Traveler Behavior Cognitive Reasoning Mechanism

Ahmed Tlili, Salim Chikhi and Ajith Abraham

January 5, 2024

Traveler Behavior Cognitive Reasoning Mechanism

Ahmed Tlili[1] , Salim Chikhi[2] , Ajith Abraham[3]
[1,2] Complex Systems Modeling and implementation (MISC Labs), Abdelhamid Mahri University, Constantine, Algeria
[3]Machine Intelligence Research Labs (MIR Labs). Scientific Network for Innovation and Research Excellence. Auburn, Washington 98071, USA
[1]{HYPERLINK "mailto:a.tlili@univ-emir.dz"}, [2] Sl.chiki@univ-constantine2.dz, [3]ajit.abrahm@ieee.org

Abstract. In this work, we use the kosko's fuzzy cognitive maps to represent the reasoning mechanism in complex dynamic systems. The proposed approach focuses on two points: the first one is to improve the learning process by providing a connection between Kosko's FCMs and reinforcement learning paradigm, and the second one is to diversify the states of FCM concepts by using an IF-THEN rules base based on the Mamdani-type fuzzy model. An important result is the creation of the transition maps between system states for helpful knowledge representation. After transition maps are validated, they are aggregated and merged as a unique map. This work is simulated under Matlab with Fuzzy Inference System Platform.

Keywords: Fuzzy Cognitive Maps, Reinforcement Learning, Traveling Salesman Problem

1. Introduction

The more intensively studied optimization problem is the Traveling Salesman Problem (TSP). TSP is ranged in the combinatorial NP-hard problem that requires more calculation time, because the number of possible circuits is extremely wide even for cases were number of cities is small. For this reason, the use of the heuristic technics is suitable. TSP, as a nonlinear NP-complete problem, is formulated as follows: A salesman visits n cities that he starts by chooses one amongst cities goes to each city and returns to the starting one. So he provides a complete tour that combines' all cities where TSP objective now is cost minimizing in energy or time. TSP, mathematically in the literature, is well characterized and described but cannot be solved with the exact methods therefore the heuristic methods are used [16]. In last decades, many studies have using FCMs formalism [1][2], to study dynamic systems, and have given hopeful results [3] [4] [5]. In this work we assume that the task performed by the traveler to find a best tour with a minimum of cost is in nature a cognitive task. based on this idea we present in this paper an approach based on FCM cognitive formalism with Reinforcement Learning (RL).

2. Literature review

TSP is one of the most studied problems in the optimization field. Among the methods developed by researchers we discuss two methods related to our approach, namely: Hopfield Neural Networks with Genetic Algorithm and Fuzzy Self-Organizing Maps.
Liu et al. [14] applied Hopfield Neural Network (HNN) with Genetic Algorithm (GA) in TSP reasoning mechanism so GA-HNN was established. In GA-HNN there is a connection between the property of the GA and the parallelism mechanism of HNNs. This connection seems be, in their work, between global stochastically searching ability of GA and self-learning ability of HNN. According to the authors the proposed method applied to TSP optimization has the advantages of convergence, precision and calculation stabilization.
Kajal and Chaudhuri [15] illustrated how the Fuzzy Self-Organizing Map (FSOM) can be used to improve TSP reasoning mechanism in the winner city search by integrating its neighborhood preserving property and the convex-hull property of the TSP. In order to improve learning at each stage, FSOM draws for all excited neurons to the input city and in the meantime excites them towards the convex-hull of cities cooperatively.

3. Theory background

3.1 Fuzzy cognitive maps

The term of Cognitive Map (CM) was introduced in 1948 by Tolman [9] and described the abstract mental representation of space built by rats trained to navigate in the labyrinth. The term of Fuzzy Cognitive Maps (FCM) as illustrated in Figure 1 was introduced by Kosko [2], to designate a simple extension of CMs by the connection between fuzzy logic and artificial neural networks. FCMs can describe the dynamic behavior of entities. They are directed graphs with nodes representing concepts categorized into sensory, motor and effectors. Arcs represent causal relationships between concepts. Each arc from one concept C_i to one concept C_j is associated with a weight ω_{ij} which reflecting a strength of causal relationship: inhibition if $\omega_{ij} <0$ or excitation if $\omega_{ij} > 0$. The activation degree for each concept is associated and it represents its state at time t, and over time can be modified. For more detail about FCMs refer to [6].

