

# GA based Optimal Itinerary Planning for Multiple Mobile Agents in Wireless Sensor Networks

A N Rani<sup>1</sup>, Lalit Dole<sup>2</sup>PG Student, Dept. of CSE, G H Raisoni College of Engineering, Nagpur, India<sup>1</sup>Assistant Professor, Dept. of CSE, G H Raisoni College of Engineering, Nagpur, India<sup>2</sup>

**ABSTRACT:** Using multiple mobile agents (MA) in WSNs significantly improves the network efficiency and scalability of its data fusion applications. In multiple MA based computing, MA Itinerary Planning plays an important role and a badly designed Itinerary Plan severely affects the network and application performance. There are several approaches and schemes proposed and presented for the mobile agent's itinerary planning in resource constrained WSNs. Genetic Algorithm based optimization schemes are explored very little for efficient MIP (Multiple Mobile Agents Itinerary Planning) solutions. Here we proposed a two phase design model (GA-OIP-MMA) with Genetic Algorithm for global optimal itinerary planning of MAs followed by local dynamic in-network route adaptation by individual MAs to design optimal MIP solutions for WSNs.

**Keywords:** mobile agent, wireless sensor networks, genetic algorithm, optimal itinerary planning, in-network route adaptation

## I. INTRODUCTION

Wireless sensor networks (WSNs) have attracted much attention in the research community over the last few years, driven by a wealth of theoretical and practical challenges and an increasing number of practical world applications. "Single Network, Multiple Applications" is a promising trend in the deployment of WSNs, because of the high cost and complexity of deploying huge no. of sensor nodes over a wide geographical area and the application-specific tasking of WSN. Such a network trend requires sensor nodes to have various capabilities to handle multiple applications. However, it is infeasible to store the programs required to run every possible application in the local memory of embedded sensors, as these devices generally have tight memory constraints. The use of mobile agents to dynamically deploy new applications in WSNs is proving to be an effective method to address this challenge.

Xu et al. [1] proposed a mobile agent computing paradigm that is suitable for collaborative processing in WSNs. Fig.1 shows Mobile agent based computing paradigm. In this paradigm, an application-specific MA, which is special software containing executable codes to be run in sensor nodes [2], is dispatched to the surveillance area. The MA traverses the area of interest to extract, collect and fuse sensory data from source nodes, and then carry the result back to the sink. The goal of the mobile agent computing is to migrate to a group of sensor nodes in a particular sequence, maximizing the information extracted while keeping resource usage to a minimum. This approach may result in a significant improvement on the network efficiency, such as information fusion, load balancing, and contention avoidance in WSNs. The benefits of the mobile agent based computing in support of collaborative processing in WSNs have been thoroughly studied in [1], [3].

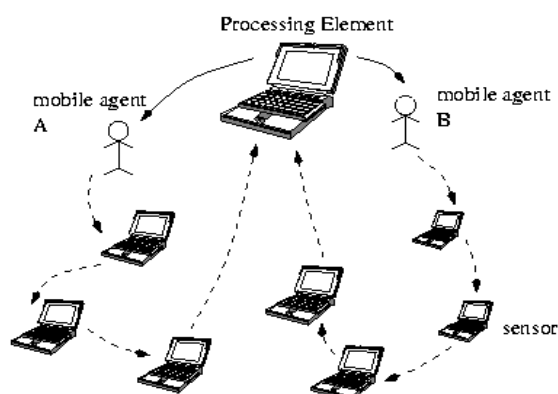


Fig. 1: MA based computing paradigm

The Fig.2 shows the data fusion patterns. The MA deployment in WSNs typically has the following design issues 1) architecture, 2) itinerary planning, 3) middleware system design and 4) agent cooperation. Among these issues, itinerary planning determines the order of source nodes to be visited by an MA, which has a significant impact on the system performance. Thus, how to find out an optimal itinerary for the MA to visit the source nodes of interest is critical.

MA-related research has become more interesting after Chen et al. [4] presented the potential benefit of using multiple MAs, such as the scalability to deal with large scale WSNs and the optimisation of task load by partitioning area among the MAs. The multiple MAs based WSN is an evolution version of the single MA-

based WSN. Instead of dispatching a single MA to the network, it enables two or more MAs to execute the cooperative pattern. The whole data fusion application is then fragmented and completed by each MA respectively. Compared to the single MA itinerary planning (SIP) problem, dispatching multiple MAs require the system to consider three key elements: (i) the quantity of MAs, (ii) grouping of source nodes for each MA, and (iii) itinerary planning of each MA. These elements are correlated to each other; thus, the computational complexity of the solution is dramatically increased. We now refer these problems collectively a Multiple MA Itinerary Planning (MIP) problem. In order to optimize the Itinerary planning schemes to sustain to rapid network inconsistencies of WSNs, they have to be treated in two phases: 1) to compute global optimal itinerary plan 2) to provide on fly in-network route adaptation techniques for individual MAs. According to this methodology, itinerary planning can be categorized as follows:

