Radio and Computational Resource Management for Fog Computing Enabled Wireless Camera Networks

Emil Eriksson, György Dán, Viktoria Fodor
School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
{emieri,gyuri,vfodor}@kth.se

***Abstract*—We consider the problem of assigning communication and computing resources of a fog computing network to sensors that may observe the same scene from multiple viewing angles. We formulate the Multi-View Assignment Problem (MVAP) as a quadratic mixed-integer problem, and show that it is NP-hard. We propose polynomial time 4-approximation based on a transformation to a General Assignment Problem with dependent profits, and based on a reduction of the number of items to an approximately dominant set of items. We show that the reduction of the set of items does not affect the solution of the problem if the set of items is exactly the dominating set. Extensive numerical results show that the proposed algorithm performs close to optimal for small systems, and scales well with the number of sensors in the system.**

I. Introduction

Computer vision systems based on affordable cameras and cheap networking technologies are becoming important building blocks of the networked society, supporting smart cities and transportation, health care and entertainment. As cheap camera sensors become a commodity, overlapping fields of views between two or more cameras can be utilized for multi-view visual processing, which allows improved reliability of visual analysis based surveillance [1], tracking [2] and recognition [3].

If computer vision systems are based on affordable wireless sensor network platforms, the processing limitations of the camera sensors and the low transmission capacity of the sensor network make the real-time processing of the visual information at the sensors, alternatively at a remote server impossible [4]. Instead, a suitable solution for timely visual analysis could be to delegate the processing to nearby, powerful network nodes, such as access points or base stations, as it was shown for single camera [5] and for multiple camera systems [6].

In this paper we address the specific challenge of remote visual processing in multi-view camera systems, where overlapping fields of views of multiple sensors need to be processed jointly, that is, at the same base station. Moreover, since the same image can be part of several multi-views, processing all these views at the same base station leads to additional gains, as each image needs to be transmitted to one single base station only, saving both time and energy. Transmitting all images to the same base station is, however, infeasible in systems with many sensor, and thus the assignment has to be optimized so as to maximize the number of multi-views processed, subject to the available communication capacity. The solution to this problem is challenging in a wireless environment, as the channel quality, and thus the required transmission times depend on the relative positions of the sensors and the base stations.

In this paper we address this problem by formulating the multi-view sensor assignment problem (MVAP), in which a set of sensors has to be assigned to computing resources located at or near base stations in such a way that the total number of multi-views processed within a time frame is maximized. The MVAP is similar to general assignment problems, widely addressed in the literature [7], [8]. However, the specific requirement of multi-view processing leads to a quadratic utility function and dependent gains in the case when a sensor contributes in several multi-views. In this paper we propose to solve the multi-view assignment problem as a series of multiple choice knapsack problems (MCKP) [9], and provide an approximation algorithm with a bounded approximation ratio. We show that the complexity of the approximation algorithm can be decreased significantly by heuristics considering only the dominating sets of multi-views without compromising the efficiency of the resulting assignment. Our numerical results show that the proposed algorithm performs well independent of the view-graph topology, and allows to process up to 30% more views compared to a greedy approximation using the same communication capacity. To the best of our knowledge we are the first to consider the multi-view assignment problem for jointly optimizing the use of wireless communication and computing resources in a fog computing infrastructure.

The paper is organized as follows. In Section II we introduce the system model and in Section III we formulate the multi-view assignment problem. In Section IV we develop the approximation algorithm and in Section V we provide analytical results. In Section VI we present simulation results and we conclude the paper in Section VII.

