DMQEA-FCM: an Approach for Preference-based Decision Support

Seung-Hwan Baek
Robotics Program
KAIST
Daejeon 305-701, Korea
Email: shbaek@rit.kaist.ac.kr

Si-Jeong Ryu
Department of Electrical Engineering
KAIST
Daejeon 305-701, Korea
Email: sjryu@rit.kaist.ac.kr

Jong-Hwan Kim, IEEE, *Fellow*
Department of Electrical Engineering
KAIST
Daejeon 305-701, Korea
Email: johkim@rit.kaist.ac.kr

***Abstract*—This paper proposes a novel algorithm, named dual multiobjective quantum-inspired evolutionary (DMQEA) algorithm augmented fuzzy cognitive map (FCM). DMQEA was developed to help users select preferable solutions out of the non-dominated ones and has been proven to be an effective way compared to other multi-objective optimization methods, such as MQEA, MQEA-PS, etc. DMQEA, in this paper, has been coupled with decision supporting tool, fuzzy cognitive maps (FCMs) to support selecting best models which can reflect users' preferences. Even though the attempts with single optimization such as genetic algorithms (GAs) or particle swarm optimization (PSO) have been frequently carried out, there have been only few attempt to incorporate FCM with multicriteria decision making algorithm, especially to reflect user's preference. This study aims to integrate DMQEA with FCM to build models according to user's preference. In robotics field, the interaction with human operators is an important issue and DMQEA-FCM can aid robots in their decision making process in the context of the interaction.**

I. Introduction

This paper attempts to optimize FCM's weight matrix with respect to multiple objective optimization based on user's preference. FCM has been used as a tool for decision support to deal with given scenarios. Basically, FCM is a weighted graph consisting of nodes which are connected by signed and directed edges (Fig. 1). FCM can represent systems or scenarios dynamically changing with time. Because of its simplicity in presenting high level images of dynamic relationships between components which have main effect on corresponding situation, FCM has been widely used in politics or economics to predict and make decision under critical situation. As acknowledged its usefulness in decision making, however, people in the field of engineering started to adopt FCM in their study, as in medicine, robotics, information technology, etc. To utilize FCM as an engineering tool, it was required to generate more accurate description of a complex situation and these requirements on FCM led to deriving various types of variants which have evolved to fit into given work frames.

Even though FCM can be considered as one of neural networks, there exists a factor which distinguishes those two models: cause-effect relationship of complex system

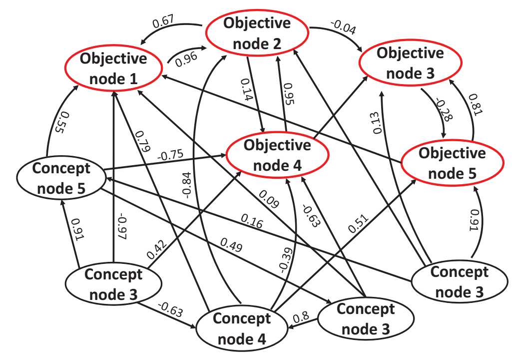


Figure 1: Structure of FCM which is used in this paper. Concept values are in oval shapes and weight values are on connecting edges. Objective nodes contains information which is used as evaluation for corresponding objectives of given problem.

[1]. FCM is specified to represent reasoning of dynamic system's operation, and especially, in robotics, its applications covers navigation, learning, and prediction [2]. For the navigation using FCM, Vaščák, implemented FCM to make it automatically adapt to environment by applying Hebbian learning [3][4]. Also for learning and prediction, Stach, W. *et al.* combined FCMs with real-coded genetic algorithm (RCGA) to model and predict time-series data at linguistic and numerical level[5]. In the algorithm, they transformed given time-series data into amplitude(position) and change(speed) information to assign them as the concept nodes of FCMs to search for the best candidate FCM model by RCGA.

Optimization techniques have been applied on FCM, such as GA or particle swarm optimization (PSO). Song *et al.* proposed an approach for FCM learning based on multi-objective particle swarm optimization [6] in addition to Parsopoulos *et al.*'s single-objective optimization using particle swarm optimization apporach [7]. There are other

