Telecommunication Systems 22:1–4, 33–59, 2003

On the Complexity of Distributed Self-Configuration in Wireless Networks

BHASKAR KRISHNAMACHARI * bkrishna@usc.edu
Department of Electrical Engineering – Systems, University of Southern California, Los Angeles, CA 90089, USA

STEPHEN WICKER wicker@ece.cornell.edu
School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853, USA

RAMÓN BÉJAR and CÈSAR FERNÀNDEZ {ramon, cesar}@eup.udl.es
Department d'Informàtica i Enginyeria Industrial, Universitat de Lleida, Jaume II, 69, E-25001, Lleida, Spain

Abstract. We consider three distributed configuration tasks that arise in the setup and operation of multi-hop wireless networks: partition into coordinating cliques, Hamiltonian cycle formation and conflict-free channel allocation. We show that the probabilities of accomplishing these tasks undergo zero-one phase transitions with respect to the transmission range of individual nodes. We model these tasks as distributed constraint satisfaction problems (DCSPs) and show that, even though they are NP-hard in general, these problems can be solved efficiently on average when the network is operated sufficiently far from the transition region. Phase transition analysis is shown to be a useful mechanism for quantifying the critical range of energy and bandwidth resources needed for the scalable performance of self-configuring wireless networks.

Keywords: self-configuration, wireless networks, distributed constraint satisfaction

Introduction

With recent advances in technology, it has become feasible to consider the deployment of large scale multi-hop wireless networks for a wide range of communication and sensing applications [Estrin et al., 14, 15; Haas et al., 21; Perkins, 35; Toh, 42]. Distributed self-configuration mechanisms are required to enable such networks to provide their desired functionality.

In this paper we will examine some self-configuration problems which relate to the formation of specialized structures on the network connectivity graph. It is well known that many graph problems, including those that we consider in this paper, are NP-complete [Garey and Johnson, 19]. This means that, in the worst-case, there are problem instances that will require computational and communication resources which increase exponentially with the size of the network.

* Corresponding author.

The most common approach, when faced with NP-hard problems, is to use algorithms that are not guaranteed to result in optimal solutions, or to generate the correct answer. These could be heuristic local search mechanisms [Michalewicz and Fogel, 33] or approximation algorithms [Hochbaum, 24]. In this paper we will focus instead on using complete, optimal algorithms, and try to identify the special conditions under which they can be used to solve the self-configuration problems efficiently. This methodology is influenced by recent work from the area of constraint satisfaction in Artificial Intelligence.

Researchers have found that there exists a critical ratio of constraints to variables in constraint satisfaction problems such as propositional satisfiability (SAT). Randomly generated problem instances with a ratio higher than this critical point are almost always unsatisfiable, while instances generated with a ratio lower than this critical point are almost always satisfiable [Cheeseman et al., 10; Mitchell et al., 34]. It has been shown that this "phase transition" in satisfiability is analogous to that which takes places in physical systems [Kirkpatrick and Selman, 25]. Furthermore, it turns out that the critical point also corresponds to a peak in the average computational cost. Problem instances which are well to the left and right of this transition are much easier to solve than those at the critical point.

There has also been some recent work in the area of multi-hop wireless networks suggesting the existence of similar critical points and phase transitions [Gupta and Kumar, 20; Krishnamachari et al., 28, 29; Sanchez et al., 38; 46]. Consider n nodes randomly placed in a given operational area. Let each node transmit with the same power, so that there is an effective communication range R within which any pair of nodes can communicate with each other. It has been found that there are critical communication ranges beyond which desired global network properties can be achieved with high probability.

This paper is primarily targeted at wireless networking researchers who are looking for a general methodology for implementing self-configuration and thinking about the complexity of distributed problem solving in these kinds of systems. We map out the connection between the critical power thresholds in wireless networks and the work on constraint satisfaction, and show through experiments that the average problem complexity can be reduced by appropriately tuning the transmission power of individual nodes. We present results on the following three NP-hard problems that are typical of self-configuration tasks in wireless networks:

- Partitioning the network into coordinating cliques – How does a collection of nodes partition itself into subgroups of completely interconnected nodes? Such problems arise in the design of sensor networks, where a collection of nodes is assigned the joint task of tracking a particular object.
- Hamiltonian cycle formation – How does a collection of nodes devise an ordering of links such that each node in the collection is visited exactly once? Such orderings are important in the creation of token ring topologies.

- Conflict-free channel scheduling – How does a collection of nodes jointly allocate the locally available spectrum in an efficient manner while avoiding conflicts between neighboring transmitters? This is the traditional problem of devising a "frequency reuse" strategy such that a given logical channel is efficiently used across a wireless network, while no two co-channel transmitters are close enough to interfere with one another.

Our first contribution is to formulate these tasks as distributed constraint satisfaction problems. This formalism makes the problems easy to solve in a distributed manner using off-the-shelf complete algorithms. Our second contribution is to show for each problem that the transmission power of the individual nodes is a control parameter with respect to which there is a zero-one phase transition in satisfiability. Finally, we establish through experiments that the computational complexity of solving the problem (i.e. finding a satisfying solution or showing that no such solution exists) undergoes an easy-hard-easy phase transition, with the hardest problems distributed near the critical threshold value. In order to design systems whose self-configuration problems are under-constrained and hence easy to solve, we need to engineer sufficient resources into the system, with "sufficiency" quantified in terms of the phase transition.

The rest of the paper is organized as follows. In section 1, we discuss the notion of phase transitions and critical thresholds for the existence of global properties in multi-hop wireless networks. In section 2, we provide background information on constraint satisfaction problems. In section 3, we discuss distributed constraint satisfaction problems (DCSPs). The notion of self-configuration and the need for suitable formalisms in wireless networks is discussed in section 4. We examine each of our self-configuration problems in section 5, formulate them as DCSPs, and present results on their complexity. Concluding comments are presented in section 6.

1. Phase transitions in wireless networks

Let us consider a wireless network of n nodes. We place these nodes randomly in a square area with a uniform, independent distribution. We use a reasonable first-order model for communication: any pair of nodes within a radius R can communicate with each other. Note the requirement that all nodes have the same communication radius is not restrictive, since this is in fact necessary for ensuring symmetric links within the network.

Figure 1 shows what happens as we increase the parameter R, for a particular configuration of $n = 10$ nodes. As expected, we see that the network graph becomes denser as the communication radius is increased. When the value of R is small, the network is quite sparse and does not form a single connected component. At the same time, one can see that this sparse network will exhibit low levels of interference. When the R is sufficiently high, we can get a complete network graph in which each node can communicate with (and at the same time interefere with) every other node. At some in-between value of R, the network becomes connected, allowing for a multi-hop path between each

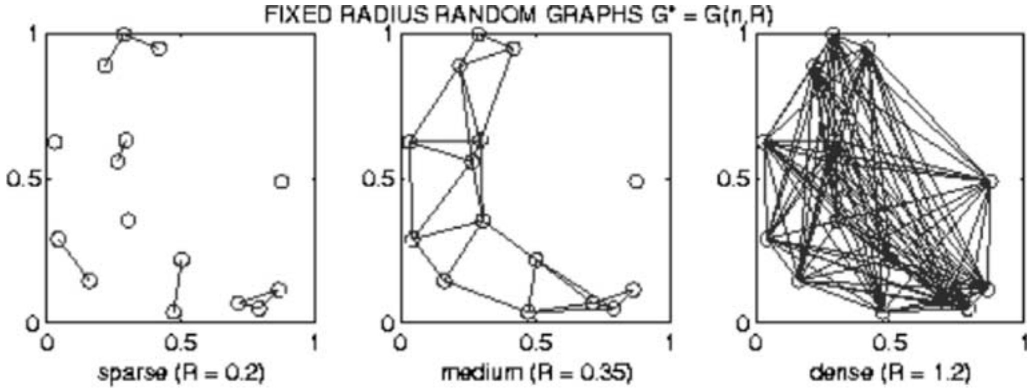

Figure 1. Fixed radius random graphs.

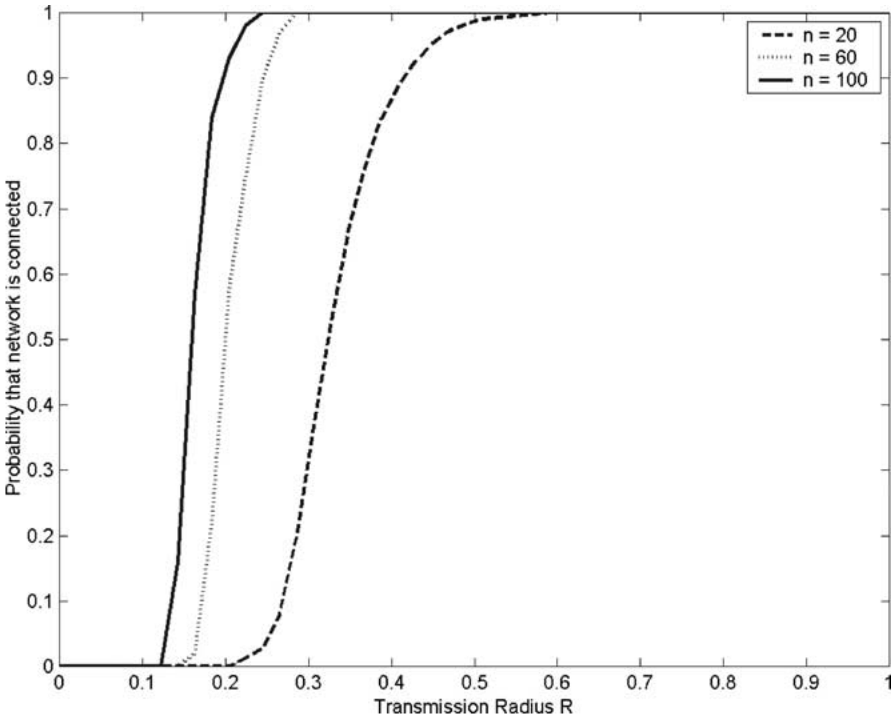

Figure 2. Phase transitions in probability of connectivity in fixed radius ad-hoc wireless networks.

pair of nodes. If we repeat this experiment, locating the nodes at random, we will get another value for the radius R at which the network became connected. For any value of R, based on this prescribed random experiment, we can evaluate the probability that the generated network is connected. It turns out that this probability transitions from nearly zero to nearly one over a small range of R values, with the transition becoming sharper with the size of the the network. This is shown in figure 2.

