



Adaptive e-Learning Multi-Agent Systems with Swarm Intelligence

Manuj Darbari

Department of Computer Science, BBD University,
Lucknow, India.

Priya Sahai

Department of Computer Science, BBD University,
Lucknow, India.

Abstract

In this paper we propose a multi-agent approach to the problem of recommending relevant and adequate study material to the learner in E-learning environment. It illustrates swarm intelligence of bee-colony optimization that enables agents to recommend learner most appropriate data content in real-time which stored in the form of case-sets. A flexibility, adaptability and interactiveness is achieved through agents that autonomously and intelligently uses swarm intelligent algorithms to recommend course structure to learners. Secondly we have suggested construction of a cache knowledge base which will be updated by swarm intelligent agents by analysing various parameters such as learners feedbacks, their educational records and other parameters.

Keywords: E-learning, swarm intelligence, multi-agent system, cache, Baye's theorem.

1. INTRODUCTION

There are number of E-learning methodologies developed so far, but none of the models are able to deal with the interactiveness where the system can intelligently recommend most appropriate course content to the user based on his educational records, feedbacks in real-time.

E-Learning, the new technology of supporting education and training, recently has been gaining a lot of attention. Content retrieval in e-learning refers to the process by which the learning content is provided by means of an electronic medium. It is an effective web-based learning paradigm, where many agents can be assigned with unique responsibilities to cope-up with the content retrieval and content delivery to various users. Multi-Agent based system[7,8,9] can manage the information stored in the e-learning environment, access permissions and granting access to various data content. Each tasks are carried out by an autonomous agent and various such agents are grouped to form a multi-agent based system.

To allow a better educational experience it is necessary to develop systems which have the capability to offer specific and personalized contents to the students in an intelligent way[10], i.e. systems capable of making decisions about which is the most suitable educational content at every time for each learner.

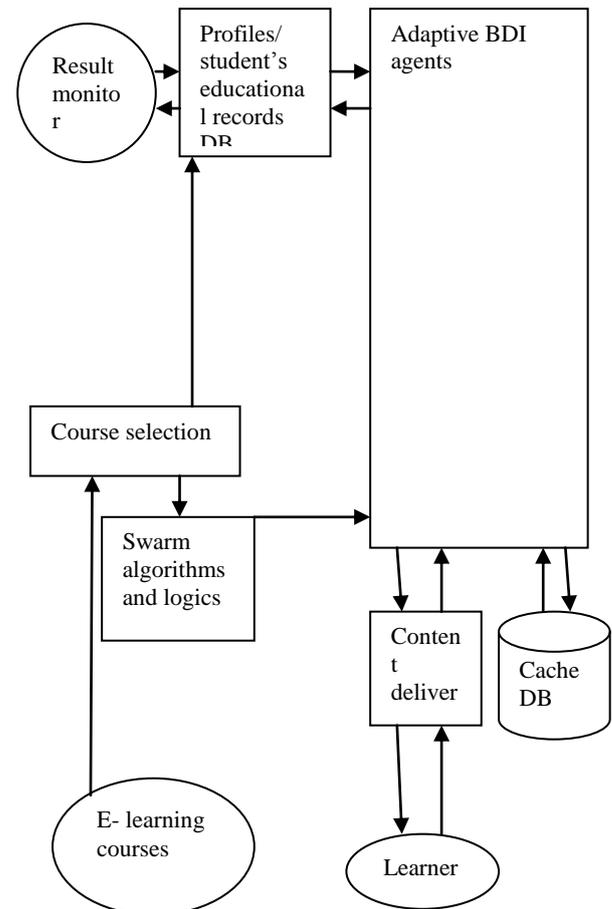


Figure 1: Basic framework for adaptive e-learning agent

Swarm intelligence can be stated as, "the emergence of coherent functional global patterns from the collective behaviours of entities interacting locally with their environment"[1]. Swarm intelligence is mostly used to solve computationally hard problems like NP-hard problems. There is certain level of abstraction at which solving of such problems can be modelled using distributed computational systems consisting interacting artificial entities [2].

In order to provide our solution in real-time we are using bee-colony optimization technique of swarm intelligence in our recommender system framework that ensures task-allocation among cooperative agents.