Kosko [2] proposed equation (1) to calculate values of each concept::

$$\mathrm{X}_{\mathrm{i}}^{k+1} = f(\sum X_j^k . \omega_{ji}) \tag{1}$$

In order to make the most of the history of the concepts, (2) was proposed:

{EMBED Equation.3} (2)

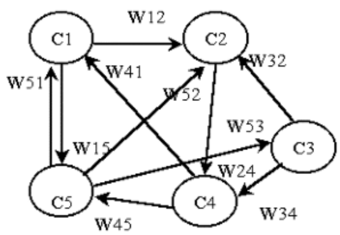

Fig 1. An FCM as a graph

3.2 Reinforcement Learning

Formal framework of reinforcement learning is defined by the Markov Decision Processes (MDP) [12] where MDP process is defined by:

- S, a finite set of states. s Є S
- A, a finite set of actions in state s. a Є A(s)
- r, a reward function. r(s, a) Є R
- P, the probability of transition from one state to another depending on the selected action. P (s '| s, a) = P_a(s, s').

The solution is to find best policy of actions that achieves the aim by maximizing rewards beginning with any initial state. In all stages, the TSP chooses an action according to these outputs. So, the environment sends either award or penalty defined by: $r_k = h\ (s_k,\ a_k,\ s_{k+1})$. In RL paradigm, there is at each stage an accumulation of costs and its allows to find total cost represented by the formula Σ h(s_k,a_k ,s_{k+1}). In [7] the expected reward is weighted by the parameter γ and come to be Σ γ h (s_i,a_i,s_{i+1}) with $0 \leq \gamma \leq 1$. The RL is to find a optimal policy π* among all possible action selection policy. The existence of optimal policy π whose is considered consequently the Bellman [10] optimality equation is satisfied:

$$V^{\pi^*} = V^*(s_i) = \max\left\{R(s_i, a) + \delta(\sum P(s_i \rightarrow s_{i+1}, a)V^*(s_{i+1})\right\} \quad \forall s \in S \qquad (3)$$

Equation (3) sets the value function of the optimal policy that RL will seek to assess:

{EMBED Equation.3} (4)

3.3 Q-Learning Algorithm

Q-learning algorithm was developed by Watkins is the is one of the most popular reinforcement learning methods based on temporal difference learning technic TD (0). Q-Learning algorithm technique is to establish a quality function represented by one value for each state-action couples and Q^{π} *(s, a)* is to reinforce estimate when choice is to starting from state s, with *a* as an action by following a policy π. In this technique [13], for any policy π and any state $s \in S$, the value of executing action *a* in state s under policy π denoted by $Q^{\pi}(s, a)$ correspond to the expected future reward starting from state s:

{EMBED Equation.3} (5)

Where $Q^{\pi}(s, a) = E\ \Sigma\gamma r_i$ and $Q^*(s, a)$ to references the optimal state-action with following policy π^* if $Q^*(s, a) = \max Q^{\pi}(s, a)$ and if we reach the $Q^*(s_i, a_i)$ for each pair state-action then we say that the agent can reach the goal starting from any initial state. The value of Q is updated by the following equation:

{EMBED Equation.3} (6)

4. Proposed Approach

Framework of the proposed method is shown In Fig. 2. The dynamic systems require a balance between exploitation and exploration processes in the search for optimal actions. An imbalance between these concepts can produce either a premature convergence, to a chaotic state, or a divergence that leads the system towards a deadlock situation. This equilibrium is achieved through reinforcement learning and performing actions based on a heuristic method.

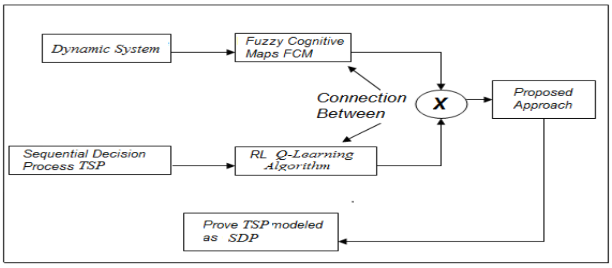

Fig 2. Framework of the proposed approach.

The proposed methode is summarized by pseudo code 1:

Step 1: Generate the output vector {EMBED Equation.3}

Step 2: In response to environment:

IF *r = 1 // Award*

{EMBED Equation.3}

{EMBED Equation.3}

{EMBED Equation.3}

IF *r = o // Penalty*

{EMBED Equation.3}

{EMBED Equation.3}

{EMBED Equation.3}

Step 3: *Stop if the system converge. Otherwise go to Step* 1.

