Emergent Intelligence: A Novel Computational Intelligence Technique to Solve Problems

Suresh Chavhan and Pallapa Venkataram
Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India

Keywords: Emergent Intelligence, MultiAgent System, Job Scheduling and Resource Allocation.

Abstract: Technological advancement and increasing globalization makes humans face many problems in day to day life, involving many possible goals and each goal is associated with multiple possible actions, each associated with many different dynamic and uncertain consequences. In real systems, the message passing mechanisms and few computational intelligence techniques (like Swarm intelligence, Multiagent System, etc.) hinder mutual cooperation and coordination of agents while solving problems in an uncertain environment, even though they are highly efficient and sophisticated. Therefore, in this paper, we propose an Emergent Intelligence technique (EIT) based problem solving. The EIT is collective intelligence of group of agents, which is an extension of multiagent system (MAS). Unlike MAS, the EIT provides independent decision making for a single task by the multiple agents with mutual coordination and cooperation. It is very useful to solve the complex and dynamic problems in uncertain environments. In this paper, we discuss EIT functioning, benefits, comparisons, and also illustration of two problems: (1) resource allocation and (2) job scheduling. Each problem is categorically analyzed and solved step by step using EIT. We measure performance of the technique by considering real time situations, and results are compared and shown the importance of EIT over MAS.

1 INTRODUCTION

Problem solving is the process of working through details of a problem to reach one or more solutions. Problems have some typical characteristics, such as complexity (large number of items, interrelations and decisions), dynamics (time considerations), in-transparency (lack of clarity of situation) and polytely (multiple goals). Problem solving may include mathematical or systematic operations and can be a gauge of an individual's critical thinking skills. The problems are solved using problem solving methods includes Swarm intelligence, Agent based system, heuristic algorithms, etc.

Agent-based systems use the set of intelligent agents for complex problem solving and decision making. Software agent systems allow a single agent intelligently interacts and collects relevant information to make a decision. In other words an agent is an authority to take decisions while solving problems, like conflict problems (Jacak, et.al., 2007), E-learning (Sun, S., et.al., 2007), medical (Fenza, G., et.al., 2012), process automation (Pakonen, A., et.al., 2007), image analysis (Bell, D.A., et.al., 2007), etc. Swarm intelligence based system solves asset management problem (Reynolds, Joshna, et. al., 2015), real time allocation problem (Johansson, F.,, et. al., 2010), scheduling problem (Kalyan V., et. al., 2004), evacuation problem (Guizzi, et. al., 2015), etc. These methods take indepedent local decisions, individual interactions, self-aware, provides local view and they concern managing the agents behavior, like migration, interaction, mobility, etc., and these behavior makes less efficient and may lead to delays and less accurate in decision making.

Hence, we have developed a technique called EI where multiple agents would take partial or full decisions on a task by using the available and historical information and the decisions are communicated to the main task originator. An EI is not attributed to an individual agent, but it is a global outcome of group of agents coordination, cooperation, collaboration and negotiation. The Emergent Intelligence is a main concern of potential users when we try to utilize agent technology for transportation planning, autonomous systems, development of smart city, smart grid computing, etc. In addition, The EI takes collaborative and collective dynamic decisions, group interactions, group aware, provides global view and also it concerns management of agents behavior and man-

Chavhan, S. and Venkataram, P.
Emergent Intelligence: A Novel Computational Intelligence Technique to Solve Problems.
DOI: 10.5220/0007244100930102
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 93-102
ISBN: 978-989-758-350-6

agement of information, such as resources, jobs, activities, tasks, etc. In literature (Rqevski, G., et. al., 2007), (W.D. Hillis, 1988), (T. Wolf, 2007), (Z. Li, et. al., 2006), and (P. J. Angeline, 1994) exhibits the guidelines on how to exploit Emergent Intelligence and Self-organization in multiagent systems to solve problems. But it is not clear, how EI technique can be used for problem solving in uncertain and dynamic environment.

In this paper, we developed and presented a novel technique, i.e., Emergent Intelligence Technique (EIT), which is an extension of multi-agent system (MAS). We have demonstrated an EIT based solution by collaborating, coordinating, cooperating and negotiating the problem solving agents. The proposed technique solves the problem by executing 6-phases sequentially: (1) Analyzes given problem, makes clear problem statement and identifies all possible tasks, subtasks, inputs and outputs; (2) Builds an EI-network for the given problem; (3) Estimates possible practical solutions for each sub-task; (4) Creates action plans; (5) Implements all action plans and (6) Whenever any changes in input repeats phases from 1-5. In this paper, we have illustrated resource allocation and job scheduling problems, each problem is categorically analyzed and solved step by step by using EIT.

The rest of the paper is organized as follows. Section 2 presents the literature survey. Section 3 discusses the EI technique. Section 4 presents the resource allocation problem using EI technique. In Section 5 job scheduling problem solving using EI technique is discussed. Section 6 case study is described; Section 7 simulation and analysis results are discussed; and conclusions are drawn in Section 8.

2 LITERATURE REVIEW

In this section, we discuss problem solving using the intelligent systems with swarm intelligence, multiagent systems, and emergent intelligence.

