# Localized Algorithms for Channels Assignment in Ad-Hoc Wireless Sensor Networks

Farinaz Koushanfar
EE249 Final Project Report
{farinaz@eecs.Berkeley.edu}

## ABSTRACT

We have developed the first systematic approach for the design of localized algorithms in wireless embedded ad-hoc sensor networks. The approach has three modular phases: information gathering, system structuring, and optimization mechanisms. The phases can be organized in a number of ways, depending on the underlying operating system and media access control mechanisms.

As a driver example for testing the approach, we addressed the channel assignment - edge-coloring problem. We first developed centralized maximally constrained - minimally constraining algorithms. After that we used this algorithm as the optimization engine in TDMA and Aloha media access protocol scenarios to develop a family of localized algorithms. Comprehensive evaluation of the algorithms on variety of benchmarks clearly indicates the effectiveness of the proposed approach and algorithms.

## 1. INTRODUCTION

### 1.1 Motivation and objectives

Wireless Embedded Sensor Networks (WESN) are complex distributed systems deployed in an ad-hoc manner. WESN consists of a number of sensor nodes, each with a significant amount of computation, communication, storage, and sensing resources. WESN provide an interface between the Internet and/or research lab and the physical, chemical, and biological world. They provide the potential to revolutionize the science world and economic models, but also pose numerous difficult and conceptually technical challenges. Among them, there is a wide consensus that localized algorithms are of the highest priority.

Localized algorithms are algorithms implemented on sensor networks in such a way that only a limited number of nodes with a limited amount of communication are conducted between the nodes during their execution. Localized algorithms operate with incomplete information, noisy data, and often under very strict communication and energy constraints. Additional design considerations include fault-tolerance, privacy and security requirements, load balancing, and unreliable synchronization. Therefore, there is a great opportunity in developing localized algorithms while there are still a number of complex new technical problems. Until now, there has been no systematic way to develop localized algorithms. As a matter of fact, only preliminary studies that marginalize operating systems and communication problems have been conducted and only on problems that are mainly related to local optimization.

Our primary goal is to develop a systematic way of designing localized algorithms. We introduce the approach that has three phases: information gathering, system structuring, and optimization mechanisms. In the first phase nodes acquire data for executing the algorithm, in the second they organize themselves according to the communication needs of the conducted computation and in the final stage they execute the computation and convey their results to a subset of other nodes. The key observation is that these three phases often have to be interleaved for optimal results. We explain each of the phases and several ways how they can be organized.

In order to make the effort tangible, we focus our attention on the channel assignment problem that can be modeled as the NP hard edge-coloring problem. We present several new versions of the formulation that model the need for efficient communication in WESN.

The optimization basis for the edge-coloring algorithm is a generalized maximally constrained - minimally constraining paradigm. In addition to the centralized algorithm, we develop several localized algorithms for different network operating systems and media access control (e.g. TDMA and Aloha) protocols. In order to test the quality of the results, we have also developed a generator of instances of the edge-coloring problem, where the optimal solution is known. Comprehensive evaluation of the algorithm clearly indicates the effectiveness of the proposed approach and algorithms.

### 1.2 Localized algorithms

In this section, we first identify the key properties and requirements for localized algorithms and informally and formally define localized algorithms. Finally, in order to establish a global role of the localized algorithms, we compare them with two other widely sued groups of algorithms: centralized and distributed.

Localized algorithms are algorithms implemented on sensor networks in such a way that only a limited number of nodes with a limited amount of communication is conducted between the nodes during their execution. Each node can communicate with only a few close neighbors directly and using multiple hops with all other nodes.

We model localized algorithms in the following way. One or more nodes initiate a request for a computation. The result of the computation should be distributed to one or more nodes (in the case of channels assignment to all nodes). Each node has to obtain all information required for communication either by using its sensors or through communication with its neighbors. The goal is to achieve the maximal objective function for the targeted optimization in such a way that all constraints are satisfied and that the minimal amount of communication, measured in terms of one hop session, is conducted. Obviously one can define here primal and dual problem where the initial objective function and amount of communication are at some requested level or optimized. It is interesting and important to compare localized algorithms with two other widely used types of algorithms. By far the most popular model of computation is the Turing machine. It has been shown that numerous other models, such as Post machine and Universal Register Machine, are equivalent in terms of computational power to this model. The most popular conceptual and economic implementations of the Turing Machine are Von-Neuman and Intel's PC respectively. Note that there is also a huge number of computational models for specific application with more rich mechanisms to express aspects such as timing and concurrency, including communicating FSM, synchronous/reactive, dataflow, process networks, CSP, Petri nets, Tagged-signal models, and Object-oriented models.