task simultaneously. These multiple MAs work in a given

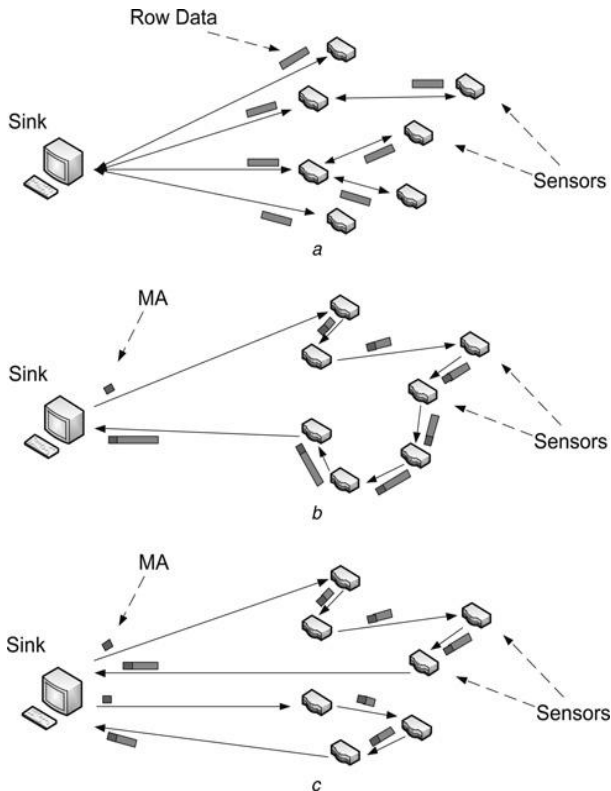


Fig. 2: Comparison of data fusion patterns in WSNs

a. Traditional b. One MA c. Multiple MAs

#### A. Itinerary planning categories

- 1) *Static planning*: In static planning, the agent itinerary is totally determined by the sink node before the agent is dispatched.
- 2) *Dynamic planning*: In dynamic planning, the mobile agent autonomously determines the source nodes to be visited and the route of migration according to the current network status.
- 3) *Hybrid planning*: In hybrid planning, the set of source nodes to be visited is decided by the sink, and the source- visiting sequence is determined dynamically by the mobile agent.

Though several GA based single mobile agent itinerary planning schemes are presented previously, the first GA based itinerary planning scheme for multiple MAs in WSNs (GA\_MIP) is proposed by Chen et al. [5]. However, ever changing network topological behaviour of WSNs and the possibility of centrally computed MA routes to outdate while actual realization of route plans are major issues still to be addressed even in GA based MIP solutions. Our research work is a step forward to attempt these issues which suggests the local in-network route adaptation technique by individual MAs while following GA computed global itinerary plan.

The remainder of the paper is organised as follows: Section-II introduces the related work in this domain, and the discussions on individual schemes are mentioned in Section-III. In Section-IV we will give our proposed scheme and discuss the proposed GA and In-Network Route Adaptation technique in detail. The conclusion and research directions are made in Section-V.

## II. RELATED WORK

In this section, we survey four representative approaches of the state of art on the MIP topic. In contrast to SIP approaches, MIP solutions have to do Source grouping i.e. to set up some groups for source nodes, which will be visited by corresponding MAs. The whole itinerary planning comprises of Source Grouping and MA routing i.e. the process of determining an itinerary for one particular MA to visit the designated group of source nodes. The basic approaches that depict the state-of-the-art MIP solutions are:

#### A. Centre location-based multiple MA itinerary planning

Chen et al. [6] first proposes a centre location-based MIP solution (CL-MIP). The main idea of CL-MIP is to deem the MIP solution as an iterative version of an SIP, which consists of four steps:

- 1) *Selection of VCL for an MA*: CL-MIP assumes that the source nodes with higher geographical relevance should be in the same group. Consequently, the agent's VCL should be selected at the centre of an area with a high source node density. In CL-MIP, the calculation of VCL is based on a gravity algorithm, which imitates the gravity accumulation in natural world.

