Locality Helps Sleep Scheduling

Jukka Suomela
Helsinki Institute for Information Technology HIIT
Department of Computer Science, P. O. Box 68, FI-00014 University of Helsinki, Finland
jukka.suomela@cs.helsinki.fi

ABSTRACT

This work studies sleep scheduling in mobile-device centric sensor networks. The objective is to maximise the lifetime by exploiting redundancy in an over-deployed network. Redundancy is described by a redundancy graph; it turns out that typical realistic redundancy graphs are members of the family of local graphs. By using a divide-and-conquer technique based on modular grids, it is shown that sleep scheduling admits a polynomial-time approximation scheme in local graphs; this is not the case in arbitrary graphs.

1. INTRODUCTION

Large-scale sensor networks [5] may be overly deployed. Some sensors may be redundant; for example, the entire detecting range of a motion detector may be covered by neighbouring sensors. Such redundancy is desired for fault-tolerance, and it may also be caused by chance in a mobile sensor network where, for example, movements of people carrying mobile sensing devices are not centrally controlled.

Over-deployment can be exploited to obtain a longer lifetime of a battery-powered sensor network. One possible approach is *sleep scheduling*. Put simply, each node can be asleep occasionally, as long as the active nodes cover the entire monitored area.

Much work on sleep scheduling focuses on the issue of preserving the connectivity of a wireless network [5, §7.2]. However, connectivity is not an issue in many mobile-device centric sensor networks. For example, cellular phones can transmit sensor measurements, alerts, and other information over a mobile data service such as GPRS, regardless of whether or not neighbouring devices are active. This work focuses on such applications.

1.1 Sleep Scheduling Problem

Let V denote the set of sensor nodes. We write $b(v)$ for the maximum time that the node $v \in V$ can be active; this is determined by the battery capacity and power consumption of the node. If a subset of nodes $A \subseteq V$ is able to cover the entire monitored area, we call A a *covering* set. The collection of all covering sets is denoted by $\mathcal{A}$. We write $A(v) = 1$ if $v \in A$ and $A(v) = 0$ if $v \notin A$. A function $x\colon \mathcal{A} \to \mathbb{R}$ is a *sleep schedule* if

$$\begin{aligned} \textstyle\sum_{A \in \mathcal{A}} A(v)x(A) &\le b(v) && \text{for all } v \in V, \\ x(A) &\ge 0 && \text{for all } A \in \mathcal{A}. \end{aligned}$$

The length of the sleep schedule is $\sum_A x(A)$. In the *sleep scheduling problem*, the objective is to find a sleep schedule of the maximum length.

A solution of the sleep scheduling problem can be used to determine when each sensor is active. For example, we can order the active sets arbitrarily, say $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$. First, the nodes that are members of A_1 are active for $x(A_1)$ units of time; the remaining nodes are asleep. Second, the nodes in A_2 are active for $x(A_2)$ units of time, and so on. This way the total lifetime of the network equals $\sum_A x(A)$, which may be considerably more than $\min_v b(v)$, and at any point in time, the entire monitored area is covered.

The sleep scheduling problem is a linear program (LP), and thus it can be solved in a time polynomial in the size of the LP. However, the number of variables in the LP equals the size of $\mathcal{A}$, which may be exponential in the number of the nodes. Thus, explicit construction of the set $\mathcal{A}$ or the LP is not feasible in the case of large-scale sensor networks.

1.2 Redundancy Graphs

This work focuses on the following case where $\mathcal{A}$ is given implicitly. Instead of the collection $\mathcal{A}$, it is assumed that a *redundancy graph* $G = (V, E)$ is given. An edge $\{u, v\} \in E$ indicates that the nodes u and v are mutually redundant: if node v is active, node u may be asleep and vice versa. The collection of covering sets $\mathcal{A}$ consists of all dominating sets of G.

Naturally, a redundancy graph is a simplification of the real world. It may be that the redundancy is not symmetric; furthermore, it may be that the interactions of the sensors cannot be described by a pairwise redundancy. However, the formulation captures, at least approximately, many practical problems. For example, in environment monitoring, a pair of sensors close to each other typically produces highly correlated measurements, and they can be considered to be mutually redundant.

WSW'06 at SenSys'06, October 31, 2006, Boulder, Colorado, USA.

Furthermore, this work focuses on the special case of uniform sensors, that is, $b(v) = 1$ for all $v \in V$. This special case of sleep scheduling turns out to be closely related to the problem of a domatic partition.

