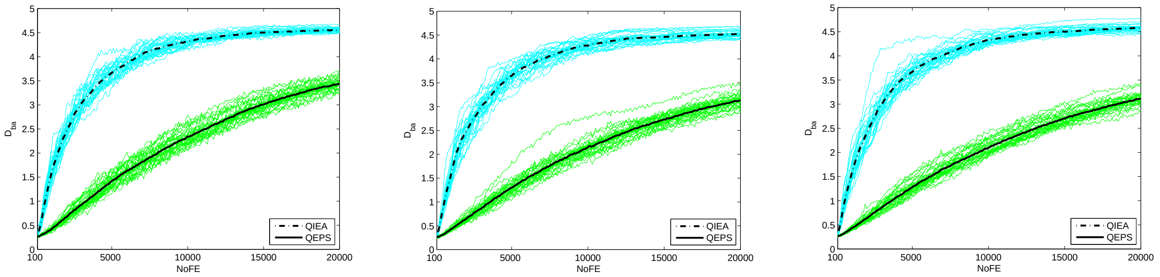


(a) 400 items

(b) 600 items

(c) 800 items

Figure 5: D_{bc} of QEPS and QIEA with items 400, 600 and 800.



(a) 400 items

(b) 600 items

(c) 800 items

Figure 6: D_{ba} of QEPS and QIEA with items 400, 600 and 800.

and the average fitness convergence (C_{fa}).

Both QEPS and QIEA use Q-bit individuals to construct a population. Thus we can apply the best Q-bit individual convergence and the average Q-bit individual convergence in a population to observe how much Q-bits approach 0 or 1 in the searching process. Their definitions are as follows.

(i) The best Q-bit individual convergence is described as

$$C_{qb} = \frac{1}{m}\sum_{j=1}^{m} \max\left\{|\alpha_{bj}|^2, |\beta_{bj}|^2\right\} \tag{9}$$

where $[\alpha_{bj}\ \beta_{bj}]^T$ is the j-th Q-bit in the best Q-bit individual corresponding to the best fitness in a population; m is the number of Q-bits in a Q-bit individual. $0.5 \leq C_{qb} \leq 1$.

(ii) The average Q-bit individual convergence is depicted as

$$C_{qa} = \frac{1}{n}\sum_{i=1}^{n} \left\{ \frac{1}{m}\sum_{j=1}^{m} \max\left\{|\alpha_{ij}|^2, |\beta_{ij}|^2\right\} \right\} \tag{10}$$

where $[\alpha_{ij}\ \beta_{ij}]^T$ is the j-th Q-bit in the i-th Q-bit individual in the population with n individuals; m is the number of Q-bits in a Q-bit individual. $0.5 \leq C_{qa} \leq 1$.

C_{qb} and C_{qa}, calculated in the Q-bit space, can be regarded as genotypic convergence measures. C_{qb} and C_{qa} have not a direct relationship to the quality of solutions. Therefore we employ the other two convergence measures, the best fitness and the average fitness, to observe the convergence rates of solutions.

(iii) The description of C_{fb} and C_{fa} is given as follows

$$C_{fb} = \max_{i=1}^{n} f_i(x) \tag{11}$$

$$C_{fa} = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{12}$$

where $f_i(x)$ is the fitness of the i-th individual. Equation 11 is listed based on a maximum optimization. For a minimum problem, C_{fb} is to find the minimal fitness among n solutions.

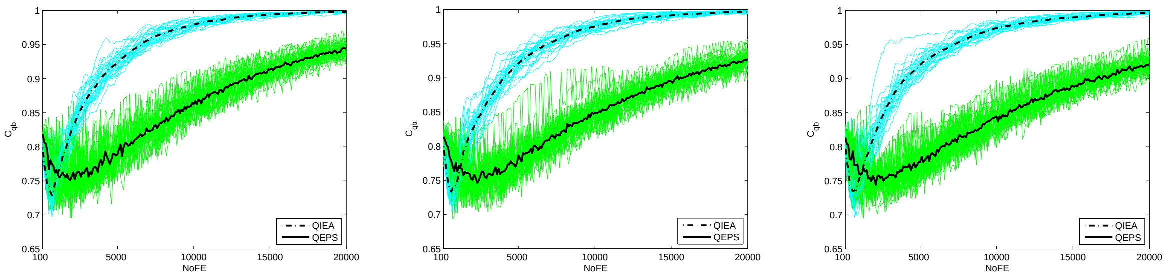
We still apply the three knapsack problems with 400, 600 and 800 items to observe the convergence performances of QEPS and QIEA. The population size, NoFE and independent runs are assigned as 20, 20000 and 30, respectively. The changes of C_{qb}, C_{qa}, C_{fb} and C_{fa} are shown in Figs.7–10, where each subfigure provides the results of 30 independently random runs (green solid lines for QEPS and cyan solid lines for QIEA) and the mean values over 30 runs (black bold solid lines for QEPS and black bold dash-dot lines for QIEA). The comparisons between QEPS and QIEA, shown in Figs.7–10, give us the following hints.