This phenomenon has been studied analytically by Kumar et al. [Gupta and Kumar, 20; 46]. Gupta and Kumar show in [Gupta and Kumar, 20] that if n nodes are placed uniformly and independently in a disc of unit area in $\mathbb{R}^2$, and each node transmits at a power level so as to cover an area of $\pi R^2 = (\log\ n + c(n))/n$, then the network is connected with probability asymptotically tending to one if and only if $c(n) \to \infty$. Xue and Kumar [46] show that for n nodes placed uniformly iid in a unit square, the network is connected with high probability when each node is connected to $\Theta(\log n)$ neighbors. These results have relied primarily on the theory of continuum percolation [Meester and Roy, 32].

Another subject that is useful in generalizing the notion of critical thresholds to other network properties besides connectivity is the theory of random graphs [Bollobás, 6]. Most of the work in this area has been done on Bernoulli random graphs, which are somewhat similar to the random graphs $G(n, R)$ that we have been using to represent wireless networks. In these graphs $G(n, p)$ we have n nodes, and there is a Bernoulli parameter p which is the independent probability of having an edge between each pair of nodes.

To quote Bollobás:

> "one of the main aims of the theory of random graphs is to determine when a given [graph] property is likely to appear ... Erdös and Rényi were the first to show that most monotone properties appear rather suddenly. In rather vague terms, a threshold function is a critical time, before which the property is unlikely and after which it is likely" [Bollobás, 6].

In 1996, Friedgut and Kalai proved that in fact *every* monotone graph property undergoes a sharp transition [Friedgut and Kalai, 17]. Let $\mu_p(A)$ be the probability that a monotone property A is satisfied by $G(n, p)$, and c_1 a universal constant, then the following is their result:

Theorem 1. If $\mu_p(A) > \varepsilon$, then $\mu_q(A) > 1 - \varepsilon$ for all $q > p + c_1 \log(1/2\varepsilon)/\log n$.

In [Krishnamachari et al., 28, 29] we have presented empirical evidence suggesting that properties which show phase transitions in $G(n, p)$ also show phase transitions in the random graphs $G(n, R)$ which represent wireless networks. From a networking perspective many of these properties are very important.

Phase transition analysis gives us a tool for analyzing and determining resource-efficient regimes of operation for wireless networks, with respect to a given global property. For example, if the global property is that of connectivity, figure 2 tells us that for a uniformly distributed network with a density of 100 nodes per unit area, the transmission power must be such that the effective communication range is more than 0.25 units (or, equivalently, that each node should have about $\pi(0.25)^2 99 \simeq 20$ neighboring nodes, ignoring edge effects). This density threshold is an energy-efficient point of operation, in that to the left of this threshold the network is disconnected with high probability, and to the right of this threshold, additional energy expenditure results in a negligible increase in the high probability of connectivity. The same is true for the phase transitions for other properties like k-connectivity, k-neighborhood, Hamiltonian cycle formation, and partition into cliques. Of course, it must be kept in mind that increasing the communication range not only makes the network graph denser, but also increases the level of interference in the network. As we shall see, this increased level of interference can make it difficult to allocate non-interfering channels to nearby nodes; the property of conflict-free channel assignment shows a reverse "one-zero" phase transition with respect to interference level. It is also important to analyze the intersection of thresholds for such conflicting properties as network connectivity and conflict-free channel allocation because this determines the feasible region of operation for a given wireless network.

In this paper, we offer yet another reason for examining such phase transitions – they also tell us about the operating conditions under which the computational and communication costs of distributed problem solving can be reduced. This brings us to the related work in the area of constraint satisfaction.

2. Constraint satisfaction problems

Constraint satisfaction is a formalism that has been used to model a large class of problems with applications in engineering design, planning, scheduling, resource allocation, and fault diagnosis [Dechter and Frost, 12]. A constraint satisfaction problem (CSP) is easy to understand. There are a number of variables, each of which has an associated domain of values. Constraints are specified on subsets of these variables restricting the set of values they can take on jointly. The objective of a CSP is to find out if each of these variables can be assigned a value from its domain in such a way that all the constraints are satisfied. It is helpful to consider the difference between a CSP and a constrained optimization problem (COP): while in a COP one wishes to find the lowest cost point in the search space which satisfies all constraints, in a CSP it suffices to find a single point in the search space which satisfies all constraints. A CSP is said to be *satisfiable* if there exists such a point, and *unsatisfiable* otherwise.

As an illustration, we briefly describe the original NP-complete problem – propositional satisfiability (SAT), which is a special kind of CSP [Garey and Johnson, 19]. Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of Boolean *variables*. Each variable x_i and its negation $\neg x_i$ constitute *literals*. A *clause* is a disjunction (OR) of one or more literals (e.g., $(x_1 \vee \neg x_2)$) and is said to be *satisfiable* if there exists some *truth assignment* of $0/1$ values to all variables such that at least one of its literals evaluates to true under that assignment. Two special cases are the unit clause, represented (l), that contains only one literal, and the *empty clause*, represented $(\square)$, which contains no literals and is by definition unsatisfiable. A conjunctive normal form (CNF) formula over X consists of the conjunction (AND) of a number of clauses, and is said to be satisfiable if there exists some truth assignment to the variables in X such that all the clauses are satisfied. An instance of SAT consists of a CNF formula Γ, and the goal is to determine if there exists a satisfying truth assignment for Γ. For example, the formula $\Gamma = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$

is satisfied by setting both x_1 and x_2 to 1; the formula $(x_1) \wedge (\neg x_1)$ is unsatisfiable since only one of the clauses can be satisfied by setting x_1 to either 0 or 1. Note that the clauses in CNF formulae represent constraints on the Boolean variables. k-SAT refers to a special case of SAT in which all clauses have exactly k literals.

In the early 90's, researchers in Artificial Intelligence found empirically that for many CSPs including SAT, as the ratio of constraints to variables is increased, the fraction of (randomly generated) instances that are satisfiable undergoes a one to zero phase transition [Cheeseman et al., 10; Kirkpatrick and Selman, 25; Mitchell et al., 34]. Further, they found that the computational cost of determining whether or not an instance is satisfiable shows an easy-hard-easy pattern, with the complexity peaking in the phase transition region. Such an empirical result for randomly generated 3-SAT problems is shown in figure 3. The plot illustrates that it is easy to solve CSPs when they are under-constrained, and easy to show that they have no solution when they are over-constrained. The hardest instances lie in the critically-constrained phase transition region.

In recent years, a number of analytical results have been developed to support these empirical findings. In 1999, Friedgut showed that for k-SAT, there exists a constant c_k such that all formulas with at most $(1-\varepsilon)c_k n$ clauses are satisfiable with high probability (i.e. with probability tending to one as n approaches infinity) and formulas with at least $(1+\varepsilon)c_k n$ clauses are unsatisfiable with high probability [Friedgut, 16]. Particular attention has been focused on 3-SAT, for which empirical evidence suggests that the $c_3 \approx 4.24$ (see figure 3): it has been shown analytically that $3.145 \leq c_3 \leq 4.506$ [Achlioptas, 1; Dubois et al., 13]. Analytical results on the complexity profile have been harder to obtain. It is known that 3-SAT remains NP-complete even if the instances are restricted to ratios between $1/3$ and $7(n^2-3n+2)$. In terms of the average complexity, however, Frieze and Suen [18] have shown that there exists a polynomial heuristic which can find satisfying solutions with high probability for instances with the ratio less than 3.003, and on the other side of the phase transition, Beame et al. [3] have shown that one can prove unsatisfiability in polynomial time with high probability if the ratio is more than $n/\log n$. A first-order algorithm-independent analysis of the deep structure of constraint satisfaction problems by Williams and Hogg shows that indeed the average computational cost should be expected to peak near the phase transition threshold [Williams and Hogg, 44].

2.1. A complete algorithm for satisfiability

Complete algorithms are frequently used to study the complexity of CSPs. An CSP algorithm is complete if it provides a satisfying solution whenever the CSP has one, or else determines that the problem is unsatisfiable. DLL is a complete algorithm that is frequently used for solving SAT problems [Davis et al., 11]. It is based on the use of the following two rules:

- *Unit-propagation rule:* Given a CNF formula Γ containing a unit clause $\{l\}$:
  1. Remove all clauses containing the literal l. When all the clauses from a formula are removed through application of this rule and the empty formula $\emptyset$ is gener-

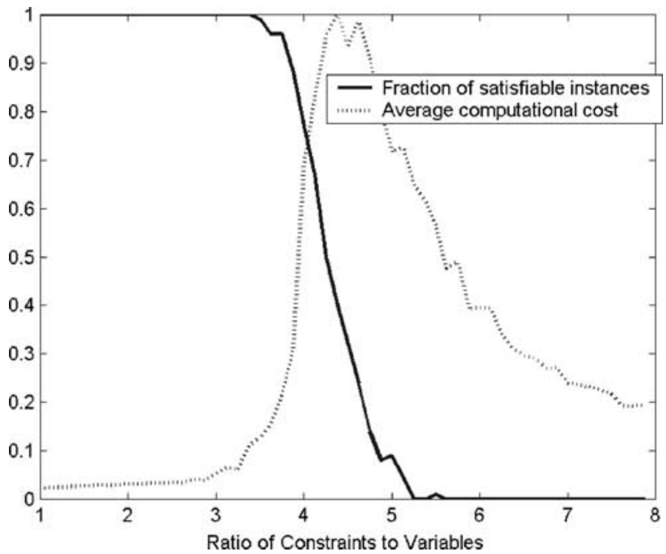

Figure 3. Phase transitions in the fraction of satisfiable problems and the average complexity for 3-SAT with 40 variables using a complete algorithm.

ated, all the clauses have been satisfied, and we have a solution to the original expression.