1.3 Fractional Domatic Partition

A *domatic partition* of graph $G = (V, E)$ is a partitioning of V into disjoint dominating sets. The *domatic number* of the graph is the maximum size of its domatic partition, that is, the maximum number of disjoint dominating sets. Determining whether the domatic number is above a given value is a well-known NP-complete problem [3]. In the corresponding optimisation problem, the objective is to find a domatic partition of the maximum size.

The maximum domatic partition problem can be formulated as an integer program

$$\begin{aligned} \text{maximise} \quad & \textstyle\sum_D x(D) \\ \text{subject to} \quad & \textstyle\sum_D D(v)x(D) \le 1 \quad \text{for all } v, \\ & x(D) \ge 0 \quad \text{for all } D, \\ & x(D) \in \mathbb{Z} \quad \text{for all } D, \end{aligned}$$

where v ranges over all vertices and D ranges over all dominating sets; again, $D(v) = 1$ if $v \in D$ and $D(v) = 0$ if $v \notin D$. Given a feasible solution x, the collection $\{D : x(D) = 1\}$ is a domatic partition.

By relaxing the integrality constraint, an LP is obtained:

$$\begin{aligned} \text{maximise} \quad & \textstyle\sum_D x(D) \\ \text{subject to} \quad & \textstyle\sum_D D(v)x(D) \le 1 \quad \text{for all } v, \\ & x(D) \ge 0 \quad \text{for all } D. \end{aligned}$$

In this work, a feasible solution of the above LP is called a *fractional domatic partition* and the maximum value of $\sum_D x(D)$ is called the *fractional domatic number*. While this terminology is not widely used, it is analogous to the case of graph colouring and the chromatic number. A solution to the analogous LP relaxation of graph colouring is called a fractional graph colouring and the optimum value is called the fractional chromatic number. The fractional graph colouring problem is a covering LP with one variable for each independent set and one constraint for each vertex; similarly, the fractional domatic partition problem is a *packing* LP with one variable for each dominating set and one constraint for each vertex.

Observe that sleep scheduling in the case of redundancy graphs and uniform sensors is the same problem as the fractional domatic partition. Interestingly, in this application, the practical problem setting motivates the LP relaxation of a well-known combinatorial problem, not the original integral version.

1.4 Related Work

Feige et al. [2] study the approximability of the domatic partition. They prove that the domatic partition in arbitrary graphs can be approximated in polynomial time within a logarithmic factor but, under plausible complexity-theoretic assumptions, no better. The approximability of the fractional version of the problem does not seem to be discussed in the literature.

Domatic partition has been applied to maximising the lifetime of sensor networks. For example, Moscibroda and Wattenhofer [6] present a distributed, randomised approximation algorithm. However, they consider the case of arbitrary redundancy graphs; thus, their analysis achieves only logarithmic approximation ratios.

Instead of a packing of dominating sets, one can also consider a packing of more general set covers. This leads into problems such as *set cover packing* and its LP relaxation. This problem has also been considered in the context of sensor networks; Slijepcevic and Potkonjak [7] call it the *set K-cover problem*.

Berman et al. [1] is one of the few works that explicitly consider the fractional version of the set cover packing problem. However, they only obtain logarithmic approximation ratios, as they focus on the general case.

1.5 Contributions

In Section 2, we extend the known results on approximability of domatic partition to the fractional case. It turns out that the fractional version is as hard to approximate as the integral version.

While dominating sets and fractional domatic partitions are hard to optimise and hard to approximate in the general case, they need not be hard in problem instances that are relevant to sleep scheduling in sensor networks. Section 3 introduces a solution: so-called *local graphs*. The family of local graphs captures realistic redundancy graphs.

As the main contribution, Section 4 shows that there is a polynomial-time approximation scheme (PTAS) for maximising fractional domatic partition in local graphs. This shows that a simple and realistic assumption on locality indeed helps solving the problem.

2. APPROXIMABILITY OF FRACTIONAL DOMATIC PARTITION

Feige et al. [2] present an algorithm for finding a domatic partition of size $\Omega(\delta/\log|V|)$ where δ is the minimum degree of the graph. A domatic partition is also a fractional domatic partition. Moreover, $\delta + 1$ is not only an upper bound on the domatic number but also an upper bound on the fractional domatic number. Thus, the fractional domatic partition can also be approximated within a factor of $O(\log|V|)$. As a corollary, the integrality gap in domatic partition (that is, the ratio of the fractional domatic number to the domatic number) is $O(\log|V|)$ because the domatic number is at least $\Omega(\delta/\log|V|)$ and the fractional domatic number is at most $\delta + 1$.

Let us show that the above-mentioned polynomial-time approximation algorithm is near the best possible. Feige et al. [2] prove that it is hard to approximate the domatic number within a factor of $(1-\epsilon)\ln|V|$. The proof can be extended to the fractional domatic number, as follows.