II. System model

We consider a system that consists of a set $\mathcal{S}$ of $S = |\mathcal{S}|$ camera-equipped sensors and N base stations. The sensors may have overlapping fields of views (FoV), which we model by a *view-graph* $\mathcal{G}_V(\mathcal{S}, \mathcal{E}_V)$, where an edge $e_{s,s'} \in \mathcal{E}_V$ represents a shared FoV between sensors s and s'. We denote by A the $S \times S$ weighted adjacency matrix of $\mathcal{G}_V$. Element $a_{s,s'}$ of A represents the portion of the FoV of sensor s that overlaps with that of sensor s'. For simplicity, we consider that $a_{s,s'}$ is independent of the view of the other sensors, that is, $a_{s,s'} = a_s$. For notational convenience, we define $\tilde{A}$ to be a binary version of A, where

$$\tilde{a}_{s,s'} = \begin{cases} 1, & a_{s,s'} > 0 \\ 0, & a_{s,s'} = 0. \end{cases} \quad (1)$$

Time is slotted with slot length T. Each sensor captures an image at the beginning of each time slot, and has to transmit the captured image to one or more of the N base stations before the end of the time slot, where the images are processed for object recognition or tracking. For a pair of sensors s and s' for which $e_{s,s'} \in \mathcal{E}_V$, the two sensors need to transmit their images to the same base station, otherwise object recognition for the view (s, s') is not possible. Note that a sensor may need to transmit to several base stations to allow the processing of all multi-views.

Sensors transmit to the base stations via wireless channels, and we consider that adjacent base stations use different frequency bands in order to avoid interference, but have a single radio unit, and can receive only from one sensor at a time. We denote by $d_{n,s}$ the distance between sensor s and base station n, and consider that the transmission capacity $c_{n,s}$ is inversely proportional to $d_{n,s}$ up to the maximum range D, above which $c_{n,s} = 0$. The fraction of a time-slot T that sensor s needs to complete transmission of $a_{s,s'}$ to base station n can be expressed as $t_{n,s} = \frac{a_s}{c_{n,s}}$. We say that $t_{n,s} = \infty$ if $d_{n,s} > D$.

III. Multi-View Assignment Problem (MVAP)

Our objective is to maximize the number of multi-views processed in each time slot by assigning sensors to base stations subject to the multi-view graph and capacity constraints. We refer to the problem as the Multi-View Assignment Problem (MVAP). Due to the requirement of multi-view processing at the same base station MVAP is a quadratic mixed-integer optimization problem, and can be formulated as follows

$$\max \sum_{s=1}^{S} \sum_{s'=s+1}^{S} \tilde{a}_{s,s'} \left(1 - \prod_{n=1}^{N} [1 - x_{n,s} x_{n,s'}]\right) \quad (2)$$

$$\text{s.t.} \sum_{s=1}^{S} t_{n,s} x_{n,s} \leq 1, \forall n \quad (3)$$

$$x_{n,s} \in \{0, 1\}, \quad (4)$$

where $x_{n,s}$ is the decision variable indicating whether sensor s transmits to base station n. Equation (3) enforces that base stations do not receive past the duration of the time-slot, and (4) ensures that the entire FoV of sensor s is transmitted.

Proposition 1. *The MVAP defined by* (2)–(4) *is NP-hard.*

Proof. We prove the NP-hardness by transforming the Densest k-Subgraph problem (DkS) [10] to MVAP with $N = 1$. DkS is an NP-Hard problem where, given the graph $G = (V, E)$, the objective is to find the k-vertex subgraph of G which contains the largest number of edges. Let the nodes and the edges of the view-graph $\mathcal{G}_V(\mathcal{S}, \mathcal{E}_V)$ be identical to those of $G = (V, E)$, and set $t_{n,s} = \frac{1}{k}, \forall s$, ensuring that in the transferred problem exactly k sensors can transmit to the base station. The resulting optimum assignment of the MVAP gives highest number of processed multi-views, and consequently, the densest k-vertex subgraph of G. As DkS is NP-hard, problem (2)–(4) is also NP-hard. □