Also, distributed algorithms are widely used. Initially, they were mainly developed with goal to make execution of programs faster or resilient against faults. They received great, additional impetus due to the Internet in the last decade. It is interesting to note that there is very little common ground between distributed and localized algorithms.

## 2. RELATED WORK

The exponential growth of the wireless and mobile appliances requires effective and efficient re-use of the limited spectrum allocated to such tasks. In particular, in cellular networks, the base stations are costly and the number of base stations can be reduced by more efficient re-use of the radio spectrum. In particular relevance to our work are the dynamic channel allocation (DCA) schemes where the main idea is to evaluate the cost of each candidate channel and then select the best candidate provided that certain interference constraints are satisfied. In distributed DCA schemes, there are either cell-based or signal strength based approaches. [Kat96] provides a comprehensive survey on proposed channel assignment schemes for cellular wireless systems. All of the proposed schemes (even the distributed ones) rely on the existence of a central base station for each cell and some sort of handshaking between different base-stations. Due to the lack of a fixed base-station, none of these schemes are applicable to the wireless sensor networks.

Ad-hoc wireless networks are formed without the assumption of a central base station. However, there are a number of centralized channel assignment schemes for ad-hoc networks. There are a number of fixed assignment protocol approaches. [Gu00][Sen00] describe a CDMA-based approach to channel allocation. [Gro00] illustrates a TDMA-based approach to the problem. [Lop99] presents a procedure for neighborhood discovery and demand based channel assignment for ad-hoc network without an optimization goal in mind.

Recently, a flurry of research activities has been directed towards sensor networks and localized algorithms for such networks [Bad93]. For example, [Meg01] provides a generic localized algorithm for sensor networks that enables two types of optimization: The first, guarantees the fraction of nodes that are contacted while optimizing for solution quality. The second provides guarantees on solution quality while minimizing the number of nodes that are contacted and/or amount of communication. [Sch01] has a dynamic sensor network MAC addressing scheme based on a distributed algorithm. Krishnamachari [Kri02] provides a distributed problem solving approach that can be used in sensor networks for medium access scheduling, Hamiltonian cycle formulation and the partitioning of the network nodes into coordinating cliques. [Zho01] has a very simple first come first serve scheme for channel assignment in sensor networks but the procedure is not optimized for the limited number of channels. To the best of our knowledge we have the first efficient localized algorithm for constrained channel assignment in sensor networks.

Edge-coloring or minimum chromatic index is one of the oldest problems in graph theory. Tait [Tai80] stated that the four-color conjecture is equivalent to coloring the edges of every bridgeless cubic planner graph with three colors. However, 10 years later Peterson [Pet91] contradicted his opinion and illustrated that there are bridge-less cubic graphs that are not three edge-colorable. A Mathematical implementation of edge coloring using the vertex coloring routines and graph line transformation is shown in Combinatorica [Ski90]. [Fio77] provides a comprehensive survey on Graph-theoretic results on edge coloring. Later on it was shown that any graph could be edge colored using at most $\Delta+1$ colors, where $\Delta$ is the maximum vertex degree [Viz64] [Gup66]. However, the complexity of the problem was not determined until Holyer [Hol81] proved that computing the edge-chromatic number is NP-complete.

A randomized reduction from the distributional tiling problem to the distributional graph edge-coloring problem was proposed in [Ven88]. [Pan97] has proposed a randomized distributed edge-coloring that is not optimized for colors and does not take into account the constraints on the number of colors. Recently, [Cre99] proved that the problem is not approximable within $4/3 - \varepsilon$, for any $\varepsilon > 0$.