- [2, Thm. 11]: The theorem is used as is, in its integral form.

- [2, Prop. 13]: The reduction is formed from the original integral problem to the fractional domatic number, in a weaker form with only the first upper bound ($p \leq |V|/r$). The same transformation is used. The lower bound holds: a domatic partition is a fractional domatic partition. The first upper bound holds: if all dominating sets are of size at least r, then the fractional domatic number is bounded by $|V|/r$.
- [2, Prop. 10]: The reduction is formed from the original integral problem to the fractional domatic number. Using the above modified proposition, the original proof applies directly. Note that the weaker form suffices; the second upper bound is not needed here.

Using the above results, we obtain the following extension to [2, Thm. 9]: for every $\epsilon > 0$, it is hard to approximate the *fractional* domatic number within a ratio of $(1-\epsilon)\ln|V|$. More precisely, there is no such polynomial-time approximation algorithm unless NP $\subseteq$ DTIME$(n^{O(\log\log n)})$.

3. LOCAL GRAPHS

A graph $G = (V, E)$ is called (d, N)-local if $V \subseteq \mathbb{R}^d$ such that $\|u - v\| < 1$ for all $\{u, v\} \in E$ and no ball of radius 1 contains more than N vertices.

Local graphs are bounded-degree graphs. However, note that local graphs need not be disk graphs; the length of each edge is bounded, but it is not required that there is an edge between a pair of nearby vertices.

With a suitable choice of N, typical redundancy graphs are $(2, N)$-local or $(3, N)$-local graphs: For each sensor node, there is one vertex in the graph, and the identity of the vertex is the physical location of the sensor node in a 2 or 3 dimensional space (after a suitable scaling of the coordinates). Distant sensors cannot be mutually redundant as the very reason of installing a sensor network is the fact that sensors measure information in their vicinity only. Furthermore, sensor devices are usually not packed in an arbitrarily dense manner; scaling up the number of sensor nodes typically means that a larger network covers a larger geographic area (that is, a *constant* N suffices for a *family* of arbitrarily large problem instances).

Note that typical redundancy graphs are not necessarily disk graphs. As a simple example, two sensors that are very close to each other may be separated by a wall, making them nonredundant.

4. FRACTIONAL DOMATIC PARTITION IN LOCAL GRAPHS

To develop a PTAS for the fractional domatic partition, we apply the approximation scheme by Garg and Könemann [4]; the same approach is used in the context of sleep scheduling by Berman et al. [1]. In this approximation scheme, we need to provide an oracle that finds a minimum-weight column of the coefficient matrix of the covering LP for an arbitrary weight vector. In our case, the columns correspond to dominating sets.

In Section 4.1, we present a PTAS for the minimum-weight dominating set in local graphs. Using this PTAS as an approximate oracle in the approximation scheme by Garg and Könemann results in an approximation algorithm for fractional domatic partition in local graphs.

4.1 Weighted Dominating Set

In the minimum-weight dominating set problem, each vertex v has a weight $w(v) \geq 0$, and the objective is to find a dominating set D of the minimum total weight $W(D) = \sum_{v \in D} w(v)$.

Fix the parameters d and N. Choose any $\epsilon > 0$. We show how to approximate the minimum-weight dominating set within an approximation ratio of $1 + \epsilon$ if the graph is (d, N)-local.

Before presenting the algorithm, we introduce some notation. For each d and ϵ, we choose an integer constant $m > 2^d d/\epsilon$. We use f and g to denote functions in the following families:

$$f\colon \{1, 2, \ldots, d\} \to \{0, 1, \ldots, m-1\},$$
$$g\colon \{1, 2, \ldots, d\} \to \mathbb{Z}.$$

We form a family of hypercubes $Q(f, g, r) = \{x \in \mathbb{R}^d : -r \leq x_k - 2(mg(k) + f(k)) < 2(m-1) + r \ \forall k\}$. Intuitively, f selects one of m^d positions for a modular grid, g selects one cube in the grid, and r is the width of a margin around each cube. Finally, let $V(f, g, r) = Q(f, g, r) \cap V$, that is, $V(f, g, r)$ consists of the vertices in the cube $Q(f, g, r)$.

Now we are ready to present the algorithm. The input consists of the graph $G = (V, E)$ and the weights $w(v)$ for each $v \in V$.