A. Multiple-choice Knapsack Problem (MCKP)

A basic building block in our solution is an approximation algorithm to the Multiple-choice Knapsack Problem (MCKP). In the MCKP a set $\mathcal{K}$ of items is partitioned into I classes; item j in class i has weight $w_{i,j}$ and value $p_{i,j}$. The objective is to select exactly one item out of each class of items so as to maximize the profit of the selected items subject to the capacity constraint c,

$$\max \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} p_{i,j} y_{i,j} \quad (5)$$

$$\text{subject to} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} w_{i,j} y_{i,j} \leq c \quad (6)$$

$$\sum_{j \in \mathcal{J}_i} y_{i,j} = 1 \quad (7)$$

$$y_{i,j} = \{0, 1\}. \quad (8)$$

Note that according to Proposition 1 the MCKP problem is NP-hard.

One way to solve the MCKP is to use a greedy algorithm that always selects the item with highest profit $p_{i,j}$, without violating the capacity constraint. The pseudo-code of the algorithm is given in Algorithm 1. This greedy algorithm has complexity $\mathcal{O}(L \log L)$, where L is the total number of items across all classes, and does not have a bounded approximation ratio.

An alternative to greedy is the polynomial time Dyer-Zemel (DZ) approximation algorithm proposed in [11], [12] for solving the MCKP. For completeness we discuss the algorithm, shown in Algorithm 2. The basic idea is to pair items of the same class and in this way iteratively remove items based on their profit-weight ratio. The DZ algorithm terminates once there is one single item left in each class (which may be the

Algorithm 1 Greedy approximation for MCKP

```
Input: $\mathcal{I}$, $\mathcal{J}_i$, $p_{i,j}$, $w_{i,j}$, $c$
Output: Set of $\mathcal{M}$ of assigned items
 1: Sort all items $j$ in all component $i$ by decreasing $p_{i,j}$
 2: Let $\mathcal{I}^+$ the set of components with assigned item
 3: $\mathcal{I}^+ = \emptyset$
 4: $\mathcal{M} = \emptyset$
 5: $W = 0$
 6: for each $\{i, j\}$, $i \notin \mathcal{I}^+$ do
 7:     if $W + w_{i,j} \leq c$ then
 8:         $\mathcal{M} = \mathcal{M} \cup j$
 9:         for $i$, $j \in i$ $\mathcal{I}^+ = \mathcal{I}^+ \cup i$
10:         $W = W + w_{i,j}$
11:     end if
12: end for
13: return $\mathcal{M}$
```

Algorithm 2 Dyer-Zemel MCKP approximation

```
Input: $\mathcal{I}$, $\mathcal{J}_i$, $p_{i,j}$, $w_{i,j}$, $c$
Output: Set $\mathcal{M}$ of assigned items
 1: Let $\mathcal{J}_i^+ = \mathcal{J}_i \; \forall \; i$
 2: while $\exists |\mathcal{J}_i^+| > 1$ do
 3:   for each $i$ do
 4:     while More than 1 unpaired item remains in class $i$ do
 5:       Pair any items $(j,k) \in i$
 6:       if $(w_{i,j} < w_{i,k} \;\wedge\; p_{i,j} > p_{i,k})$ then
 7:         $\mathcal{J}_i^+ = \mathcal{J}_i^+ \setminus k$, pair $j$ with new item
 8:       else if $(w_{i,j} > w_{i,k} \;\wedge\; p_{i,j} < p_{i,k})$ then
 9:         $\mathcal{J}_i^+ = \mathcal{J}_i^+ \setminus j$, pair $k$ with new item
10:       else if $(w_{i,j} > w_{i,k}) \vee (w_{i,j} == w_{i,k} \;\wedge\; p_{i,j} < p_{i,k})$
          then
11:         Change the order of $j$ and $k$
12:       end if
13:     end while
14:     for each Pair $(j,k) \in i$ do
15:       Let $\alpha_{ijk} = \frac{p_{i,k} - p_{i,j}}{w_{i,k} - w_{i,j}}$
16:     end for
17:     Let $\alpha = \text{median}(\{\alpha_{ijk}\})$
18:     for each $i$ do
19:       Let $\mathcal{M}_i = \text{argmax}_{j \in i} p_{i,j} - \alpha w_{i,j}$
20:       Let $a_i = \text{argmin}_{j \in \mathcal{M}_i(\alpha)} w_{i,j}$
21:       Let $b_i = \text{argmax}_{j \in \mathcal{M}_i(\alpha)} w_{i,j}$
22:     end for
23:     if $\sum_{i=1}^{t} w_{i,a_i} \leq c < \sum_{i=1}^{t} w_{i,b_i}$ then
24:       return $\mathcal{M} = \cup_i \mathcal{M}_i$
25:     else
26:       for each $i$ do
27:         for each Pair $(j,k)$ do
28:           if $\sum_{i=1}^{t} w_{i,a_i} \geq c \wedge \alpha_{ijk} \leq \alpha$ then
29:             $\mathcal{J}_i^+ = \mathcal{J}_i^+ \setminus k$ (remove heavier item)
30:           end if
31:           if $\sum_{i=1}^{t} w_{i,a_i} < c \wedge \alpha_{ijk} \geq \alpha$ then
32:             $\mathcal{J}_i^+ = \mathcal{J}_i^+ \setminus j$ (remove lighter item)
33:           end if
34:         end for
35:       end for
36:     end if
37:   end for
38: end while
39: return $\mathcal{M}$
```

