Channel Assignment in Multi-Radio Wireless Mesh Networks : A Graph-Theoretic Approach

Srikrishna Sridhar*, Jun Guo†, Sanjay Jha†
* Computer Science and Information Systems, Birla Institute of Technology and Science (BITS), Pilani India
Email: f2005097@bits-pilani.ac.in
† School of Computer Science and Engineering, The University of New South Wales, Australia
Email: jguo@cse.unsw.edu.au; sjha@cse.unsw.edu.au

***Abstract*— In this paper, we propose a load-based scheme for assigning channels to radio interfaces in multi-radio, multi-channel wireless mesh networks. We first construct a model for channel assignment as an optimization problem with the goal of minimizing the overall network interference. The problem is proven to be NP-Hard. We then apply the Lagrangian relaxation method to obtain lower bounds as well as near-optimal feasible solutions for large size networks. We further present a meta-heuristic based on genetic algorithms, which can yield good quality solutions for very large networks. With these two centralized approaches as the benchmark, we propose a fully distributed algorithm in order to tackle the channel assignment problem practically. Our extensive simulation experiments demonstrate that the distributed algorithm performs competitively and can serve as a practical and scalable solution to the channel assignment problem.**

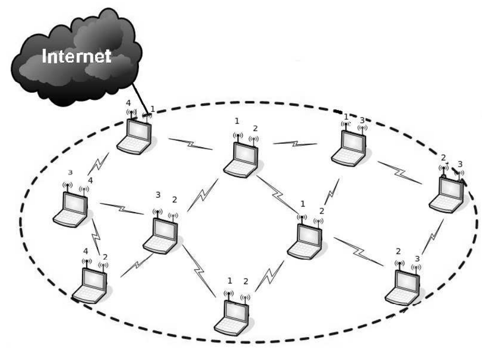

Fig. 1. A wireless mesh network with two radios and four channels.

I. Introduction

Physical layer technologies have undergone noteworthy changes in the recent past. Despite these changes, today's wireless LAN still cannot match its wired counterparts in providing sustained bandwidth. With the inclusion of all the overheads like 802.11 headers, errors and MAC contention, the actual throughput available to applications is drastically reduced [1]. The transmission rates are also known to fall rapidly with increasing distance between the nodes under consideration. The situation is further intensified in multi-hop wireless mesh networks [2] due to the increased interference between adjacent nodes as well as interference between neighboring paths [3]. To account for this, the IEEE 802.11 standards provide multiple overlapping frequency channels to support multiple simultaneous transmissions in the same interference region. Figure 1 illustrates the use of multiple channels in a multi-radio wireless mesh network. While these multiple channels offer a way of minimizing interference, they raise additional issues of channel assignment for maximizing overall throughput.

Channel assignment deals with the assignment of channels to radio interfaces with the goal of minimizing the total network interference. Most of the existing approaches target the multi-radio environment because it is known that equipping each node with multiple radio interfaces can more effectively utilize the spectrum [4], [5].

One of the approaches to handle channel assignment is to change channels on-demand, i.e., on a per-packet basis [6]–[9]. However, such *dynamic* channel assignment schemes require frequent channel switchings within each node. This is known to cause delays of the order of a few milliseconds [10]. They also require high speed synchronizations among nodes during transmission/receival over a particular channel, which is difficult to achieve without modifying the 802.11 MAC. For these reasons, an approach in this direction might not be practical. Another commonly suggested approach is *static* channel assignment [4], [11] which is done on the premise of ease of adaptability in commodity 802.11 hardware. Although these processes are referred to as *static*, they can easily be extended to *semi-dynamic* by refreshing the channel assignment at regular fixed time intervals, depending on the network load stability and predictability. *Hybrid* approaches [12]–[14] apply *semi-dynamic* / *static* schemes to fixed interfaces and *dynamic* channel assignment schemes to switchable interfaces. Although they still suffer from channel switching delays of the *dynamic* approaches, they can work well in highly unstable networks. Strictly speaking, our approach in this work addresses the *semi-dynamic* channel assignment problem which can alternatively be used in any one of the *hybrid* approaches.

In this paper, we propose a load-based scheme to address the channel assignment problem. This scheme is an important element of the joint channel assignment and routing problem that we are currently investigating. We adopt a graph-theoretic approach and formulate the problem as an integer linear programming (ILP) problem. The channel assignment problem is proven to be NP-Hard. We then apply the Lagrangian relaxation method [15] to obtain lower bounds as well as near-optimal solutions. We further present a meta-heuristic based on genetic algorithms(GA's) which can yield good quality solutions for very large networks. With these centralized approaches as the benchmark, we propose a fully distributed algorithm in order to tackle the channel assignment problem practically.

The rest of the paper is organized as follows. After giving a brief overview of related work in Section II, Section III provides the ILP formulation of the channel assignment problem. In Section IV we apply the Lagrangian relaxation method to obtain lower bounds and present the Lagrangian heuristic to find near-optimal feasible solutions. In Section V we propose the meta-heuristic while Section VI presents the distributed algorithm. In Section VII we evaluate the performance of our algorithms using extensive simulation experiments. We conclude our paper in Section VIII and discuss future directions.

II. Related Work

As mentioned previously *dynamic* and *hybrid* approaches have significant shortcomings for channel assignment based on our requirements. We will now look at *static/semi-dynamic* channel assignment in greater detail.

Das *et al.* [16] proposed various optimization models, however their approaches are not scalable due to the exponential complexity. Ramachandran et. al. in [17] developed a measurement based centralized approach to handle channel assignment for radios instead of links. This approach plainly handles the binding of a radio to a channel leaving the interface-channel binding procedure unsolved. Although the polling procedure for traffic bandwidth estimation suggested in [17] requires less time in comparison with those suggested in [18], it still requires diverting traffic to a *default channel* for a considerable amount of time. Apart from the above stated shortcomings, all these approaches suffer from being centralized and therefore difficult to implement in a real network.

Several authors have proposed distributed algorithms to aid practicality. In [1], the authors proposed a distributed channel assignment scheme for a tree-based traffic pattern with gateways as the root nodes. This scheme may however lead to inefficient channel assignments and routing in a more generic peer-to-peer enterprise network. The authors of [19] proposed a distributed algorithm based on minimizing the interference using partially overlapping channels as explored by [20]. This algorithm treats channel assignment independent of network load. However, it is a well known fact that load-aware channel assignment improves the network throughput [11]. Thus, we are motivated to design a scheme that can more intricately acknowledge the dependence of channel assignment on network load.