- 2) *Source grouping*: After selecting the VCL, the next step is to specify the visiting area, typically a circle/oval centred at the VCL with an assigned radius. All the source nodes in the group will be included in the visiting list of the corresponding MA.
- 3) *MA routing*: Within one group of source nodes, the itinerary of the MA is planned the same as the SIP problem, whereby existing SIP solutions can be applied, such as local-closest-first (LCF), global-closest-first (GCF) [7], genetic approach [8], mobile agent-based directed diffusion [9], itinerary-energy-minimum-for-first-source selection (IEMF) algorithm and the itinerary-energy minimum- algorithm [10] and so on. In CL-MIP solution, IEMF is adopted because of its higher efficiency and lower computational complexity.
- 4) *SIP iteration*: If there are uncovered source nodes, the next VCL will be calculated based on the remaining set of source nodes. The previous process will repeat until all the source nodes have been assigned to an MA.

#### B. Direction-based multiple MA itinerary planning

The proposal in [11] induces an alternative perspective on the step of source grouping, the angle gap-based MIP (AG-MIP). The main idea of AG-MIP is based on the criteria that most of the information-relevant source nodes are located in the same direction from the view of the sink. Therefore the source grouping method is direction oriented. The proposed scheme connects the sink and all source nodes with beelines, and the angle gaps between the beelines become a critical factor to describe the relevant degree among the source nodes. There is a VCL, which is determined by the two nodes with minimal angle gap. AGMIP does not utilise shape of circle for source grouping. Instead, the nodes within a particular angle gap threshold  $\alpha$  around one central location should be included in the same group. This approach may result in some isolated source nodes that are located near the group. They will finally be considered as a special group after several iterations.

#### C. Tree-based multiple MAs itinerary planning

The proposal in [12] models the MIP problems as a totally connected graph (TCG). In the TCG, vertices are the sensor nodes in the network, and the weight of an edge is derived from the estimated hops between the two end nodes of the edge. The authors calculate an MST with the TCG, and suggest that all source nodes in a particular sub tree in the MST should be considered as a group. Furthermore, the authors introduce a balancing factor while calculating the weights in the TCG, to effect the forming of a balanced minimum spanning tree (BST). The balancing factor achieves flexible control on the trade-off between energy cost and task duration. By adjusting balance factor, the quality-of-service (QoS) requirements in the term of delay can be satisfied for a large range of applications while trying to reduce the energy cost to a minimal level. This approach algorithm is called BST-based MIP (BST-MIP). The main contribution of BST-MIP is that it analyses the critical impact of the geographic positions of source nodes on the source grouping, and then evaluates the MST approach. There are some other tree-based methods for solving MIP problems, including NOID [13], CBID [14] and TBID [15].

NOID proposes a heuristic algorithm to suggest an appropriate number of MAs that minimises the overall data fusion cost, and to construct near-optimal itineraries by adopting the constrained minimum spanning tree problem. In NOID, the geographical distant is considered as the main factor for the cost weight between two source nodes, which is different from hop-account-oriented BST-MIP. In order to effectively include the nodes that are far away from the centre, NOID uses a trade-off function for balancing. From the formatted tree, by migration and connection operations, near-optimal itineraries are planned.

CBID applies a tree structure with branches for planning the itineraries. While deriving the tree structure, CBID follows a greedy approach to include the node, which will make the total cost minimal. Furthermore, one MA will be cloned as two slave MAs at certain branch joint to visit individual tree branches for collecting data until the leaf nodes. Finally, slave MAs will go reversely and forward collected data to the MA at the joint then to the sink.

Another tree-based heuristic method, TBID, consists of two procedures: partitioning the whole WSN coverage around the sink into concentric zones, and building the MA itineraries from inner zones to outer ones. TBID also utilises greedy-like approach to always select the nearest node to form the binary tree.

#### D. Genetic algorithm-based multiple MAs itinerary planning

The genetic algorithm-based multi-agent itinerary planning (GA-MIP) algorithm is proposed in [5]. The authors substantiate the proposed GA approach by encoding how many MAs are dispatched and which sensors are covered by individual MAs. The main features of this work can be summarised as follows:

† GA-MIP first proposes a novel two-level coding method for GA to solve MIP problem. The coding represents the source grouping and MA routing solution within a unique gene.

† In each evolutionary iteration, the crossover and mutation operators are always the essential elements. GA-MIP provides a set of novel crossover and mutation operators according to the two-level coding.

† Fitness function, sometimes called select operator, sets up a criterion for selecting the better genes to survive. It is the most critical issue in GA design, which controls the direction of convergence. GA-MIP constructs a performance-aware fitness function that can derive a near optimal solution for the MIP problem.

### III. DISCUSSION

The contributions of CL-MIP include that it first proposes a four-step generic framework for solving the MIP problem, which reutilises the existing SIP solutions. The proposed gravity algorithm precisely describes the density centre of the groups of source nodes, which is the basis of source grouping. However, there are still unsolved issues as follows:

† The determination of the sets of source nodes is only based on geographical information. The load balancing among MAs is not considered.