2.1 MAS based Problem Solving

Authors in (Jacak, et.al., 2007) solved conflict problems by coordinating and negotiating multiagent systems actions to achieve a common goal. An agent consists of many autonomous components in order to perceive and react to its environment, plan and execute an action, and intelligently it negotiates with other agents. In (Gundersen, et.al., 2005) authors have developed a prototype of multi-agent system based approach to construct plans based on available resources in the environment, which dynamically plans and solves assigned problems. Also, MASs have been used to solve problems, such as E-Learning (Sun, S., et.al., 2007), medical (Fenza, G., et.al., 2012), process automation (Pakonen, A., et.al., 2007), image analysis (Bell, D.A., et.al., 2007). These kind of systems have traditional benefits of concurrent and distributed problem solving strategies (Bala M., 2008). Novice users suffer from their incapability to combine individual statements and constructs related to flowchart, HIPO chart, IPO chart, algorithm, etc., into valid programs () and (Aris., T.N., 2012). In (Johansson, F.,, et. al., 2010) authors have focused on investigating the possibility to use multi-agent systems as a new agent model for computational problem solving which is utilized by visual programming as the mode of programming to make it easier for novice programmers.

2.2 SI based Problem Solving

Particle swarm optimization is especially useful for rapid optimization of problem involving multiple objectives and constraints in dynamic environments. Work in (Johansson, F., et. al., 2010) particle swarm optimization has applied to real time allocation problems and discussed the allocation of weapons for defensive purposes. Authors in (Reynolds, Joshna, et. al., 2015) swarm intelligence is used for the autonomous asset management problem in electronic warfare. The particle swarm optimization speed provides fast optimization of frequency allocations for receivers and jammers in highly complex and dynamic environments. In (Kalyan V., et. al., 2004) authors have presented a swarm intelligence based approach for optimal scheduling problems in sensor networks. Authors have developed a methodology and cost function to solve the graph partitioning problem. The swarm intelligence algorithm solves the problem and emerges with an optimal schedule. Work in (Guizzi, et. al., 2015) authors have discussed the swarm intelligence based solutions to evacuation problems. Authors have determined the optimum path during evacuation process by using swarm intelligence's algorithms (both ant colony and particle swarm optimization).

2.3 EI based Problem Solving

Authors in (Rzevski, G., et., al., 2007) described scheduler behavior using emergent intelligence in multi-agent systems for not only transportation domain and all other logistics applications. Research in

(T.Wolf, 2007) explained how self-organizing emergent behavior exhibits in multiagent systems. Authors in (Li, et., al., 2006) carried out the survey of emergent behavior and its impacts on agent-based systems. Authors in (P.J. Angelone, 1994) demonstrated and provided the guidelines on how to exploit Emergent Intelligence to extend the problem solving capabilities.

3 EMERGENT INTELLIGENCE TECHNIQUE

EIT is collective intelligence of a group of agents. The EIT is an extension of MAS where agents group activities and individual independent decision making are incorporated. The group of agents cooperatively (whenever agents are compatible towards completing tasks execution), coordinately (whenever there is need of agents joint actions for executing tasks) and collaboratively (whenever agent has insufficient ability to carry out tasks) interacts among themselves to provide dynamic independent decisions for the given problems. The EIT can be used for executing independent tasks, which could be parallelly executed and provides partial (or complete) solution.

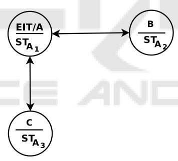


Figure 1: Scenario of 3 nodes network with task and its sub-tasks.

The EIT is illustrated as follows: Let us consider the task t_A in 3-nodes network as shown in Figure 1. As per the objectives of the task, it is possible to split the task t_A into 3-subtasks (st_{A_1}, st_{A_2} and st_{A_3}) which can be independently solved by 3 agents. Since the task is initiated at node A where EIT is deployed in it, which creates 3-agents A1, A2 and A3 and migrates them to all 3-nodes A, B and C, respectively. These agents independently solve the task t_A by considering all relevant information available locally and globally. Finally, the decision is communicated to the main task solving agent A1 as per the following equation

$$D(t_A) = D(st_{A_1}) + D(st_{A_2}) + D(st_{A_3}) \qquad (1)$$

where $D(st_{A_1})$, $D(st_{A_2})$ and $D(st_{A_3})$ are the partial or full decisions taken at nodes A, B and C, respectively. The same problem would have been solved by the MAS exclusively at node A by collecting relevant information from nodes B and C.

The EI technique is elaboratively discussed in the Section 3.2.

In the literature, we can observe that some of the attempts made to use EIT for solving problems (Rzevski, G., et., al., 2007, W.D. Hillis, 1988, T.Wolf, 2007, Z. Li, et., al., 2006, P.J. Angelone, 1994). But it is not clear, how this technique can be used for the problem solving in an uncertain domain.

3.1 Definitions

In this section, we present some of the definitions useful to discuss the proposed technique.

1. Task and its subtasks: Task can be decomposed into independent subtasks. These sub-tasks are assigned to agents and they execute and produce results. These results are passed to the main task originator agent and it takes final cumulative decision.