2. Delete all occurrences of the complementary literal $\neg l$ in clauses of the formula (by the rule of the excluded middle, the complementary literal cannot be satisfied). This portion of the unit-propagation rule can produce new unit clauses, since we may delete a literal from a clause with two literals. The unit-propagation rule should be applied again with the new unit clauses.

- *Branching rule:* Reduce the problem of determining whether a CNF formula Γ is satisfiable to the problem of determining whether $\Gamma \cup \{l'\}$ is satisfiable or $\Gamma \cup \{\neg l'\}$ is satisfiable, where l' is a literal occurring in Γ.

The unit-propagation rule can be seen as a simplification rule, while the branching rule is a splitting rule that divides the problem into two subproblems. DLL returns true if the input CNF formula Γ is satisfiable, and false when the formula is unsatisfiable. First, it repeatedly applies the unit-propagation rule, until there are no more unit clauses, resulting in a simplified formula Γ'. It then selects a literal l' of Γ', applies the branching rule and recursively solves the problem of deciding whether $\Gamma' \cup \{l'\}$ is satisfiable or $\Gamma' \cup \{\neg l'\}$ is satisfiable. As such subproblems contain a unit clause, the unit-propagation rule can be applied again. DLL terminates either when some subproblem is shown to be satisfiable by obtaining the empty CNF formula or when all the subproblems are shown to be unsatisfiable by deriving the empty clause ($\square$) in all of them. The empty clause is derived when the unit-propagation rule deletes the unique literal of a unit clause.

The application of the branching rule can be interpreted as the construction of a search tree. Although the DLL algorithm only works for the SAT problem, there exist similar complete search algorithms that work for more general CSPs [Dechter and Frost, 12]. Other alternatives to complete algorithms are stochastic local search algorithms that obtain the solution through a series of local, randomized, moves through the search space [Selman et al., 40]. Local search algorithms are often faster at solving satisfiable instances, but cannot detect if a problem has no solution, and are not always guaranteed to find the solution even if one does exist.

3. Distributed Constraint Satisfaction Problems (DCSPs)

DCSPs extend the constraint satisfaction formalism to the framework of distributed problem solving [Yokoo et al., 47]. They provide a good formalism for modelling and reasoning about constraint satisfaction problems that are *per se* of a distributed nature, where there is no easy or practical way to solve them in a centralized manner. In a DCSP, there is a set of n agents $A = \{1, 2, \ldots, n\}$. Each agent has its own variables with associated domains. There are *intra-agent* constraints between the variables of each individual agent, and *inter-agent* constraints between the variables of different agents. A satisfying solution to the DCSP is an instantiation of values to the variables of each agent such that every intra and inter-agent constraint is satisfied.

To satisfy the inter-agent constraints in a DCSP, agents need to use some communication mechanism for exchanging the values of their variables with other agents. Therefore in DCSPs, a communication cost is added to the computational effort associated with a centralized CSP. In the realm of multi-hop network design, this is an important consideration as each communicated message incurs some radio energy cost. One measure of the communication complexity for a DCSP is the total number of messages exchanged by the agents in order to solve the problem or to detect that no solution exists. Very often the communication complexity is proportional to the computational complexity, which can be measured by the time required to solve a problem instance.

Figure 4 gives an example of a satisfiable DCSP (one that has at least one solution). This DCSP consists of three agents, with one binary variable for each agent. The inter-

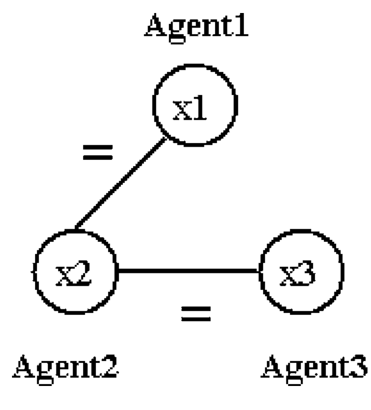

Figure 4. Satisfiable DCSP.

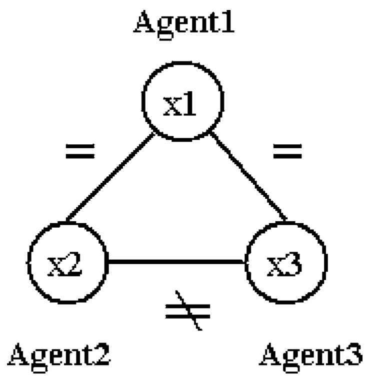

Figure 5. Unsatisfiable DCSP.

agent constraints are represented in the figure as edges with a binary relation symbol. The relation symbol specifies the relation that must hold between the variables of the two connected agents. A possible solution for this DCSP is for all agents to set the same value (0 or 1) to their variables. Figure 5 gives an example of an unsatisfiable DCSP. There is no possible solution for this DCSP, because the fact that $x_1 = x_2$ and $x_1 = x_3$ must be true *implies* that $x_2 = x_3$ should also be true, which would violate the inter-agent constraint between agents 2 and 3.

3.1. Complete algorithms for solving DCSP

A number of complete algorithms have been developed for solving DCSPs, such as the asynchronous backtracking algorithm (ABT), the asynchronous weak commitment search (AWC), and the distributed backtracking algorithm (DIBT) [Hamadi et al., 23; Yokoo et al., 47]. Basically these algorithms generalize centralized backtracking search to the distributed setting. Because we are working in an asynchronous environment (we assume no central control in a flat, multi-hop network) the agents decide for themselves when to change the values assigned to their variables. At the beginning, all the agents choose a value for all their variables such that their intra-agent constraints are satisfied (they can achieve this using any existing centralized CSP algorithm). Before the search can proceed, it is necessary to assign a unique identifier number to every agent. This identifier is used to establish a priority order between agents, such that one agent has a greater priority than other if its identifier is smaller. Given an inter-agent constraint, one between two agents, the higher priority agent may change the values of those variables in the constraint that belong to him. It must inform the other agent about any change to the variables by sending an *information* message. When the other agent receives the information message, it must try to find an assignment to its own variables such that all the inter-agent constraints that it has with higher priority agents, and its own intra-agent constraints, are satisfied. If it changes the value of some of its variables, it will send information messages to all lower priority agents with whom it has inter-agent constraints. However, if it is unable to change the values of its variables, it will send a *backtracking* message to the lowest priority agent among all its higher priority agents that have an unsatisfied inter-agent constraint. This message tells the higher priority agent that it must try to find a different value for the variable that is causing a conflict with the lower priority agent, because the latter cannot do anything to resolve the conflict.

To ensure completeness, the algorithms must never revisit any previously considered "bad" solution, and never fail to explore a potentially good solution. For example, agents in ABT record *nogoods*, each of which may correspond to several inconsistent assignments.

To a first approximation the complexity of a DCSP solver is proportional to the complexity of a CSP solver. Studying the computational complexity of a CSP using a centralized algorithm provides a strong indication of the communication and computational complexity of a distributed version of the same problem. We will make use of this relation in presenting our results on the complexity of DCSPs that arise in wireless networks.

4. Self-configuration in wireless networks

Multi-hop wireless networks for communication and sensing are characterized by a lack of centralized pre-configured infrastructure. This is chiefly due to the ad hoc and possibly unattended nature of their deployment, as well as the requirement of scalable performance. Under such conditions, it is necessary for the wireless nodes to first collaborate with each other and *self-configure* themselves into a functioning network before they can perform their principal information routing and dissemination tasks.

The recent literature has pointed out the importance of developing self-configuration protocols for this space of networking applications [Estrin et al., 15; Haas et al., 21]. For example consider the treatment of node localization as an adaptive distributed self-configuration problem in [Bulusu et al., 7] and the set of self-organizing algorithms for configuring and maintaining a multi-hop wireless network described in [Sohrabi et al., 41].

One of the standing challenges in this domain has been to come up with suitable general, unifying formalisms for distributed self-configuration. Such formalisms would (a) simplify the efficient implementation of self-configuration protocols and (b) provide a systematic mechanism for studying complexity issues and identifying relevant tuning parameters.

We argue in this paper that one such unifying formalism is the notion of distributed constraint satisfaction. We will show in the next section that a number of distinct self-organization tasks in multi-hop networks can be mapped to this formalism, enabling the use of off-the-shelf complete distributed solvers. More importantly, these mappings enable us to leverage results from the area of constraint satisfaction in order to determine parameters which impact the efficiency of problem solving. We shall see that there are tunable individual node-level parameters that can have a critical impact on the solvability and complexity of such tasks in multi-hop wireless networks.

5. Distributed constraint satisfaction in wireless networks

We now consider three specific problems in distributed wireless networks: the partitioning of nodes into coordinating cliques, the formation of Hamiltonian cycles, and conflict-free channel scheduling. These problems are all known to be NP-hard, so unless P = NP, we expect the communication and computational complexity in these problems to be exponential in the worst case. A clear understanding of the complexity of these tasks is important if we wish to incorporate the tasks into self-configuring multi-hop wireless networks.

In this section we formalize these tasks as distributed constraint-satisfaction problems and show that they each have a "complexity-tuning" parameter over whose range they exhibit a 0–1 phase-transition in the probability of being satisfiable, and a corresponding easy-hard-easy profile in average case complexity. Most interesting from the view-point of application is the fact that in each case we can move the system into the easy and satisfiable portion of the transition curves by adding resources (in the form of additional bandwidth or energy). This is the region under which the communication and computation complexity is the lowest and the distributed problem solving task can be performed most efficiently.