Distributed algorithms and applications arise in a variety of different disciplines in computer science and optimization algorithms. They find use in traditional computer science theory [Lam78] [Gal83], in parallel and distributed computations and algorithms [Ray88][Ber89][Lam90][Tel94][Lyn96], in concurrent languages and related computational models [Hoa85][Mil89], in operating system applications [Kis92][Ben93], in client-server research and development community [Dow98], in distributed artificial intelligence [Rum86][Dur89] and in particular distributed sensing [MIT82]. Each of the above mentioned types of distributed algorithms, have algorithms very specific for their target tasks. Although a number of basis and premises are similar in distributed nature, they are very different for certain sorts of tasks. In particular, we did not encounter an already proposed scheme that would be suitable for the wireless ad-hoc sensor network. In wireless sensor networks, the first issue to address is how the key functional parameters are affected by the selection of the algorithms and the underlying mechanisms. Based on the optimization objective, the distributed algorithm differs to more efficiently address the needs of each objective and its accompanying constraints. Furthermore, the key issues are the amount of computation at each component and then inter-communication between different components. This communication/computation trade-off is the major driving force in designing distributed wireless sensor network algorithms and is always present. The other main component in distributed algorithms for our target networks is the crucial need for security and privacy in such applications.

Bitner and Reingold [Bit75] were the first to propose in computer science community, the systematic use of the maximally constrained variable selection paradigm. They use it to guide search within generic backtracking search for the optimal solution. Detailed analysis of this approach is given in [Pur83].

The first to use the maximally constrained rule as part of fast heuristic program was Brelaz for graph coloring [Bre79]. He developed exceptionally fast and good heuristical graph coloring algorithm, by superimposing a maximally constrained - minimally constraining objective function on the of the optimal, but with exponential run time, graph coloring algorithm by Randall-Brown [Ran72].

In CAD literature, most constrained least constraining paradigm has been widely used, most under the name of force-directed heuristics [Has87, Pau89, Mo01]. A more global picture of the role of maximally constrained minimally constraining approach as an efficient heuristics is given in [Pea84].

# 3. PRELIMINARIES

## 3.1 The Network layered architecture

In essence, each network can be divided into two major components: the physical network and the logical network. The ever-increasing expansion of the network throughout the world has resulted in increased complexity of the logical network. The common method used to deal with the complexity of the network is to layer the network into discrete functional entities common to all communication tasks. Perhaps the most popular networking standard is the OSI that was proposed by the IEEE ISO community. In this standard, complex networks typically have up to seven layers. These layers are (in order): physical layer, data link layer, network layer, transport layer, session layer, presentation layer and the application layer. Not every network implements all the layers in the ISO standard. Our understanding in sensor network is that because of the power and memory constraints of the nodes, the networks should be as simple as possible and thus the network only consists of a few layers: physical layer, data link layer, network layer and possibly the transport layer. The physical layer provides the physical medium for data transmission. The data link layer is responsible for frame transmission from one node to the other and also monitors for errors in the physical layer. The network layer makes decisions for routing and relaying packets from one destination to the other through the network. Our algorithm fits into the network architecture exploration that should be done in order to enable the routing for the network layer. The transport protocol is the layer that decides where to route the data based on the information. Like any other complex network, in WESN the network operating system (NOS) is responsible for task management, resource allocation, memory and interface managements. We are developing algorithms that would best fit into the NOS structure.

## 3.2 The MAC layer protocols

As mentioned above, our algorithm is working on the top of the data link layer to provide a basis for the networking layer. To be more precise, the data link layer is itself divided into two sub-layers: 1) Logical Link Control (LLC) and 2) Medium Access Control (MAC). The LLC part ensures reliable data transfer over the physical link and manages the physical layer. MAC is the protocol that determines at each point, which node should have access to the transmission medium. The choice of the MAC protocol has a detrimental effect on the network discovery and assignments of the channels within the network.

