Near Optimal Multi-Application Allocation in Shared Sensor Network

Eric You Xu; Abusayeed Saifullah; Yixin Chen; Chenyang Lu; Sangeeta Bhattacharya

Washington University in St. Louis

Sept 22, 2010, Chicago
Outline

1. Introduction
2. Problem Formulation
3. Optimization Algorithm
4. Evaluations
Outline

1. Introduction
2. Problem Formulation
3. Optimization Algorithm
4. Evaluations
Shared wireless sensor network as an integrated infrastructure
Allocation Applications in a Shared Sensor Network

- Shared wireless sensor network as an integrated infrastructure
- Allocating multiple applications in a shared WSN
  - more cost effective
  - more flexible
Fundamental Challenges

Two fundamental challenges

1. Achieve high qualities of monitoring (QoM)
Fundamental Challenges

Two fundamental challenges

1. Achieve high qualities of monitoring (QoM)
2. Subject to severe resource constraints

- 8MHz 16-bit CPU
- 10KB RAM
- 250 Kbps radio
Two fundamental challenges

1. Achieve high qualities of monitoring (QoM)
2. Subject to severe resource constraints

These make allocation a constrained nonlinear optimization problem
Our Contributions

- Multi-application allocation in shared sensor networks as an optimization problem
  - maximize Quality of Monitoring subject to resource constraints
  - non-linear, discrete, and no closed-form objective function
Our Contributions

- Multi-application allocation in shared sensor networks as an optimization problem
  - maximize Quality of Monitoring subject to resource constraints
  - non-linear, discrete, and no closed-form objective function
- Submodular and monotonic objective function
  - exploits common properties of QoM functions
  - empirical validation based on a real-world data set
Our Contributions

- Multi-application allocation in shared sensor networks as an optimization problem
  - maximize Quality of Monitoring subject to resource constraints
  - non-linear, discrete, and no closed-form objective function
- Submodular and monotonic objective function
  - exploits common properties of QoM functions
  - empirical validation based on a real-world data set
- Efficient multi-application allocation algorithm
  - $(1/3 - \eta)$-approximation bound in terms of QoM
  - suitable for handling multi-dimensional resource constraints
A Concrete QoM Example – Intel Lab Data

The Intel Lab Data

- collected in Intel Berkley Research Lab
- temperature and humidity readings from 54 sensors in 36 days

Source: Intel Lab
Attributes for Quality of Monitoring

Quality of Monitoring

- maps a set to a real value
- monotonic
- diminishing return (also called *submodular*)

Source: Intel Lab
Variance Reduction

- **Intuition**: to *estimate* the sensor readings in the other nodes with *high confidence* based on the reading from a few sensors.
Variance Reduction – A Representative QoM Objective

Variance Reduction

- **Intuition**: to *estimate* the sensor readings in the other nodes with *high confidence* based on the reading from a few sensors
- **Goal**: minimize the variance of the estimation
  - a nonlinear function with no closed-form
Verification of Attributes

Verify submodularity and monotonicity for variance reduction

<table>
<thead>
<tr>
<th></th>
<th>Submodularity</th>
<th>Monotonicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empirically</td>
<td>99.8%</td>
<td>98.6%</td>
</tr>
<tr>
<td>Theoretically</td>
<td>N/A</td>
<td>Proved</td>
</tr>
</tbody>
</table>

- empirically sampled 30 million instances
- tested for both humidity and temperature applications
Outline

1. Introduction
2. Problem Formulation
3. Optimization Algorithm
4. Evaluations
CPU/Memory constraints for sensor $j$ are intuitive

We have

$$\sum_{t=1}^{p} a_{t,j} m_{t,j} \leq M_j$$

where $a_{t,j}$ is a *binary variable* and $m_{t,j}$ is the *resource requirement* for application $t$. 
Bandwidth Constraint

- Bandwidth required in one sensor node:
Bandwidth Constraint

- Bandwidth required in one sensor node:
  - routing for other nodes
Bandwidth Constraint

- Bandwidth required in one sensor node:
  - routing for other nodes
  - allocated applications
Bandwidth Constraint

- Bandwidth required in one sensor node:
  - routing for other nodes
  - allocated applications
  - bandwidth lost due to interference
    - interference detection based on the RID protocol [Zhou INFOCOM'05]
Bandwidth Constraint

- Bandwidth required in one sensor node:
  - routing for other nodes
  - allocated applications
  - bandwidth lost due to interference
    - interference detection based on the RID protocol [Zhou INFOCOM’05]

- $B_{routing} + B_{allocated} + B_{interference} \leq C$
Outline

1. Introduction
2. Problem Formulation
3. Optimization Algorithm
4. Evaluations
Overview of the Algorithm

- General for any monotonic and submodular objectives
Overview of the Algorithm

- General for any monotonic and submodular objectives
- \((1/3 - \eta)\)-approximation bound
Overview of the Algorithm