2. Cluster: A cluster is logically partitioned depending upon the nodes degree of connectivity greater than or equal to 2. The creation of clusters in the given problem network is given as follows: (i) Select a node (where problem need to be solved) from the given problem network, (ii) Agent at the selected node finds the degree of connectivity (i.e., all 1-hop away nodes), (iii) Agents at these 1-hop away nodes calculate their own degree of connectivity, if they have more than 2 then remove from the list of the selected nodes, (iv) End nodes or nodes which are having only one degree of connectivity are used to form the cluster at the selected node, and (v) Nodes which are having multiple degree of connectivity will also follow the above steps to form their respective cluster. Algorithm 1 explains the procedure to find the degree of nodes, cluster head nodes, end nodes, and forms the clusters of the given network.

Algorithm 1:
Begin
Let n be the total number of nodes in the given problem network
Select a node (where problem need to be solved) from n
/*Identification of degree of nodes in the given problem network*/
Initialize $D(i) = 0$ (degree of node i).
for $i = 1$ *to* n **do**
    for $j = i+1$ *to* n **do**
        if node i is 1-hop neighbor of j **then**
            $D(i) = D(i) + 1$
        end if
        Degree of connectivity of node i is $Deg_i = D(i)$
    end for
    $D(i) = 0$
end for
/* Identification of highest degree of nodes*/
Initialize $m = 0$
for $i = 1$ *to* n **do**
    for $j = i+1$ *to* n **do**
        $m = max(Deg_i,\ max(m, Deg_j))$
    end for

Highest degree of node is m
end for
/*Formation of clusters*/
for $i = 1$ *to* n **do**
if $Deg_i \geq 2$ **then**
Node i is cluster head
for $j = 1$ *to* Deg_i **do**
if degree of node j is < 2 **then**
Node j is the end node.
end if
Cluster i is formed with all the 1-hop away end nodes of node i
end for
end if
end for
End

3.EI-Network: It is formed by developing static agent at each of cluster heads. These agents are connected as per the requirements in the problem. The EI-network is formed as follows: (i) EIT running on a cluster head node (where problem need to be solved) creates an EI-network, (ii) EI-network consists of cluster head nodes and their connectivity to solve the given problem, (iii) EIT creates and sends agents to cluster head nodes in the EI-network, and (iv) In every cluster the cluster head interacts and shares the analyzed information. For example Figure 2 shows the EI-network with having 5-cluster head nodes, where EIT is running on node A. EI-network formation is explained in the Algorithm 2.

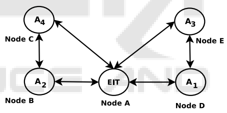


Figure 2: EI-network.

Algorithm 2:
Begin
Let A be the node where problem need be solved in the given problem network
Algorithm 1 provides the number of cluster head nodes (N_c).
EI-technique runs on a node A
for $A_i = 1$ *to* A_{N_c-1} **do**
Cluster head node A makes connection with cluster head node A_i
end for
for $A_i = 1$ *to* $A_{N_c} - 1$ **do**
Cluster head node A creates agent and sends to cluster head node A_i
Migrated agent interacts with cluster head node A_i and shares the analyzed information of the given problem
end for
End

3.2 EI-Technique Functioning

Any given problem in terms of goals and barriers can define many problems and splitting a big problem into more manageable sub-problems. The EI-technique solves a given problem in following six phases:
(P1) Analyze the problem, make clear problem statement and identify all inputs and outputs
(i) It analyzes the given network, available and required information, (ii) It identifies the number of inputs, tasks/subtasks and outputs, and (iii) It formulates the problem statement using the available and required information of the given problem along with the constraints, such as activity and history database information.
(P2) Build the EI-network for the given problem:
(i) Creation of clusters, which is explained in Section 3.1(2) and (ii) EI-network formation, which is explained in Section 3.1(3).
(P3) Estimate the practical solution for each sub-task:
(i) Each sub-task is assigned to a cluster-head in the built EI-network, (ii) EIT makes the cluster head node's agents to cooperate, coordinate, collaborate and negotiate among them to collect the resources needed for each sub-task, and (iii) Practical solution is estimated, of each sub-task, by referring to the activity and resource utilization database along with the available and collected resources.
(P4) Create an action plan:
(i) Every agents parallelly refers to history database to estimate accurate requirements of each of the node, (ii) In each cluster, the cluster head node's agent collects the relevant excess resources from nodes in the cluster, and (iii) In each cluster, cluster head node's agent parallelly allocates the resources.
(P5) Implementation of an action plan:
(i) Interaction among cluster head node and other nodes in the cluster, (ii) Accurate estimation of requirements using history database, (iii) Excess relevant resources collection from the one-hop away nodes, and (iv) Each cluster head node's agent parallelly allocates the resources.
(P6) Whenever a change in input repeat Phases from 1-5: In this phase, if any change in input or change in external environment condition it makes to repeat Phases from 1-5 again.