The work in [21] is most closely related to ours. There, the authors chose to approximate the channel assignment problem as a solution to the *Max-K-Cut* problem on the conflict graph using Tabu search [22]. However, their approach does not offer theoretical lower bounds for accuracy estimations.

III. Problem Formulation

In this section, we present an ILP formulation of the channel assignment problem. The later sections will then describe different approximation and heuristic schemes to offer good quality polynomial time solutions for this problem.

A. Network Model

A wireless mesh network can be modeled as a connected graph $G = (V, E)$, where V is the set of N mesh nodes and $E \subset V \times V$ is the set of wireless links. We assume that each node uses omni-directional antennas and all wireless links are bi-directional. A wireless link exists between nodes i and j if the distance between the two nodes, $d_{i,j}$, is smaller than R_t, where R_t is a fixed transmission range. For simplicity, we assume that a transceiver has the same receiving and transmission range. Thus, in our context, each edge $(i, j) \in E$ represents an undirected edge of the graph G.

Let the set of channels supported by the 802.11 spectrum be denoted as K, where $K = \{1, 2, \ldots, k\}$, and the number of radios on each node as $M_i \le |K|, \forall i \in V$. We assume that all channels are orthogonal, so the interference exists between two links if they are within interference range and are assigned the same channel. We believe that our model can be easily extended to account for non-orthogonal channels. To model the interference we consider a conflict graph $G_c = (V_c, I)$, where $V_c = E$ and $I \subset E \times E$. Two links (i, j) and (u, v) interfere with each other if they operate on the same channel and any of the quantities $d_{u,i}$, $d_{v,i}$, $d_{u,j}$, $d_{v,j}$ is smaller than sR_i, where R_i denotes the fixed interference range. Let $I_{i,j} \subset I, \forall (i, j) \in E$, denote the set of all links in the network within the interference range of link (i, j).

Let $\mathbf{L}$ be the load matrix of the network. Thus, $L_{i,j}$ is the expected traffic on link (i, j). This flow estimate of network traffic can be obtained using tools like the CoMo project [23].

We also make the following assumptions while modeling the channel assignment problem in wireless mesh networks.

- The traffic flow on the network is relatively stable over a period of time and is easy to predict. This is a fairly reasonable assumption for enterprise networks which are designed for balanced network flows.
- Nodes are generally static. This ensures no major topology changes during the course of channel assignment.

B. ILP Formulation

The problem of multi-radio channel-assignment is the assignment of at-most M_i channels to each node from the set K such that the sum of weights of potentially interfering links is a minimum.

First, we define a channel assignment matrix $\boldsymbol{C}$ as

$$C^k_{i,j} = \begin{cases} 1 & \text{if link } (i,j) \text{ uses channel } k \\ 0 & \text{otherwise} \end{cases}$$

The channel-assignment problem can be modeled as an extended version of the edge scheduling problem with an additional constraint of limited number of radios per node. To account for this constraint we define a radio usage matrix $\boldsymbol{X}$ in the following manner

$$X_i^k = \begin{cases} 1 & \text{if node } i \text{ has a radio tuned to channel } k \\ 0 & \text{otherwise} \end{cases}$$

The binding of link (i,j) to channel k must force the binding of both the corresponding radios to the same channel. Thus, we have

$$C_{i,j}^k \leq X_i^k \quad \forall\ k \in K, (i,j) \in E \tag{1}$$

$$C_{i,j}^k \leq X_j^k \quad \forall\ k \in K, (i,j) \in E \tag{2}$$

The number of channels that can be supported on each node is limited by the number of radios it has. This can be expressed as

$$\sum_{k \in K} X_i^k \leq M_i \quad \forall\ i \in V \tag{3}$$

We also ensure that each link in the network is assigned one channel to meet the requirement of topology preservation. Several models have questioned this assumption. However, we believe that, without assigning a channel to each link, complicated issues of topology-preservation would be raised, which can have adverse effects on higher layers. Therefore, this constraint is enforced by

$$\sum_{k \in K} C_{i,j}^k = 1 \quad \forall (i,j) \in E \tag{4}$$

To indicate if interference exists on link (i,j), we define binary variables $Y_{i,j}$ as

$$Y_{i,j} = \begin{cases} 1 & \text{if interference exists on link } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

In our model, we assume that interference exists on link (i,j) if any link in the set $I_{i,j}$ is assigned the same channel as (i,j). This can be ensured by the following constraint.

$$C_{i,j}^k + C_{u,v}^k - 1 \leq Y_{i,j}$$

$$\forall k \in K,\ (u,v) \in I_{i,j},\ (i,j) \in E \tag{5}$$

The optimization criterion of the channel assignment problem is to minimize the interference throughout the network. Unlike [19], the objective function of our load-based interference model is a weighted summation giving due importance to the link traffic. Hence, we have

$$Z = \min \sum_{(i,j) \in E} L_{i,j} Y_{i,j} \tag{6}$$

C. Interference Estimation and Modeling

The load matrix $\mathbf{L}$ is determined by an iterative procedure based on predictable load estimations which is a reasonable assumption for enterprise networks. We define a time frame T_r to measure link load. The average number of packets sent and received in the previous *n* time frames would serve as the expected load matrix for the next time frame. The averaging is done to prevent unexpected load spikes from affecting the system adversely.

Misra *et al.* [20] have formally modeled the degree of overlap between partially overlapping channels. We believe that our model can be easily extended to incorporate partially overlapping channels.

Control information can be transfered using a default channel thoughout the network. This fairly simple solution for transferring control information has been disputed by Naveed *et al.* in [24] where they propose an alternative topology preserving assignment for default channel. However, this cluster based channel assignment scheme is targeted mainly at broadcast scenarios and may result in poor channel assignment schemes in a more generic peer-to-peer network. Disrupting topology can also have adverse effects on the upper layers, especially routing [25].

IV. Lagrangian Relaxation

The channel assignment problem we consider in this paper is NP-hard (see Theorem 1 in the Appendix). Therefore, we resort to approximation approaches to obtain good quality solutions in polynomial-time. In this section, we apply the Lagrangian relaxation method to the ILP model. This approach will not only provide lower bounds but also generate near-feasible solutions which can be made feasible and near-optimal by some judicious tinkering.