1. Find the pairs (f, g) such that $V(f, g, 2)$ is nonempty; in the remainder of the algorithm, only these pairs (f, g) need to be considered. The pairs can be found in polynomial time: consider each vertex v in turn and find the pairs (f, g) such that the cube $Q(f, g, 2)$ contains the v; the number of such pairs is bounded by a constant for each v.
2. Use exhaustive search to find a set $D(f, g) \subseteq V(f, g, 2)$ of the smallest weight that dominates all vertices in $V(f, g, 1)$. Note that the size of $V(f, g, 2)$ is bounded by a constant; all exhaustive searches can be performed in polynomial time.
3. Let $D(f) = \bigcup_g D(f, g)$. Choose a function f^* that minimises $W(D(f^*))$. Let $D = D(f^*)$. Output the set D.

Let us now prove the correctness of this algorithm. Let D^* be a minimum-weight dominating set. First, observe that for each (f, g), the set $D^* \cap V(f, g, 2)$ dominates all vertices of $V(f, g, 1)$. Thus, the total weight of $D(f, g)$ is bounded by $W(D^* \cap V(f, g, 2))$.

Second, we show that the set D is a dominating set. Consider any $v \in V$. For each f, there is a g such that $v \in V(f, g, 1)$. Thus, v is dominated by some $D(f^*, \cdot)$ and by $D = D(f^*)$.

Third, we show that the total weight of D is bounded by $(1+\epsilon)W(D^*)$. For each dimension $k \in \{1, 2, \ldots, d\}$ and for each value $i \in \{0, 1, \ldots, m-1\}$, let $P_k(i) = \{x \in \mathbb{R}^d : -2 \leq x_k - 2(mj+i) < 0, j \in \mathbb{Z}\}$. Let $U_k(i) = P_k(i) \cap V$ and $U(f) = \bigcup_k U_k(f(k))$.

For each k, the sets $P_k(\cdot)$ partition the space into m parts. Thus, there is a function f' such that $W(D^* \cap U_k(f'(k))) \leq W(D^*)/m$ for all k, implying $W(D^* \cap U(f')) \leq dW(D^*)/m$.

Let $Z(f,g) = V(f,g,2) \setminus V(f,g,0) \subseteq U(f)$. For each (f,v), there are at most 2^d functions g such that $v \in Z(f,g)$. We obtain

$$\begin{aligned} W(D) &\leq W(D(f')) \\ &= W(\textstyle\bigcup_g D(f',g)) \\ &\leq \textstyle\sum_g W(D(f',g)) \\ &\leq \textstyle\sum_g W(D^* \cap V(f',g,2)) \\ &= \textstyle\sum_g W(D^* \cap V(f',g,0)) \\ &\quad + \textstyle\sum_g W(D^* \cap Z(f',g)) \\ &\leq W(D^*) + 2^d W(D^* \cap U(f')) \\ &\leq (1 + 2^d d/m) W(D^*) \\ &\leq (1+\epsilon) W(D^*), \end{aligned}$$

completing the proof.

5. CONCLUSIONS AND FUTURE WORK

This work reported early research results on the problem of sleep scheduling in mobile-device centric sensor networks. The main focus was on the concept of local graphs; this family of graphs admits a PTAS for fractional domatic partition, which is equal to the problem of sleep scheduling in the case of redundancy graphs and uniform sensors. Assumptions on locality indeed help with sleep scheduling.

As sleep scheduling in local graphs is a new concept, it opens up several possibilities for research. This work only scratched the surface by considering some theoretical results and centralised algorithms; some directions for future research include developing practical distributed algorithms and studying the limits of distributed algorithms in this setting.

6. ACKNOWLEDGEMENTS

Thanks to Petteri Kaski and Aleksi Penttinen for discussions and to Patrik Floréen and Tanja Säily for commenting on the work.

This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the author's views.

7. REFERENCES

[1] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky. Power efficient monitoring management in sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC, Atlanta, GA, USA, March 2004)*, pages 2329–2334, Piscataway, NJ, USA, 2004. IEEE.

[2] U. Feige, M. M. Halldorsson, G. Kortsarz, and A. Srinivasan. Approximating the domatic number. *SIAM Journal on Computing*, 32(1):172–195, 2002.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, NY, USA, 1979.

[4] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proc. 39th Annual Symposium on Foundations of Computer Science (FOCS, Palo Alto, CA, USA, November 1998)*, pages 300–309, Washington, DC, USA, 1998. IEEE Computer Society.

[5] B. Krishnamachari. *Networking Wireless Sensors*. Cambridge University Press, Cambridge, UK, 2005.

[6] T. Moscibroda and R. Wattenhofer. Maximizing the lifetime of dominating sets. In *Proc. 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS, Denver, CO, USA, April 2005)*, page 242b, Washington, DC, USA, 2005. IEEE Computer Society.

[7] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *IEEE International Conference on Communications (ICC, Helsinki, Finland, June 2001)*, pages 472–476, Piscataway, NJ, USA, 2001. IEEE.