empty set), and the capacity constraint is not violated. The algorithm has an approximation ratio of 3 and complexity $\mathcal{O}(L)$.

IV. Approximation Algorithm for MVAP

In this section we develop a 4 approximation algorithm with complexity $O(N^2L)$ to the MVAP problem. The algorithm consists of two phases. In the first phase we transform the MVAP problem into a Generalized Assignment Problem with Dependent Profits (GAPDP), which allows the selection of items from the same class at multiple knapsacks. In the second phase we solve the GAPDP by treating it as a series of MCKPs.

We begin by describing the first phase. Let $\mathcal{I}$ be the set of connected components in $\mathcal{G}_V$. For each $i \in \mathcal{I}$ we define the class of items $\mathcal{J}_i$ as the power set of the sensors in i, including the empty set $\emptyset$. We will reduce the cardinality of this set in Section IV-B. For a subset of sensors $j \in \mathcal{J}_i$ we denote the subgraph induced by the sensors by $\mathcal{G}_{i,j}$. The profit of item j is the number of shared views, that is, the number of edges in $\mathcal{G}_{i,j}$, $p_{i,j} = |E(\mathcal{G}_{i,j})|$. The weight $w_{n,i,j}$ of $j \in \mathcal{J}_i$ depends on the base station n, $w_{n,i,j} = \sum_{s \in j} t_{s,n}$.

We formulate the GAPDP as

$$\max |\bigcup_{n} \bigcup_{y_{n,i,j}=1} E(\mathcal{G}_{i,j})| \tag{9}$$

$$\text{subject to} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} w_{n,i,j} y_{n,i,j} \leq 1, \forall n \tag{10}$$

$$\sum_{j \in \mathcal{J}_i} y_{n,i,j} = 1, \forall n, i \tag{11}$$

$$y_{n,i,j} = \{0,1\}, \tag{12}$$

where $y_{n,i,j}$ indicates whether item j in class i is assigned to knapsack n, (9) aims at maximizing the number of multi-views covered by all assigned items, and (10)-(12) are the transformed constraints (3), (4), and the additional constraint on one assigned item per component.

The solution of the MVAP, which is the matrix of binary values $x_{n,s}$, is then given as $x_{n,s} = \sum_{s \in j, j \in \mathcal{J}_i, i \in \mathcal{I}} y_{n,i,j}$ for all n and s.