We first relax the complicating constraints (5). The Lagrangian problem can be stated as,

$$\begin{aligned} Z_d(\boldsymbol{\lambda}) = \min \sum_{(i,j) \in E} L_{i,j} Y_{i,j} \\ + \sum_{(i,j) \in E} \sum_{k \in K} \sum_{(u,v) \in I_{i,j}} \lambda_{i,j,u,v}^k (C_{i,j}^k + C_{u,v}^k - 1 - Y_{i,j}) \end{aligned} \tag{7}$$

$$\text{subject to} \quad (1), (2), (3) \text{ and } (4),$$

where $\boldsymbol{\lambda} = \{\lambda_{i,j,u,v}^k \geq 0 \mid k \in K, (i,j) \in E, (u,v) \in I_{i,j}\}$ is the set of Lagrangian multipliers associated with (5).

Expanding (7), we have

$$\begin{aligned} Z_d(\boldsymbol{\lambda}) = \min \sum_{(i,j) \in E} \Big(L_{i,j} - \sum_{k \in K} \sum_{(u,v) \in I_{i,j}} \lambda_{i,j,u,v}^k \Big) Y_{i,j} \\ + \sum_{(i,j) \in E} \sum_{k \in K} \sum_{(u,v) \in I_{i,j}} \lambda_{i,j,u,v}^k C_{i,j}^k \\ + \sum_{(i,j) \in E} \sum_{k \in K} \sum_{(u,v) \in I_{i,j}} \lambda_{i,j,u,v}^k C_{u,v}^k \\ - \sum_{(i,j) \in E} \sum_{k \in K} \sum_{(u,v) \in I_{i,j}} \lambda_{i,j,u,v}^k \end{aligned} \tag{8}$$

Let

$$\tilde{\lambda}_{i,j}^k = \sum_{(u,v)\in I_{i,j}} \lambda_{i,j,u,v}^k \tag{9}$$

and

$$\hat{\lambda}_{i,j}^k = \sum_{(u,v)\in I_{i,j}} \lambda_{u,v,i,j}^k \tag{10}$$

Lemma 1: Given a channel-assignment matrix $\boldsymbol{C}$ and the set of Lagrangian multipliers $\boldsymbol{\lambda}$,

$$\sum_{(i,j)\in E}\sum_{k\in K}\sum_{(u,v)\in I_{i,j}} \lambda_{i,j,u,v}^k C_{u,v}^k = \sum_{(i,j)\in E}\sum_{k\in K} \hat{\lambda}_{i,j}^k C_{i,j}^k \tag{11}$$

Proof: Owing to the symmetry of the interference set, we can observe that if $(u,v)\in I_{i,j}$ then $(i,j)\in I_{u,v}$. Thus, by interchanging the summations over (i,j) and (u,v) in the left-hand side of (11), we have

$$\begin{aligned}&\sum_{(i,j)\in E}\sum_{k\in K}\sum_{(u,v)\in I_{i,j}} \lambda_{i,j,u,v}^k C_{u,v}^k \\ &= \sum_{(i,j)\in E}\sum_{k\in K}\sum_{(u,v)\in I_{i,j}} \lambda_{u,v,i,j}^k C_{i,j}^k \\ &= \sum_{(i,j)\in E}\sum_{k\in K} \hat{\lambda}_{i,j}^k C_{i,j}^k\end{aligned}$$

where the final equivalence follows from (10). ■

Letting, $\bar{\lambda}_{i,j}^k = \hat{\lambda}_{i,j}^k + \tilde{\lambda}_{i,j}^k$, (8) can be written as

$$\begin{aligned}Z_d(\boldsymbol{\lambda}) = \min &\sum_{(i,j)\in E}\Big(L_{i,j} - \sum_{k\in K}\sum_{(u,v)\in I_{i,j}} \lambda_{i,j,u,v}^k\Big)Y_{i,j} \\ &+ \sum_{(i,j)\in E}\sum_{k\in K}\bar{\lambda}_{i,j}^k C_{i,j}^k \\ &- \sum_{(i,j)\in E}\sum_{k\in K}\sum_{(u,v)\in I_{i,j}} \lambda_{i,j,u,v}^k\end{aligned} \tag{12}$$

We first observe that

$$\min \sum_{(i,j)\in E}\Big(L_{i,j} - \sum_{k\in K}\sum_{(u,v)\in I_{i,j}} \lambda_{i,j,u,v}^k\Big)Y_{i,j}$$

is a trivial problem. Letting

$$u_{i,j} = L_{i,j} - \sum_{k\in K}\sum_{(u,v)\in I_{i,j}} \lambda_{i,j,u,v}^k \tag{13}$$

we have

$$\min \sum_{(i,j)\in E} u_{i,j} Y_{i,j} = \sum_{(i,j)\in E} \min(0, u_{i,j})$$

where

$$Y_{i,j} = \begin{cases} 1 & \text{if } u_{i,j} < 0 \\ 0 & \text{otherwise}\end{cases}$$

Therefore, (7) reduces to

$$\begin{aligned}Z_d(\boldsymbol{\lambda}) = &\sum_{(i,j)\in E}\Big(\min(0,u_{i,j}) - \sum_{k\in K}\sum_{(u,v)\in I_{i,j}} \lambda_{i,j,u,v}^k\Big) \\ &+ \min \sum_{(i,j)\in E}\sum_{k\in K}\bar{\lambda}_{i,j}^k C_{i,j}^k\end{aligned} \tag{14}$$

$$\text{subject to } (1),(2),(3) \text{ and } (4).$$

To solve (14), the optimal $\boldsymbol{C}$ must solve

$$\min \sum_{(i,j)\in E}\sum_{k\in K}\bar{\lambda}_{i,j}^k C_{i,j}^k \tag{15}$$

$$\text{subject to } (1),(2),(3) \text{ and } (4).$$

However, it can be shown that (15) is still NP-hard due to the set of complicating constraints (1) and (2). Thus, we further relax (1) and (2), and the resulting Lagrangian problem is