In fixed assignment MAC protocol, allocation of the time and frequency is fixed and does not change with the changes in the network. Every node has its own slot in time or frequency through which it can transmit data on the channel to the other nodes. The most regular and uniform assignments of this type are FDMA (Frequency Division Multiple Access-a portion of the bandwidth is allocated to each node) and TDMA (Time Division Multiple Access-a portion of the time in each cycle is allocated to a node). The random assignment protocol on the other hand is very good for bursty (vs. uniform) traffics. But since the assignment is random, two or more nodes might collide in accessing the channel and in this event; they should both back off and start at some other random time until they do not collide. Examples of such approach are ALOHA and slotted ALOHA. In this work, we select one protocol from each class and develop a localized algorithm for each specific protocol. Our localized algorithm is developed for TDMA and ALOHA protocols.

# 4. MCMC ALGORITHMS for CHANNELS ASSIGNMENT

## 4.1 Problem formulation

The problem can be informally stated as follows: We are given a field F instrumented with wireless ad-hoc nodes $s_i \in S$, where the number of nodes N is known. Two sensors are able to communicate if their distance is smaller than the communication range R of the nodes. We are assuming that the nodes are forming a network, but the topology of the network is unknown (i.e. nodes are not informed about the other nodes in their communication range). If two nodes communicate to each other, there is a communicating edge (e) between them. The frequency bandwidth $\Delta f$ allocated to the network is fixed. If we assume that the bandwidth is slotted to a number of channels c, and we assume a uniform slotting, the frequency width of each communication channel is $\Delta f/c$ and is thus fixed. If we number the channel slots from 1 to k, we have at most k channels available in the network. The problem is now to assign this limited number of channels to the communicating edges in the network in such a way that the number of edges that have channels assigned to them is maximized. We map this problem to the edge-coloring problem where we map the nodes to the vertices of a graph and communication edges are the edges of the graph. The colors 1,..,k correspond to channels 1,…,k respectively. This problem can now be formally defined:

PROBLEM: GRAPH *Edge K*_COLORABILITY

**INSTANCE:** Graph G(V,E), positive integer $K \leq |E|$

**QUESTION:** *Is G colorable?* I.e. does there exist a function f : $E \rightarrow 1,2,3,\ldots,K$ such that $f(e_1) \neq f(e_2)$ whenever $e_2$ , $e_1 \in V$?

Note that the above-formulated problem is a decision problem, although our original problem was an optimization (maximization) problem. The optimization problem can be mapped into the decision problem using the binary search.

## 4.2 Centralized algorithm

We are attempting to make a maximally constrained minimally constraining (MCMC) centralized algorithm for the problem formally stated above. For solving this problem, we form an objective function (OF) for each edge in the graph. This OF is used in order to evaluate the edges of the graph for coloring using the MCMC strategy. Note that by the terms immediate neighborhood or adjacent edges or neighboring edges we mean all the edges that share a vertex with a target edge. K is the maximum number of colors available. We first define the best potential node for the coloring procedure. Assume we have started the coloring procedure and some of the edges in the graph are already colored. Furthermore, assume that each node $v_i$ has $m_i$ edges connected to it, from which $\chi_i$ edges are already colored. The potential for the node $v_i$ is denoted by $\Theta(v_i)$ and is defined as the degree of freedom of that node to get different colors. More precisely, $\Theta(v_i)$ is defined using the following procedure:

For $\forall v_i$ : if $(m_i-\chi_i)\leq K$, then $\Theta(v_i)= m_i-\chi_i$

Otherwise, $\Theta(v_i)=K$

After we define $\Theta(v_i)$ for each node, we select the nodes with higher $\Theta$ as the better candidates (more constrained) than the other nodes. Next, we define criteria for finding the best node among those candidates and the most constrained associated edge of the selected nodes to color. Furthermore, we define an objective function for each edge. This objective is dependent on the probability of using each color on the target edge. We now define the procedure to find the probability for using a certain color $c_i$ for each edge $e_i$, that has $\lambda_i$ neighboring edges. Furthermore, assume that the neighboring edges have already used $\mu_i$ colors ($\mu_i\leq K$). The critical palette for each edge, is the set $\varsigma_i$ that contains all of the colors that has already been used on the neighboring edges to $e_i$. For each color $c_j$, the probability of using $c_j$ on edge $e_i$ is denoted by $G_{ij}$ and is defined as followed:

For $\forall$ $e_i,c_j$ : if $c_j \in \varsigma_i$, $G_{ij}=0$, otherwise, $G_{ij}= 1 \big/ (K-\mu_i)$