3.3 Benefits of EI Technique

Benefits of EIT over multiagent system (MAS) in randomness and uncertain problem domains are as follows:
(1) In EIT, distributed agents take partial or full decisions, whereas in MAS, decisions are unpredictable,
(2) A problem is being automatically decomposed among distributed agents in EIT, whereas in MAS, each agent chooses to whether to participate,
(3) Task and its sub-tasks are distributed and decentralized action selection are done in EIT, whereas in MAS, agents keep their own copies a task/sub-task that might include others,

(4) In EIT, distributed agents take the benefits of latest resource available, history database and also new nodes coming into picture during execution of tasks, whereas in MAS, takes the benefits of only available resources during execution of tasks,
(5) EIT reduces complexity (space and time) due to the concise and natural modeling of problem domain: (i) In MAS, takes $O(n \times m)$ time complexity and $O(n+m)$ space complexity, where n is the total number of agents (or nodes) and m total number of tasks need to be executed. (ii) In EIT, takes $O(n \times log(m))$ time complexity and $O(n+log(m))$ space complexity.
(6) In EIT, brings autonomy, flexibility, adaptiveness, robustness, self-organization and evolution, whereas in MAS, brings only autonomy,adaptiveness and robustness and
(7) EIT replenishes environment by creating autonomous regenerating feedback loop by interaction among group of agents.

3.4 Comparisons of Computational Intelligence Techniques

In this subsection, we present comparisons among Computational Intelligence (CI) techniques, that is MAS, Swarm Intelligence (SI) and EIT. In MAS approach agents coordinate their knowledge and activities and reason the processes of coordination autonomously. SI is used to model simple behaviors of individuals, and local interactions with environment and neighboring individuals, in order to obtain more complex behaviors that can be used to solve complex problems. Table 1 shows the comparisons among MAS, SI and EIT.

4 EXAMPLE 1. RESOURCE ALLOCATION PROBLEM SOLVING

Problem: A network consists of 13 nodes A, B, C, D, E, F, G, H, I, J, K, L and M (See Figure 3) and each of them have 10, 20, 15, 5, 20, 10, 15, 20, 25, 15, 6, 15 and 25 amount of resources, respectively. Each of these nodes requires 25, 15, 5, 4, 30, 8, 5, 30, 15, 10, 20, 10 and 15 amount of resources.

We demonstrate EI-technique functioning to solve the above problem by executing following 6-phases:

(P1) Analyze the problem, make clear problem statement and identify all inputs and outputs:
(i) Analyze the problem: (1) There are 13 nodes in the network A to M, (2) Resources available at nodes A, B, C, D, E, F, G, H, I, J, K, L and M are

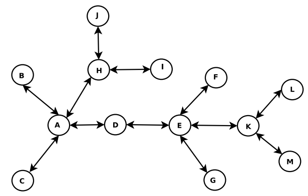


Figure 3: Scenario of a network with 13 nodes.

10, 20, 15, 5, 20, 10, 15, 20, 25, 15, 6, 15 and 25, respectively, (3) Required resources at each node are: 25, 15, 5, 4, 30, 8, 5, 30, 15, 10, 20, 10 and 15.
(ii) Problem statement: (1) Choose a node (say A) to run the EI technique (2) Create clusters with the nodes having high degree of connectivity. (3) Collect excess resources from its cluster nodes and from cluster head nodes A, E, H and K. (4) Check the genuinity of resource requirements using history. (5) Allocate required resources at nodes A, E, H and K.

(P2) Build an EI-network for the given problem by deploying agents at each cluster head by EIT:

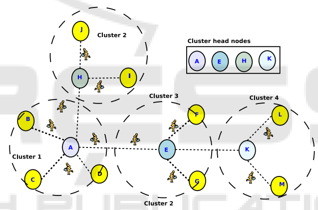


Figure 4: Clusters formation of resource allocation problem.

- Cluster formation (See Figure 4): Algorithm 1 is used to create the following 4 clusters of the given problem network. EIT forms cluster 1 at node A, cluster 2 at node H, cluster 3 at node E and cluster 4 at node K as shown in Figure 4.
- EI-network formation: The EIT running on node A uses Algorithm 2 to create an EI-network and deploys agents for interaction to share the analyzed information of the given problem (given by Phase 1) as shown in Figure 5.

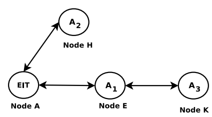


Figure 5: EI-network of resource allocation problem.

(P3) Estimate practical solution of a problem
(i) Resource collection for estimating solution of a problem: (1) EIT running on node A indicates its resource requirements parallelly to cluster head

Table 1: Comparisons of EI, MAS and SI.

Multiagent System	Emergent Intelligence Technique	Swarm Intelligence
Presence of decentralization control	Presence of partial decentralization and centralization control	Presence of decentralization control
Self-aware	Group-aware	Self-aware
Provides local view	provides global view	Provides local view
Takes individual decisions	Takes collaborative dynamic decisions	Takes local indepedent decisions
Individual interactions	Group Interactions	One to one interactions
Concerns management of agents behavior, like migration, interaction, mobility, etc.	Concerns management of agents behavior, like migration, interaction, mobility, etc. and management of information, such as resources, activities, etc.	Concerns management of independent agents behavior, like migration, interaction, mobility, etc.