978-1-5090-0626-7/16/$31.00 ©2016 IEEE

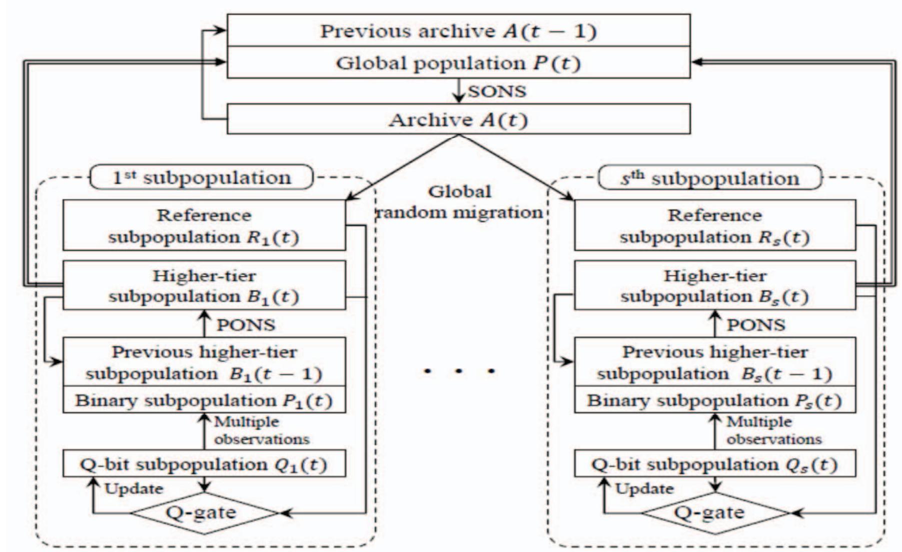


Figure 2: Structure of DMQEA.

approaches for learning by implementing evolutionary strategies (ES) as well. In robotics, researchers are paying more intensive attentions onto human robot interaction (HRI). So far, there have been many efforts to select candidate FCMs based on learning algorithms, as indicated above. However, to implement and develop in the future, it should be considered in the point of view of HRI, as in artificial emotion forecasting algorithm using FCM [8]. In HRI, preference of users has been known to be one of the necessary attributes to be considered.

Dual multi-objectives quantum-inspired evolutionary algorithm (DMQEA) (Fig. 2) has been proposed and proven to be efficient by investigating two metrics: hypervolume and diversity [9]. Application of DMQEA to arrange sensors is introduced in [10]. In this paper, we combined FCM with DMQEA to implement preference of users in selection of best fitting model for given situation. Simulation has been performed by MATLAB. Reference FCM has been selected randomly and candidate FCMs are chosen as the ones that best reflect user's preference.

This paper is organized as follows. Section II presents background knowledge of FCM and QEA, followed by DMQEA. Section III describes DMQEA-FCM algorithm and Section IV for simulation by Matlab, result and discussion. Finally, the conclusion remarks are followed in Section V.

II. DMQEA-FCM

A. FCM

FCM is a graphical diagram which is suitable to represent dynamic aspect of systems. It is comprised of concept nodes and edges, which, respectively, represent system's important attributes and weights connecting those attributes nodes. Therefore, the structure of FCM itself is considered simple, relative to other ways of system modeling techniques. The point here is how to describe dynamics of system in the way the attributes affect each other, and accordingly the two components, i.e. the concept nodes and edges for level of affection, of FCM should be correctly chosen and organized to best represent the system. Main characteristic of edges of FCM is that it is uni-directional to link any sets of two nodes, as the bridges connecting cities do. The traffic become different at each time in both directions, and the level of influence hit on each city by the traffic gradually changes. As in this analogy, FCM considers the concept nodes' influences on each other as different individuals.

Even though the approaches to arranging and assigning values to the concept nodes and edges entirely depends on problem definition, in most paper, it follows the basic way of updating FCM nodes, which is weighted sum method (Eq. 1).

$$c_i^{t+1} = f\left(c_i^t + \sum_{j=1, j\neq i}^{N} e_{ij} * c_j^t\right) \tag{1a}$$

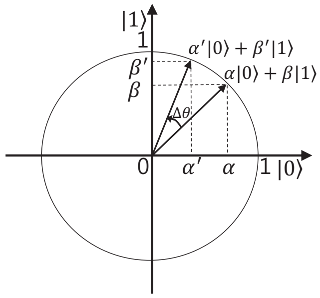


Figure 3: A Q-bit can be described as a point on the circle. $|\alpha|^2$ and $|\beta|^2$ respectively represent probabilities of a Q-bit found in the "0" state and "1".

$$f(x) = \left(1 + e^{-\lambda x}\right)^{-1} \tag{1b}$$

Eq. 1a describes the process of updating concept nodes. The values of nodes are multiplied by weights and summed together. In this paper, the values of nodes are limited to $[0, 1]$ by applying logistic function symmetric to point 0.5 as in Eq. 1b. e_{ij} in Eq. 1a is a weight value connecting node j to node i and this value covers the range of $[-1, 1]$; if node j negatively influence node i it means the e_{ij} is negative and *vise versa*.