After we define the $G_{ij}$ associated with each color for the whole neighborhood of each of the target edges, we calculate the cumulative probability $P_{ij}$ of using the color $c_j$ on the edge $e_i$. $P_{ij}$ defines the probability of using the color $c_j$ on edge $e_i$ according to our criteria.

$$P_{ij} = G_{ij} \Bigg/ \left( G_{ij} + \sum_{k=1}^{\lambda_i} G_{kj} \right) \qquad \textbf{Eq. 1}$$

The cumulative probability $P_{ij}$ has the advantage of having wider perspective and being broader than the probability $G_{ij}$. Since $P_{ij}$ takes into account not only the probability of using one color, but also the damaging effect that selecting each color for a candidate edge, would have on its neighborhood. In other words, $P_{ij}$ seeks the minimally constraining color for one edge. Another interesting artifact is $(1-P_{ij})$ which is the probability of not using the color $c_j$ on edge $e_i$ according to our criteria. Once we define $P_{ij}$ for each of the colors $c_j$ on $e_i$, we can calculate the probability $U_i$ of staying un-colored for $e_i$ as followed:

$$U_i = \coprod_{j=1}^{K} P_{ij} \qquad \textbf{Eq. 2}$$

Now that we have calculated the color-ability probability for each edge $e_i$ associated with the potential nodes (the ones with the highest $\Theta(v_i)$), we rank the potential nodes with respect to the color-ability probabilities of their associated edges. If a node has many edges that are hard to color, that node is most constrained and should be the next node to be finally selected and then we define the most constrained edge of this node (the edge with the least $U_i$ ) and color it with the color that has the highest $P_{ij}$ on that edge. We proceed with the algorithm until the colorability $U_i$ of each edge $e_i$ in G goes to zero.

## 4.3 Localized deterministic algorithm

The deterministic algorithm assumes having a fixed assignment protocol at the MAC layer. We selected TDMA as our fixed assignment protocol. In TDMA, each period of time is divided into n slots, 1,…,n. (n is the number of nodes in the network). For simplicity, assume that the indices of the time slots, match the indices of the nodes, i.e the node $v_i$ has the time slot i. Each node has access to the channel to communicate to its neighboring nodes at its specific time slot and no other time.

The phase-I of the procedure is to discover the local neighborhood for each of the nodes in the network. During the first cycle, every node broadcasts an identification message on its own time slot. The nodes that can receive the signal from the transmitting node are the ones that share an edge with the sender node. The receiving nodes update their neighbor list by denoting the sender as their neighbors. After a cycle through all of the nodes in the network, each node in the network has the information about all of its immediate neighbors. The second cycle has the same form as the first one. The difference is that the nodes broadcast to their neighboring nodes not only their identities but also the information they gathered about their neighborhood in the last cycle. After this cycle, all of the nodes in the network have the information about their 2-hop neighborhood. The procedure can repeat to up to k cycles, after which every node will acquire the information on its k-hop neighborhood. As described above, in our case 3-hop information would give us enough perspective to evaluate the nodes and the edges in the graph in phase-II.

Phase-II of the algorithm first evaluates the nodes and edges in the network and then assigns the channels to the most constrained edges of the graph. The basic methodology, node objective function and maximally constraining analysis is very similar to the centralized algorithm described before, although there are a number of fundamentally different issues that arise because of the localized nature of the strategy. In the centralized strategy, the stopping criterion was when the colorability of all of the edges became zero. In the TDMA strategy, the stopping criterion is having two identical rounds of TDMA, without improving the number of colored edges during a round.

The main part of the algorithm is very similar to the centralized case. The only critical issue here is that nodes do not have information about all of the nodes in the network but just their local neighborhood. We go through the nodes in the network one by one. For each node, we calculate its potential Өto the other nodes in its neighborhood. If the node has the highest potential in its local area it will attempt to color its most constrained edge with the least constraining color as discussed in the centralized case, and then it will send updates to its neighboring edges. The only issue that arises here is the ordering. Since the nodes that are responsible for coloring are not able to update their 2-hop neighborhood, some of the nodes might not receive the critical decisions before the next round terminates. To avoid such confusions, we restrict the nodes to color the most constrained edge that occurs after them according to the ordering of the TDMA slots.