† The efficiency of source grouping by a circle is questionable. Circle-shaped source grouping is not a generic solution considering those scenarios with irregularly distributed nodes. In addition, the radius of the source nodes grouping will also strongly effect the performance of CL-MIP, although its optimal value is still not measured or analysed explicitly.

† The work merely transforms an MIP problem to a repetition of SIP problems by a four-step framework. This greedy approach may lead to a substantially sub-optimal MIP solution.

The benefit of AG-MIP is that by using the angle gap to split the whole network into non- overlap sectors, the contention and interferences among different MA flows will be potentially reduced. However, the AG-MIP is still constrained within the four-step framework proposed in CL-MIP. The same as CL-MIP, the angle-oriented grouping method is an optimisation to specific applications, which cannot be counted as a generic solution.

The source nodes grouping algorithm in BSTMIP becomes more generic, which can be applied to a variety of sensor nodes deployments. Same as CL-MIP and AG-MIP, this scheme uses the four step framework proposed in CL-MIP.

NOID outperforms LCF and GCF, but it suffers from low working speed and high computational complexity. CBID performs much better with respect to metrics of itinerary length and response time. However, CBID faces to scalability problem: as the size of the WSN increases, more branches will be created, which will degrade the performance significantly because of the interference. Also, the reversed trips of slave MAs consume unnecessary cost (for about one more time). TBID generates low itineraries for MAs, but it has similar shortages as CBID, that is, the doubly consumed energy in the reverse routes, and the interference among the huge amount of branches. Furthermore, all tree-based schemes lack dynamical recovery for transmission failures.

Although extensive simulations have been carried out to show the better performance of GA-MIP in terms of the delay and energy consumption, it doesn't provides in-network route adaptation technique which is necessary to address the issues like outdated local sub-optimal route plan realization by individual MAs, in-tolerant network node failure inconsistencies and lack of QoS support for real time applications etc in WSNs.

### IV. PROPOSED WORK

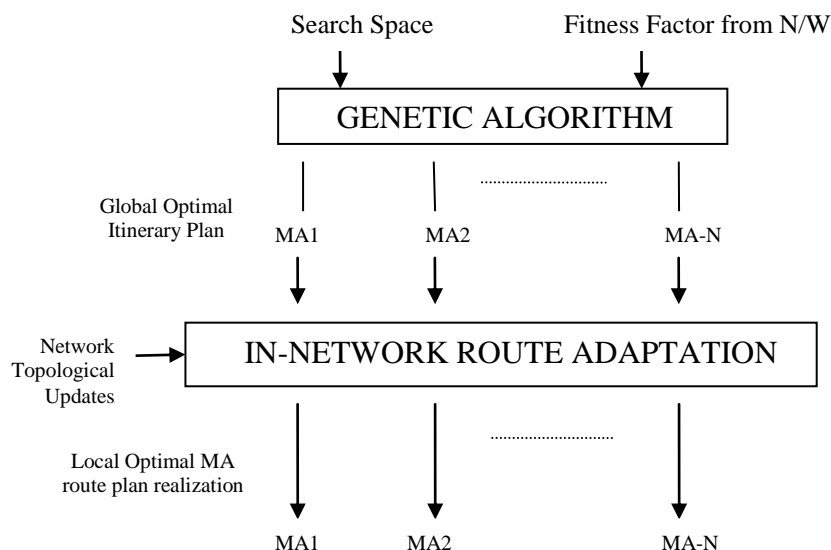


Fig 3: GA-OIP-MMA model

In this section, we introduce a two phase design model in Fig.3 (GA-OIP-MMA) to produce optimal MIP solutions with Genetic Algorithm for global optimal itinerary planning of MAs followed by local in-network route adaptation by individual MAs. Our proposed procedure uses a genetic algorithm to determine the number of agents and their itineraries, followed by techniques for in-network adaptation to unpredictable situations like node failure. The possible implementations of GA and In-Network Route Adaptation techniques for the proposed design model are discussed in detail.

Genetic Algorithm [16] (GA) is adaptive heuristic search algorithm based on the evolutionary theory of genetic and natural selection, which will produce the fittest survival. In a GA system, each solution to the problem was described as

an individual instance with genetic information in the nature. The solutions produce children that inherit mixture characteristics from their parents. An opportunistic mutation may happen to generate new individuals. Through the evaluation by a fitness function, the better individuals could survive. As time goes on, the survivals contain the excellent genes which represent the better solutions to the problem. Kuzuo Sugihara [17], R Ramakrishnan [18] proposed different encoding techniques to improve GA performance without depending on the network feasible nodes. Here we can use GA-MIP algorithm proposed by Chen et al. [5] as base GA algorithm in first phase of the model. They proposed a two level encoding scheme for GA implementation.