2. RELATED WORK

We are motivated by the work of Priya Sahai, Manuj Darbari on Adaptive e-learning using Granularised Agent framework [3]. This paper illustrates a multi-agent system using a



granularised approach for solving the problem of recommendation overload to provide relevant courses to users in real time. It proposed dynamic binding of clusters to form granules ensuring right course delivery to the student based on his educational record, choices and helps setting his learning pace too.

The work by Sorin Ilie, Costin Badica ,Multi-agent distributed framework for swarm intelligence[4]. It presents a multi-agent distributed framework for swarm intelligence .Their framework could solve graph search problems on a computer network, they have compared the results of the distributed implementations bee-colony and ant –colony algorithms to conclude that ACS algorithm is faster than BCO.

We used CASIS ,case-based recommender system from Task allocation in Case-based Recommender Systems [5]. CASIS, combines the use of swarm intelligence in the task allocation among cooperative agents applied to a case-based recommender system to help the user to plan a trip.

We have used the concept of Intuitive Bayes Theorem from Mario F. Triola by her work on Baye’s Theorem [6].Her work contributes that when calculation is complicated enough to create an abundance of opportunities for errors and/or incorrect substitution of the involved probability values. Fortunately there is another approach that is much more intuitive and easier called Intuitive Baye’s theorem. It states that: “Assume some convenient value for the total of all items involved, then construct a table of rows and columns with the individual cell frequencies based on the known probabilities.”

3. PROPOSED WORK

In order to provide course structure to learner in real time we have used Bee- colony optimization technique in our recommender system .It’s key features are updating the best known solution for a particular query and secondly, saving it in the form of case sets at the server website.

Whenever a learner enters it’s queries the agent responsible for the content retrieval will check all the existing case –sets to derive the best known course set that should be recommended to the user. In case no previous saved case-sets matches the user query for a particular request then agent will select randomly the best solution by matching the query for the particular request leading to content delivery.

Algorithm for Basic Bee-colony implementation in our framework can be stated as follows:

AGENT_INIT()

1.V ← **MANAGED_DATASETS()**

2. for v ∈ V do

3. **CREATE_ENTITY(V)**

ADJUST_ATTRIBUTES(entity)

1.if **RECOMMENDATION_COMPLETED(entity)** then

2. **DATABASE_SEARCH(entity)**

3. **BROADCAST_SOLUTION(entity)**

4. **INITIALIZE(entity)**

BEST_DATASET(entity)

1.best_dataset =**RANDOM_CHOICEMATCH()**

2.**ADD_to_BESTSOLUTION(best_dataset,entity)**

3. return best_dataset.

HANDLE_DATABASE(entity)

1.**UPDATE_BESTSOLUTION(entity).**

Entity here can refer to the message transfer between agents searching for **BEST_RECOMMENDER_SOLUTION**.