## 4.4 Localized random algorithm

In this section, we present the localized algorithm that assumes Aloha media access control policy. We assume that t tokens are used by network operating system, i.e. they are the processes that simultaneously run. Note that any k token algorithm can be easily emulated by a single token algorithm with equal or lower communication cost, but longer run time. The algorithm has three major phases: (i) Neighborhood discovery; (ii) Token membership assignment; and (iii) Simultaneous coloring within each token domain. We assume random time attempt of transmission and continuous reception at each node. The neighborhood discovery is conducted in the same way as in the case of TDMA-based localized algorithm. The only change is that, we use as stopping criteria at each node the elapsed time. Therefore, we guarantee the level of energy consumption at potential degradation of performance. Specifically, each node waits until it does not hear from each of its neighbors $t_n$ times, where $t_n$ is the number of neighbors. The most interesting and unique aspect of the Aloha-based centralized algorithm is token membership resolution procedure. There are t tokens, each assigned in a random way to one of the nodes in the network. The goal is to assign membership of each node in the network to one of the tokens in such a way that all tokens cover approximately the same number of nodes, that communication distance between nodes with the same token is minimized, and that a cluster of nodes with the same token has one-hop communication to as few as possible other clusters. We proceed in the following way. Each token is sent to one of its neighbors. With each token also travels information about the cardinality of the current corresponding cluster. Every time, when the token is sent to a node that already has membership, the token obtains an estimate about the size of the clusters that already tried to assign that node. The distance of the original node and the relative number of nodes is used to either request that this node is reassigned or just not further consider for membership in this token set. The reassignment is done when two tokens are at the same node.

## 5. EXPERIMENTAL RESULTS

In order to evaluate the performance of the above-defined procedures on the real graphs, we need to generate a graph G for which we already know the optimal edge-coloring solution. We use this graph to experiment our three types of algorithms described earlier. Assume that we have N nodes in the graph, called $n_1, \ldots, n_N$. We have maximum of K colors $c_1, \ldots, c_K$ to color this graph. In order to necessitate using at least K colors to color the edges of the graph, we first randomly select a node $n_i$ in the graph and connect it to K other random nodes. Now, edges associated with $n_i$ need at least K-colors to be colored. After that, we start to assign the colored edges between the nodes of the graph G. If an edge colored in $c_c$ assigned to two nodes has a neighboring edge with the same color $c_c$, that edge would not accept that color. This random assignment would continue for a user specified number of edges E. Now, we have a graph G that is fully colored and we are certain that the minimum number of colors to color such a graph is K. We can further complicate the target graph by adding an obfuscation phase to the graph generation. During this phase, we add uncolored edges between the nodes in G that already have K edges associated with them. The added edges are not colorable since they already have neighbors in every possible color. Thus, the final generated graph G would still be an optimal solution for coloring the maximum number of edges of the graph using at most K colors.

In order to evaluate the quality of developed centralized and localized algorithms we conducted two type of experiments. In the first experiment, we compared the number of colored edges after the application of the centralized, TDMA and Aloha localized algorithms vs. the number of colored edges in the optimal solution. Table 1 shows the obtained results. In all examples, we used 10 colors.

The first two columns indicate the number of nodes and edges in the graphs. The next column indicates the number of colored edges in the optimum solution. After that we show, in the three columns, the results for centralized, TDMA and Aloha localized algorithms. It is apparent, that in all cases the new algorithms performed exceptionally well.

| Nodes | Edges | Optim al | Centr alized | TDM A | ALO HA |
|---|---|---|---|---|---|
| 100 | 1500 | 1000 | 1000 | 1000 | 952 |
| 200 | 3000 | 2000 | 2000 | 1983 | 1911 |
| 250 | 3750 | 2500 | 2500 | 2490 | 2386 |
| 500 | 10000 | 10000 | 9998 | 9973 | 9354 |
| 1000 | 20000 | 20000 | 19999 | 19922 | 18356 |
| 1000 | 30000 | 20000 | 19983 | 19902 | 18019 |

**Table 1 - Experimental results**

In the second experiment, we generated 100 nodes in fields of various sizes with various communication ranges. We again used 10 colors. Each node was able to communicate with its neighbors, within a specified range. We measured the lower bound on percentage of obtained potential. The potential is defined as the number of edges incident to the node, unless the node has more edges then colors. In that case, the potential is set to the number of colors. The centralized algorithm achieved at average 94% of the potential on 20 test examples. On the same set of examples, TDMA and Aloha achieved 91.8 % and 86.7% of the potential, essentially indicating that more complex algorithms are not needed.