5.1. Partition into coordinating cliques

In wireless sensor network, sensing or other tasks may need to be distributed among the various nodes organized together as coordinating groups. One such example is in the task of monitoring the environment for a pre-specified phenomenon. If several nodes are selected to perform this task together, it may be desirable that these nodes form a communication clique. In other words, any node in the coordinating group should be able to communicate directly over the wireless link with any other. For example, such a situation arises in the sensor tracking where it is required that a number of nodes participate and coordinate their actions jointly in order to track mobile nodes [Bejar et al., 4]. The partitioning of nodes into such cliques has other applications in multi-hop wireless networks, such as cluster formation [Cano and Manzoni, 8] and geography-informed routing [Xu et al., 45].

Let us consider this problem further. Given $n = qk$ wireless nodes, each with a transmitting radius R forming the network graph $G = G(V, E)$, the objective is to partition the graph into q communicating cliques of size k each. This can be formulated as a DCSP as shown in table 1. Each node has an associated agent which has $k - 1$ variables $\{x_{i,1}, \ldots, x_{i,k-1}\}$ which can each take on values from 1 to n. The variable $x_{i,l}$ represents the lth of node i's neighbors in the clique it will be part of. The DCSP formulation allows us to represent a global problem in terms of local variables and constraints. When each of these variables is assigned a value that satisfies all constraints, each node will have a local representation of the coordinating clique to which it belongs. If the DCSP is unsatisfiable, the network cannot be partitioned into distinct cliques of size k.

Figure 6 shows an unsatisfiable instance of this problem on a sparse network consisting of nine nodes which is to be partitioned into three coordinating cliques of size

Table 1
DSCP formulation for partitioning network into coordinating cliques.

Agent i	Node i
Variables of agent i	$x_{i,l}, l = 1, 2, \ldots, k-1$
Domain of variable $x_{i,l}$	$\{1, \ldots, n\}$
Intra-agent constraints	1. (Only neighbors in clique) $x_{i,l} = j \iff (i, j) \in E, i \neq j$
	2. (Uniqueness) $\forall l \neq l', x_{i,l} \neq x_{i,l'}$
Inter-agent constraints	(Symmetry) $\forall (i, j) \in E, \exists l$ s.t. $x_{i,l} = j$
	$\iff \exists l'$ s.t. $x_{j,l'} = i$ AND $\forall (m \neq l), \exists m'$ s.t. $x_{i,m} = x_{j,m'}$

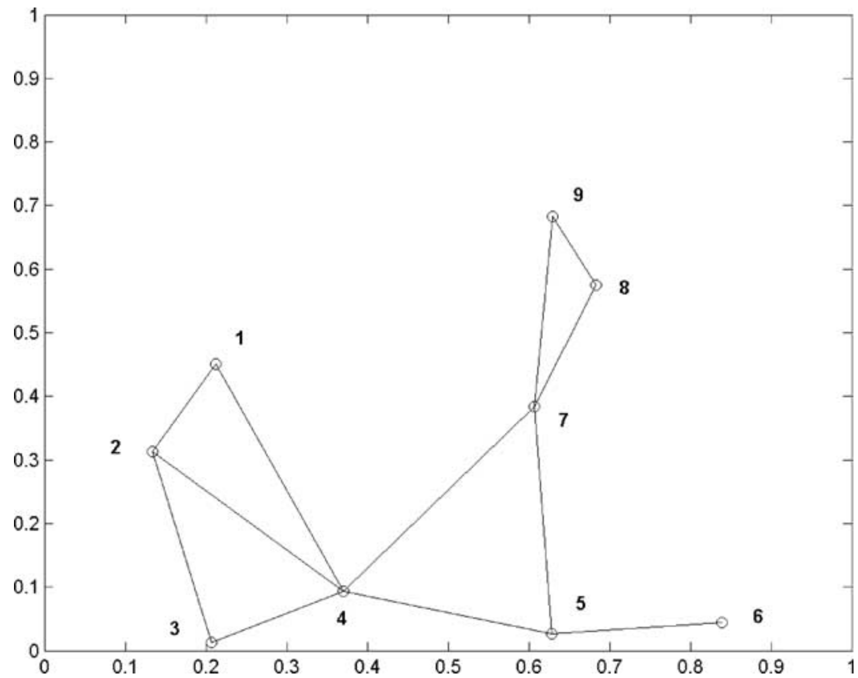

Figure 6. Unsatisfiable partition into coordinating cliques with small transmission radius ($n = 9$, $k = 3$, $R = 0.40$).

three each. This instance is clearly unsatisfiable because node 6 has only one neighbor and hence cannot communicate/coordinate with two other nodes. If we increase the transmission radius of each node, we get a denser network graph as shown in figure 7. This graph represents a satisfiable instance of the problem. The dashed edges represent one possible partition of the graph into three 3-cliques. Also shown in the figure is the corresponding, satisfiable, value assignment to the variables of each node agent.

The problem of partitioning a graph into isomorphic subgraphs is known to be NP-hard for any connected subgraph with more than 3 nodes [Garey and Johnson, 19]. For a given set of nodes positioned arbitrarily, the difficulty of obtaining a partition in

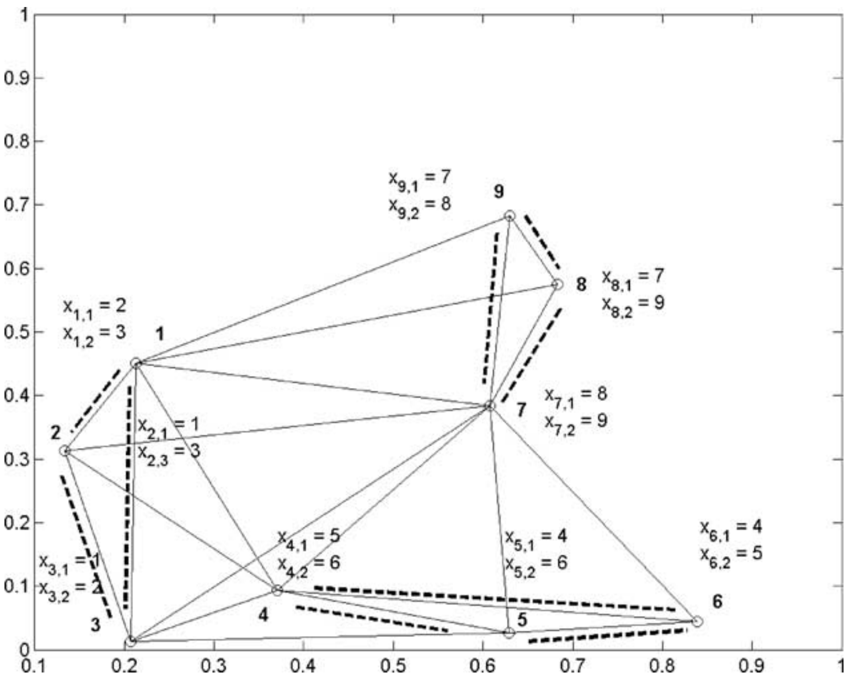

Figure 7. Satisfiable partition into coordinating cliques with large transmission radius ($n = 7$, $k = 3$, $R = 0.55$).

this problem is dependent on the density of the network graph, which in turn is affected directly by the transmission radii of the nodes. Figure 8 shows the phase transitions in both probability of partition and the average complexity profile for this problem based on 100 problem instances ($n = 9$, $k = 3$) for each value of the transmission radius R ranging from 0 to $\sqrt{2}$. The solutions were computed using a Regular-SAT centralized backtracking solver [Bejar et al., 5]. It can be seen that there is a critical transmission power level above which the problem has a satisfying solution with high probability and below which there is rarely a satisfying solution. The computational complexity is seen to peak near the phase transition region. If we operate the network sufficiently to the right of the phase transition, the average computational cost for this problem can be significantly reduced.

We now turn to our second sample problem.

5.2. Distributed Hamiltonian cycle formation

Consider the following task in a wireless sensor network: a set of nodes that form a connected network component wish to form a Hamiltonian cycle in a distributed manner. Recall that in a Hamiltonian cycle, each node in the graph is visited exactly once. The formation of such a cycle is useful, for example, when forming a token ring in the network, and also forms the basis of some other distributed algorithms such as leader selection [Lynch, 31]. Another application is in optimal one-to-one broadcasting where

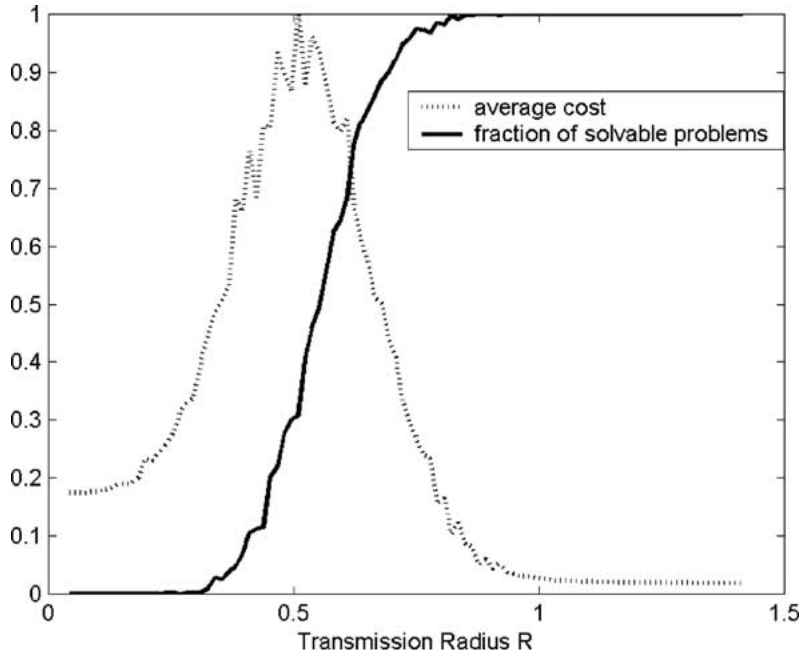

Figure 8. Phase transitions in the fraction of satisfiable problems and the average complexity for the problem of partitioning a network into coordinating cliques using a complete search algorithm with a simple pruning heuristic.