Since this paper aims to demonstrate that FCM can be combined with preference-based multi-objective selection algorithm, DMQEA, we followed as well the weighted sum method for nodes update (Eq. 1). If the problem requires to adopt some other advanced versions of FCM, it can be implemented into our proposed preference-based multi-criteria decision making algorithm, once the evaluation concept nodes are clearly defined. Hence, it is assumed that there exists a system which can be well-represented by this basic concept nodes update rule and guaranteed for convergence.

B. QEA

QEA, one of the evolutionary algorithms, is the base of DMQEA by providing fundamental search mechanism. QEA is a single objective optimization algorithm, whereas MQEA a multiple objectives optimization algorithm and DMQEA a multiple objective optimization with consideration on user's preference. It is a stochastic evolutionary optimization algorithm based on concepts and principles of quantum computing, such as a quantum bit (Q-bit) and superposition of states [11]. QEA has an advantage of balancing between exploration and exploitation through its inherent probabilistic mechanism.

1) Representation: QEA uses a probabilistic representation based on the concept of Q-bits and superposition of states. A Q-bit is the smallest unit of information and the state of a Q-bit is represented as

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2}$$

where α and β are complex numbers and $|\alpha|^2 + |\beta|^2 = 1$. $|\alpha|^2$ and $|\beta|^2$ respectively represent probabilities of a Q-bit found in the "0" state and "1" state, which can be illustrated as a unit vector on the two dimensional space, as shown in Fig. 3. A Q-bit individual is a string of Q-bits and is defined as follows:

$$\mathbf{q} = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array}\right] \tag{3}$$

where m is the number of Q-bits. Q-bit individuals at generation t with the size of population n is represented as $Q(t) = \{\mathbf{q}_1^t, \mathbf{q}_2^t, ..., \mathbf{q}_n^t\}$. Algorithm 1 shows a standard procedure of QEA.

Algorithm 1 Procedure of QEA

```
begin
t ← 0
initialize Q(t)
make P(t) by observing the states of Q(t)
evaluate P(t)
store the best solutions among P(t) into B(t)
while not (termination condition) do
    t ← t + 1
    make P(t) by observing the states of Q(t − 1)
    evaluate P(t)
    update Q(t) using Q-gates
    store the best solutions
    among B(t − 1) and P(t) into B(t)
    store the best solution b among B(t)
    if (global migration condition) then
        migrate b to B(t) globally
    else if (local migration condition) then
        migrate b_j^t in B(t) to B(t) locally
    end if
end while
end
```

C. DMQEA

Dual multiobjective quantum-inspired evolutionary algorithm (DMQEA) is a search algorithm for solutions which can best reflect user's preference. It sorts and filters solutions through two stages sorting mechanisms: primary objectives non-dominated sorting (PONS) and secondary objectives non-dominated sorting (SONS). PONS is a sorting mechanism which is first performed based on problem's objectives. PONS does not implement user's preference but processes non-dominated sorting on given objectives. By PONS, DMQEA can filter off dominated solutions from each subpopulation. Best solutions from previous generation of each subpopulation are evaluated along with solution candidates (populations), which have been populated anew. Filtered populations from each subpopulation are gathered together to comprise global population. Therefore, the global population is already an elite group with equal consideration level on objectives.

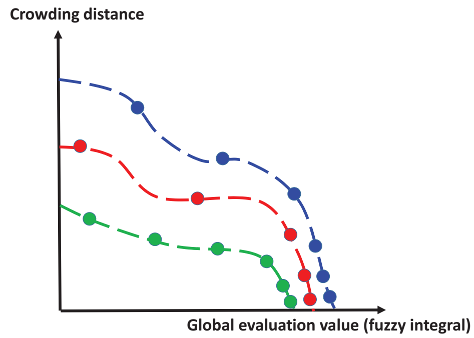


Figure 4: secondary objective non-dominated sorting (SONS). In the smaller crowding distance area, the denser collection of solutions is observed.

SONS is performed to organize primarily selected solutions by user's preference. In the sense that preference reflecting level is evaluated on selected solution, it remarkably reduces computation time. Fuzzy integral is used to globally evaluate the PONS-selected solutions. Because the PONS-selected solutions do not have dominating relationships on each other, by introducing fuzzy integral based on user-defined preference, dominating relationships was made between the PONS-selected solutions. However, if the preference-based fuzzy integral itself had been used as global evaluation, it could have brought us to local minimum values. To avoid the trivial problem, DMQEA introduces crowding distance. Crowding distance is defined as the average distance of evaluations (objectives) values between a solution's closest two neighbors. Therefore, by selecting the solutions with large crowding distance, eventually it gets to select solutions with far existing neighbors. Continuing to select solutions of comparable global evaluation values but with larger crowding distance allows efficient exploration of the solution space. In SONS stage, fuzzy integral values, which is global evaluation values, becomes one of the objectives for non-dominated sorting along with the second objective, crowding distance as in Fig. 4. The first ranked (1-Tier) solutions by SONS are randomly migrated to be reference solutions to update Q-bit individuals, which acts a role as a threshold to turn a binary bits to 0 or 1.