## 6. CONCLUSION

We developed the first approach for the design and evaluation of localized algorithms in wireless embedded ad-hoc sensor networks. The approach has three phases: information gathering, system structuring, and optimization mechanisms. We demonstrated the approach on the channel assignment - edge-coloring problem. We used a new maximally constrained - minimally constraining centralized algorithm as the optimization engine in TDMA and Aloha media access protocol scenarios to develop localized algorithms. The algorithms perform exceptionally well on a number of test examples.

## 7. REFERENCES

[Bad93] Badrinath, B.R. et. al. Impact Of Mobility On Distributed Computations. ACM Operating Systems Review, April 1993.

[Ben93] Bender, M. et. al. Unix For Nomads: Making Unix Support Mobile Computing. In Proceedings of the USENIX Symposium on Mobile & Location Independent Computing, 1993.

[Ber89] Bertsekas, D.P. Tsitsiklis, J.N. Parallel and distributed computation. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[Bit75] Bitner, J.R. Reingold, E.M. Backtrack programming techniques. Communications of the ACM, vol.18, (no. 11), pp. 651-6, 1975.

[Bre79] Brelaz, D. New methods to color the vertices of a graph. Communications of the ACM, vol.22, (no. 4) pp. 251-6, 1979.

[Cre99] Crescenzi, P. Kann, V. Silvestri, R. Trevisan, L. Structure in approximation classes, SIAM Journal of Computing (no. 28), pp. 1759-1782, 1999.

[Dow98] Downing, T.B. Java RMI: Remote Method Invocation. IDG Books Worldwide, New York, 1998.

[Dur89] Durfee, E.H. Lesser, V.R. Corkill, D.D. Trends in Cooperative Distributed Problem Solving. IEEE Transactions on Knowledge and Data Engineering, vol.1, pp. 63-83, 1989.

[Fee99] Feeney, L.M. A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks SICS Technical Report T99:07.

[Fio77] Fiorini, S. Wilson, R. Edge-colourings of graphs. Research Notes in Mathematics 16, Pitman, London, 1977.

[Gal83] Gallager, R.G. Humblet, P.A. Spira, P.M. A Distributed Algorithm For Minimum Spanning Tree. ACM Transactions on Programming Languages and Systems, pp. 66-77, 1983.

[Gro00] Gronkvist, J. Assignment methods for spatial reuse TDMA. Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc'00), pp. 119-24, 2000.

[Gu00] Gu, D. L. Ly, H. Hong, X. Gerla, M. Pei, G. Lee, Y-Z. C-ICAMA, a centralized intelligent channel assigned multiple access for multi-layer ad-hoc wireless networks with UAVs. 2000 IEEE Wireless Communications and Networking Conference, vol.2, pp. 879-84, 2000.

[Gup66] R. P. Gupta. The chromatic index and the degree of a graph. Notices of the American Mathematics Society, (no. 13), pp. 66T-429, 1966.

[Has87] Hasan, N. Liu, C.L. A force-directed global router. Advanced Research in VLSI, 1987.

[Hoa85] Hoare, C.A.R. Communicating Sequential Processes, Prentice-Hall International, 1985.

[Hol81] Holyer, I. The NP-completeness of edge colorings. SIAM Journal of Computing, (no. 10), pp. 718-720, 1981.

[Kat96] Katzela, I. Naghshineh, M. Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey. IEEE Personal Communications, vol.3, (no. 3), pp. 10-31, 1996.

[Kis92] Kistler, J. Satyanarayanan, M. Disconnected Operation In The Coda File System. ACM Transactions on Computer Systems, 1992.