Table 2
DSCP formulation for Hamiltonian cycle formation.

Agent i	Node i
Variables of agent i	$FROM_i$, TO_i, and $HOPCOUNT_i$
Domain of variables	1. $FROM_i$: $\{1, \ldots, n\}$
	2. TO_i: $\{1, \ldots, n\}$
	3. $HOPCOUNT_i$: $\{0, \ldots, n-1\}$
Intra-agent constraints	1. (Origin) $HOPCOUNT_1 = 0$
	2. (Uniqueness) $TO_i \neq FROM_i$
Inter-agent constraints	1. (Link from Neighbor) $FROM_i = j \Longleftrightarrow (i, j) \in E, i \neq j$
	2. (Link to Neighbor) $TO_i = j \Longleftrightarrow (i, j) \in E, i \neq j$
	3. (Symmetry) $FROM_i = j \Longleftrightarrow TO_j = i$
	4. (Increment) $HOPCOUNT_i = (HOPCOUNT_j + 1) \bmod n$

nodes only send messages to one of their neighbors [Seddigh et al., 39]. If a Hamiltonian cycle is established, any node in a one-to-one network can send a broadcast message to all the nodes in the network in sequential order, with a minimal number of data packets, and even get an acknowledgement of the successful receipt of the message by all nodes.

We can represent the problem of forming a Hamiltonian cycle as a DCSP as shown in table 2. Again, we associate a distinct agent with each node. Each agent has three variables $FROM_i$, TO_i and $HOPCOUNT_i$. The first two variables help track the preceding and succeeding nodes in the cycle, while the *HOPCOUNT* variable is used to provide a sequence number from the origin. The intra and inter-agent constraints guarantee consistency in the assignment of these variables. Once it is thus specified, a complete DCSP algorithm can be used to solve this problem in a distributed manner. When all the agents' variables are given satisfying assignments, they have an internal, localized representation of the global Hamiltonian cycle structure. If an instance is found to be unsatisfiable, then no Hamiltonian cycle exists in the network.

Figure 9 shows an unsatisfiable instance of this problem on a small, sparse network graph which contains no Hamiltonian cycles. No assignment of values to the variables of nodes 1, 6, and 7 will satisfy their respective intra-agent constraints, since they each have only one neighbor. Figure 10, on the other hand, shows a satisfiable instance of this problem on a denser network with a higher transmission radius. A particular solution is indicated in this figure using dashed lines, along with the corresponding constraint-satisfying values to the variables of each node agent.

Figure 11 shows that phase transitions occur in the existence of Hamiltonian cycles in wireless networks as the transmission radius R is increased. The average complexity profile is shown in figure 12 for a network with $n = 100$ nodes, with a 100 instances tested at each value of R. A specialized HCP-solver written by Vandegriend is used to generate this profile [Vandegriend, 43]. The profile shows the characteristic easy-hard-easy profile and confirms that Hamiltonian cycles can be determined efficiently if the network is operating well to the right of the transition region.

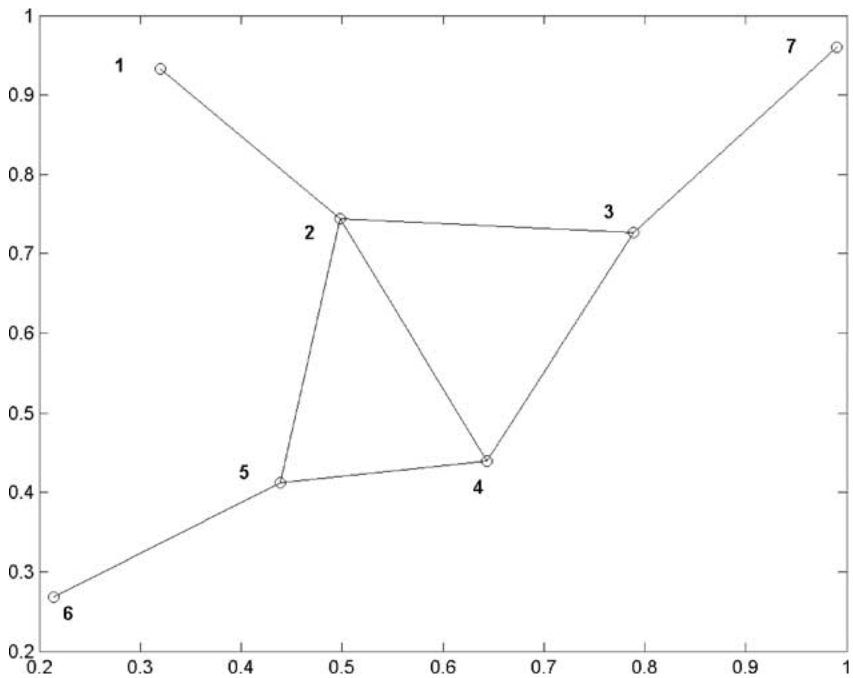

Figure 9. Unsatisfiable Hamiltonian cycle formation with small transmission radius ($n = 7$, $R = 0.40$).

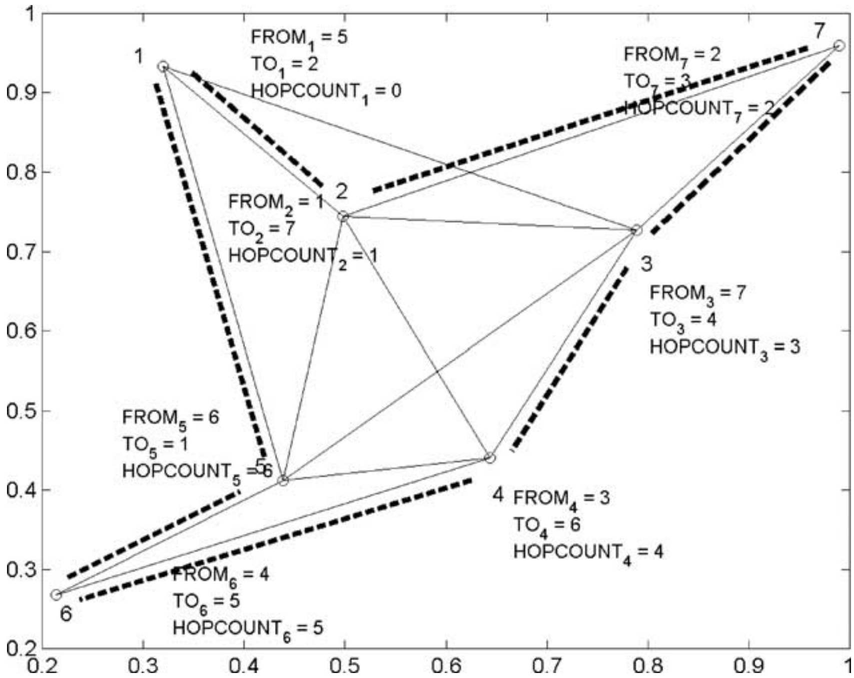

Figure 10. Satisfiable Hamiltonian cycle formation with a larger transmission radius ($n = 7$, $R = 0.55$).

Again, the graph density at which the phase transition occurs in this problem corresponds to a critical amount of per-node energy consumption. When sufficient energy resources are provided to the system, we see that the problem enters the under-constrained regime where a solution exists with high probability and the solution complexity is low.

5.3. Conflict-free channel scheduling

We now turn to a third and final problem that once again reflects the impact of transmission power on self-configuration: conflict free channel scheduling. One of the primary advantages of having limited-power wireless nodes is that they can be assigned time or frequency channels that can be spatially reused. Nodes that are sufficiently far away from each other can be assigned the same channel. Depending on whether the wireless nodes act as base-stations or as multi-hop relays, this problem becomes essentially a graph coloring problem with a one-hop or two-hop coloring constraint. We assume that links in the network graph now represent the interference between nodes. In the case of nodes acting as base-stations, the problem is referred to as channel allocation or frequency assignment, and the goal is to ensure that no nodes within one hop of each other may share the same channel [Cao and Singhal, 9; Hale, 22]. In the case of multi-hop wireless networks, this problem is known as broadcast scheduling, and the goal is to assign channels to nodes while ensuring that no nodes within two hops of each other share the same channel [Bao and Garcia-Luna-Aceves, 2, Pond and Li, 36; Lloyd, 30;

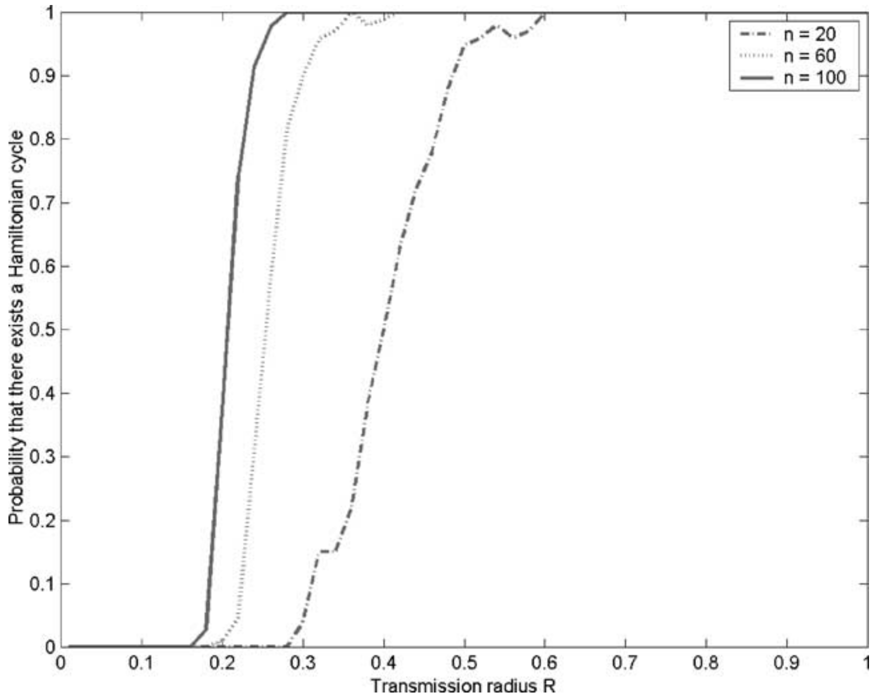

Figure 11. Phase transition in the fraction of satisfiable problems for Hamiltonian cycle formation ($n = 20, 60, 100$).

