Statement of Work

Group 27

Lucas Wojciechowksi, Ariel Weingarten, Alexander Maguire,
Austin Dobrik, Dane Carr

May 24, 2014

1 Problem Characterization

2 Problem Formulation

2.1 Givens

Suppose you have a number of mobile sensors

$$sensors = \{sensor_i | i \in [1, n_{sensors}]\}$$

Using these mobile sensors, you are to peform a number of tasks

$$tasks = \{task_i | i \in [1, n_{tasks}]\}$$

Each task has an inherent "priority"

$$priority(task_i) \in [0, 1]$$

Sensors are not homogeneous; some are more "skilled" at certain tasks than others

$$skill(sensor_i, task_j) \in [0, 1]$$

2.2 Problem

Our objective is to find an injective mapping of *task*s to *sensor*s that optimizes the overall expected performance. This mapping can be expressed as a vector

$$S = [s_1, ..., s_n]^T$$

$$s_i = j \Rightarrow sensor_i \text{ performs } task_j$$

Our objective is to maximize our objective function, $performance(S)$

$$performance(S) = \sum_{i=1}^{n} skill(sensor_i, task_{s_i}) \cdot priority(task_{s_i})$$

3 Problem Modeling

The complex task allocation problem can be modeled as a linear assignment problem. The solution can be represented as a bipartite graph $G = (S, T, E)$. The vertex sets S and T each have n vertices, and E contains n edges. $x_{i,j}$ represents an edge between S_i and T_j.

A solution can be represented as a matrix X_G, where

$$x_{i,j} = \begin{cases} 1 & : x \in E \\ 0 & : x \notin E \end{cases}$$

Let $p(t_i)$ equal the priority of task i. Let $s(s_j, t_i)$ equal the skill of sensor j at task i. The performance can be represented as a matrix

$$P = \begin{bmatrix} p(t_0) \cdot s(s_0, t_0) & \cdots & p(t_n) \cdot s(s_0, t_n) \\ \vdots & \ddots & \vdots \\ p(t_0) \cdot s(s_n, t_0) & \cdots & p(t_n) \cdot s(s_n, t_n) \end{bmatrix}$$

The performance of a solution can be calculated by multipling a solution matrix and the performance matrix, then summing the elements of the resulting matrix.

E.g.

$$X = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.2 & 0.4 & 0.1 \\ 0.6 & 0.8 & 0.9 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}$$

$$XP = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.2 & 0.4 & 0.1 \\ 0.6 & 0.8 & 0.9 \\ 0.3 & 0.2 & 0.5 \end{bmatrix} = \begin{bmatrix} 0 & 0.4 & 0 \\ 0.6 & 0 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

$$performance(X) = \sum_{i=1}^{3} \sum_{j=1}^{3} XP_{i,j} = 1.5$$

4 Reduced Size Problem

5 Real Problem