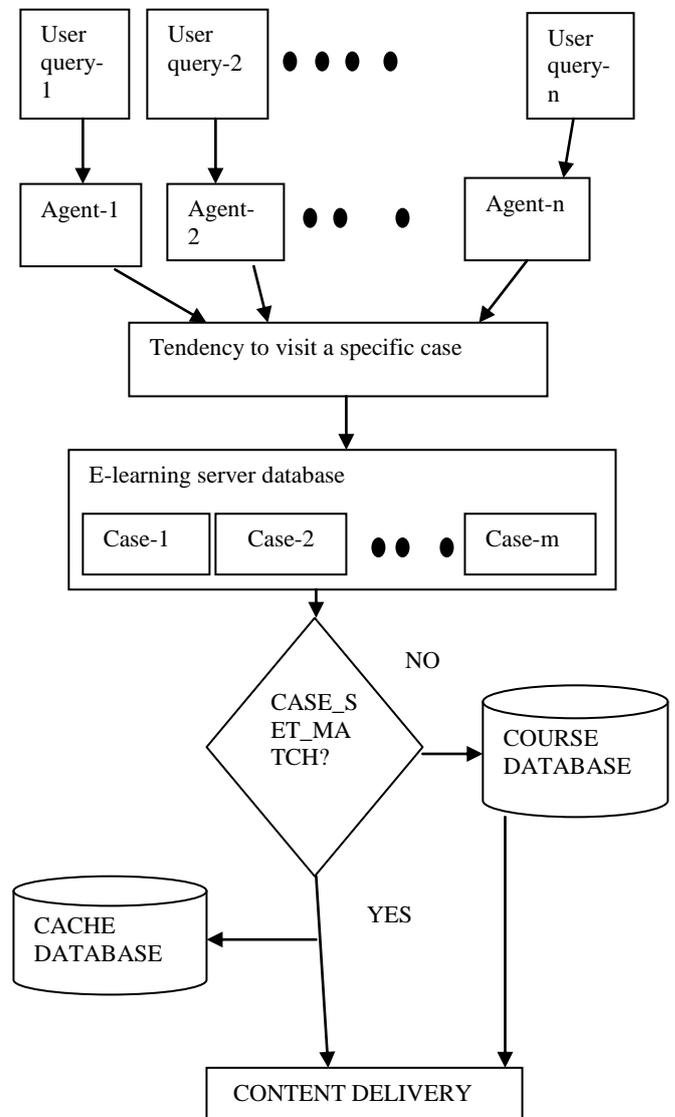


Figure 2: Architecture for swarm intelligent adaptive E-learning system

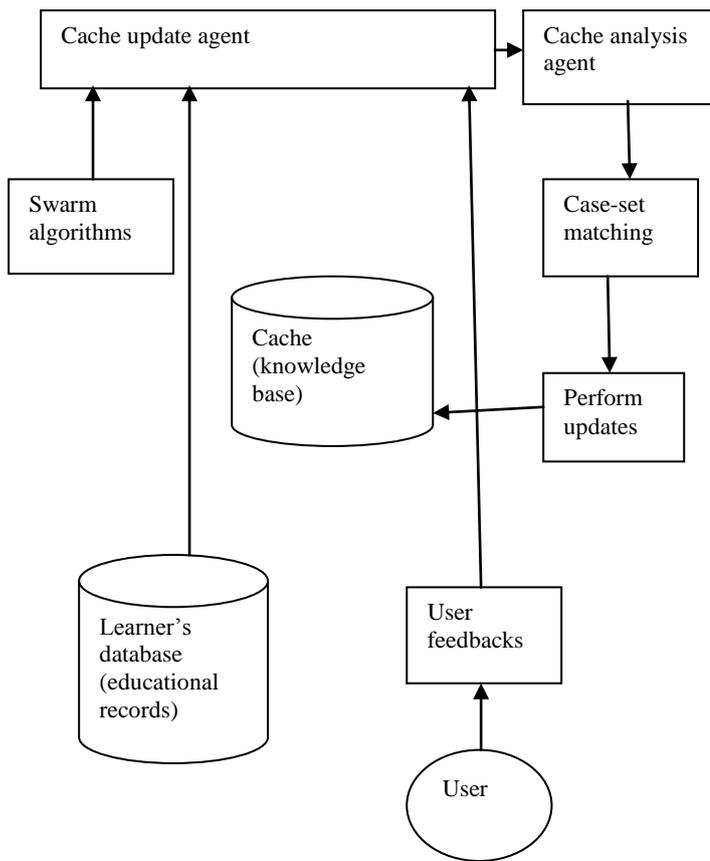


Figure 3: Cache update system

The multi-agent system will maintain a cache ,this cache contains all the best-known solutions. Whenever a user query is checked –in the content retrieval agent will search the cache for the best match solutions .In case if no solution is found then the query is forwarded to the next agent that will retrieve the best course data set to the learner.

An agent is also responsible for the database update in cache. This updation process is completed by analysing data from three sources:-

- Feedback from user/learner.
- User results from his saved educational record.
- Performance measures for swarm algorithms.

We are using Camazine and Sneyd model [7] to finally derive an algorithm for the adaptive recommender system in e-learning environment. Data-sets or case-sets here refers to the chapter or clusters of course chapters belonging to a particular domain. Various variables used in algorithm can be stated as following:

f_a^d : probability of selecting a case-set A for the query

f_b^d :probability of selecting a case-set B for the query

f_x^a : probability of abandoning case set A and moving ton next case set.

F_x^b : probability of abandoning case set B and moving ton next case set.