III. DMQEA-FCM

From the previous sections, the roles and structures of FCM and DMQEA are explained respectively. FCM models to represent dynamic systems and DMQEA explores search space for solutions which best reflect user's preference. Therefore, by combining the two techniques, users can help themselves design dynamic systems which reflect their preferences. This idea can be projected into robots operating in dynamic situation to autonomously make decision for its operator's intention. The overall algorithm is shown in Algorithm 2.

The procedure of DMQEA-FCM is as following.

Algorithm 2 DMQEA-FCM Algorithm

Some of FCM concept nodes are nominated as the nodes which corresponds to problem's objectives, respectively

```
1:  Initialize all variables
2:  Initialize Q-bit individuals Q(t) for subpopulations
3:  Initialize and run reference FCM until it converges
4:  for iteration = 0 to iteration_max do
5:      for generation = 0 to generation_max do
6:          for subPopulation = 0 to subPopNum do
7:              Observe P(t) by multiple observation on Q(t)
8:              Compose P(t) and B(t − 1)
9:              Decode the combined P(t)
10:             Evaluate the decoded P(t) (run candidate FCMs)
11:             Calculate the errors between candidate FCMs and reference FCMs
12:             Perform PONS on P(t) into B(t)
13:         end for
14:         Compose global population gPop(t) from each subPopulation
15:         Combine gPop(t) with archive(t − 1)
16:         Perform SONS to obtain archive(t)
17:         Generate refSubPop(t) for each subpopulation
18:         Update Q(t) of each subpopulation
19:     end for
20: end for
```

Reading through Algorithm 2 along with Fig. 2 helps for better comprehension. Procedure is further explained line by line.

1) Initialize required variables, such as sub-population number and size, size of global population, concept node number of FCM etc.
2) Initialize Q-bit individuals as many as the sub-population numbers.
3) Generate reference FCM and run it until it converges. Reference FCM's concept nodes values at each time step are referred as target values for candidate FCMs. That is, at each time step, best candidate FCMs survives which the most tracked reference FCM's nodes of the time step and be another candidates for the next time step coming up.
4) Iteration number is the reference FCM's time step made until its convergence. For instance, if it updated its nodes 10 times, before the reference FCM converges, the iteration number becomes 10.
5) Generation number means the times Q-bit individuals get updated. At every update of Q-bit individual

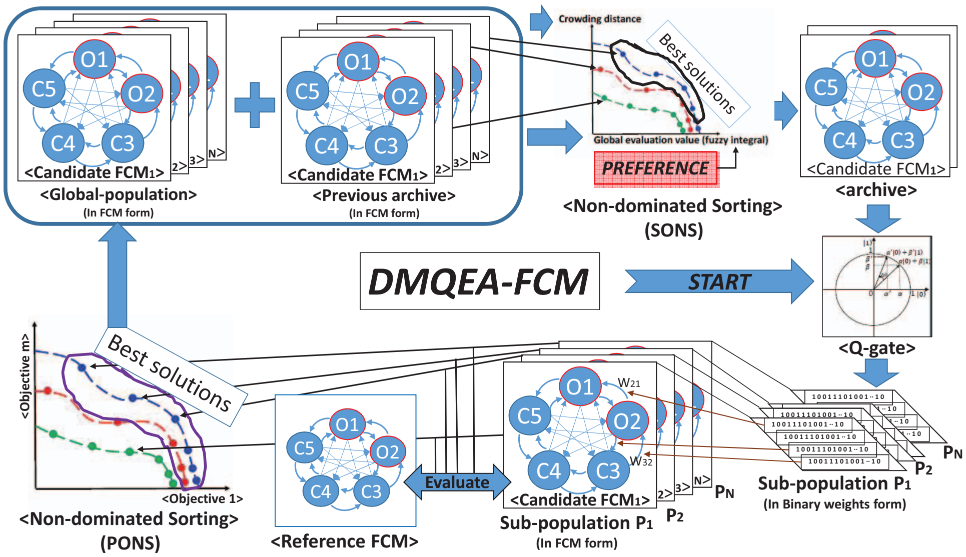


Figure 5: Diagram of DMQEA-FCM. Reference FCM generates target concept values and candidate FCMs tracks the values. Nodes with O are objective nodes, which represent objectives of given problem. Nodes with C are generic concept nodes.

updates, global population gets evolved toward target FCM.