E. Encoding

They encode the Source Node Sequence and the Source Node Group into numbers as the genes for genetic evolution. Fig 4 shows the encoding example.

- 1) *Code of Source Node Sequence*: The Code of Source Node Sequence is an array of the source nodes' identifiers, which implies the order for a MA to visit the corresponding source nodes. One MA can have only one final Code of Source Node Sequence for its visiting itinerary.
- 2) *Code of Source Node Group*: The Code of Source Node Group is an array of integer numbers, in which each number indicates the number of source nodes that are allocated in the corresponding Source Node Group.

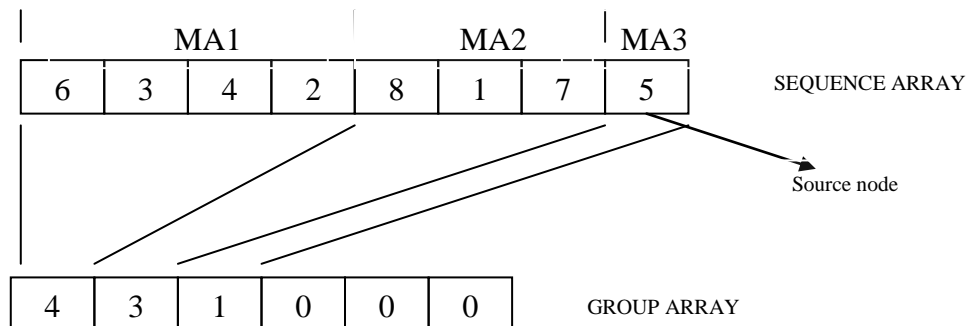


Fig. 4: The Encoding Example with 3 MAs and 8 Source Nodes.

F. Operators

As conventional operators for GA, they implemented the *Crossover*, *Mutation* and *Selection* operators.

- 1) *Crossover Operator*: Crossover operator is an essential operator in GA. It imitates the way of natural biological evolution. There are several crossover schemes have been proposed, such as one-point crossover and multi-point crossover. In their approach, crossover is only applied between the sequence arrays attach with the same group array. The two sequence segments which are located in the same positions in the two sequence arrays will be exchanged each other and duplicates are eliminated in the crossover procedure.

In their approach [5], crossover is only applied between the *sequence arrays* attach with the same *group array* as shown. They randomly select a non-zero element (denoted by m) in *group array*. Corresponding to its associated *sequence array*, m is mapping to a sequence segment. The two sequence segments which are located in the same positions in the two *sequence arrays* will be exchanged each other in crossover procedure.

Given one parent as follows:

Second parent as follows:

- *Sequence array 1* : 2-3-5-4-/1-6-8-/7-9      and      • *Sequence array 2* : 1-2-5-3-/6-7-9-/8-4 and
- *Group array 1* : 4-/3-/2      and      • *Group array 2* : 4-/3-/2.

where the crossover portions are enclosed by two “/” signs, the crossover operator produces the following child:

- *Sequence array 1* : 2-3-5-4-6-7-9-1-8      and      *Group array 1* : 4-/3-/2
- *Sequence array 2* : 1-2-5-3-6-8-7-9-4      and      *Group array 2* : 4-/3-/2

- 2) *Mutation Operator*: The mutation operator is used to keep the variety of the genes so that the discovery of new solutions is possible. In their approach, both sequence array mutation and group array mutation are implemented. For the sequence array, the operator randomly selects two elements in the array and switches their positions. For the group array, similar operation is performed that a non-zero element with larger value will be decreased by 1 and a smaller element will be increased by 1. The elements in group array is in numerical order, thus a sort for the group array should be performed after the random mutation. It requires a sorting to produce the final group array. Otherwise, the solution repetition cannot be avoided.

Before mutation :

Group array 1 : 3-3-1-1-/1-/0-0-0-0

Sequence array 1 : 2-3-5-4-/1-/6-8-/7-9

After Mutation :

Sequence array 1 : 2-3-1-4--5-6-8--7-9

Group array 1 : 3-3-2-1-0-0-0-0-0

3) *Selection Operator*: For the purpose of selecting the better genes to survive for the evolution at the next round, they propose a fitness function to evaluate the performance of a gene. As to obtain an efficient solution, they employ the same criterion to estimate the communication cost of an itinerary, as in [19].