(i) The three subfigures corresponding to the respective knapsack problems with 400, 600 and 800 items, in each of Figs.7–10, show respectively similar changes for QEPS and QIEA.

(ii) It can be seen from the results shown in Figs.7–8 that C_{qb} and C_{qa} have similar tendencies, to be specific, QIEA converges much faster in Q-bit space than QEPS and quickly arrives at the maximal value 1, which implies that no further improvement of solutions in QIEA can be gained at the second half of evolutionary processes. The drastic convergence easily makes QIEA trapped in local extrema and consequently a premature end of the evolutionary process appears. On the contrary, C_{qb} and C_{qa} of QEPS go up much slower than those corresponding to QIEA with respect to NoFE and finally mount up to around 0.9 for values of NoFE in the region of 20000, which suggests that the solutions can be further improved if more NoFE is provided.

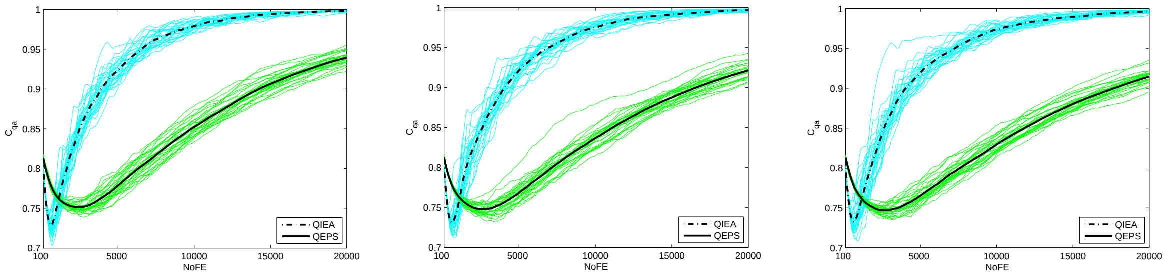
(iii) In Figs.9–10, QIEA has faster increases of C_{fb} and C_{fa} than QEPS and then stays at a relatively flat level after a certain NoFE, while QEPS goes through a slower start than QIEA and then rapidly goes beyond QIEA and keeps an ascending trend. Thus QEPS obtains better solutions than QIEA. The observations in Figs.9–10 can also be derived from the results in Figs.7–8. Additionally, it is worth noting, according to the six subfigures of Figs.9–10, that QEPS has better performance than QIEA in terms of the consistency of the results obtained for 30 independent runs when mean values are considered. This suggests that QEPS has better robustness properties than QIEA.

The convergence and population diversity are often conflicting features for population-based search methods. Rapid convergence usually results in a fast loss of population diversity, whereas better varieties of individuals produce more possibilities to improve solutions. The dynamic behaviors can be observed from the changes of population diversity, shown in Figs.1–6, and convergence performance in Figs.7–10. The diversity and convergence analysis above indicate that QEPS can achieve a better trade-off between convergence and diversity than QIEA, i.e., better balance between exploration and exploitation than QIEA. This better balance of these two essential features of any evolutionary approach is the principal explanation of the fact that QEPS achieves high quality solutions, better than QIEA if NoFE is large enough. For example, NoFE is greater than 10000 for the knapsack problems with 400, 600 and 800 items, which corresponds to 100 evolutionary generations.