nodes E, H and K, (2) Nodes E, H and K parallelly estimate their excess resources and they provide to node A, (3) Each agent refers to history database parallelly to estimate accurate requirements of each node, and (4) Each cluster head node parallelly shares its excess resources to the requested cluster head nodes depending upon the history database.
(ii) Practical solution at node A refers to history database which gives an accurate solution and is given as

$$P_{i,A} = \frac{R^{requir}_{i,C_1,A} + max(H_{i,C_1,A}(t) - R^{requir}_{i,C_1,A}, 0)}{R^{avail}_{i,C_1,A} + \delta R^{exc}_{i,C_1,C_2,C_3,C_4}} \quad (2)$$

where (a) δ is scaling factor and its value is 1 when $R^{requir} > R^{avail}$ and is 0 otherwise, (b) $R^{requir}_{i,C_1,A}$ is the i^{th} resource requirements by node A in cluster 1, and (c) $R^{exc}_{i,C_1,C_3,C_4}(t)$ is the i^{th} excess resources available in cluster 1 (C_1), 2 (C_2), 3 (C_3) and 4 (C_4) is given as

$$R^{exc}_{i,C_1,C_2,C_3,C_4}(t) = \sum_{c=1}^{C}\sum_{j=1}^{n} \Delta(R^{avail}_{c,j,i}(t) - [R^{pred}_{c,j,i}(t) + R^{alloc}_{c,j,i}(t)])$$

where $\Delta = 1$ when $R^{requir} < R^{avail}$ and $\Delta = 0$ otherwise where C is the total number of cluster in the given network, n is the total number of resources and $H_{i,C_1,A}(t)$ is the i^{th} resource history database of node A in cluster 1.

(P4) Create an action plan
(i) Cluster head node A parallelly indicates resource requirements to all other cluster head nodes in the EI-network, (ii) Cluster head nodes (other than A) check genuinity of resource requirements by referring to the history database, (iii) Node A negotiates resource requirements with nodes E, H and K, and collects negotiated excess resources, and (iv) Node A allocates resources using available and collected resources and is given as

$$R^{alloc}_{i,A}(t) = n(P) * \frac{R^{avail}_{i,A}(t) + \sum_{c=1}^{C}\sum_{i=1}^{n} R^{coll}_{i,c}(t)}{(1+n) * H_{c,i,A}(t)} \quad (3)$$

where $n(P)$ is the predicted demands at node A.

(P5) Implementation of an action plan:
(i) EIT makes cluster head nodes (A, E, H and K) agent to cooperate, coordinate, collaborate and negotiate among them to collect required resources and support for achieving their goals, and (ii) Resources are allocated at cluster head node A.

(P6) Whenever a change in input repeat phases from 1-5: In this phase, if any changes in inputs like occurrence of new events and resources along with the change in external environment condition it makes to repeat phases from 1-5 again.

5 EXAMPLE 2: JOB SHOP SCHEDULING PROBLEM

Problem: A network consists of 15-nodes A, B, C, D, E, F, G, H, I, J, K, L, M, N and P (See Figure 6), resources available in these nodes are 10, 20, 15, 5, 20, 10, 15, 10, 5, 10, 15, 10, 15, 10 and 15, respectively. Jobs available at nodes A, D and E are J_1, J_2 *and* J_3, respectively and these jobs need to be schedule in a specific order. Each of these jobs require a specific amount of resources, that is J_1 requires 30, J_2 requires 20 and J_3 requires 40, before scheduling at their respective nodes. We demonstrate EI-technique

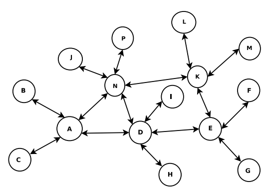


Figure 6: Scenario of a network with 15 nodes.

functioning by solving the above problem by executing following 6-phases:

(P1) Analyze the problem, make clear problem statement and identify all inputs and outputs
(i) Analyze the problem: (1) There are 15 nodes in the network A to I, (2) Resources available at

nodes A, B, C, D, E, F, G, H, I, J, K, L, M, N and P are 10, 20, 15, 5, 20, 10, 15, 10,5, 10, 15, 10, 15, 10 and 15, respectively, (3) There are 3 jobs J_1, J_2 *and* J_3 at nodes A, D and E, respectively, and (4) Job J_1 at node A requires 30, J_2 at node D requires 20 and J_3 at node E requires 40 amount of resources for scheduling at their respective nodes.
(ii) Problem statement: (1) Choose a node (say A) to run the EI technique, (2) Create clusters with the nodes having high degree of connectivity, (3) Collect excess resources from its cluster nodes and from cluster head nodes A, E, D, N and K, (4) Check the genuinity of resource requirements using history, and (5) Allocate required resources at nodes A, E, D, N and K.

(P2) Build an EI-network for the problem by deploying agents at each cluster head using EIT:
(i) Cluster formation: The cluster formation Al-

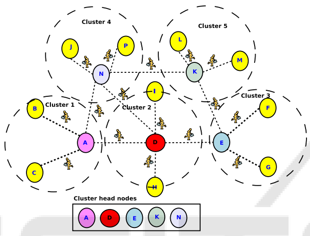


Figure 7: Clusters formation of job scheduling problem.

gorithm 1 is used to create following 5 clusters of the given problem network. EIT forms cluster 1 at node A, cluster 2 at node D, cluster 3 at node E, cluster 4 at node N and cluster 5 at node K as shown in Figure 7.
(ii) EI-network formation: EIT running on node A uses Algorithm 2 to create the EI-network and deploys agents for interaction to share analyzed information of the given problem (given by Phase 1) as shown in Figure 8.

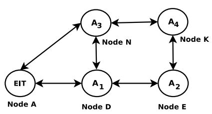


Figure 8: EI-network of the job scheduling problem.