$$\begin{aligned}\bar{Z}_d(\boldsymbol{\beta},\boldsymbol{\gamma}) = \min &\sum_{(i,j)\in E}\sum_{k\in K}\bar{\lambda}_{i,j}^k C_{i,j}^k \\ &+ \sum_{i(i,j)\in E}\sum_{k\in K}\beta_{i,j}^k (C_{i,j}^k - X_i^k) \\ &+ \sum_{(i,j)\in E}\sum_{k\in K}\gamma_{i,j}^k (C_{i,j}^k - X_j^k)\end{aligned} \tag{16}$$

$$\text{subject to (3) and (4).}$$

where $\boldsymbol{\beta} = \{\beta_{i,j}^k \geq 0 \mid k\in K, (i,j)\in E\}$ is the set of Lagrangian multipliers associated with (1) and $\boldsymbol{\gamma} = \{\gamma_{i,j}^k \geq 0 \mid k\in K, (i,j)\in E\}$ is the set of Lagrangian multipliers associated with (2). Rearranging the terms in (16), we have

$$\begin{aligned}\bar{Z}_d(\boldsymbol{\beta},\boldsymbol{\gamma}) = \min &\sum_{k\in K}\sum_{(i,j)\in E}(\bar{\lambda}_{i,j}^k + \beta_{i,j}^k + \gamma_{i,j}^k) C_{i,j}^k \\ &- \sum_{k\in K}\sum_{(i,j)\in E}\beta_{i,j}^k X_i^k \\ &- \sum_{k\in K}\sum_{(j,i)\in E}\gamma_{j,i}^k X_i^k\end{aligned} \tag{17}$$

Let

$$\theta_{i,j}^k = \bar{\lambda}_{i,j}^k + \beta_{i,j}^k + \gamma_{i,j}^k \tag{18}$$

and

$$\phi_i^k = \sum_{j:(i,j)\in E}\beta_{i,j}^k + \sum_{j:(j,i)\in E}\gamma_{j,i}^k \tag{19}$$

Thus, we have

$$\bar{Z}_d(\boldsymbol{\beta},\boldsymbol{\gamma}) = \min \sum_{k\in K}\Big(\sum_{(i,j)\in E}\theta_{i,j}^k C_{i,j}^k - \sum_{i\in V}\phi_i^k X_i^k\Big) \tag{20}$$

$$\text{subject to (3) and (4).}$$

This problem reduces to two sub-problems, i.e.

$$\min \sum_{k\in K}\sum_{(i,j)\in E}\theta_{i,j}^k C_{i,j}^k, \text{ subject to } (4) \tag{21}$$

and

$$\max \sum_{k\in K} \sum_{i\in V} \phi_i^k X_i^k, \text{ subject to (3)} \tag{22}$$

We observe that (21) is trivially solved by choosing k with the smallest $\theta_{i,j}^k$ for each link (i,j) and setting $C_{i,j}^k = 1$. Also (22) is easily solved by choosing the M_i largest values of ϕ_i^k for each node i and setting the corresponding X_i^k's as 1.

A. Evaluating Multipliers

It is clear that the best choice for the vector $\boldsymbol{\lambda}$ would be an optimal solution to the problem

$$Z_D(\boldsymbol{\lambda}^*) = \max_{\boldsymbol{\lambda}} Z_d(\boldsymbol{\lambda}) \tag{23}$$

To evaluate $\boldsymbol{\lambda}$, we use the subgradient method [26]. Given an initial value of $\boldsymbol{\lambda}^0$, a sequence $\boldsymbol{\lambda}^T$ is generated, $\forall (i,j) \in E,\ (u,v) \in I_{i,j},\ k \in K$, using

$$(\lambda_{i,j,u,v}^k)^{T+1} = (\lambda_{i,j,u,v}^k)^T + s_k[(C_{i,j}^k)^T + (C_{u,v}^k)^T - 1 - (Y_{i,j})^T]$$

where $\boldsymbol{C}^T$ is the optimal solution vector and s_k is a positive scalar step size given by

$$s_k = \frac{\epsilon(Z^* - Z_d(\boldsymbol{\lambda}^T))}{\|[(C_{i,j}^k)^T + (C_{u,v}^k)^T - 1 - (Y_{i,j})^T]\|^2}$$

Here, ϵ is a scalar satisfying $0 < \epsilon \leq 2$ and is determined by initiating $\epsilon = 2$ and halving it whenever Z_d has failed to decrease for a fixed number of iterations. Z^* is an upper bound, obtained by applying the Lagrangian heuristic which we will discuss in more detail in the following section.

Since (20) is obtained by Lagrangian relaxation as well, we evaluate $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, $\forall (i,j) \in E,\ k \in K$, in a similar manner using

$$(\beta_{i,j}^k)^{T+1} = (\beta_{i,j}^k)^T + t_k[(C_{i,j}^k)^T - (X_i^k)^T]$$

where

$$t_k = \frac{\bar{\epsilon}(\bar{Z}^* - \bar{Z}_d(\lambda^T))}{\|[(C_{i,j}^k)^T - (X_i^k)^T]\|^2}$$

and

$$(\gamma_{i,j}^k)^{T+1} = (\gamma_{i,j}^k)^T + t_k^{'}[(C_{ij}^k)^T - (X_j^k)^T]$$

where

$$t_k^{'} = \frac{\tilde{\epsilon}(\bar{Z}^* - \bar{Z}_d(\lambda^T))}{\|[(C_{i,j}^k)^T - (X_j^k)^T]\|^2} \tag{24}$$

B. Generating Feasible Solutions

Solving the Lagrangian problem of $Z_D(\boldsymbol{\lambda}^*) = \max_{\boldsymbol{\lambda}} Z_d(\boldsymbol{\lambda})$ provides lower bounds which can serve as a benchmark for solution quality. Although it is rare that feasible solutions will be discovered, we often find that the solutions obtained by solving the Lagrangian problem are nearly feasible and can be made feasible with careful manipulations. We propose a two phase policy in order to convert nearly feasible solutions into near-optimal feasible solutions.

The first phase, *Make-Feasible*, described in Algorithm 1, is designed to alter an infeasible solution for (15) into a feasible one. *Make-Feasible* uses the solution vector $\boldsymbol{C}$ of (15) and calculates $\bar{M}_i, \forall i \in V$, the number of radios required by each node to make the assignment $\boldsymbol{C}$ feasible. Since the radio-usage $\boldsymbol{X}$ and the channel-assignment $\boldsymbol{C}$ are solved as two independent problems in (15), $\bar{M}_i$ may or may not be lesser than M_i. If $\bar{M}_i < M_i$ the algorithm does not alter the channel assignment for links corresponding to i. However, every node whose $\bar{M}_i > M_i$ is added to a List.