$A(f_1^a)$: probability of selecting case set A from cache to recommend

$B(f_1^b)$: probability of selecting case set B from cache to recommend

For updation process of the cache knowledge base in recommendation system we have used the following variables:

$$A(f_1^a) = \frac{D_a d_a}{D_a d_a + D_b d_b}$$

And

$$B(f_1^b) = \frac{D_b d_b}{D_a d_a + D_b d_b}$$

Where:

A, B : Case-sets

D_a and D_b : number of user preferences from feedback given by them.

d_a and d_b : most recommended case sets or most hits/favoured cases.

Educational record of the learner is analysed by the agent and their parameters discussed above are used to recommend and update the database and cache.

Probability of selecting a particular case set uses following levels or grades appointed to them through various sources illustrated earlier. The grades ranges from :-

- Poor
- Satisfactory
- Above average
- Good
- Excellent or best.

Now, for algorithm a term that is most commonly used is Tendency(T). It is the probability to visit a specific case[5].Instead of visiting all cases from the case-base the agent will visit a case stored in the case base according to the tendency associated to that case. Then further analysis of that case set is done to decide whether it should be recommended to the user or not.



$$T_{ij} = \frac{S_{ij}^2}{S_{ij}^2 + \theta_{ij}^2}$$

where S_{ij} is the stimulus associated with case-set(i) and demand (j) of user. θ_{ij} refers to answer threshold with case i and demand j. θ_{ij} and S_{ij} are updated as following:

Case-1: If the agent selects a certain case that it visited for recommendation

$$\theta_{ij} = \theta_{ij} - \alpha, \text{ if } \theta_{ij} > 0$$

$$\theta_{ij} = 0, \text{ otherwise}$$

$$S_{ij} = S_{ij} + \alpha$$

Case-2: If the agent does not selects or ejects a particular case from visiting.

$$\theta_{ij} = \theta_{ij} + \alpha$$

$$S_{ij} = S_{ij} - \alpha$$

θ_{ij} and S_{ij} are working as a reinforcement to the cases[5] that are most suitable for the user query, ensuring that the agent does not visit any case which is not relevant to the demand or query that is being handled. According to the recommendation algorithm, when user inputs his preferences the agents are stimulated.

Recommendation algorithm :

C_b : case-base (c_1, c_2, \dots, c_n)

D_b : demand base (d_1, d_2, \dots, d_n)

$$\theta_{ij} = 1$$

$$S_{ij} = 1$$

For $b=1,2,3, \dots, n$ do

$$d_1 = \text{random}(D_B)$$

$$c_1 = \text{random}(C_B)$$

$$T_{ij} = \frac{S_{ij}^2}{S_{ij}^2 + \theta_{ij}^2}$$

If (T_{ij} met) then

begin

repeat

$$p_{x_{ij}}^c = \text{Euclidean distance}(q, C_{ij})$$

if ($(p_{x_{ij}}^c)$ not met) then begin

$$p_{D_{ij}}^c = 1 - p_{x_{ij}}^c$$

if ($(p_{D_{ij}}^c)$ met) then begin

$$\theta_{ij} = \theta_{ij} - \alpha$$

$$S_{ij} = S_{ij} + \alpha$$

End

else keep searching

end if

end

else begin

$$\theta_{ij} = \theta_{ij} + \alpha$$

$$S_{ij} = S_{ij} - \alpha$$

Repeat

$$P_{F_{xj}}^c = \frac{Dc_{xj}}{\sum_{y=1}^M D_y}$$

If ($(P_{F_{xj}}^c)$ met) then

$$C_i = C_x$$

End if

Until ($(C_i = C_x)$)

End

End if

Until(agent visiting case set i and user query j matches or fulfils parameters, set criteria for recommendation to the demand j of the learner)

End

End if

End for

Return highest level matching case-set.

4. ANALYSIS

Baye's theorem illustrates how the conditional probability of each of a set of possible causes for a given observed outcome can be computed from knowledge of the probability of each cause and the conditional probability of the outcome of each cause. Originated in mid 19th cent and named after Thomas Bayes (1702–61), English mathematician. It is commonly known as "Bayes' Rule" given as:



$$P\left(\frac{A}{B}\right) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) * P\left(\frac{B}{A}\right)}{P(B)}$$

In order to select the best course material for the user we have used intuitive Baye’s theorem. We are going to use Intuitive Baye’s Theorem to illustrate our swarm intelligent agents activities in recommendation process and content extraction. Sometimes in artificial intelligent systems calculations is complicated enough to create an abundance of opportunities for errors and/or incorrect substitution of the involved probability values[6] . Thus, we use this much more easier and intuitive approach.