During the GA evolution, the  $i^{\text{th}}$  gene corresponds to a planned itinerary, whose cost is denoted by  $E_1(i)$ ,  $i = 1...k$ , where  $k$  is the size of the search space. After the crossover and mutations, the number of genes is increased to  $(1+\alpha)k^5$ . In the implementation, the select operator selects first  $k$  genes according to their better  $E_1$  s.

However, for better performance of GA, we recommend the following operations also to be implemented for the mentioned reasons.

4) *Inversion Operator*: The crossover operator has a wide span of activity in the feasible solution space, while, due to the pressure of genetic selection and natural elimination, the mutation operation also has difficulty in carrying out local search activity (especially at the late stage of computation when the genetic algorithm tends to converge). The inversion operation is a special form of mutation which is designed to carry out a reordering operation and improve the local search for the genetic algorithm. At any time, two inversion points are selected randomly to determine the inversion portion of the individual. The inversion operation is executed by reversing the order of the inversion portion of the original individual. Given one parent as follows:

- Sequence array 1 : 0-5-7-1/-2-8-9-/-6-3 and
- Group array 1 : 4-/-3-/-2.

where the inversion portions are enclosed by two “/” signs, the inversion operator produces the following child:

- Sequence array 1 : 0-5-7-1/-9-8-2-/-6-3 and
- Group array 1 : 4-/-3-/-2.

5) *Replacement Policy*: The last step in GA is to create a new population using an appropriate replacement policy. From crossover and gene mutation operations, we can get  $k$  offspring chromosomes. We utilize the best  $k$  chromosomes (i.e., the chromosome with the highest fitness score) to generate a new population. However, when creating new population by crossover, there is a chance to lose the fittest chromosome. Therefore, an elitism strategy, in which the best chromosome (or a few best chromosomes) is retained in the next generation’s population, is used to avoid losing the best candidates. The GA stops and returns the current fittest solution until the number of total generations  $K$  is reached or the best fitness score does not change for continuous  $L$  generations.

#### G. Parameter selection for GA

Some parameters should be set up for the genetic algorithm, such as the number of iteration, the size of search space, the ratio of crossover and mutations for *sequence array* and *group array*. Using the larger number of iteration times and the bigger search space, the lower estimated cost can be obtained. However, the more iteration times and the bigger search space will cause higher computational complexity. Thus, there is a tradeoff between the performance and the computational complexity. The table-1 shows the GA-MIP [5] simulated parameters.

TABLE I SIMULATION PARAMETERS FOR GA-MIP

GA Iteration Times	1500
GA Search Spaces	300
Sequence Crossover Ratio	0.9
Sequence Mutation Ratio	0.4
Grouping Mutation Ratio	0.2

Figures 5 & 6 shows the visualization of GA-MIP with different no of MAs.

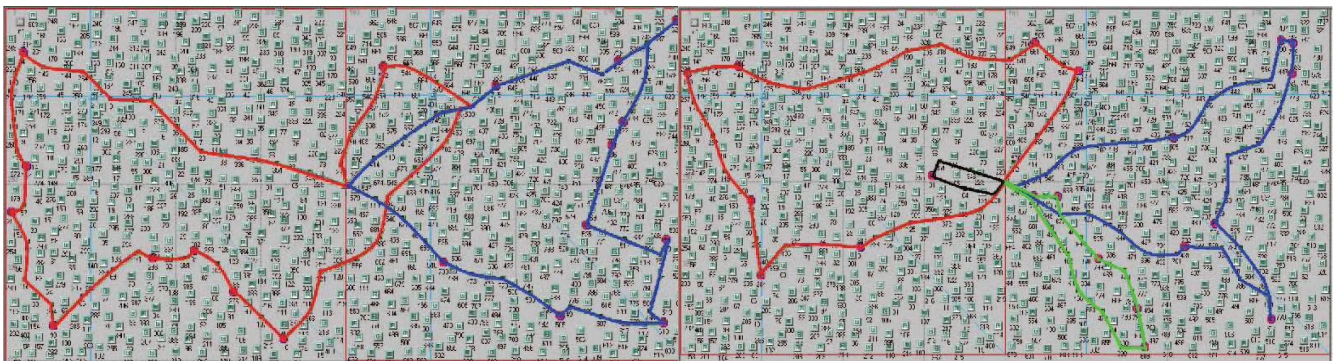


Fig. 4: The GA-MIP with 3 MAs

Fig. 5: The GA-MIP with 4 MAs

#### H. In-Network route adaptation

1) *Need for In-Network route adaptation:* The sensor network's topology may change due to nodes running out of energy, desynchronizing, or being affected by external forces (e.g., being moved by the wind). Continuously monitoring the network topology is energy consuming. The topology of WSNs changes over time. As a result, the network view used by GA may be outdated. Determining the network topology at the collection point is time and energy consuming, which makes it infeasible to continuously collect the network topology. Thus, the computed routes by GA, given to the mobile agents may not always be completely realizable by individual MAs while routing with in local network topology.