6) The procedures coming after this line will be iterated to cover the entire sub-population.
7) Binary vector form sub-population individuals $P(t)$ are generated multiple times based on one Q-bit individuals and filtered to match numbers of individuals with the one of sub-population. Users can perform just one time observation, instead of multiple times.
8) Combine $P(t)$ and previous best individuals $B(t-1)$.
9) Transform the binary individuals $P(t)$ into real numbers for candidate FCMs by decoding. The decoded real numbers are the weights of candidate FCMS.
10) Evaluate all the objective fitness values by running candidate FCMs. Some of candidate FCMs' nodes are the objective nodes.
11) Objective fitness values previously obtained are the errors between reference FCM and candidate FCMs. The smaller the errors are, the better the fitness values are.
12) Perform non-dominated sorting (PONS) on sub-pupolation individuals (candidate FCMs) with objective fitness values. Result is stored in $B(t)$
13) End of For loop on Sub-population.
14) Combine all the sub-populations to compose global population $gPop(t)$.
15) Combine generated $gPop(t)$ with previous generation's final result $archive(t-1)$.
16) Perform SONS on the combined individuals. At this step, calculate two secondary objectives: crowding distance and Choquet fuzzy integral based on given preference (Fig. 4). In this paper, it aims to maximize both crowding distance and fuzzy integral value, leaving the rank of tiers in Fig. 4 goes higher (better solution) towards top-right corner. Half of the top ranked individuals in a row are stored in $archive(t)$.
17) Migrate randomly the individuals of $archive(t)$ to $refSubPop(t)$ for all sub-populations.
18) Update Q-bit individuals by referring $refSubPop(t)$.
19) End of For loop on generation
20) End of For loop on iteration

Combination of QEA as an single-objective optimization algorithm and FCM for decision support are proposed to train the weight matrix, which is an array of weights connecting concept nodes. Even though there have been several EA techniques, such as GA or particle swarm optimization (PSO), which are implemented into FCM to train the model to fit into the problem of interest, no attempt has been done to augment FCM with QEA in training its weights. Known to better perform over conventional GA, implementing QEA onto FCM can be a chance to develop a better modeling algorithm.

IV. Experiments

A. FCM setting

Structure of FCM is depicted in Fig. 1 and applied to reference FCM and candidate FCMs for simulation. The experiment has been demonstrated using MATLAB. As explained in previous section, the target concept values are generated by running the randomly created FCM (reference FCM) (Fig. 5) and the sizes of concept nodes and weight matrix are 5 and 5x5, respectively. For each iteration, candidate FCMs run 100 times (100 generations at one iteration), and since reference FCM converges in 10 iterations the iteration number is 10 as well. Initial weights are randomly selected by probabilistic operation of generating $P(t)$ binary weights out of Q-bit individual in DMQEA.

Some of the nodes of FCM are considered as the ones representing problem objectives. The other nodes are attributes or properties which affect systems and should be defined depending on given problem, such as sensor values in navigation problem [3] or data itself and speed of change of data in time [5]. As depicted in Fig. 6, FCM values alter at each iteration and converge into certain values, once given initial nodes values. Fig. 6 contains target node values at each iteration. The five values at each time step, which is the iteration number in Algorithm 2, are the target values to candidate FCMs.

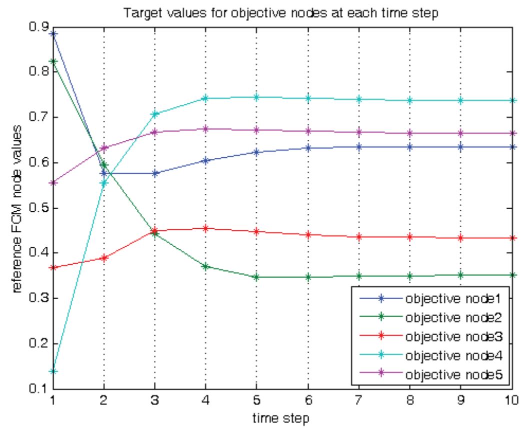


Figure 6: Change in objective nodes' values in Reference FCM as time steps. Five values at each time step are defined as target values of candidate FCMs. At each time step, which is the iteration number in Algorithm 2, the candidate FCMs with the closest to target values survives after given generation number to proceed further.

Table I: Averages of partial evaluation values on archives of cases for different preferences. The lower the preference is, the preferable it is.