- General for any monotonic and submodular objectives
- \((1/3 - \eta)\)-approximation bound
- Efficient: linear to network size
Overview of the Algorithm

- General for any monotonic and submodular objectives
- $(1/3 - \eta)$-approximation bound
- Efficient: linear to network size

Three steps of our Fractional Relaxation Greedy (FRG) algorithm
Overview of the Algorithm

- General for any monotonic and submodular objectives
- $(1/3 - \eta)$-approximation bound
- Efficient: linear to network size

Three steps of our Fractional Relaxation Greedy (FRG) algorithm

1. Separation
Overview of the Algorithm

- General for any monotonic and submodular objectives
- \((1/3 - \eta)\)-approximation bound
- Efficient: linear to network size

Three steps of our Fractional Relaxation Greedy (FRG) algorithm

1. Separation
2. Relaxation
Overview of the Algorithm

- General for any monotonic and submodular objectives
- $(1/3 - \eta)$-approximation bound
- Efficient: linear to network size

Three steps of our Fractional Relaxation Greedy (FRG) algorithm

1. Separation
2. Relaxation
3. Rounding back
Separation

- Separating variables to ‘heavy’ and ‘light’ elements by their weights
  - Heavy element affects optimality greatly
Separation

- Separating variables to ‘heavy’ and ‘light’ elements by their weights
  - Heavy element affects optimality greatly
- Finding optimal assignments for ‘heavy’ elements by enumeration
Relaxation

- Relax ‘light’ variables to fractional values
Relaxation

- Relax ‘light’ variables to fractional values
- Do local greedy adjustment in a $k$-neighborhood

Finally arrive at a local maximal point $y^*$ and $f(y^*) \geq 1/2$.
Relaxation

- Relax ‘light’ variables to fractional values
- Do local greedy adjustment in a $k$-neighborhood
- Finally arrive at a local maximal point $y^*$ and $f(y^*) \geq 1/2\text{Opt}$
Rounding Back

- Probabilistically round fractional values back to binary values
**Rounding Back**

- Probabilistically round fractional values back to binary values
- Put ‘heavy’ and ‘light’ elements together
Rounding Back

- Probabilistically round fractional values back to binary values
- Put ‘heavy’ and ‘light’ elements together
- Expected objective value is at least \((1/3 - \eta)\text{Opt}\)
1 Introduction

2 Problem Formulation

3 Optimization Algorithm

4 Evaluations
Evaluation on Simulated Data

- Network with 50 and 100 nodes
- Allocating 5, 10, and 15 applications
- Baseline for our comparison
  - First-fit bin-packing
  - Simulated annealing with 30 minutes running time
Results on Simulated Data

Performance comparison of three algorithms in a 50-node network
Performance comparison of three algorithms in a 100-node network
Scalability Results on Simulated Data

Solution Time versus different network sizes (for 10 applications)
Multi-application allocation in shared sensor networks as an optimization problem

- a difficult optimization problem
- objective function is usually nonlinear and of no closed-form
Conclusion

- Multi-application allocation in shared sensor networks as an optimization problem
  - a difficult optimization problem
  - objective function is usually nonlinear and of no closed-form
- We exploit two common properties
  - submodularity
  - monotonicity
  - validated using real data set
Conclusion

- Multi-application allocation in shared sensor networks as an optimization problem
  - a difficult optimization problem
  - objective function is usually nonlinear and of no closed-form
- We exploit two common proprieties
  - submodularity
  - monotonicity
  - validated using real data set
- Efficient multi-application allocation algorithm
  - linear to network size
  - also suitable for other submodular optimization problems
Related works in application-resource allocations

- Sensor selection as single application allocation
  - utility-based sensor selection [Bian IPSN’06]
  - market-based sensor selection [Mainland NSDI’05]
  - submodular sensor selection [Krause IPSN’06]
Related works in application-resource allocations

- Sensor selection as single application allocation
  - utility-based sensor selection [Bian IPSN’06]
  - market-based sensor selection [Mainland NSDI’05]
  - submodular sensor selection [Krause IPSN’06]
- Multi-application allocation with no approximation bounds
  - market-based allocation with linear objectives [Ma ATSN’07]
  - integrated system with simple allocation algorithms [Bhattacharya RTAS’10]
Related works in submodular optimization

- $(1 - 1/e)$ approximation algorithm for single knapsack constraint [Sviridenko ORL’04]
- $1/5$ approximation algorithm for general linear constraints [Lee STOC’09]
- $(1 - 1/e)$ approximation bound for multiple knapsack constraints [Kulik SODA’08]
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Objective</th>
<th>Average Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bin-packing</td>
<td>1016</td>
<td>0.29</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>1066</td>
<td>1800</td>
</tr>
<tr>
<td>Our FRG Algorithm</td>
<td>1041</td>
<td>1.69</td>
</tr>
</tbody>
</table>

Table 1: Solution quality and time (in seconds) of three algorithms under bandwidth constraints.