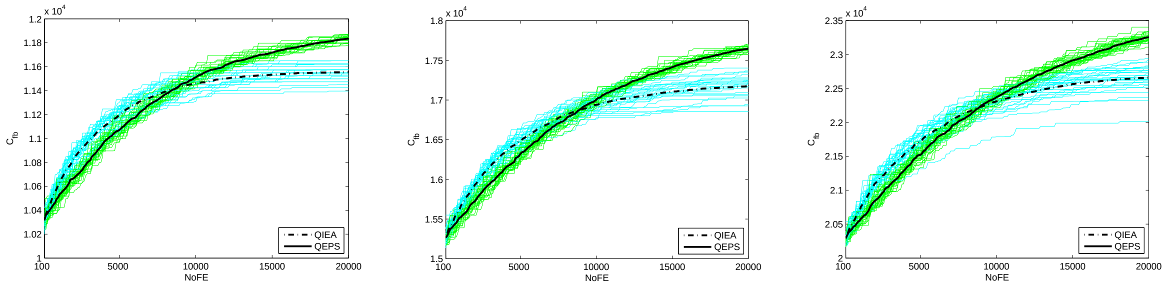
(a) 400 items (b) 600 items (c) 800 items

Figure 7: C_{qb} of QEPS and QIEA with items 400, 600 and 800.



(a) 400 items (b) 600 items (c) 800 items

Figure 8: C_{qa} of QEPS and QIEA with items 400, 600 and 800.

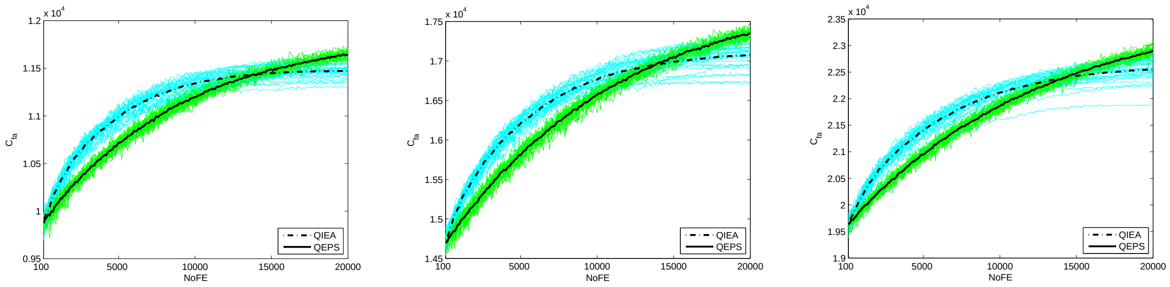


(a) 400 items (b) 600 items (c) 800 items

Figure 9: C_{fb} of QEPS and QIEA with items 400, 600 and 800.

3.3 Correlation between Measures and Fitness

The goal of MIEAs is to find the optimal solution of an optimization problem, so the relationship between the diversity and convergence measures and fitness is very important for improving



(a) 400 items

(b) 600 items

(c) 800 items

Figure 10: C_{fa} of QEPS and QIEA with items 400, 600 and 800.

the algorithm performance with respect to diversity and convergence. In this subsection, we introduce a tendency correlation coefficient to evaluate the importance of the eight measures: D_{qbw}, D_{qa}, D_{hbw}, D_{hm}, D_{bc}, D_{ba}, C_{qb} and C_{qa}.

The tendency correlation coefficient of two sequences, $S_1 = (s_1^1, s_2^1, \cdots, s_L^1)$ and $S_2 = (s_1^2, s_2^2, \cdots, s_L^2)$, is defined as

$$\rho = \frac{\sum_i (\Delta S_1 \cdot \Delta S_2)}{\sqrt{\sum_i \Delta S_1{}^2 \cdot \sum_i \Delta S_2{}^2}} \tag{13}$$

where $\Delta S_1 = (\Delta s_1^1, \Delta s_2^1, \cdots, \Delta s_{L-1}^1)$ and $\Delta S_2 = (\Delta s_1^2, \Delta s_2^2, \cdots, \Delta s_{L-1}^2)$, where Δs_k^1 and Δs_k^2 $(k = 1, 2, \cdots, L-1)$ are respectively

$$\Delta s_k^1 = s_{k+1}^1 - s_k^1 \tag{14}$$

$$\Delta s_k^2 = s_{k+1}^2 - s_k^2 \tag{15}$$

The tendency correlation coefficient ρ varies in the range between -1 and 1. The maximal value 1 and minimal value -1 mean that the two sequences S_1 and S_2 have identical and completely opposite tendencies, respectively. Thus a larger absolute value of ρ indicates a stronger tendency correlation.

We use the tendency correlation coefficient to analyze the relationship between each of the eight measures, D_{qbw}, D_{qa}, D_{hbw}, D_{hm}, D_{bc}, D_{ba}, C_{qb} and C_{qa}, and each of two kinds of fitness, C_{fb} and C_{fa}. Experimental results are listed in Table 1, where each datum is calculated by using the statistical results of three knapsack problems with 400, 600 and 800 items in Sections 3.1–3.2.