Algorithm 1 Make-Feasible (Graph G)

Require: $\boldsymbol{C}$: Channel Assignment Matrix
  K^* : Least used channel in $\boldsymbol{C}$
  $\boldsymbol{B_i}$: Sorted List of channels by $\bar{\boldsymbol{\beta}}$, for each node i
Ensure: A Feasible Solution C*
  List $\Leftarrow \phi$
  for all Node $i \in V$ **do**
    Compute $\bar{M}_i$ for the given $\boldsymbol{C}$
    if $\bar{M}_i > M_i$ **then**
      Add-to-list(List, node i)
    end if
  end for
  Unassigned $\Leftarrow \phi$
  while List $\neq \phi$ **do**
    i $\Leftarrow$ Element in List with maximum $\bar{M}_i - M_i$
    $CurChnl \Leftarrow$ First(B_i)
    while $\bar{M}_i = M_i - 1$ **do**
      Choose $j : (i,j) \in E$ and $C_{i,j}^{CurChnl} = 1$
      Add-to-list(List, node j)
      Unassigned $\Leftarrow$ Unassigned $\bigcup$ link (i,j)
      $C_{i,j}^{CurChnl} \Leftarrow 0$
      OldR $\Leftarrow \bar{M}_i$
      Recompute $\bar{M}_i$
      if $\bar{M}_i < OldR$ **then**
        $CurChl \Leftarrow$ Next(B_i)
      end if
    end while
    Remove-from-list(List, node i)
  end while
  for all link $(i,j) \in$ Unassigned **do**
    $C_{i,j}^{K^*} \Leftarrow 1$
  end for
  $\boldsymbol{C}^* \Leftarrow \boldsymbol{C}$

For every node i in the List, *Make-Feasible* performs the following tasks

- "Unassigns" the channels for the links (i,j) until $\bar{M}_i = M_i - 1$. Channels chosen for un-assignment are done in increasing order of $\bar{\boldsymbol{\beta}}$. This is done to merge the solutions for $\boldsymbol{C}$ and $\boldsymbol{X}$ as closely as possible.
- For each unassigned link (i,j), add the node *j* to the List if it not already present.

Once the list is empty, *Make-Feasible* assigns all the unassigned channels to a pre-defined default channel K^*, the channel with least number of total assignments in $\boldsymbol{C}$.

The second phase uses the solution determined by Make-

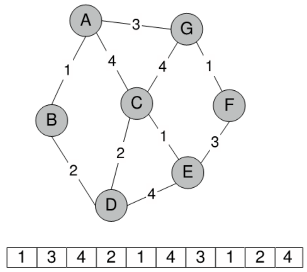

Fig. 2. Chromosome representation for four channels.

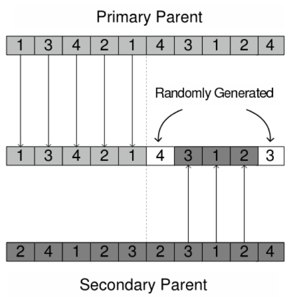

Fig. 3. Structural crossover for the modified GA.

Feasible to obtain a strong upper-bound for (12) taking into account the network interference. In this phase, we sort the interfering links in decreasing order of the network load values $L_{i,j}$. For every link (i, j) in the sorted list, we choose a random feasible channel that causes link (i, j) to not interfere. If no such channel exists, the channel assignment for link (i, j) remains unaltered.

Though the heuristic described above has a worst case running time of $O(|K||E|ln|E|)$ per iteration, we often find the solutions to the Lagrangian problem as near feasible requiring very few manipulations.

V. Genetic Algorithm

The solutions obtained by the Lagrangian relaxation method may not be competitive for very large networks. In this section, we propose a modified GA which can obtain good quality solutions of the channel assignment problem for very large networks. GAs are population-based stochastic search and optimization approaches inspired by the mechanism of natural selection which obeys the rule of "survival of the fittest" [27]. GAs have been extensively used for solving various real-world complex optimization problems due to their broad applicability, ease of use and global perspective [28].

The channel assignment of the graph's edges is represented as an ordered non-binary string, y_1,y_2...$y_{|E|}$ where $|E|$ is the number of edges on the graph and each $y_i \in \{1, 2..K\}$. Here y_i represents the channel assigned to the i^{th} edge. Figure 2 illustrates a possible assignment scheme for a graph with 10 edges. Since this representation does not ensure the feasibility of a solution, we modify our encoding scheme as follows.

Algorithm 2 Select-Channel (node i)

Require: L^* : List of unassigned links $(i, j) \in I_i^{'}$
  C^* : Channel assignment matrix of links $(i, j) \in I_i^{'}$
  O_i : Ownership Set of links owned by i
Ensure: (min_k, e) : Most suitable channel min_k for an unassigned link e.
  $min \Leftarrow \infty$
  $min_k \Leftarrow$ Default channel
  $e \Leftarrow$ Random link in L^*
  for all $k \in K$ **do**
    $OF(k) = \sum_{(i,j)\in I_i - e} Y_{i,j} L_{i,j} + Y_e(k) L_e$
    if $OF(k) < min$ **then**
      $min_k \Leftarrow k$
      $min \Leftarrow OF(k)$
    end if
    Remove-from-list(L^*, e)
  end for

Initially all radios are assumed to be untuned and all channels unassigned. A link is selected at random and a random feasible channel k is assigned to it. The channel k must satisfy one of the following conditions.

- Both transceivers contain a radio that is already tuned to k.
- One transceiver is already tuned to k while the other still contains atleast one untuned radio.
- Both transceiver are not tuned to k and both still contain atleast one untuned radio.

If no such channel exist, the process is restarted.

We propose modified crossover and mutation operators that are specific to the structure of this problem. While simpler GA operators exist, they are not constraint satisfying. Other approaches like [29] to handle constrained optimization problems with penalty functions use a large number of scaling factors which require tedious experiments for evaluation.

The mechanism of structural crossover is illustrated by Figure 3. One parent is called a primary parent while the other is termed as the secondary parent. A random point is chosen to break the strings. The channel assignments from the primary parent are maintained in the child. However, for the secondary parent, channels are chosen to ensure that the resulting offspring bears maximum structural similarity to its parents, in other words, it contains as many common channels as its secondary parent as possible, while maintaining feasibility. The roles of the parents are reversed to generate the second offspring.