Moreover applications such as target tracking, response systems to application specific events like fire scenario, mobile sensors might require a dynamic routing to follow a moving target. To cope with these inconsistencies, when following its route, every mobile agent makes its own local decisions to adapt its route during runtime.

Thus, in order to solve the formulated minimization problem we propose a model containing two phases: an initial route planning phase done by a genetic algorithm and an in-network route adaptation phase that is done by each agent autonomously after it has been injected into network.

2) *In-Network route adaptation Technique:* If an agent cannot successfully migrate from the  $i$ th node of its proposed route, to the  $(i + 1)$ th, it incrementally tries to migrate to the  $j$ th node such that  $j > (i + 1)$ . Given a typically dense WSN, at least one of the nodes that follow the  $(i+1)$ th node on the agent itinerary may eventually be in the  $i$ th node radio's range. Migrating to that node will allow the agent to resume its exploration. Figure 4.7 presents an example. Furthermore, if the underlying mobile agent middleware supports multi-hop routing (e.g., geographic routing) such as Agilla [20], this solution is even more robust since an agent can migrate to any node in the WSN regardless of distance.

Although mobile agents currently only adapt to individual network failures, they could also adapt to application specific events. For example, on a fire scenario, agents could adapt their routes either to advance towards the fire, or to avoid landing on nodes that are likely to get burned. Moreover, mobile agents could adapt their routes in the case the collection point moved. This is an extension of our work we are currently considering.

#### V. CONCLUSION

Multiple MAs in WSNs greatly improves network performance and enhance the application scalability. In this paper we discussed the issues related to deployment of multiple MAs into WSNs and the state-of-the art MIP solutions are reviewed. Many of these MIP solutions are multi-step simple iterative versions of SIP (Single MA Itinerary Planning) solutions and can't be adaptive to large scale WSNs. Some MIP solutions (GA-MIP etc) are targeted for idle MIP solution, but doesn't address local network inconsistencies, network node failures, delay QoS requirements for real world applications and MA route adaptation in case of application specific events and mobility of data collection point etc. Hence we proposed a MIP solution (GA-OIP-MMA) that uses Genetic Algorithm for global optimal itinerary planning of MAs followed by local in-network route adaptation by individual MAs. The proposed model is adaptive to real world dynamic environment of WSNs for quick convergence and supports In-Network Route Adaptation of MAs [21] for QoS demanding WSN applications.

In view of various constraints of the WSN environment and QoS requirements, several issues of multiple MAs-based WSNs still have to be addressed in MIP solutions. We herein list those potential research issues as follows:

- † Exploration of ACO algorithms to design optimal MIP solutions for WSNs.
- † Delay Synchronization among MAs in real time applications [22] that demand low round trip delays.
- † Itineraries merging of MAs and cloning of MAs at certain locations for efficiency improvement
- † Node mobility support to implement in Mobile Sensor Networks [23] [24]
- † Reliable and failure tolerant to MA routing failures of nodes through itinerary backup and recovery mechanisms
- † Online itinerary planning of MAs to operate in Internetwork environment.

#### ACKNOWLEDGMENT

We take this opportunity to express our deepest gratitude and appreciation to all those who have helped us directly or indirectly towards the successful completion of this paper.

#### REFERENCES

- [1]. Y. Xu, H. Qi, "Distributed computing paradigms for collaborative signal and information processing in sensor networks", International Journal of Parallel and Distributed Computing, vol.64, no.8, pp.945-959, 2004.
- [2]. Gonzalez, S., Chen, M., Leung, V.C.M, "Applications of mobile agents in wireless networks and mobile computing", Elsevier, vol. 82, pp. 113-163, 2011.
- [3]. Y. Xu, H. Qi, and P. T. Kuruganti, "Mobile-agent-based computing model for collaborative processing in sensor networks", IEEE Global Communications Conference (GLOBECOM) vol. 6, 2003.
- [4]. Chen, M., Gonzalez, S., Zhang, Y., Leung, V.C. "Multi-agent itinerary planning for sensor networks", Proc. IEEE Int. Conf. Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine 2009), 2009.