It can be seen from experimental results in Table 1 that the three diversity measures, D_{qa}, D_{hm} and D_{ba}, have stronger tendency correlation with fitness than the other five measures. So the improvement of D_{qa}, D_{hm} and D_{ba} could be a promising way to enhance the algorithm performance. In the column of D_{ba}, positive tendency correlation coefficients show that D_{ba} and fitness have similar trends, which implies that a larger value of D_{ba} may correspond to a better fitness. While in the columns of D_{qa} and D_{hm}, negative values demonstrate that the two population diversity measures have contrary trends with fitness, which might mean that smaller values of D_{qa} and D_{hm} could result in better solutions.

Table 1: Correlation between fitness and diversity, convergence measures.

Measures			D_{qbw}	D_{qa}	D_{hbw}	D_{hm}	D_{bc}	D_{ba}	C_{qb}	C_{qa}
QIEA	400 items	C_{fb}	-0.7335	-0.7804	-0.7706	-0.9251	-0.4812	0.9180	0.1627	0.1707
		C_{fa}	-0.6894	-0.7174ĄĄ	-0.7004	-0.8643	-0.6312	0.8500	0.1579	0.1606
	600 items	C_{fb}	-0.6878	-0.6976	-0.7737	-0.9064	-0.3604	0.8921	0.2663	0.2412
		C_{fa}	-0.6548	-0.6818	-0.7792	-0.8940	-0.4993	0.8764	0.2591	0.2493
	800 items	C_{fb}	-0.6812	-0.6664	-0.8153	-0.9086	-0.4291	0.8969	0.2914	0.2865
		C_{fa}	-0.6491	-0.6794	-0.8174	-0.9171	-0.4466	0.9035	0.2436	0.2749
QEPS	400 items	C_{fb}	-0.3260	-0.8679	-0.3008	-0.7910	-0.1042	0.7715	-0.0007	0.0061
		C_{fa}	-0.2004	-0.6672	-0.2155	-0.6895	-0.4590	0.6175	0.0220	0.0246
	600 items	C_{fb}	-0.3004	-0.8260	-0.3192	-0.8121	-0.0638	0.7960	0.0522	0.0427
		C_{fa}	-0.2300	-0.7009	-0.2148	-0.7635	-0.3809	0.7067	0.0231	0.0691
	800 items	C_{fb}	-0.2531	-0.8121	-0.2878	-0.8472	-0.0524	0.8356	0.1060	0.0573
		C_{fa}	-0.2057	-0.7123	-0.2370	-0.7767	-0.3477	0.7336	0.0575	0.0766

4 Examples

In this section, a knapsack problem is described and more experiments are conduced to further verify the observations in the preceding section.

4.1 Knapsack Problem

A knapsack problem can be described as the process of selecting from among various items those that are most profitable, given that the knapsack has limited capacity [12, 20, 21]. The knapsack problem is to select a subset from the given number of items so as to maximize the profit $f(x)$:

$$f(x) = \sum_{i=1}^{k} p_i x_i \tag{16}$$

Subject to

$$\sum_{i=1}^{k} w_i x_i \leq C_k \tag{17}$$

where k is the number of items; p_i is the profit of the i-th item; w_i is the weight of the i-th item; C_k is the capacity of the given knapsack; and x_i is 0 or 1. This paper uses strongly correlated sets of unsorted data: w_i =uniformly random [1, 50], $p_i = w_i + 25$. The average knapsack capacity C_k is applied.

$$C_k = \frac{1}{2} \sum_{i=1}^{k} w_i \tag{18}$$

4.2 Experiments and Results

In this subsection, we carry out the experiments on 15 knapsack problems with 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2200, 2400, 2600, 2800 and 3000 items to compare the performance of QEPS and QIEA. In the experiments, QEPS and QIEA use 20 individuals and 30 independent runs are performed for each case. QEPS uses the OLMS, where 15 elementary membranes and the maximal number 9 of iterations for each elementary membrane are considered, according to the investigation in [12]. The stopping condition for QEPS and QIEA is set as follows: 20000 NoFE for the first four knapsack problems; 30000 NoFE for the three knapsack problems with 1000, 1200 and 1400 items; 40000 NoFE for the four knapsack problems with 1600, 1800, 2000 and 2200; 60000 NoFE for the last four knapsack problems. The best, worst and average solutions over 30 independent runs are recorded and listed in Table 2. We also provide the results, shown in Table 3, for each problem when NoFE is one-tenth of the prescribed value.