[Kri02] Krishnamachari, B. Bejar, R. Wicker, S.B. Distributed Problem Solving and the Boundaries of Self-Configuration in Multi-hop Wireless Networks. To appear in the Proceedings of the Hawaii International Conference on System Sciences (HICSS-35), 2002.

[Lam78] Lamport, L. Times, Clocks, And The Ordering Of Events In A Distributed System. CACM, pp. 558-565, 1978.

[Lam90] Lamport, L. Lynch, N. Distributed Computing: Models And Methods. Handbook of Theoretical Computer Science, Elsevier Science Publishers, pp. 1158-1199, 1990.

[Lop99] Lopez-Rodriguez, D. Perez-Jimenez, R. Distributed method for channel assignment in CDMA based ad-hoc wireless local area networks. IEEE MTT-S International Topical Symposium on Technologies for Wireless Applications, Canada, pp. 11-16, 1999.

[Lyn96] Lynch, N. Distributed Algorithms. Morgan Kaufman, San Francisco, 1996.

[Meg01] Meguerdichian, S. Slijepcevic, S. Karayan, V. Potkonjak, M. Localized Algorithms In Wireless Ad-Hoc Networks: Location Discovery and Sensor Exposure. Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'01), 2001.

[Mil89] Milner. R. Communication and Concurrency. Prentice Hall, New York, 1989.

[MIT82] MIT Lincoln Laboratories. Workshop on Distributed Sensor Networks. 1982.

[Mo01] Mo, F. Tabbara, A. Brayton, R.K. A timing-driven macro-cell placement algorithm. Proceedings 2001 IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp. 322-327, 2001.

[Pan97] Panconesi, A. Srinivasan, A. Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. SIAM Journal of Computing (no. 26), pp. 350-368, 1997.

[Pau89] Paulin, P.G. Knight, J.P. Force-directed scheduling for the behavioral synthesis of ASICs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, June 1989, vol.8, (no. 6), pp. 661-79.

[Pea84] Pearl, J. Heuristics: intelligent search strategies for computer problem solving. Reading, Mass: Addison-Wesley Pub. Co., 1984.

[Pur83] Purdom, P.W. Jr. Brown, C.A. An analysis of backtracking with search rearrangement. SIAM Journal on Computing, vol.12, (no.4), pp. 717-33, 1983.

[Ran72] Randall-Brown, J. Chromatic scheduling and the chromatic number problems, Management Science, Vol. 19, (no. 4), pp. 456-463, 1972.

[Ray88] Raynal, M. Distributed Algorithms and Protocols, John Wiley and Sons, 1988.

[Rum86] Rumelhart, D. E. McClelland, J. L. Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press, 1986.

[Sch01] Schurgers, C. Kulkarni, G. Srivastava, M.B. Distributed Assignment of Encoded MAC Addresses in Sensor Networks. Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'01), pp. 295-298, 2001.

[Sen00] M. Sengoku, H. Tamura, K. Mase, S. Shinodu. A routing problem on ad-hoc networks and graph theory. International Conference on Communication Technology Proceedings (ICCT'00), Beijing, China, 2000.

[Ski90] Skiena, S. Implementing Discrete Mathematics. Addison-Wesley, Redwood City, CA, 1990.

[Tai88] Tait, P. On the coloring of maps. Proceedings of the Royal Society, Edinburg, Sect A, (no. 10), pp. 501-503, 1880.

[Tel94] Tel, G. Introduction to Distributed Algorithms. Cambridge University Press, Cambridge, U.K., 1994.

[Ven88] Venkatesan, R. Levin, L. Random instances of a graph-coloring problem are hard. In Proceedings of the 20th Annual Symposium on Theory of Computing, ACM Press, pp. 217-222, 1988.

[Ven91] Venkatesan, R. Average-Case Intractability. Ph.D. Thesis (Advisor: L. Levin), Boston University, 1991.

[Viz64] Vizing, V. G. On an estimate of the chromatic class of a p-graph (in Russian). Diskret. Analiz, (no. 3), pp. 23-30, 1964.

[Zho01] Zhong, L. Shah, R. Guo, C. Rabaey, J. An ultra-low power and distributed access protocol for broadband wireless sensor networks. Networld+Interop: IEEE Broadband Wireless Summit, Las Vegas, May 2001.