Ramaswami and Parhi, 37]. The two-hop constraint is required to avoid the hidden node terminal problem whereby an intermediate node experiences collisions due to simultaneous broadcasts from two distinct neighbors. In both cases the problem is known to be NP-complete based on a reduction from the graph coloring problem.

Let us begin with a graph $H = H(V, E)$ in which links represents the coloring constraints between nodes. In the case of the one-hop constraint problem, this graph is the same as the communication graph of the network $G(n, R)$. Even the two-hop constraint problem can be reduced to this model by placing edges between nodes that are two hops away in the original communication graph G. Now, let each node i in the network have a specified traffic demand for t_i channels. Let C be the total number of channels available in the network. Note that if the total bandwidth is kept constant then increasing C reduces the throughput available in each channel, whereas if the per-channel throughput is kept constant, then increasing C increases the bandwidth required. The goal is now to find an assignment of t_i distinct channels for each node i such that no two neighboring nodes i and j share the same channel. This can be formulated as a DCSP as seen in table 3.

Associate an agent with each node, with t_i multi-valued variables $\{x_{i,1}, \ldots, x_{i,t_i}\}$ for each agent i, corresponding to the allocated channels. These variables can take on values from 1 to C. The intra-agent constraint here is that each of the variables within an agent must take on distinct values. The inter-agent constraints take the form that if

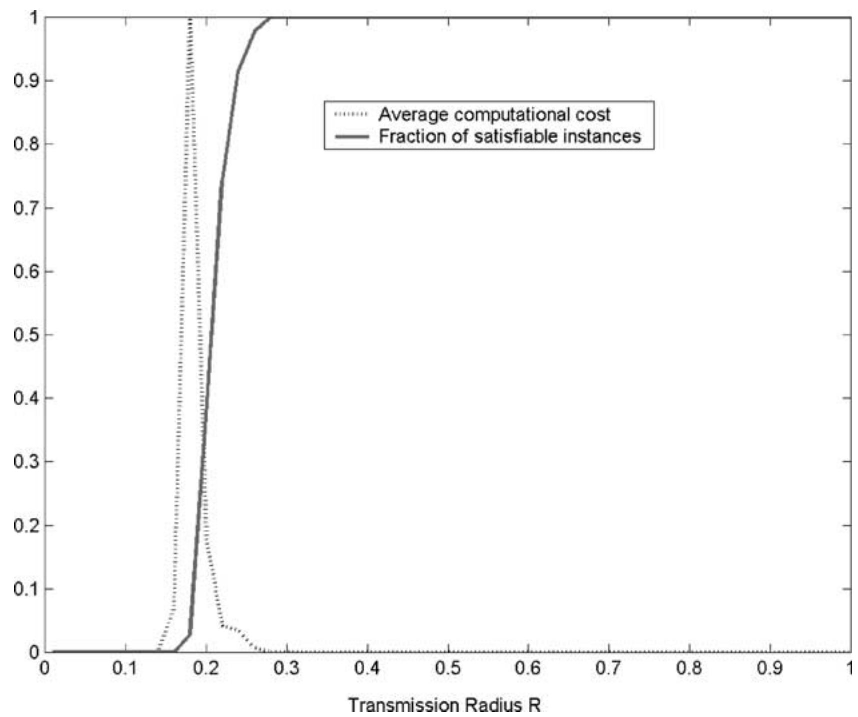

Figure 12. The average complexity profile for forming a Hamiltonian cycle using a complete search algorithm ($n = 100$).

Table 3
DSCP formulation for conflict-free channel allocation.

Agent i	Node i
Variables of agent i	$x_{i,l},\ l = \{1, \ldots, t_i\}$
Domain of variable $x_{i,l}$	$(1, \ldots, C)$
Intra-agent constraints	(Uniqueness) $\forall l \neq l',\ x_{i,l} \neq x_{i,l'}$
Inter-agent constraints	(Interference Constraint) $(i, j) \in E, i \neq j \Longrightarrow \forall(l, l'), x_{i,l} \neq x_{j,l'}$

there are two neighboring (interfering) nodes i and j, their variables must not take on the same values.

Formulated as a DCSP, this problem can be solved using one of the distributed backtracking algorithms described in the previous section. Although the communication and computational costs involved can be exponential in the number of nodes in the worst case, as we have discussed before, the average complexity can be within tolerable limits provided the system as a whole is under-constrained.

Figure 13 shows a satisfiable instance of this problem on a small, sparse graph. Variable assignments that satisfy all constraints are indicated in the figure. Figure 14, on the other hand, is an unsatisfiable instance of this problem on a dense graph. Since there are only three channels available, and the nodes 2, 3, 4, and 5 form a clique of size 4,

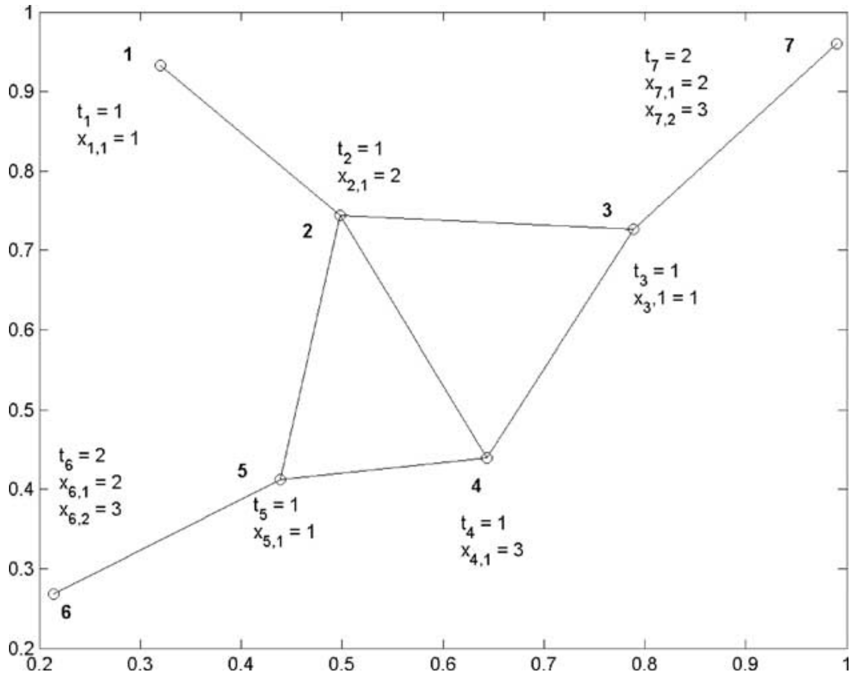

Figure 13. Satisfiable channel scheduling with small transmission radius ($n = 7$, $C = 3$, $R = 0.40$).

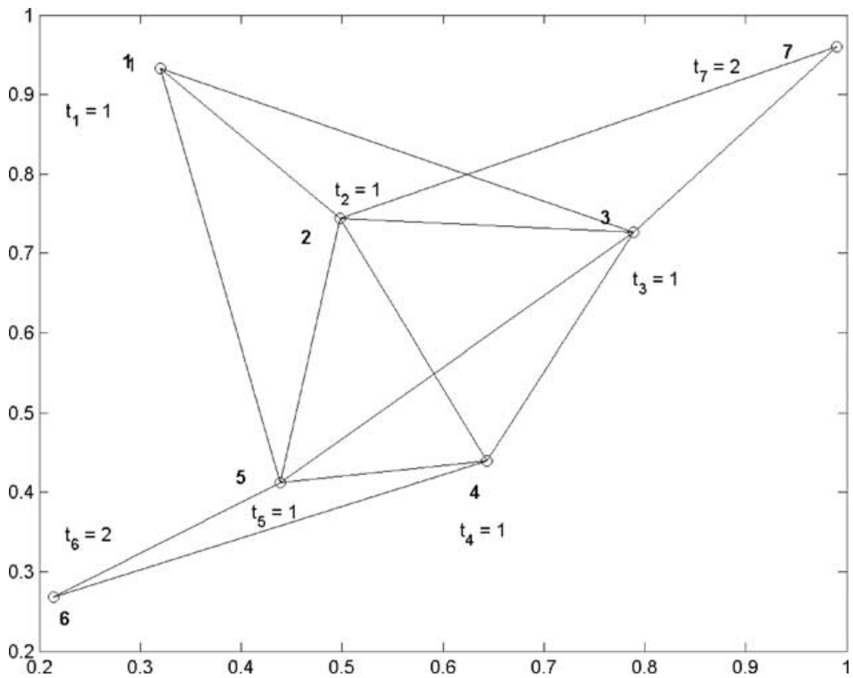

Figure 14. Unsatisfiable channel scheduling with a larger transmission radius ($n = 7$, $C = 3$, $R = 0.55$).