Several facts can be obtained from the results in Tables 2 and 3.

(i) The results in Table 3 show that QIEA achieves better results than QEPS when NoFE is only one-tenth of the prescribed value for each of the 15 knapsack problems.

(ii) The statistical data in Table 2 show that QEPS is superior to QIEA in terms of the quality of solutions. Even the worst solution of each problem obtained by QEPS is better than the best one provided by QIEA.

(iii) It seems that there is a conflict between the two conclusions in (i) and (ii). Actually these conclusions can be explained by using the analysis of population diversity and convergence in Section 3. The advantage of QIEA over QEPS in Table 3 comes from the fact that QIEA has a faster convergence speed and shows more rapid changes of population diversity than QEPS. Slower changes can also be regarded as a better balance, between diversity and convergence, leading to improved searching capabilities for QEPS, which results in the superiority of QEPS over QIEA, as shown in Table 2. Moreover, these experimental results provide sufficient details explaining the behavior of QEPS and QIEA, as stated in Section 3.

Table 2: Experimental results for QEPS and QIEA, where NoFE for each problem is prescribed.

Items	QIEA			QEPS		
	Best	Average	Worst	Best	Average	Worst
200	5885	5786	5359	5959	5945	5909
400	11650	11553	11396	11873	11837	11778
600	17403	17173	16851	17702	17647	17575
800	22940	22659	22010	23403	23257	23109
1000	28673	28333	27954	29531	29373	29198
1200	34399	33984	33424	35441	35292	35061
1400	39560	39149	38488	40886	40722	40364
1600	45277	44864	44423	47242	47018	46672
1800	50784	50163	49506	52772	52600	52395
2000	56453	55879	55129	58775	58543	58065
2200	61645	61175	59820	64513	64230	63680
2400	66683	65984	64981	70402	70015	69726
2600	72546	71992	71497	76621	76245	75296
2800	77511	76734	75924	81918	81486	80683
3000	83294	82608	82020	88207	87657	87044

5 Conclusion

The dynamic behavior analysis of MIEAs is very illustrative for a better understanding of the role played by P systems in the context of evolutionary algorithms. This paper discusses

Table 3: Experimental results for QEPS and QIEA, where NoFE for each problem is one-tenth of the prescribed value.

Items	QIEA			QEPS		
	Best	Average	Worst	Best	Average	Worst
200	5574	5476	5359	5496	5395	5306
400	10964	10836	10687	10744	10666	10548
600	16004	15928	15775	15914	15721	15549
800	21234	21096	20860	21230	20839	20686
1000	26755	26571	26309	26501	26274	26124
1200	32148	31925	31707	31849	31635	31460
1400	36990	36832	36553	36754	36482	36179
1600	42918	42588	42352	42634	42270	42038
1800	47878	47559	47353	47428	47121	46788
2000	53376	53068	52803	53001	52654	52433
2200	58646	58335	58102	58271	57957	57622
2400	63441	63195	62819	63226	62806	62328
2600	69342	69077	68676	69196	68755	68420
2800	74051	73613	73269	73524	73207	72833
3000	79630	79236	78743	79361	78886	78483

for the first time significant aspects of developmental processes occurring in MIEAs using a comparative approach in an experimental context, whereby a set of population diversity and convergence measures were introduced and analyzed. This work not only provides some answers to the difficult and intriguing question of what are the benefits of using P systems in evolutionary algorithms, but also suggests possible improvements for MIEAs. On the basis of this work, a promising research might start with a multi-objective optimization framework utilizing concepts and principles of P systems in a systematic and consistent way.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61170016, 61373047), the Program for New Century Excellent Talents in University (NCET-11-0715) and SWJTU supported project (SWJTU12CX008)

Bibliography

[1] Păun, Gh. (1998); Computing with membranes, *J. Comput. Syst. Sci.*, ISSN: 0022-0000, 61: 108-143.

[2] Păun, Gh.; Rozenberg, G.; Salomaa, A. (2010); The Oxford Handbook of Membrane Computing, *Oxford University Press.*

[3] Pan, L.; Păun, Gh.; (2009); Spiking neural P systems with anti-spikes, *Int J Comput Commun*, ISSN: 1841-9836, 4: 273-282.

[4] Zhang, X.; Wang, J.; Pan, L. ; (2009); A note on the generative power of axon P systems, *Int. J. Comput. Commun.* ISSN: 1841-9836, 4: 92-98.