5. Case Study: Symmetric Traveling Problem

In the dynamic systems theory, we can model the TSP same as a sequential decision process SDP [11], designated by the sextuplet $\Gamma = \{\Gamma, S, A_p, P, Q, W\}$. An alternative, between others, is to consider that a set of states S is composed by all cities for solving *TSP*. Dimension of S here is equivalent to the instance size of the problem The efficient understanding the power of proposed approach is presented in example of TSP with 5 cities shown on Fig. 3. All action a_{ij} are being to visiting the city s_j from city s_i, and the number associated to each arc corresponds to the distance between cities:

1. Γ: iteration set instants denoted by $\Gamma = \{1 \dots n\}$, where the number of n cities that form a route for TSP corresponds to cardinality of Γ.
2. S: set of states represented by $S = \{s_1, \dots, s_n\}$, with each state s_i, $i=1,\dots,n$ corresponds to a city.
3. A_p: The set of possible action {EMBED Equation.3}
4. *P:* transition probability function between states $s \in S$ with the elements $p_{ij}(s_j|s_i, a_{ij})$

   is the probability to reach state s_j were it is in state s_i choose action a_{ij}..
5. *Q: one pair of (*state, action) value measures quality function denoted by{EMBED Equation.3}.
6. *W:* weight matrix between concepts and is a function of {EMBED Equation.3}in $\Re$ relating a weight {EMBED Equation.3} to pair{EMBED Equation.3}. The best way to initialize the connection weights is to take W_{ij} inversely proportional to the distance between cities {EMBED Equation.3} = $1/d_{ij}$.

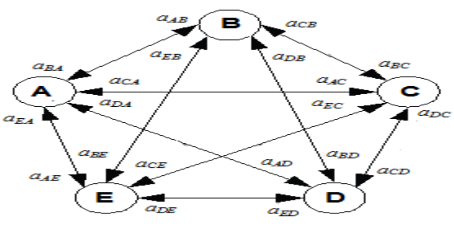

Fig 3. Graph of the example TSP with 5 cities.

In summary, the main objective is to find shortest path of visiting n cities exactly one time and returning to the initial city [13]. The mathematical description is:

$$\text{Minimize } \sum d_{ij}x_{ij} \quad (8)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,2,\ldots,n \quad (9)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,2,\ldots,n \quad (10)$$

Where d_{ij} represents distance between cities i and j; in the permutation matrix, decision variable indicate that the path is from city i to city j; be a sign of the route which isn't chosen by the salesman. Equation (8) represent the objective function, (9) and (10) are the constraints to ensure that each city will be visited only one. One solution to the problem, a tour visiting all cities and return to the started city, can be encoded as a permutation matrix, i.e., a binary square matrix containing exactly one '1' per column and row. In this matrix, a line represents a city and a column indicates the order of visiting this city's during a tour. For Fig 3, one possible tour BDAECB is shown in Table 1.

Table 1. One accepted solution for 5-cities TSP.

	A	**B**	**C**	**D**	**E**
A	0	0	1	0	0
B	1	0	0	0	0
C	0	0	0	0	1
D	0	1	0	0	0
E	0	0	0	1	0

The dynamics of fuzzy cognitive map is guided at each step in the evolution of the system with allowed actions to move from one state s_i to state s_j, i.e. at the heart of the construction of the commercial traveler solution is constrained by behavioral adaptation in a given step made that certain actions are not available to go from state s_i to state s_j. The possible actions set is denoted by $A_p = A_p(s_1) \cup \ldots \cup A_p(s_n)$ with $A_p(s_i) = \{ a_{ij}, a_{ik}, \ldots, a_{in} \}$. For example, if in step k one has the partial following solution sol_p: $s_i \rightarrow s_j \rightarrow s_k$ with $i > j > k$, then the possible actions in this stage to advance to the next stage are: $A_k(s_k) = \{a_{kr},\ r \neq i \text{ and } r \neq j\}$. In this case the states si and sj with respectively a_{ki} *and* a_{kj} actions are not feasible, and this is to prevent the passage through the same state (the same city) more than once accordingly to respect constraints.

6. Hybrid Learning Fuzzy Cognitive Maps HLFCM

The inference mechanism, by IF-THEN rules, start after fuzzyification process of the input data is accomplished. The search for the best solution at each step, the system is in a state represented by concepts of FCMs constructed at this stage and we called transition card. The traveler arrived at this stage is always seeking to transit to a future city (state), among the possible cities by optimizing the reward of the environment and respecting the constraints imposed, a city is visited once and only once, by adapting his behavior by removing actions that are not permitted at this stage.