- [5]. Cai,W., Chen,M., Hara,T., Shu,L., “GA-MIP: genetic algorithm based multiple mobile agents itinerary planning in wireless sensor network,” Proc. Fifth Int. Wireless Internet Conf. (WICON), 2010
- [6]. Chen, M., Gonzalez, S., Zhang, Y., Leung, V.C. “Clone based Multi-agent itinerary planning for sensor networks,” Proc. IEEE Int. Conf. Communications, 2009.
- [7]. Qi, H., Wang, F.: “Optimal itinerary analysis for mobile agents in AdHoc wireless sensor networks”, Proc. IEEE Int. Conf. Communications (ICC 2001), 2001.
- [8]. Wu, Q., Rao, N.S.V., Barhen, J., et al.: “On computing mobile agent routes for data fusion in distributed sensor networks,” IEEE Trans. Data Eng., vol.16, pp.740–753, 2004.
- [9]. Chen, M., Kwon, T., Yuan, Y., Choi, Y., Leung, “Mobile agent based directed diffusion in wireless sensor networks”, EURASIP J. Appl. Signal Process., vol.1, pp.219–242, 2007.
- [10]. Chen, M., Leung, V., Mao, S., Kwon, T., Li, M.: “Energy-efficient itinerary planning for mobile agents in wireless sensor networks”, Proc. IEEE Int. Conf. Comm. (ICC 2009), 2009.
- [11]. Chen, M., Gonzalez, S., Leung, V., “Directional source grouping for multi-agent itinerary planning in wireless sensor networks”, Proc. Int. Conf. ICT Convergence (ICTC), 2010.
- [12]. Chen, M., Cai, W., Gonzalez, S., Leung, V.C., “Balanced itinerary planning for multiple mobile agents in wireless sensor networks,” Proc.Second Int. Conf. Ad Hoc Networks (ADHOCNETS2010), 2010.
- [13]. Gavalas,D., Mpitzopoulos, A., Pantziou, G., Konstantopoulos, C. “An approach for near-optimal distributed data fusion in wireless sensor network”, Springer Wirel. Netw., vol.16, no.5, pp.1407–1425, 2009.
- [14]. Mpitzopoulos, A., Gavalas, D., Konstantopoulos, C., Pantziou, G.: “CBID: a scalable method for distributed data aggregation in WSNs”, Hindawi Int. J. Distrib. Sens. Netw., 2010.
- [15]. Konstantopoulos, C., Mpitzopoulos, A., Gavalas, D., Pantziou, G.: “Effective determination of mobile agent itineraries for data aggregation on sensor networks”, IEEE Trans. Knowl. Data Eng., vol.22, no.12, pp. 1679–1693, 2010.
- [16]. M. Mitchell, “An Introduction to Genetic Algorithms”, MIT Press, 1998.
- [17]. K.Sugihara, Smith,J, “GA for adaptive motion planning of an autonomous mobile robot”, Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '97), pp. 138-143, 1997.
- [18]. R.Ramakrishnan, Zein Sabatto S, "Multiple Path Planning for a Group of Mobile Robot in a 2D Environment using Genetic Algorithms", IEEE Transactions of Robotics and Systems, pp. 65-71, 2001.
- [19]. M. Chen, V. Leung, S. Mao, T. Kwon, and M. Li, “Energy-Efficient itinerary planning for mobile agents in wireless sensor networks”, Proc.IEEE International Conference on Communications(ICC 2009), pp. 1–5, 2009.
- [20]. C.-L. Fok, G.-C. Roman, and C. Lu, “Rapid development and flexible deployment of adaptive wireless sensor network applications”, in International Conference on Distributed Computing Systems (ICDCS'05), 2005.
- [21]. Daniel Massaguer, ChienLiang Fokz, Nalini Venkatasubramanian, GruiaCatalin Romanz, Chenyang Luz.V “Exploring Sensor Networks using Mobile Agents”, AMAS'06, 2006.
- [22]. Stoina, P., Quintero, A., Cobo, L., “Transport protocol for real-time communication in wireless sensor networks,” Int. J. Ad Hoc Ubiquit. Comput, vol. 5, no.2, pp.86–102, .. 2010.
- [23]. Huang, C.J.,Wang, Y.W., Lin, C.F., Hu, K.W., Chang, T.Y.: “Direction sensitive routing protocols for underwater wireless sensor networks,” J. Internet Technol., vol.11, no.5, pp.721–730, 2010.
- [24]. Coronato, A., Pietro, G.D., Chao, H.C., Park, J.H.: “A framework for pervasive applications applied to intra- vehicular sensor network applications,” Mobile Netw. Appl., vol.15, no.1, pp.137–147, 2010.



**A N RANI** received her BE degree in Computer Technology from Nagpur University (India) and pursuing ME in Mobile Technology from G H Raisonni College of Engineering (India). Her research interests include distributed database systems in advanced Adhoc Communication Networks, Wireless Sensor Networks, IP mobility and Mobile Computing.