[5] Lu, C.; Zhang, X.; (2010); Solving vertex cover problem by means of tissue P systems with cell separation, *Int J Comput Commun*, ISSN: 1841-9836, 5: 540-550.

[6] Zhang, X.; Luo, B.; Pan, L.; (2012); Small universal tissue P systems with symport/antiport rules, *Int J Comput Commun*, ISSN: 1841-9836, 7: 173-183

[7] Zhang, G.X.; Liu, C.X.; Rong, H.N. (2010)Łť Analyzing radar emitter signals with membrane algorithms, *Math. Comput. Model*, ISSN: 0895-7177, 52: 1997-2010.

[8] Zhang, G.X.; Cheng, J.X.; Gheorghe, M.; Meng, Q. (2013); A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems, *Appl. Soft Comput*, ISSN:1568-4946, 13:1528-1542.

[9] Păun, Gh.; Pérez-Jiménez, M.J. (2006); Membrane computing: brief introduction, recent results and applications, *Biosystems*, ISSN: 0303-2647, 85: 11-22.

[10] Nishida, T.Y. (2004); An application of P-system: A new algorithm for NP-complete optimization problems, *Proc. of WMSCI*, 109-112.

[11] Huang, L.; He, X.; Wang, N.; Xie, Y. (2007); P systems based multi-objective optimization algorithm, *Prog. Nat. Sci.*, 17: 458-465.

[12] Zhang, G.X.; Gheorghe, M.; Wu, C.Z. (2008); A quantum-inspired evolutionary algorithm based on P systems for knapsack problem, *Fund. Inform.*, ISSN: 0169-2968, 87: 93-116.

[13] Cheng, J.X.; Zhang, G.X.; Gheorghe, M.; Zeng, X.X. (2011); A novel membrane algorithm based on differential evolution for numerical optimization,*Int. J. Unconv. Comput.*, ISSN: 1548-7199, 7: 159-183.

[14] Zhang, G.X.; Gheorghe, M.; Li, Y.Q. (2012); A membrane algorithm with quantum-inspired subalgorithms and its application to image processing, *Nat. Comput.*, ISSN: 1567-781, 11: 701-717.

[15] Zhang, G.X.; Zhou, F.; Huang, X.L.; Cheng, J.X.; Gheorghe, K.; Ipate, F.; Lefticaru, R. A novel membrane algorithm based on particle swarm optimization for solving broadcasting problems, *J. Univers. Comput. Sci.*, ISSN: 0948-695X, 18: 1821-1841.

[16] Xiao, J.H.; Zhang, X.Y.; Xu, (2012); J. A membrane evolutionary algorithm for DNA sequences design in DNA computing. *Chinese Sci. Bull.*, ISSN: 1001-653, 57:698-706.

[17] Xiao, J.H.; Jiang, Y.; He, J.J.; Cheng, Z. (2013); A dynamic membrane evolutionary algorithm for solving DNA sequences design with minimum free energy, *MATCH Commun. Math. Comput. Chem.*, ISSN: 0340-6253, 70: 971-986.

[18] He, J.J.; Xiao, J.H.; Shi, X.L.; Song, T. (2013); A membrane-inspired algorithm with a memory mechanism for knapsack problems, *J. Zhejiang U.-SCI. C*, ISSN: 1869-1951, 14: 612-622.

[19] Xiao, J.H.; Huang, Y.F.; Cheng, Z; He, J.J.; Niu, Y.Y. (2014); A hybrid membrane evolutionary algorithm for solving constrained optimization problems. *Optik*, ISSN: 0030-4026, 125: 897-902.

[20] Han, K.H; Kim, J.H. (2002); Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE T. Evolut. Comput.*, ISSN: 1089-778X, 6: 580-593.

[21] Zhang, G.X. (2011); Quantum-inspired evolutionary algorithms: a survey and empirical study, *J. Heuristics*, ISSN: 1381-1231, 17: 303-351.

[22] Burke, E.K.; Gustafson, S.; Kendall, G. (2004); Diversity in genetic programming: an analysis of measures and correlation with fitness, *IEEE T. Evolut. Comput.*, ISSN 1089-778X, ISSN 1089-778X, 8: 47-62.

[23] Herrera, F.; Lozano, M. (1996); Adaptation of genetic algorithm parameters based on fuzzy logic controllers. *Genetic Algorithms and Soft Computing*, 95-125.

Author index