(P3) Estimate practical solution of given problem:
(i) Resource collection for estimation of solution of the problem: (1) EIT running on node A parallelly indicates its resource requirements to cluster head nodes E, D, N and K, (2) Nodes E, D, N and K parallelly estimates their excess resources and they provide to node A, (3) Each agent parallelly refers to the history database to estimate accurate requirement of each node, and (4) Each cluster head node parallelly shares their excess resources to the requested cluster head nodes depending upon history database. (ii) Practical solution of problem of job J_1 by referring to the history database at node A is given as

$$P_{i,J_1,A} = \frac{R^{requir}_{i,J_1,A} + +max(H_{i,j,A}(t) - R^{requir}_{i,J_1,A}, 0)}{\delta R^{exc}_{i,J_1,C_1} + R^{avail}_{i,J_1,A}} \quad (4)$$

where (a) $\delta = 1$ when $R^{req} > R^{avail}$ and $\delta = 0$ otherwise, (b) $R^{requir}_{i,J_1,A}$ is i^{th} resource requirement by node A for scheduling the job 1, (c) $R^{exc}_{C_1}(t)$ is excess resources available in cluster 1 (C_1) is given as

$$R^{exc}_{C_1}(t) = \sum_{j=1}^{n} \Delta(R^{avail}_{j,i}(t) - [R^{pred}_{j,i}(t) + R^{alloc}_{j,i}(t)])$$

where $\Delta = 1$ when $R^{requir} < R^{avail}$ and $\Delta = 0$ otherwise. (d) $H_{i,J_1,A}(t)$ is the history database of node A. (iii) Similarly for the jobs J_2 and J_3 at nodes D and E, respectively are also estimated.

(P4) Create an action plan:
(i) Cluster head nodes D, E, K and N coordinates with each other, and provides the maximum resources to node A, (ii) These cluster head nodes check genuinity of resource requirements by referring to the history database of node A, (iii) Node A negotiates resource requirements with nodes D, E, K and N, and collects negotiated excess resources, (iv) Collected excess resources and available resources at node A are used for resource allocation to schedule the job J_1, and is given as

$$R^{alloc}_{i,A}(t) = n(T_1) * \frac{R^{avail}_{i,A} + \sum_{m=1}^{n} R^{coll}_{i,C_1,C_2,C_3}(t)}{(1+n) * H_{i,A}(t)} \quad (5)$$

where $n(T_1)$ is the number of sub-tasks of job J_1 at node A. (v) For scheduling of the Job J_2, the resources need to be allocated to it is given as

$$R^{alloc}_{i,D}(t) = n(T_2) * \frac{R^{avail}_{i,D} + \sum_{m=1}^{n} R^{coll}_{i,C_1,C_2,C_3}(t)}{(1+n) * H_{i,D}(t)} \quad (6)$$

where $n(T_2)$ is the number of sub-tasks of job J_2 at node D. (vi) For scheduling of the Job J_3, the resources need to be allocated to it is given as

$$R^{alloc}_{i,E}(t) = n(T_3) * \frac{R^{avail}_{i,E} + \sum_{m=1}^{n} R^{coll}_{i,C_1,C_2,C_3}(t)}{(1+n) * H_{i,E}(t)} \quad (7)$$

where $n(T_3)$ is the number of sub-tasks of job J_3 at node E.

(P5) Implementation of action plan:
EIT makes cluster head nodes (A, E, D, N and K) agent to cooperate, coordinate, collaborate and negotiate among them to collect required resources and support for achieving their goals.

- Scheduling of Job J_1 at node A: Cluster head node A parallelly collects excess resources of nodes in its cluster, allocates required resources and schedules the job J_1 to the desired destination.

- Scheduling of Job J_2 at node D: Cluster head node D parallelly collects excess resources of nodes in its cluster, allocates required resources and schedules the job J_2.
- Scheduling of Job J_3 at node E: Cluster head E parallelly interacts with all nodes of cluster 3, collects their excess resources, allocates required resources and schedules the job J_3.

(P6) Whenever changes in input value repeat phases from 1-5:
In this phase, occurrences of any new jobs and resources along with changes in external environment condition makes to repeat phases from 1-5 again.

6 SIMULATION AND RESULTS ANALYSIS

In this section, we describe the scenarios (Figures 3 and 6) and performance assessment of the EIT and MAS by means of conducting the simulation experiments by integrating NS-2 with MobileC agent platform (B. Chen, et. al., 2010, mobilec, 2015). In the subsequent subsections, we describe simulation scenarios, results analysis and comparisons.

6.1 Simulation Scenario

We have considered 13 and 15 number of nodes simulation scenario in NS2. We have measured the performance of EIT and MAS by running the simulation 600 seconds. The MobileC agent platform is installed at each cluster head node in the given network, which deploys static agent. The static agent creates and dispatches mobile agents to each cluster head node in the given problem network. In NS-2, at every nodes resources are stored in a trace file and given to MobileC agent platform, where agents collect resources and provides to the neighborhood cluster head nodes.

In NS-2 we have created cluster (which consists of all neighborhood nodes), in each cluster all nodes interact with each other. Agents are deployed by using MobileC agent platform, which makes to collect and share available, analyzed, collected and historical resource information. The performance evaluation of the proposed system carried out using the simulation parameters as shown Table 2 on a dual-CPU Intel Core i5-2400 at 3.10 GHz Desktop computer with 12-GB RAM running Fedora version 25.