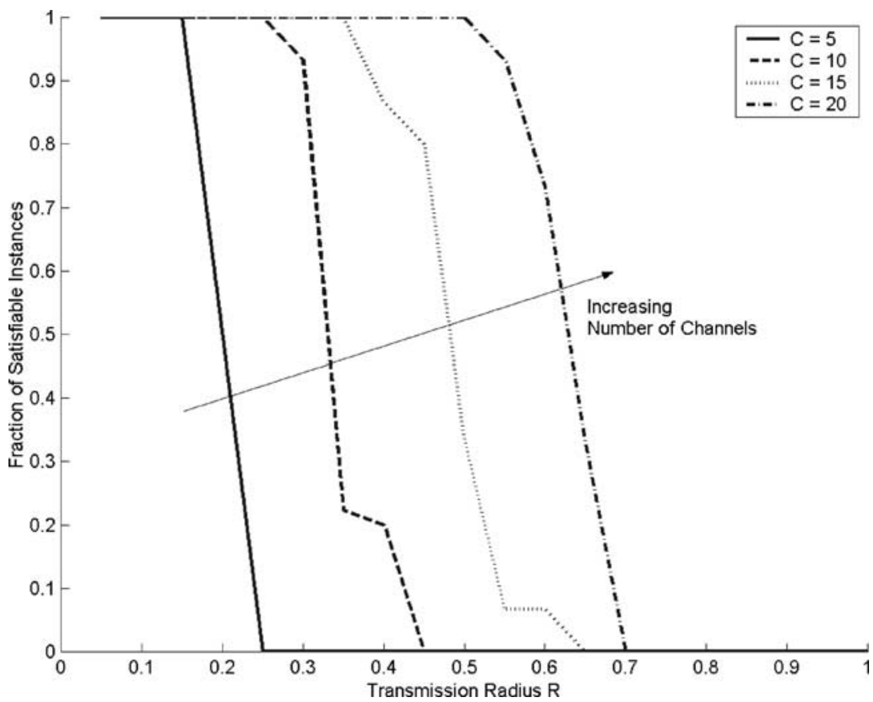

Figure 15. Phase transitions in the fraction of satisfiable problems for the channel scheduling problem ($n = 25$).

it is not possible for them to assign values to their respective variables without violating inter-agent constraints. For both the one-hop and two-hop formulations, the density of the constraint graph H is controlled by the transmission range R.

Thus for a given traffic level per node, there are two parameters that affect the problem complexity and satisfiability: the transmission radius R, and the total number of channels available C. To study this problem, we implemented the asynchronous backtracking (ABT) algorithm, which is a DCSP solver, along with an event-driven network simulator package. We present illustrative results for the one-hop constraint formulation of this problem, with $t_i = 1$ for all $n = 25$ nodes. A total of 15 random instances are generated and tested at each value of the transmission range R. As figure 15 shows, there is a one-zero phase transition in the probability that C channels suffice to perform channel scheduling. When the number of channels available is increased, the phase transition threshold moves to the right, allowing for denser networks. This is intuitive, for adding bandwidth resources to this system makes it easier to provide a non-conflicting schedule to the nodes.

It should be noted that the transitions for channel scheduling are in reverse, when compared with the phase transitions we examined for the clique-partition and the Hamiltonian cycle formation problems. Increasing the transmission range had a positive effect for the previous tasks as it increases connectivity, but a negative effect on channel scheduling as it increases the level of interference within the network.

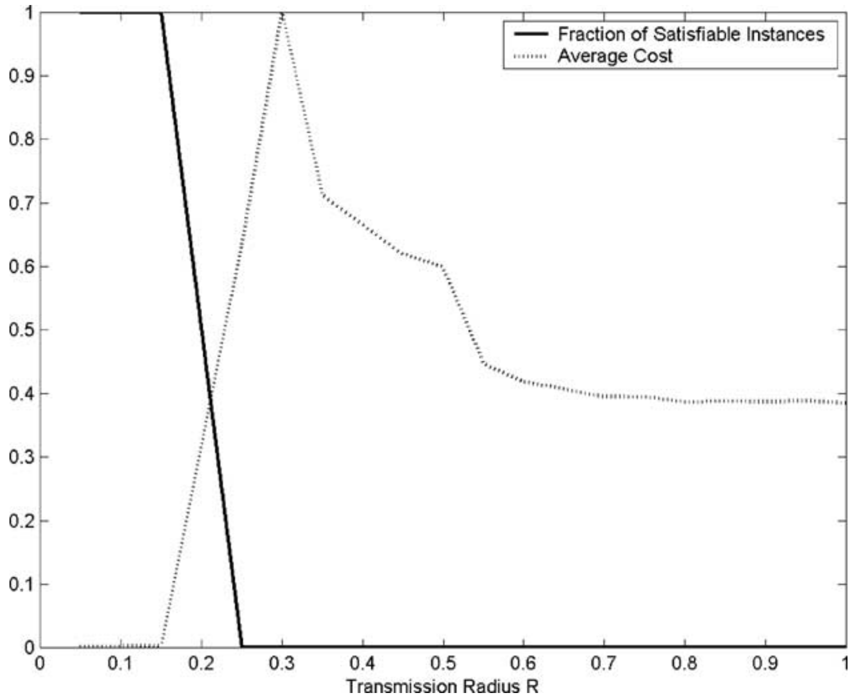

Figure 16. The average complexity profile for channel scheduling using the distributed ABT algorithm ($n = 25$, $C = 5$).

Figure 16 shows the average computational complexity profile obtained using the ABT distributed algorithm. As with the other constraint satisfaction problems, here too we find that the phase transition region coincides with a peak in the average complexity. For this problem, the way to simplify solution-complexity is to operate well to the left of the phase transition region, where the instances are satisfiable with high probability and efficiently solvable.

5.4. Comments

A couple of comments about the results we have presented in this section are in order. The first concerns the fact that our results on average complexity are all obtained through simulation experiments. The second has to do with the fact that although the distributed constraint satisfaction methodology and algorithms are particularly relevant for larger networks, we have considered only small and moderate-sized networks (less than 100 nodes) in our experiments.

It is possible to prove some bounds on the location of the phase transition thresholds in wireless networks and the reader is encouraged to look at [Krishnamachari et al., 26, 28] for some relevant analytical results. However, the focus of this paper has been on the average complexity of distributed problem solving. As we pointed out in section 2, obtaining theoretical results on the average complexity profile of constraint satisfaction problems is in general very difficult and indeed there are very few related analytical results in the literature. The general methodology for research in this direction has been to perform statistical experiments, as for example in [Mitchell et al., 34]. This is the approach we have used in this paper.

In this section we have shown results for a network with 100 nodes for the Hamiltonian cycle problem, and for the channel allocation problem we have used a distributed algorithm on a network with 25 nodes. The reason we have not investigated larger networks has to do with computational restrictions: the problems investigated are NP-complete. The point of the study is precisely to show that if the network is not tuned properly (to operate on the correct side of the phase transition threshold), the self-configuration problems may not be solved easily. Hence we have had to present results even for configurations that are hard (i.e. near the phase transition region), where the algorithms may take exponentially long times to solve. The reader should also note the computational overhead due to the need to average statistical results from multiple runs for each configuration in our study. The conclusions from these experiments are still useful for larger networks. The simulation results show that one can observe the easy-hard-easy computational profiles even with these relatively moderate-sized networks. These profiles only become more pronounced as the size of the network increases.

6. Conclusions

In this paper, we have examined three self-configuration tasks in wireless networks: partition into coordinating cliques, formation of Hamiltonian cycles, and conflict-free channel scheduling. In particular, we explored the impact of varying the transmission radius on the solvability and complexity of these problems.

In the case of the first two tasks, partition into cliques and Hamiltonian cycle formation, we saw that the probability that these tasks can be performed undergoes a transition from zero to one. When the transmission range is past the phase transition region, almost every network graph generated by the random location of nodes satisfies the desired global property. Before this region, the desired property is rarely satisfied. As we discussed in section 1, these phenomena are closely related to phase transitions in Bernoulli random graphs. In these cases, the critical transmission range corresponds to an energy-efficient operating point.

However, in the third task – conflict-free channel scheduling, we found that the transition occurs in reverse; there is a critical transmission range below which almost all network graphs generated by the random location of nodes can be allocated the available number of channels, and beyond which the desired property is rarely satisfied. We also showed that adding bandwidth resources by using more channels shifts the transition curves to the right, allowing for denser networks.

Our study suggests that the transmission range for nodes should be chosen by considering the intersection of thresholds for the various properties involved. Such a methodology would yield the critical range of resources required for feasible operation of a self-configuring network.

In wireless networks where there is need for scalability and there is no central processor responsible for the configuration of the network, there is a need for localized and distributed algorithms. We showed how each of the self-configuration tasks can be formulated as a distributed constraint satisfaction problem. Once they are formulated in this manner, it is possible to employ complete DCSP algorithms such as the asynchronous backtracking algorithm (ABT) to perform the task.

Finally, we also explored the complexity of these tasks. All three problems are known to be NP-hard, so that in the worst case, unless P = NP, some problem instances can require computational and communication resources that are exponential in the size of the network. We made the connection to research from the AI community which has shown that phase transition thresholds also correspond to a characteristic easy-hard-easy profile of the average computational complexity. We showed through experimental results for each of our tasks that when the network is operated in the satisfiable region these tasks can be solved efficiently on average. Thus the phase transition approach tells us that the transmission range of nodes can be used as a parameter to tune, and therefore bound, the complexity of self-configuration tasks.

We should note that much of the work presented here is of a preliminary and empirical nature. There are any number of directions in which this work can be extended.

The phase transition results presented here rely upon the assumption that the nodes are located at random with a uniform distribution in the area. While this is a reasonable assumption, particularly when considering static snapshots of mobile nodes with anisotropic mobility patterns, other location distributions could be examined. It is our belief that while the details of the critical threshold point may be different, the qualitative behavior will be the same. This remains to be evaluated.

Another interesting question is how to incorporate the phase transition methodology into functioning networks. One approach is to perform the analysis offline to determine the power settings of nodes before they are deployed in the operational environment. A more sophisticated approach would be to incorporate these results into online, adaptive mechanisms. This is also an open area for research and development.

In general the configuration of an multi-hop wireless network is a multidimensional constrained problem. The goal is to identify the boundaries of the under-constrained region of the problem and then use efficient algorithms to identify solutions that fall within that region. Our work has suggested that phase transition analysis can play an important role in attaining this goal.