A. Iterative GAPDP approximation

To solve the GAPDP we propose the Highest Marginal Profit (HMP) algorithm, which is an iterative greedy algorithm. Its pseudo-code is shown in Algorithm 3. HMP solves the MCKP for all base stations in each iteration, and keeps the assignments to the base station n^* with the highest profit. Then, the profit of all items is updated to represent the marginal gains given the completed assignments. Let us denote the set of base stations already considered by $\mathcal{N}^+$. Then the profit of each item j is decreased by the number of multi-views already assigned at the base stations in $\mathcal{N}^+$. That is, $p_{i,j} = |E(\mathcal{G}_{i,j}) \setminus \cup_{n \in \mathcal{N}^+, y_{n,i,k}=1} E(\mathcal{G}_{i,k})|$.

Proposition 2. *The approximation ratio of the HMP algorithm is $1+\alpha$, where α is the approximation ratio of the algorithm for solving the MCKP.*

Proof. The proof is based on the local ratio technique [13], following the steps in [8]. □

Depending on the algorithm used to solve the MCKP, there may be spare capacity available at some or all base stations. By running the HMP algorithm multiple times, each time considering just the remaining capacity of the base stations and the un-processed multi-views, it may be possible to process additional multi-views. As no previously assigned sensor is removed, the total profit is non-decreasing, and thus, running the HMP algorithm multiple times does not decrease the approximation ratio given by Proposition 2.

B. Dominating set for GAPDP

Recall that each class in the GAPDP corresponds to a connected component of M sensors, and could contain 2^M items in the worst case, which makes the MVAP transformation to GAPDP and the solution of GAPDP exponential in the

Algorithm 3 HMP algorithm for approximating the GAPDP, with marginal gain maximizing base station sequence. The time complexity is $\mathcal{O}(N^2 f(S))$.

```
Input: $\mathcal{I}$, $\mathcal{J}_i$, $p_{i,j}$, $w_{n,i,j}$, $T$
Output: $\{y_{n,i,j}\}$
 1: $\mathcal{N}^+ = \emptyset$
 2: while $\mathcal{N}^+ \neq \mathcal{N}$ do
 3:     for $\forall n \notin \mathcal{N}^+$ do
 4:         Solve $\mathcal{M}_n$ = MCKP($\mathcal{I}$, $\mathcal{J}_i$, $p_{i,j}$, $w_{n,i,j}$, $T$)
 5:         $p_n = \sum_{j \in \mathcal{M}_n} p_j$
 6:     end for
 7:     $n^* = \arg\max_n p_n$
 8:     set $y_{n^*,i,j} = 1$ for $j \in \mathcal{M}_{n^*}, j \in \mathcal{J}_i$
 9:     $\mathcal{N}^+ = \mathcal{N}^+ \cup n^*$
10:     $p_{i,j} = |E(\mathcal{G}_{i,j}) \setminus \bigcup_{n+ \in \mathcal{N}^+, y_{n,i,j'}=1} E(\mathcal{G}_{i,j'})|$
11: end while
```

component size M. To reduce the exponential complexity, we propose to include only the set of *dominating items* in $\mathcal{J}_i$.

Definition 1. *An item j is dominated if there exists another item k such that $w_{n,i,j} \geq w_{n,i,k}$ and $p_{i,j} \leq p_{i,k}$ [14]. If two items have exactly the same weight and profit, we can arbitrarily declare one of them to be dominated without affecting the total profit and weight of the items assigned to base station n. If j is not dominated by any other items, we say that it is a* dominating item.

Definition 2. *We call the set of* dominating items *in a class a* dominating set.