For structural mutation, the algorithm generates a random link (i, j) and switches its channel to another random feasible channel. If no such channel exists, the channel assignment is unaltered. Our GA uses roulette wheel selection and the fitness function used to evaluate the individuals is the same as the objective function described in (6).

VI. Distributed Algorithm

In order to solve the channel assignment problem practically, we propose a fully distributed algorithm based on hill climbing. Our proposed algorithm requires that each node i maintains the following information:

- $I_i^{'}$: Interference set $\bigcup_{(i,j)\in E} I_{i,j}$.
- $\boldsymbol{C}^*$: Channel Assignment matrix for all nodes in $I_i^{'}$
- $\boldsymbol{X}^*$: Radio Usage matrix of all neighbours.

Each node $i \in V$ must assign channels to each link incident on it. To ensure consistent convergence, we define that, each link (i, j) be owned by either node i or node j depending on whichever node has a greater cumulative total expected traffic. We propose a two phase policy for assigning each link. The first phase, called *Select-Channel* while the second is a detailed protocol for assigning each channel chosen by *Select-Channel*.

The first phase, *Select-Channel*, as explained by Algorithm 2, is a method by which each node chooses the channel for each link owned by it, that causes the best improvement in the local interference. Initially, all channels are assumed to be unassigned. For each node i, *Select-Channel*, chooses a random unassigned link (i, j), owned by i, from $I_{i,j}^k$ and assigns a channel *k* to it based on the following conditions.

- Changing the channel of link (i, j) to k does not violate the interface constraints at nodes *i* and *j*.
- The binding of channel k to the link (i, j) causes the largest possible decrease in the local objective function.

The second phase of our algorithm describes a detailed protocol for assigning a channel. For example, if node $i \in V$ wants to tune a link (i, j) owned by it to a channel k, it sends a *Channel-Request* message to all its neighbors. Node j responds with a *Channel-Approve* message provided the channel assignment requested by i does not violate the interface constraints otherwise it sends a *Channel-Reject* message. If either response is not received by node i, it waits for a random period of time before retransmitting.

If the node i receives a *Channel-Approve* message, it sends a *Channel-Assign* message to all members of its interference set indicating that it is tuning all transmissions over link (i, j) to channel k. To minimize errors due to packet loss, each of its neighbours retransmits the packet.

If the node i receives a *Channel-Reject* message, it eliminates the channel k and chooses another channel that satisfies the above mentioned conditions. In order to ensure that our distributed algorithm converges, each link can be assigned a channel by its owner node only once

Note: In Algorithm 2, $Y_e(k) = 1$*, if link* e *interferes on channel* k*.*

VII. Experimental Results

We conduct extensive experiments to evaluate the performance of our algorithms in large networks. The network topologies used in our experiments are randomly generated. The number of channels $|K|$ vary from 3 to 8 while the number of radios M_i of each node *i* is a random quantity between 2 and $|K|$.

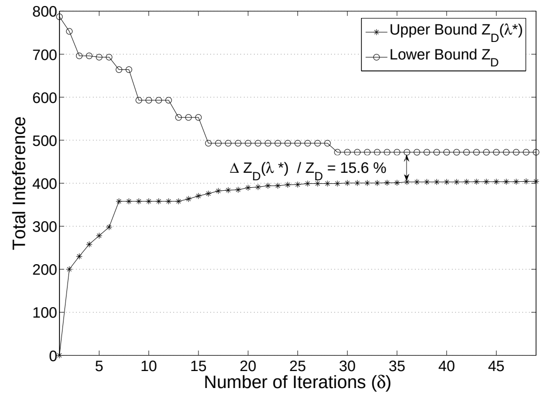

Fig. 4. Convergence of $Z_D(\boldsymbol{\lambda}^*)$ and Z_D.

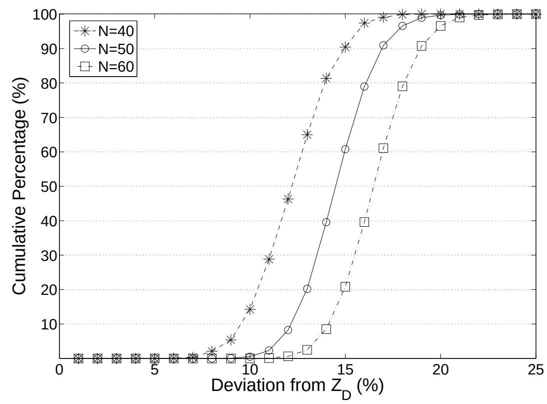

Fig. 5. Deviation of $Z_D(\boldsymbol{\lambda}^*)$ from Z_D.

A. Lagrangian Estimates

In this section, we report our results for 10,000 independent experiments for three different network sizes (N=40, 50, 60). Since our proposed solutions use randomized approaches, independent runs of the same experiment can converge to different solutions. Hence, each experiment consists of separate evaluations of the same topology.

Let $Z_D(\boldsymbol{\lambda}^*)$ denote the minimum upper bound Z* and Z_D the maximum lower bound $Z_d(\boldsymbol{\lambda})$ in (IV-A). The average deviation of $Z_D(\boldsymbol{\lambda}^*)$ from Z_D for all network sizes was calculated to be 15.34%. Figure 4 shows a typical plot of convergence of the two bounds.

Figure 5 explains the effect of the network size on the quality of the bounds. We observe that for N=40, over 80% the solutions are within 13% of Z_D, and in larger networks within 17%. This indicates that $Z_D(\boldsymbol{\lambda}^*)$ is consistent and can serve as a good benchmark for performance evaluation of our distributed algorithms for this range of network sizes.

Another interesting observation is that the percentage deviation of $Z_D(\boldsymbol{\lambda}^*)$ from Z_D steadily increases with the network size. The average deviation increases from 12.35% for N=40, to about 16.5% for N=60. Hence for much larger networks, we expect the genetic algorithms to perform as well as the

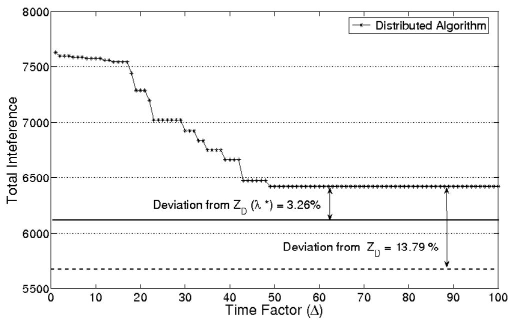

Fig. 7. Convergence of the distributed algorithm.