Table 2: Simulation Parameters.

Parameter	Values
Simulation area	2400 $\times$ 2400
Simulation time	600 sec.
Network simulation	NS-2
Agent platform	Mobile-C
Number of mobile agents	50
Number of static agents	9
Number of nodes)	15
Communication technology	IEEE 802.11a
Network transfer rate	2 Mbps
Data processing rate	100 Mbps

6.2 Results Analysis

In this subsection, we discuss performance analysis of MAS and EIT in job scheduling and resource allocation problems.

Throughput is the amount of information received successfully at the cluster head node from the agent, which is measured in bits per second (bps). In the simulation, the amount of information received successfully and lost during migration of agents from one node to other to reach the desired cluster head nodes are determined and generated the data values at varied time intervals is given in Table 3. Simulation result 9 shows the variation of throughput of MAS, SI and EIT at varying simulation time. Throughput shows the significant improvement in the case of EIT as compared to MAS and SI. Figure 10(a) shows data

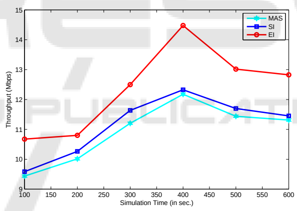


Figure 9: Throughput.

collection time needed by MAS and EIT. When an individual agent is present than both technique will take same amount of time to collect data. When there will be set of agents, then the MAS takes more time for data collection, because in the beginning these agents considers only their own goal and later they collaborate among them. But in the case of EIT, group of neighbor agents from the beginning cooperates and collaborates to achieve the system goal rather than the individual goal. Hence, EIT takes less time compared to MAS for data collection.

Due to the diverging interest of each agent in MAS, stores all resources till completion of job and hence it reserves space. Therefore, the space complexity of MAS is $O(n^2+m)$, where n is total number of agents and m is total number of resources used for execution of problem tasks t. EI technique agents have converging interests and they release resources after completion of each task and hence requires less space compare to MAS. Therefore, the space complexity of EI becomes $O(n+log(m))$. Figure 10(b) shows the memory consumption by each agent during problem solving process in the MAS and EIT. Due to the divergence and resource reservation policy in the MAS consumes more memory than the EIT and is shown in Figure 10(b). Figure 10(c) shows the time complexity for taking dynamic decisions by MAS and EIT. In MAS, each agent takes their own decision for achieving their self-goal, therefore for achieving the complete system goal takes $O(n^2+m+t)$ time and EIT based decision takes $O(n^2+log(m+t))$. Because in the case of EIT all agents collaborate and cooperate depending upon requirements and abilities for achieving whole system goal. Figure 10(d) shows resource allocating using MAS and EIT. It shows clearly that amount of resources available, required, estimated accurate requirements and allocation for 3-different tasks. Figure 10(e) shows execution time of 3-tasks of resource allocation problem. MASs have either diverging information or interests, or both and reserves the resources till completion of resource allocation to a task. In EIT, execution time is less as compared to MAS because tasks are self-organized in such away that there should not be any conflicts for resource allocation during execution of tasks independently. Self-organization realizes global goal of system during random fluctuations of environment or conditions. Figure 10(f) shows time taken to self-organize cluster of agents to achieve global goal of system during any change in environment.

MAS will adapt using current available data and history information for executing set of tasks. EIT makes agent to adapt continuously not only using history database and current available data, and also using neighbor agents current and history information for executing set of tasks. Figure 10(g) sows that MAS takes less adaptability time than EIT.

We have considered 3-jobs and estimated the execution time require to arrange and schedule them in proper order to their desired destination and is shown in Figure 10(h). MAS takes more execution time for job scheduling problem because it re-executes the same job twice when we decompose its problem into sub-problems and execute them independently. In the case of EIT, it decomposes the jobs problem in such away that they should not be any dependency on others. Hence, EIT parallelly executes decomposed problems of jobs and it requires less execution time as compare to MAS.

Table 3: Throughput of EIT, MAS and SI.

Simulation Time (in sec.)	MAS	SI	EIT
100	9.4367	9.5846	10.6734
200	10.0134	10.2643	10.80143
300	11.6038	11.2365	12.49871
400	12.1764	12.3251	14.4768
500	11.4379	11.6982	13.0179
600	11.3127	11.4523	12.8263

7 CONCLUSIONS

We presented the novel EIT for solving problems in uncertain environments and explained the functioning and phases involved. The resource allocation and job-shop scheduling problems have solved using EIT. Analyzed the performances of EIT, MAS and SI by considering performance measures, such as space complexity, time complexity, data collection time, throughput, time taken for self-organization and adaptation. These performance measures are shown that the EIT performance better than the MAS and SI for solving problems in uncertain environments.

REFERENCES

Jacak, Witold, and Karin Pröll. ”Heuristic approach to conflict problem solving in an intelligent multiagent system.” In International Conference on Computer Aided Systems Theory, pp. 772-779. Springer, Berlin, Heidelberg, 2007.

Gundersen, Odd Erik, and Anders Kofod-Petersen. ”Multiagent based problem-solving in a mobile environment.” Norsk Informatikkonferance (2005): 7-18.