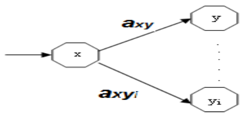

Fig 4. Transition map as a sub trip.

x is the new state, $a_{xy}....a_{xyi}$ possible actions at step k and $y_1....y_i$ are possible state or cities to visits. The adaptation of behavior is also guided at each step by using the parameter transition between state s_i, this parameter is equal to 0 if the state is not previously visited and equal to 1 if the state is already visited.

$$\delta = \begin{cases} 1 \text{ if the state is visited} \\ 0 \text{ if state is not visited} \end{cases} \qquad (11)$$

For TSP the fuzzy rules can be designated as:

$Rule_k$: IF x_1 *is* s_1 *and* x_2 *is* s_2*...and* x_k *is* s_k THEN y_k *is* O_k Where $x_1, x_2, ..., x_k$ are the input at step k. $s_1, s_2, ...s_k$ the membership function of the fuzzy rules represents states or cities and y_k the output of the rule $Rule_k$ designated by membership function O_k.

This fuzzy rule is also known as Mamdani fuzzy type model or linguistic fuzzy model. For example, in our case study TSP of 5 cities, the fuzzy rule associated for the transition card at step k can became as:

Table 2. Fuzzy rule processes.

IF				THEN
x_1	x_2	x_3	x_4	y
A	*C*	*D*	*E*	**B**
A	**D**	**E**	**B**	**C**

In this example, the traveler's will take a choice between two actions that lead to two states or two different cities (represented by concepts in LFCM). if in step 3 the salesman person is in the city D knowing that the initial starting state A and was the city he passed is the city C, the next possible cities or states are the city B or the city E, so the traveler must choose the next city to be visited in next step. For this the balance between exploration and exploitation is assured gradually based on the data of the table of the function Q values and the probability of each possible action at each stage.

Table 3. Fuzzy rules at step 3.

IF			*THEN*
x_1	x_2	x_3	y
A	*C*	*D*	*B*
A	**C**	**D**	**E**

In this step the traveler has visited the cities A, C and D and must choose the next city to visit. Here there are two options either to go to the city B or city E. based on the constructed transition map Fig 6 , the choice is guided by the probabilities of possible actions at this level and the value of the function Q if it has already taken this path.

Table 4. Output solution vector.

Input vectors	**Output vectors**	**Iteration**
1 0 0 0 0	1 0 1 0 0	1
1 0 1 0 0	1 0 1 1 0	2
1 0 1 1 0	1 1 1 1 0	3
1 1 1 1 0	**1 1 1 1 1**	**4**

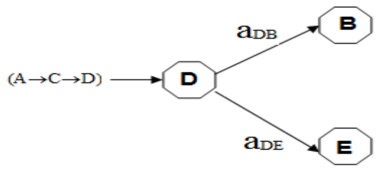

Fig 5. Transition map at step 3.

In this stage traveler has two possible actions namely a_{DB} and a_{DE}. Their corresponding Q-values and probabilities are initially depicted in the next table 5 as follows:

Table 5. Action probabilities and Q-function values.

a_i	$P(a_i)$	$Q(s_i, a_i)$	*Value*
a_{db}	p_{db}	$(B\ a_{db})$	Q_{db}
a_{de}	p_{de}	$(E\ a_{de})$	Q_{de}

The Q-function values of (state, action) initially receives a null value for all items, i.e., $Q(s_i, a_{ij}) = 0$, and a table of action probabilities initially receives a $1/n$ value for everyone actions at each associated state, and n is the number of actions at this state. At everyone iteration, the updates of Q-value and probability actions are made using pseudo code described in pseudo code 1. The Q value is rounded to 1 for the winner concept, which means this concept is activated, after the environment's response on the action giving the best result.

7. Experimental Results

The targeted objective here is behavioral adaptation in decision making during an autonomous entity reasoning mechanism. Tests were carried out using two instances of the TSPLIB library [8]: Burma14 and Ulysses16.

Table 6. TSP instances information.

instance	Number of cities	Optimal solution
Burma14	14	3323
Ulysses16	16	6859

After 20 runs on each city set, all statistics for HLFCM were generated and shown in table described below:

Table 7. Statistics comparison.