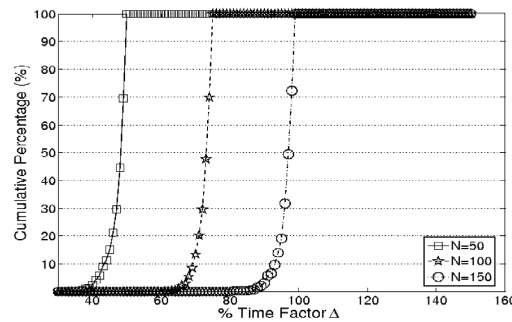

Fig. 8. Convergence times for the distributed algorithm

Lagrangian heuristic in providing good quality solutions.

B. Performance Evaluation

The performance evaluation of our algorithms aim at observing the following quantities

- Deviation of the total interference from $Z_D(\boldsymbol{\lambda}^*)$.
- Scalability of the algorithms for very large network topologies.
- Channel-wise distribution of the total interference.

To measure the quality of our algorithms, we compare the total interference calculated by the distributed and genetic algorithms with $Z_D((\boldsymbol{\lambda}^*)$. We observe that, for over 30,000 experiments, less that 5% of the solutions were better than the $Z_D(\boldsymbol{\lambda}^*)$, this further emphasizes the quality of these upper-bounds. Figure illustrates the percentage deviation of our proposed schema from $Z_D(\boldsymbol{\lambda}^*)$. For N=40, we observe that over 75% of the solutions are within 5% of $Z_D((\boldsymbol{\lambda}^*)$. In addition to that almost 10% and 20% of the solutions for the genetic and distributed algorithms respectively are within 0.1% of $Z_D(\boldsymbol{\lambda}^*)$. We can clearly observe a trend with increasing network sizes, i.e N=50 and N=60, the percentage of solutions for the genetic algorithms within 0.1% of $Z_D(\boldsymbol{\lambda}^*)$ rises steadily and reaches as high as 18% for the N=60 case. We expect that the genetic algorithm can outperform even the Lagrangian heuristics for very large networks of over 300 nodes. We can also observe that the distributed algorithm can serve as steady accurate and scalable solution to the channel assignment problem.

Investigating the convergence property can offer good insight into the scalability of the distributed algorithm. We conduct extensive simulations for very large network sizes. A typical plot of the convergence of the distributed algorithm is shown in Figure 7. Here we report the experiments for 10,000 independent runs of three very large networks (N = 50, 100, 150). As mentioned earlier, the algorithm might take a different number of iterations to converge for independent runs of the same network topology. We account for this by running each experiment 10 times. The time factor Δ is the number of iterations required for convergence. Figure 8 clearly indicates that the algorithm converges very quickly even for very large network sizes thus demonstrating good strength in scalability.

One motivation behind our randomized approaches is to ensure an even channel-wise distribution of the network interference. This is another important measure of solution quality. To measure the quality of the solutions in terms of per-channel interference, we compare our algorithms with an ideally balanced network. We conduct 1000 different experiments with $|K| = 6$ and N=50. To eliminate the negative effects of random radio numbers, we fix $M_i = 4 \ \forall i \in V$. Figure 9 clearly illustrates that the deviation from an ideal channel distribution does not exceed 5% for the distributed as well as the genetic algorithms. This can be attributed to the randomized nature of both algorithms.

VIII. Conclusions and Future Work

Exploiting multiple channels can drastically improve throughput in multi-radio wireless mesh networks. However, this raises an important channel assignment issue for achieving efficient channel utilization. In this paper, we have proposed a load-based scheme for assigning channels to radio interfaces in multi-radio, multi-channel wireless mesh networks. We adopted a graph-theoretic approach and formulated an ILP model for describing the channel assignment problem mathematically and proved it to be NP-hard. We then applied the Lagrangian relaxation method to obtain lower bounds as well as near-optimal feasible solutions for large size networks. We also presented a meta-heuristic using a genetic algorithm which can yield good quality solutions for very large networks. We further proposed a fully distributed algorithm which allows us to tackle the channel assignment problem practically. Our extensive simulation experiments have demonstrated that the distributed algorithm performs competitively and thus can serve as a practical and scalable solution to the channel assignment problem. In our future work, we will extend the methodologies and algorithms proposed in this paper to address a more challenging problem of joint channel assignment and routing, which is currently under our investigation. We also plan to conduct a more comprehensive study to evaluate the performance of our proposed algorithms in a real world test bed.

Appendix

Theorem 1: The channel assignment problem is NP-hard.

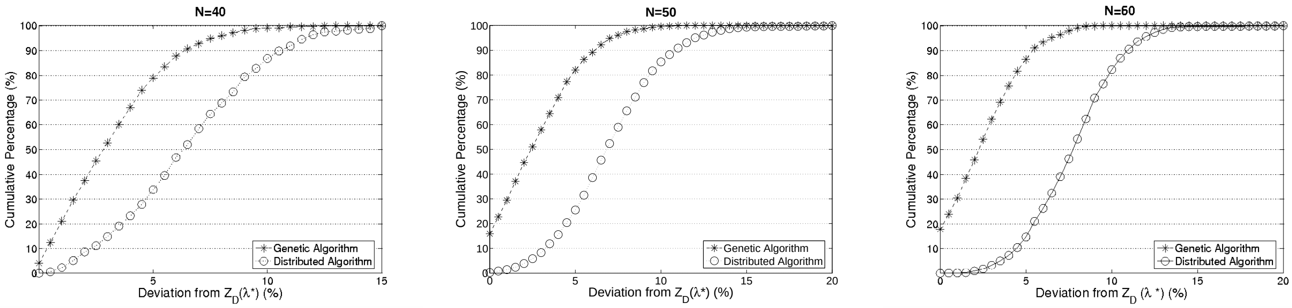

Fig. 6. Deviation of the algorithms from $Z_D(\boldsymbol{\lambda^*})$.