Acknowledgements

We are thankful to Dr. Carla Gomes and Dr. Bart Selman for their guidance and support, and to Dr. Ivan Stojmenovic, Jason Hill, Zhen Nie and our anonymous reviewers for their helpful comments and suggestions.

References

[1] D. Achlioptas, Setting two variables at a time yields a new lower bound for random 3-SAT, in: *Proc. of 32nd Annual ACM Symposium on Theory of Computing*, 2000.

[2] L. Bao and J.J. Garcia-Luna-Aceves, Collision-free topology-dependent channel access scheduling, in: *21st Century Military Communications Conference Proceedings, MILCOM 2000*, pp. 507–511.

[3] [Beame et al.] P. Beame et al., On the complexity of unsatisfiability proofs for random formulas, in: *Proc. of 30th Annual ACM Symposium on Theory of Computing*, 1998.

[4] R. Bejar et al., Distributed constraint satisfaction in a wireless sensor tracking system, in: *Workshop on Distributed Constraint Reasoning, IJCAI-01*, 2001.

[5] R. Bejar et al., Capturing structure with satisfiability, in: *Proc. of 7th International Conference on Principles and Practice of Constraint Programming (CP2001)*, 2001.

[6] B. Bollobás, *Random Graphs* (Academic Press, New York, 1985).

[7] N. Bulusu et al., Scalable coordination for wireless sensor networks: Self-configuring localization systems, in: *Proc. of the 6th International Symposium on Communication Theory and Applications (ISCTA'01)*, Ambleside, UK.

[8] J.C. Cano and P. Manzoni, A low power protocol to broadcast real-time data traffic in a clustered ad hoc network, in: *Proc. of IEEE GlobeCom*, 2001.

[9] G. Cao and M. Singhal, Efficient distributed channel allocation for mobile cellular networks, in: *Proc. of 7th International Conference on Computer Communications and Networks*, 1998, pp. 50–57.

[10] P. Cheeseman, B. Kanefsky and W.M. Taylor, Where the really hard problems are, in: *Proc. of IJCAI-91*, Vol. 1, 1991, pp. 331–337.

[11] M. Davis, G. Logemann and D. Loveland, A machine program for theorem-proving, Communications of the ACM 5 (1962) 394–397.

[12] R. Dechter and D. Frost, Backtracking algorithms for constraint satisfaction problems, Technical Report, Information and Computer Science Department, UC Irvine (1999); http://www.ics.uci.edu/∼csp/r56-backtracking.pdf.

[13] O. Dubois, Y. Boufkhad and J. Mandler, Typical random 3-SAT formulae and the satisfiability threshold, in: *Symposium on Discrete Algorithms*, 2000.

[14] D. Estrin et al., Instrumenting the world with wireless sensor networks, in: *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, 2001.

[15] D. Estrin et al., Embedded, everywhere: A research agenda for networked systems of embedded computers, National Research Council Report (2001).

[16] E. Friedgut, Sharp thresholds of graph proprties, and the k-SAT problem, J. Amer. Math. Soc. 12(4) (1999) 1017–1054.

[17] E. Friedgut and G. Kalai, Every monotone graph property has a sharp threshold, Proc. Amer. Math. Soc. 124 (1996) 2993–3002.

[18] [Frieze and Suen] A.M. Frieze and S. Suen, Analysis of two simple heuristics on a random instance of k-SAT, Journal of Algorithms 20 (1996) 312–355.

[19] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979).

[20] P. Gupta and P.R. Kumar, Critical power for asymptotic connectivity in wireless networks, in: *Stochastic Analysis, Control, Optimization and Applications*, eds. W.M. McEneany et al. (Birkhäuser, Boston, MA, 1998) pp. 547–566.

[21] Z.J. Haas et al., eds., IEEE Journal on Selected Areas in Communications 17(8) (1999). Special Issue on Wireless Ad Hoc Networks.

[22] W.K. Hale, Frequency assignment: theory and applications, Proceedings of the IEEE 68(12) (1980) 1497–1514.

[23] Y. Hamadi, C. Bessière and J. Quinqueton, Backtracking in distributed constraint networks, in: *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, 1998, pp. 219–223.

[24] D. Hochbaum, *Approximation Algorithms for NP-Hard Problems* (Brooks/Cole, 1996).

[25] S. Kirkpatrick and B. Selman, Critical behavior in the satisfiability of random Boolean expressions, Science 264(5163) (1994) 1297–1301.

[26] B. Krishnamachari, Phase transitions, structure, and complexity in wireless networks, Ph.D. Thesis, Cornell University (2002).

[27] B. Krishnamachari, R. Bejar and S.B. Wicker, Distributed problem solving and the boundaries of self-configuration in multi-hop wireless networks, in: *Proc. of 35th Hawaii International Conference on System Sciences*, 2002.

[28] B. Krishnamachari et al., Critical density thresholds in distributed wireless networks, a chapter for book in honor of Ian Blake (to appear). Available online at http://www.krishnamachari.net/papers/densityChapter.pdf.

[29] B. Krishnamachari, S.B. Wicker and R. Bejar, Phase transitions in wireless ad-hoc networks, in: *Proc. of IEEE GlobeCom 2001*, 2001.

[30] E.L. Lloyd, Broadcast scheduling for tdma in wireless multi-hop networks, in: *Wireless Networks and Mobile Computing Handbook*, ed. I. Stojmenovic (Wiley, New York, 2001).

[31] N.A. Lynch, *Distributed Algorithms* (Morgan Kauffman, San Mateo, CA, 1996).

[32] R. Meester and R. Roy, *Continuum Percolation* (Cambridge Univ. Press, 1996).

[33] Z. Michalewicz and D.B. Fogel, *How to Solve It: Modern Heuristics* (Springer, Berlin, 1999).

[34] D. Mitchell, B. Selman and H. Levesque, Hard and easy distributions of SAT problems, in: *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Jose, CA, 1992, pp. 459–465.

[35] C. Perkins, *Ad Hoc Networking* (Addison-Wesley, Reading, MA, 2002).

[36] L.C. Pond and V.O.K. Li, A distributed time-slot assignment protocol for mobile multi-hop broadcast packet radio networks, in: *Proc of MILCOM*, 1989, pp. 70–74.

[37] R. Ramaswami and K.K. Parhi, Distributed scheduling of broadcasts in a radio network, in: *Proc. of INFOCOM*, 1989, pp. 497–504.

[38] M. Sanchez, P. Manzoni and Z.J. Haas, Determination of critical transmission range in ad-hoc networks, in: *Multiaccess Mobility and Teletraffic for Wireless Communications Workshop (MMT'99)*, 1999.

[39] M. Seddigh, J.S. Gonzales and I. Stojmenovic, RNG and internal node based broadcasting algorithms for wireless one-to-one networks, ACM Mobile Computing and Communications Review 5(2) (2001).

[40] B. Selman, H.A. Kautz and B. Cohen, Noise strategies for improving local search, in: *Proc. of the National Conference for Artificial Intelligence (AAAI'94)*, 1994.

[41] K. Sohrabi et al., Protocols for self-organization of a wireless sensor network, IEEE Personal Communications 7(5) (2000) 16–27.

[42] C.K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems* (Prentice-Hall, Englewood Cliffs, NJ, 2001).

[43] B. Vandegriend, Finding Hamiltonian cycles: algorithms, graphs and performance, Masters Thesis, Department of Computer Science, University of Alberta, Canada. Available online with source code for algorithms at http://www.cs.ualberta.ca/∼basil/research.html.

[44] C.P. Williams and T. Hogg, Exploiting the deep structure of constraint problems, Artificial Intelligence 70 (1994) 73–117.

[45] Y. Xu, J. Heidemann and D. Estrin, Geography-informed energy conservation for ad hoc routing, in: *Proc. of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (ACM MobiCom)*, 2001.

[46] [Xue and Kumar] F. Xue and P.R. Kumar, The number of neighbors needed for connectivity of wireless networks (2002, submitted).

[47] M. Yokoo et al., The distributed constraint satisfaction problem: formalization and algorithms, IEEE Transactions on Knowledge and Data Engineering 10(5) (1998) 673–685.

Bhaskar Krishnamachari is an Assistant Professor in the Department of Electrical Engineering-Systems at University of Southern California. He obtained his Ph.D. from Cornell University in May 2002, where he was a recipient of a three-year graduate fellowship and the prestigious Olin Presidential Fellowship. He currently heads the Autonomous Networks Research Group at USC. His research is focused on discovering fundamental principles and designing efficient mechanisms for self-configuration and information extraction in large-scale wireless sensor networks.

Stephen B. Wicker is Professor and Associate Director for Research of the School of Electrical and Computer Engineering. He is a co-author of Codes, Graphs, and Iterative Decoding (Kluwer, 2003), Turbo Coding (Kluwer, 1999), and Reed–Solomon Codes and Their Applications (IEEE Press, 1994). He is also the author of Error Control Systems for Digital Communication and Storage (Prentice-Hall, 1995). Professor Wicker teaches and conducts research in wireless information networks, digital communication systems, and error control coding. His current work focuses on complex adaptive networks, with an emphasis on the interplay between game theory, graph theory, and the design of distributed control algorithms. Professor Wicker has served as a consultant in wireless telecommunication systems and error control coding for various companies and governments in North America, Europe, and Asia. He is also the recipient of several teaching awards.

Ramón Béjar received his Bachelor in computer engeneering in the University of Lleida (UdL) in 1994 and a Master in computer engeneering in the University of the Balearic Islands (UIB), Spain, in 1996. Finally, he received his PhD in computer science in the University of Barcelona (UAB) in 2000. His main research topics are the SAT problem and constraint satisfaction problems.

Cèsar Fernàndez got his Ph.D. in telecommunications engineering in June 2001 from the Universitat Politecnica de Catalunya (Spain). From 1990 to 1992 he worked as a communication engineer involved in European Space Agency projects. In 1992, he joined the Computer Science Department at the University of Lleida, where he is currently an Assistant Professor. His research interests are in the field of distributed constraint satisfaction problems.