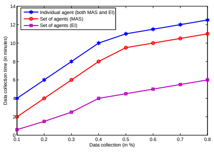
Maryam Rajabi, Teh Noranis Mohd Aris, “The Role of Multi-Agent in Computational Problem Solving Environments”, In the 4th International Conference on Electrical Engineering and Informatics, Procedia Technology, 2013, pp.1103-1109.

Sun, S., Joy, M. & Griffiths, N. The Use of Learning Objects and Learning Styles in a Multi-Agent Education System. Journal of Interactive Learning Research; 18 (3); 2007. p. 381.

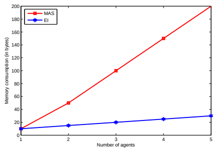
Fenza, G., Furno, D., Loia, V. Hybrid Approach for Context-aware Service Discovery in Health care Domain. Journal of Computer and System Sciences 78; 2012. p. 123.

Pakonen, A., Tommila, T., Pirttioja, T., Selionen, I. OWL based information agent services for process monitoring. Emerging Technologies and Factory Automation (ETFA, IEEE Conference) , Patras, Greece; 2007. p. 9–16.

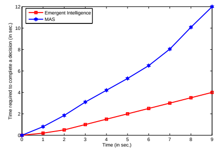
Bell, D. A., Beck, A., Miller, P., Wu, Q. X., Herrera. A.: Video Mining –Learning Patterns of Behavior via an Intelligent Image Analysis System. The 7th International Conference on Intelligent Systems Design and Applications; 2007. p. 460-464.



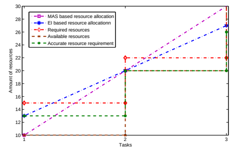
(a) Data collection time needed by MAS and EI



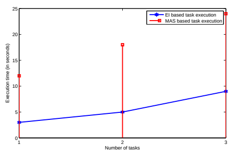
(b) Memory consumption by each agent during solving problem: MAS and EIT



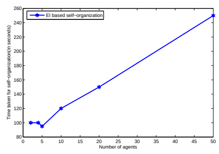
(c) Time complexity of the problem: MAS and EIT



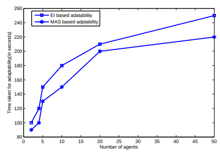
(d) Resource allocation using MAS and EI technique



(e) Taks execution time using MAS and EIT: Resource allocation problem



(f) Time taken for self-organization of clusters of a problem



(g) Adaptability of MAS and EIT for problem solving

(h) Execution time using EI and MAS: Job scheduling problem

Figure 10: Simulation and Analysis Results.

De Weerdt, M., Bos, A., Tonino, H., and Witteveen, C. A resource logic for multi-agent plan merging. Annals of Mathematics and Artificial Intelligence 37 (1–2); 2003. p. 93.

Bala M. Developing Intelligent Agent Applications with JADE and JESS. Springer-Verlag Berlin Heidelberg Knowledge-Based Intelligent Information and Engineering Systems Lecture Notes in Computer Science. 5179; 2008. p. 236-244 .

Kordaki, M. A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilotformative evaluation. Computers & Education 54(1); 2010. p. 69-87.

Aris, T.N. An Agent-Based Conceptual Model For Computational Problem Solving. The International Conference on Informatics and Applications (ICIA2012) – Malaysia; 2012. p. 240-248.

Reynolds, Joshua, Lauren Christopher, Russ Eberhart, and Patrick Shaffer. "Using computational swarm intelligence for real-time asset allocation." In Computational Intelligence for Security and Defense Applications (CISDA), 2015 IEEE Symposium on, pp. 1-5. IEEE, 2015.

Johansson, F.; Falkman, G., "Real time allocation of defensive resources to rockets, artillery, and mortars," Information Fusion (FUSION), 2010 13th Conference on, vol., no., pp.1-8.

Kalyan V., and Lisa Ann Osadciw. "Optimal scheduling in sensor networks using swarm intelligence." In Proceedings of 38th Annual Conference on Information Systems and Sciences, pp. 17-19. Princeton University Princeton, New Jersey, 2004.

Guido Guizzi1, Francesco Gargiulo1, Liberatina Carmela Santillo1 , and Hamido Fujita, " Swarm Intelligence in Evacuation Problems: a Review".

Rzevski, G. and Skobelev, P, "Emergent intelligence in large scale multi-agent systems". International Journal of Education and Information Technology, 2007, Vol. 1(2), pp.64-71.

W. D. Hillis, 'Intelligence as an emergent behavior; or, the songs of eden', Daedalus (1988), pp. 175–189.

T. Wolf, 'Analyzing and engineering self-organizing emergent applications', 2007.

Z. Li, C. H. Sim, M. Y. H. Low, 'A survey of emergent behavior and its impacts in agent-based systems', in: Industrial Informatics, IEEE International Conference on, 2006, pp.1295-1300.

P. J. Angeline, 'Genetic programming and emergent intelligence', Advances in genetic programming 1994, Vol.1, pp.75-98.

B. Chen, H. H. Cheng, 'A review of the applications of agent technology in traffic and transportation systems', Intelligent Transportation Systems, IEEE Transactions on 11 (2) (2010) pp. 485–497.

'A multi-agent platform for mobile c/c++ agents', http://www.mobilec.org/overview.php, accessed on June, 2015.