Instances And optimal TSPLIB solution	Classical FCM solution	Deviation Classical FCM/Optimal solution	HLFCM solution	Deviation HLFCM/ Optimal solution
Burma14 (Optimal solution 3323)	4624	34.15%	3334	0,33%
Ulysses16 (Optimal solution 6859)	8726	27,21%	6873	0,20%

The comparison, described in Table 7 and shown on Fig 6, between conventional FCM and FCM with hybrid learning shows that FCMs are able to learn from experiences and use their historical past in a very optimal way to model and simulate of the dynamic systems. At all iteration, one concept is active, i.e. its value is equal to 1, and the value of other concepts of the transitional card is initialized to 0. Evolution of the modeled system is performed by the reasoning mechanism implemented using the inference process described by the pseudo code of the pseudo code 1.

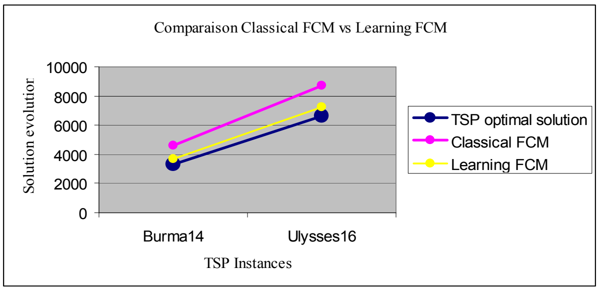

Fig 6. Classical and hybrid learning FCM evolution Solution.

8. Conclusion

In this paper, the study of traveler behavior was focused on the cognitive reasoning mechanism induced by traveler and the TSP is taken here just as a representative example. Studies of salesman traveler behavior in computer science and other related science are most important for many reasons, for example, to optimize both travel related cost and time consumption. In the last two decades, many attempts have been made to give best solution using heuristic techniques. The discussed method solving in this paper is based on classical Kosko's FCM type improved by a connection with RL. An heuristic way of updating concepts output value is presented. Based on fusion of the temporal transition maps, the whole FCM parameters were obtained and which led to more best results. Naturally the behavioral adaptation is a cognitive task that autonomous entities apply to adapt to their dynamic environment, so in this work we have targeted the TSP reasoning mechanism. In future works, we aim to test our approach on several instances of the TSP and from a mathematical point of view. We plan to improve the approach by formulating a standard model those implements, in a general manner, the reasoning mechanism of autonomous entities.

References

1. Axelrod R., (1976). ‘Structure of decision’. Princeton press..
2. Bart Kosko, (1986). ’ Fuzzy Cognitive Maps’, IJMM Studies, 24:65-75,.
3. Maikel L., Ciro R., Maria M., Garcia R. B & Koen V (2010). ‘FCMs for modeling complex systems’. Springer.
4. Stylios C D. & Peter P.G. (2004). ‘Modeling complex systems using FCM’. IEEE .
5. Tarkov M.S., (2015). 'Solving the TSP using a RNNs’. JNAA, Springer.
6. Buche C. A., Parenthoen M. & Tisseau J. (2010). ‘FCMs for the simulation of individual adaptive behaviors’ Wiley & Sun.
7. Sutton R. S. & Barto A. G. (2005). ‘Reinforcement Learning: An Introduction’. Book. The MIT Press, London, England .
8. Web TSP page”, http://comopt.ifi.uni- heidelberg.d/software/tsplib/index.html/.

9. Tolman E. (1948). 'Cognitive maps in rats and men'. Review vol. 55.
10. Thomas J. (2010). 'Dynamic Macroeconomic Theory', Section 1.1-1.4.
11. Leon M, Nápoles G., Bello R., Mkrtchyan I., Depaire B. & Vanhoof K. (2013). 'Tackling Travel Behavior: An approach based on FCMs'. IJCIS, Vol. 6.
12. Jasmin E., Imthias T.P., Jagathy V.P.R. (2011). 'Reinforcement Learning approaches to economic dispatch problem'. Elsevier.
13. Donald D., (2010). 'TRAVELING SALESMAN PROBLEM, THEORY AND APPLICATIONS'. InTech Janeza publisher, 51000 Rijeka, Croatia, Copyright ©.
14. Liu J., Qiu W. (2008). 'GA-Hopfield Network for Transportation Problem'. IEEE.
15. Kajal D. & Chaudhuri A. (2010). 'A study of TSP Using Fuzzy Self Organizing ap'. TSP Theory and Applications, Prof. Donald Davendra Ed. ISBN: 978-953-307-426-9, Intech Books.