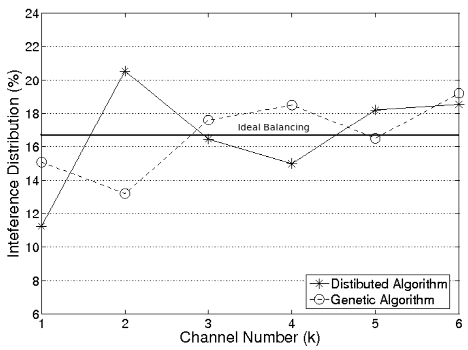

Fig. 9. Channel wise distribution of the interference function.

Proof: It is easy to see that the decision version Π of the channel assignment problem is in the class of NP. This is because a non-deterministic algorithm needs only to guess a channel assignment solution binding radios to channels and check in polynomial time if it satisfies all constraints of the channel assignment problem.

In order to show that the channel assignment problem is NP-hard we reduce a specific version of the problem to the well known problem of graph coloring . Let us assume the Load Matrix $\boldsymbol{L}$ such that $L_{i,j} = 1 \forall (i,j) \in E$. We further assume that the number of radios M_i, is equal to the degree of each node $i \in V$ in the graph G. The interference model assumed in this proof consists of all adjacent nodes $e \in V_c$ of the conflict graph G_c. Hence, we are required to minimize the number of nodes in G_c which have at least one color in common with their neighbors.

Let us assume there exists an algorithm $\boldsymbol{A}$ which can evaluate $f(G_c, K) = m$, where m is the minimum number of nodes in G_c which have at least one color in common with thier neighbors, in polynomial time. Now we can iteratively evaluate $f(G_c, K+1) = m^{'}$, $f(G_c, K+1) = m^{''}$ and so on until we get $f(G_c, K^*) = m^*$ where $m^* = 0$. Clearly K^* represents the chromatic number $\chi(G_c)$. Since $\boldsymbol{A}$ can evaluate f in polynomial time, the chromatic number, K^*, can also be evaluated in polynomial time. This is clearly a contradiction because graph coloring is NP-complete [31]. ■

ACKNOWLEDGMENT

This work was done when Srikrishna Sridhar was a visiting student at the Network Research Labs, The University of New South Wales.

REFERENCES

[1] A. Raniwala and T.-C. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *IEEE Infocom*, vol. 3, 2005, pp. 2223 – 2234.

[2] I. F. Akyildiza, X. Wang, and W. Wang, "Wireless mesh networks : a survey," *Computer Netowrks*, pp. 445–487, 2005.

[3] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on, 2000*, vol. 46, 2000.

[4] M. Marina and S. Das, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," vol. 1, 2005, pp. 381–390.

[5] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *IEEE Broadnets*, 2004, pp. 344 – 354.

[6] P. Bahl, R. Chandra, and J. Dunagan, "Connecting to multiple IEEE 802.11-based networks using a single wireless card," in *IEEE Infocom*, vol. 2, 2004, pp. 882–893.

[7] J. Shi, T. Salonidis, and E. Knightly, "Starvation mitigation through multi-channel coordination in CSMA based wireless networks," in *MOBIHOC*, 2006, pp. 214–225.

[8] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *ACM international symposium on Mobile ad hoc networking and computing*, 2004, pp. 222–233.

[9] S. Wu, C. Lin, Y. Tseng, and J. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for mobile ad hoc networks," in *ISPAN*, 2000.

[10] R. Chandra and P. Bahl, "Multinet: Connecting to multiple IEEE 802.11 networks using a single wireless card," in *IEEE Infocom*, 2004.

[11] A. Raniwala, K. Gopalan, and T.-C. Chiueh, "Centralized channel assignemnt and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE*, vol. 8, no. 2, pp. 50–65, 2004.

[12] S. Wu, Y. Tseng, C. Lin, and J. Sheu, "A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks," in *IEEE Computer Journal*, 2002, pp. 419 – 424.

[13] W. Hung, K. Law, and A. Leon-Garcia, "A dynamic multi-channel MAC for ad hoc LAN," in *Biennial Symposium on Communications*, 2002.

[14] P. Kyasanur and N. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *IEEE WCNC*, vol. 4, 2005, pp. 2051 – 2056.

[15] M. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 50, pp. 1861–1871, 1981.

[16] A.K.Das, H. Das, R.Vijayakumar, and S.Roy, "Optimization models for fixed channel assignment in wireless mesh networks with multiple radios," in *IEEE SENCON*, 2005, pp. 463–474.

[17] K. N. Ramachandran, E. M. Belding, K. Almeroth, and M. Buddhikot, "Inteference-aware channel assignment in multi-radio wireless mesh networks," in *IEEE Infocom*, 2006, pp. 1–12.

[18] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, and A. Rao, “Estimation of link interference in static multi-hop wireless networks,” 2005, pp. 28–32.
[19] B. Ko, V. Misra, J. Padhye, and D. Rubenstein, “Distributed channel assignment in multi-radio 802.11 mesh networks,” in *WCNC*, 2007, pp. 58–72.
[20] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh, “Exploiting partially overlapping channels in wireless networks : Turning peril into advantage,” in *ACM/USENIX Internet Measurement Conference*, 2004.
[21] A. P. Subramaniana, H. Gupta, and S. Das, “Minimum-interference channel assignment in multi-radio wireless mesh networks,” in *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON ’07.*, 2007, pp. 481–490.
[22] J. Hao and L. Perrier, “Tabu search for the frequency assignment problem in cellular radio networks,” 1999.
[23] The CoMo Project, http://www.como.intel-research.net/.
[24] A. Naveed, S. Kanhere, and S. Jha, “Topology control and channel assignment in multi-radio multi-channel wireless mesh networks,” in *IEEE Mobile Adhoc and Sensor Systems*, 2006, pp. 1–9.
[25] T. Liu and W. Liao, “On routing in multichannel wireles mesh networks : Challenges and solutions,” *IEEE Network*, vol. 22, pp. 13–18, 2008.
[26] M. Held, P. Wolfe, and H. Crowder, “Validation of subgradient optimization,” *Mathematical Programming*, vol. 6, pp. 62–88, 1974.
[27] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.
[28] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
[29] Z. Michalewicz, “Genetic algorithms, numerical optimization, and constraints,” in *Sixth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1995, pp. 151–158.
[30] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
[31] M. Held, P. Wolfe, and H. Crowder, “Reducibility among combinatorial problems,” *Complexity of Computer Computations*, pp. 85–103, 1972.