Preference [obj1 obj2 obj3 obj4 obj5]	Average (100 individuals) [obj1 obj2 obj3 obj4 obj5]
[1 1 1 1 1]	[0.9751 0.9617 0.9464 0.9766 0.7800]
[1 1000 1000 1000 1000]	[0.9878 0.8780 0.8709 0.9535 0.6898]
[1000 1 1000 1000 1000]	[0.9653 0.9720 0.8803 0.9756 0.6998]
[1000 1000 1 1000 1000]	[0.9250 0.8734 0.9854 0.9487 0.6815]
[1000 1000 1000 1 1000]	[0.9669 0.9034 0.9220 0.9958 0.7185]
[1000 1000 1000 1000 1]	[0.9467 0.9468 0.9356 0.9662 0.9408]

B. DMQEA setting

DMQEA comprises of two levels of population: subpopulation and global population. Global population is composed of four subpopulations and each subpopulation includes 25 Q-bit individuals. Because each Q-bit individual corresponds to each subpopulation individual $P(t)$ (even with multi-observation, only one best $P(t)$ is selected prior to its combination with $B(t-1)$), $P(t)$ and, eventually, $B(t)$ possess 25 individuals. Global population number is 100.

C. Result

Fig. 7 shows result of running DMQEA-FCM algorithm. (7b) represents the system where the first objective has 1000 times higher preference than other objectives. In Fig. (7a), preferences on every objective are the same. In Fig. (7c), the second objective is preferable than the other objectives and in Fig. (7d), the third objective is preferable than the other objectives. In Fig. (7e), the fourth objective is preferable than the other objectives. In Fig. (7f), the fifth objective is preferable than the other objectives. Solid lines are the result of reference FCM, while dashed ones are of the candidate FCMs. These individuals are the ones which have the highest global evaluation (fuzzy integral value) value in the corresponding archives. As depicted in Fig. 7, it is clear that the preferable objective arrives the closest to target value. Also, it is clearly observed that not all the node values converge at the same rate to their targets. The node representing the fourth objective is observed to easily converge to target than the others do, while the fifth objective is observed more detached than the others are. This depends on the methods that has chosen FCM for given problem updates concept and weights values.

Because DMQEA adopts probabilistic approach in selection of its best tier solution, comparison of the best solutions selected for five different preferences does not guarantee its validation. According to it, average values of partial evaluation values of all the 100 individuals in final archives of five different preference cases are compared and tabulated in Table I.

Table I indicates it is likely that preferable objectives have higher partial evaluation values over others. As previously commented, it depends on system's dynamics structure which is preferable for certain objectives and approaches that FCM models the system. However, for certain, the most preferable objective's score is higher than its scores in the other preference cases.

D. Discussion

In considerable cases where experts are not involved, FCMs which update weights by learning mechanism are used the most frequently. However, it is impractical to implement FCM into space searching algorithms for weights, such as GAs, PSO, or QEA, etc, if the FCM is the type of updating its weights for learning. Once it starts updating its weights it is not possible to track back to find weights which best fit to system, as it alters by communicating with concept nodes as well. Parsopoulos *et. al* [12] implemented PSO into FCM as a classification approach for Autism classification and Stach *et al.* [5] proposed to couple FCM with GA search algorithm to learn patterns of data and reproduce it. In both papers, concept nodes were updated in processing, but not the weights.

In proposed algorithm, even though the weights of candidate FCMs are not altered during optimization process, by targeting at every iteration reference FCM's concept node values, it keeps driving the algorithm to select preferable solutions.

The number of generation and the rating range of user's preference should be appropriately selected according to given problem. Even though the preference strength of 1 to 1000 has been used in this paper, much lower rate, such as 1 to 50, works fine. Regarding the number of generation, even though lower generation number is more suitable to observe that preferable objective's values converge to target better than the others, convergence rate for the whole FCM model performs poorly.

V. Conclusion

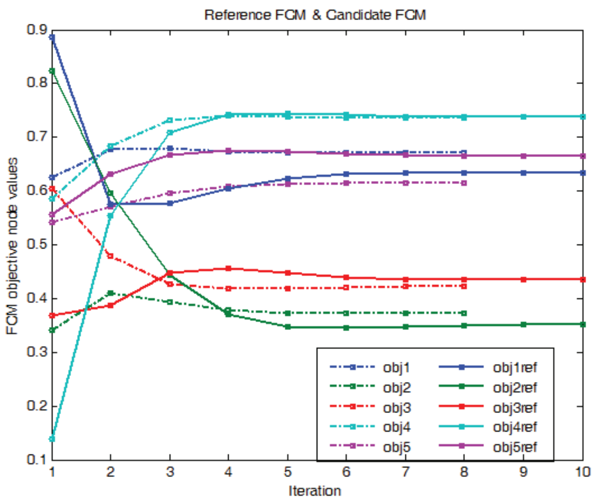
In this paper, we developed and presented decision support algorithm for multiple-objectives optimization using DMQEA-FCM. A reference FCM was randomly generated and run until it converged. Out of its concept nodes, five of them were defined as the nodes which represent objectives of given system. The temporal sequence of reference FCM's evaluation node values was the target values for candidate FCMs to follow. The proposed approach aims to select better FCM models to reflect user's preference, and the result well selected the FCM which best represent user's preference. Classic FCM approach performs well to represent control systems to decide related parameters. However, in the robotics research field, it can be utilized as a high level decision making tool in interacting with human operators or users. Robot's operational behavior can be adjusted for specific group of users or a user by introducing this DMQEA-FCM. For the further research, virtual agent application of the proposed algorithm in robot soccer simulation will be implemented. A team of robots will decide which tactic they would go for based on the context of current situation, using DMQEA-FCM.