According to Intuitive Baye’s theorem we assume some convenient values for the total of all items, then we construct a table of rows and columns with the individual cell frequencies based on the known probabilities.

Assuming that some values for the number of requests being entered in the e-learning system, is 100 and the probability of rejecting a case-set and not visiting at all is 50% (table-1). Thereby, visiting a case-set probability is taken as 50%. When an agent rejects a particular case-set from visiting at all it chooses a new random case-set to visit then which is further selected based on the given calculations:

$$P_{F_{xj}}^c = \frac{D_{cxj}}{\sum_{y=1}^M D_y}$$

where, D_{cxj} is already calculated by the cache update analysis agent.

Agent will visit any case-set or data set only if the calculated value of $P_{F_{xj}}^c$ is ≥ 0.8 . It’s an algorithm parameter that accounts for the efficiency of the recommendation system and ensuring that suggested course content is apt for user’s demands.

Now assuming that for any demand D_{cxj} is 8 and $\sum_{y=1}^M D_y$ is 9 then, on rounding off we get the value of $P_{F_{xj}}^c$ as 0.89. In case if $P_{F_{xj}}^c < 0.8$ then the entire process of searching restarts for another case-set.

Table 1 :Intuitive Baye’s Theorem values for algorithm.

characteristics of V	V (visit a case-set)	\bar{V} (does not visit a case-set)	Total
Characteristics of A			
A(probability of accepting a case-set)	$0.8 \times 50 = 40$	0	40
\bar{A} (probability of rejecting a case-set)	$0.2 \times 20 = 10$	$1 \times 50 = 50$	60
Total	50	50	100

We can easily address the question that what is the probability of our algorithm successfully giving the course content to the user when a *case-set* is visited. Restrict the table to the column of visit a case-set , then we can find the probability of accepting the case-set in that column. Among the 50 case-sets visited there are 40 accepted , so the probability we seek is $40/50 = 0.8$. That is, $P(A/V) = 0.8$.

The above value(0.8) satisfies our parameter value generated from our algorithm is efficient enough to suggest a course-set to the user in real-time mode.

5. CONCLUSION

The analysis brings to a conclusion by proposing a recommendation swarm algorithm for a multi-agent e-learning system that is adaptable , interactive ,flexible and highly intelligent . We also stated that our given algorithm satisfies the given two necessary qualities in an e-learning environment:

- (1) It’s adoptive nature.
- (2) High accuracy value.

6. REFERENCES

- [1] A.P.Englebrecht, Fundamentals of computational swarm intelligence, Wiley, 2005.
- [2] S. Ilie, C. B̃adic̃a, Multi-agent approach to distributed ant colony optimization, Science of Computer Programming.
- [3] Sorin Iliea, Costin B̃adic̃a, 2003. Multi-agent distributed framework for swarm intelligence,International Conference on Computational Science, ICCS.
- [4] Priya Sahai, Manuj Darbari,,2014 .Adaptive e-learning using Granulerised Agent Framework, international Journal of Scientific & Engineering Research, Volume 5, Issue 2, February,ISSN 2229-5518.
- [5] Task allocation in Case-based Recommender Systems: A swarm intelligence approach by Fabiana Lorenzi, Daniela Scherer dos Santos, Denise de Oliveira and Ana L.C. Bazzan,2007.
- [6] Mario F. Triola, Baye’s Theorem, pearson education,2010.
- [7] M Darbari, N Dhanda , Applying Constraints in Model Driven Knowledge Representation Framework,2010. International Journal of Hybrid Information Technology 3 (3)4.
- [8] M Darbari, S Medhavi, AK Srivastava Development of effective Urban Road Traffic Management using workflow wechniques for upcoming metro cities like Lucknow (India Development 2 (2),4,2008.
- [9] N Dhanda, M Darbari, NJ Ahuja,,2012 Development of Multi Agent Activity Theory e-Learning (MATEL) Framework Focusing on Indian Scenario International Review on Computers & Software 7 (4),2.
- [10] A. Tiwari, P. Patel , V.K. Singh , A. Srivastava. Implementing Requirements for the hospital management system using multi- agent.