Acknowledgment

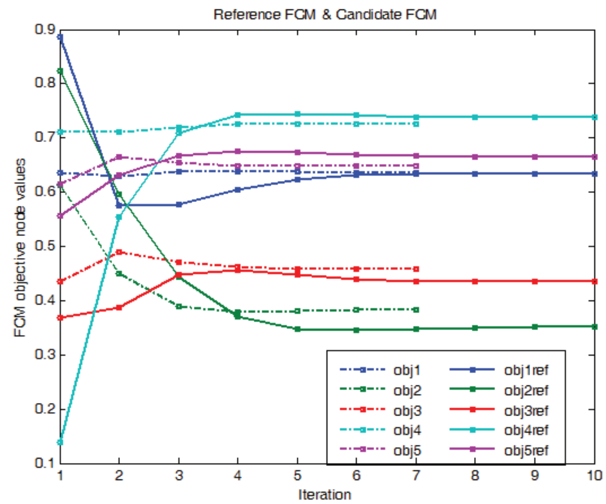
This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea MSIP (No.B0101-15-0551, Technology Development of Virtual Creatures with Digital DNA).

References

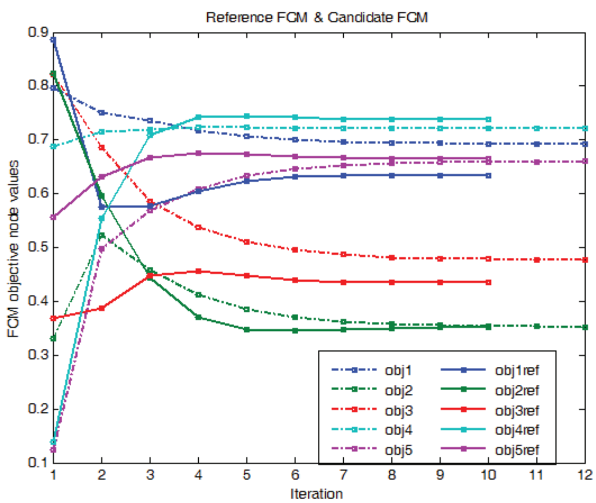
[1] B. Kosko, "Fuzzy cognitive maps," *International Journal of man-machine studies*, vol. 24, no. 1, pp. 65–75, 1986.
[2] E. Papageorgiou, J. L. Salmeron, *et al.*, "A review of fuzzy cognitive maps research during the last decade," *Fuzzy Systems, IEEE Transactions on*, vol. 21, no. 1, pp. 66–79, 2013.
[3] J. Vaščák, "Approaches in adaptation of fuzzy cognitive maps for navigation purposes," in *Proceedings of 8th IEEE International Symposium on Applied Machine Intelligence and Informatics SAMI*, 2010, pp. 31–36.
[4] E. Papageorgiou, C. D. Stylios, and P. P. Groumpos, "Active hebbian learning algorithm to train fuzzy cognitive maps," *International journal of approximate reasoning*, vol. 37, no. 3, pp. 219–249, 2004.
[5] W. Stach, L. Kurgan, W. Pedrycz, *et al.*, "Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps," *Fuzzy Systems, IEEE Transactions on*, vol. 16, no. 1, pp. 61–72, 2008.
[6] H. Song, C. Miao, and Y. Miao, "Fuzzy cognitive map learning based on multi-objective particle swarm optimization," *International Journal of Computational Cognition*, vol. 6, no. 3, pp. 51–59, 2008.
[7] K. E. Parsopoulos, E. Papageorgiou, P. P. Groumpos, M. N. Vrahatis, *et al.*, "A first study of fuzzy cognitive maps learning using particle swarm optimization," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 2. IEEE, 2003, pp. 1440–1447.
[8] J. L. Salmeron, "Fuzzy cognitive maps for artificial emotions forecasting," *Applied Soft Computing*, vol. 12, no. 12, pp. 3704–3710, 2012.
[9] S.-J. Ryu, J.-H. Kim, and K.-B. Lee, "Dmqea: Dual multiobjective quantum-inspired evolutionary algorithm." in *IJCCI (ECTA)*, A. C. Rosa, J. J. M. Guervós, and J. Filipe, Eds. SciTePress, 2014, pp. 207–214.
[10] S.-J. Ryu, R. Datta, and J.-H. Kim, "Dual multiobjective quantum-inspired evolutionary algorithm for a sensor arrangement in a 2d environment," in *Robot Intelligence Technology and Applications 3*. Springer, 2015, pp. 57–66.
[11] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 6, pp. 580–593, 2002.
[12] P. Oikonomou and E. I. Papageorgiou, "Particle swarm optimization approach for fuzzy cognitive maps applied to autism classification," in *Artificial Intelligence Applications and Innovations*. Springer, 2013, pp. 516–526.



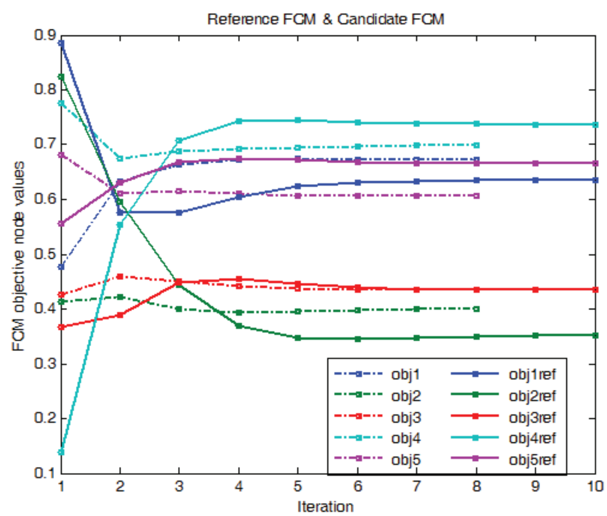
(a) Preference=[1 1 1 1 1]



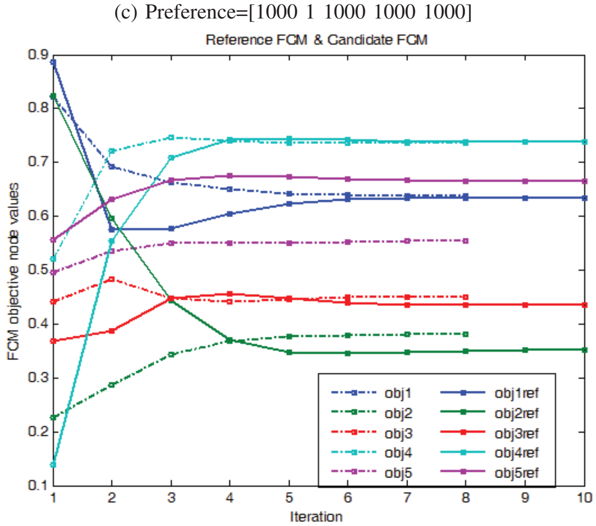
(b) Preference=[1 1000 1000 1000 1000]



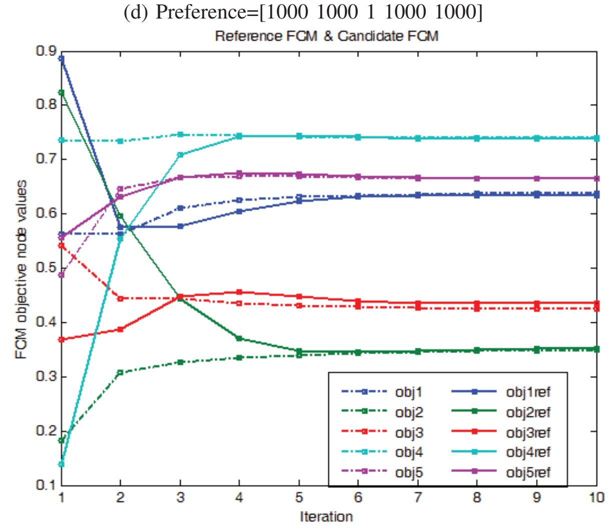
(c) Preference=[1000 1 1000 1000 1000]



(d) Preference=[1000 1000 1 1000 1000]



(e) Preference=[1000 1000 1000 1 1000]



(f) Preference=[1000 1000 1000 1000 1]

Figure 7: Result of running DMQEA-FCM with different preferences. (a) First objective preferable. (b) Second objective. (c) Third objective. (d) Fourth objective (e) Fifth objective. (f) Equally preferable. Solid lines are the target lines to be tracked. Preference is rated as the lower number is preferable.