Proposition 3. *The optimal solution when considering only the* dominating set *in each class is equivalent to the optimal solution when retaining the full set of items.*

Proof. Assume that in the solution of the MCKP, item j is selected from class i. If any item k satisfies $p_{i,k} \geq p_{i,j}$ and $w_{i,k} \leq w_{i,j}$, then item j is not a *dominating item* and the solution is not optimal, as it can be improved by substituting item j for item k. If such a substitution is not possible, then the solution can not be improved and, by definition, item j is part of the *dominating set*. □

To construct the *dominating set* of items in a class without performing an exhaustive search, we propose the greedy Remove Sensor with Lowest Ratio (RSLR) algorithm described in Algorithm 4. The RSLR algorithm starts from the complete set of sensors and iteratively removes the sensor with the lowest profit-to-weight ratio.

Proposition 4. *For a connected component that is a complete graph the RSLR algorithm computes the set of* dominating items.

Proof. The item containing all M sensors of the subgraph is clearly a dominating item. In a complete graph of size M, every vertex has $(M-1)$ edges connecting it to other vertices. After removing any vertex from the complete graph, the resulting subgraph is also a complete graph containing $\frac{(M-1)(M-2)}{2}$ edges, that is, for any items j, k, $p_{i,j} = p_{i,k}$. As Algorithm 4 removes the sensor with the greatest transmission time $t_{s,n}$ at each iteration, the resulting item j will satisfy $w_{i,j} \leq w_{i,k}$ which, by definition, makes j a dominating item. By induction, the items found in later iterations are also *dominating items*. □

V. Analytic results

The approximation algorithm we propose to solve the MVAP consists of two main steps: the transformation of the problem to the GAPDP, and the solution of GAPDP through the conversion to a series of MCKPs. Transforming the MVAP to a GAPDP does not affect the optimal solution, as long as all the *dominating sets* of items are included in the GAPDP. The approximation is in the solution of the GAPDP. The DZ algorithm (Algorithm 2) solves each MCKP with an approximation ratio of 3, which gives a resulting approximation ratio of 4 when solving the complete GAPDP using the HMP approximation algorithm (Algorithm 3). The greedy MCKP approximation algorithm (Algorithm 1) does not have an approximation ratio.

Transforming the MVAP to the GAPDP involves generating L items, where $S \leq L \leq 2^S$. The complexity of solving the resulting GAPDP using the HMP approximation algorithm is $\mathcal{O}(N^2 \cdot f(S))$, where $f(S)$ is the complexity of the algorithm used for solving the individual MCKPs. The resulting complexity when using the greedy MCKP approximation algorithm to solve the MCKP is $\mathcal{O}(L + N^2 L \log L) = \mathcal{O}(N^2 L \log L)$, while it is $\mathcal{O}(L + N^2 L) = \mathcal{O}(N^2 L)$, when using the DZ MCKP approximation algorithm.

To decrease the computational complexity, we also propose to consider only the set of *dominating items* for MCKP. This does not change the approximation ratio if the exact set of *dominating items* is found, but it decreases the computational complexity significantly. The RSLR Algorithm (Algorithm 4) finds an approximate set of *dominating items* with cardinality S and runs in $\mathcal{O}(M^2)$ time for a group of M sensors. When using the approximate sets of *dominating items*, the overall time complexity is $\mathcal{O}(L^2 + N^2 S \log S)$ and $\mathcal{O}(L^2 + N^2 S)$, for the greedy algorithm and for the DZ approximation algorithm, respectively.

Algorithm 4 RSLR algorithm for dominating set construction. Time complexity is $\mathcal{O}(M^2)$.

```
Input: $\mathcal{I}, t_{n,s}$
Output: $\mathcal{J}_i^+$
 1: $\mathcal{Z} = \{s \in \arg\max_j |E(\mathcal{G}_{i,j})|\}$
 2: $\mathcal{J}_i^+ = \{\mathcal{Z}\}$
 3: while $|\mathcal{Z}| > 2$ do
 4:     $s' = \arg\min_s \frac{deg(s)}{\sum_{s \in \mathcal{Z}} t_{n,s}}$
 5:     $\mathcal{Z} = \mathcal{Z} \setminus \{s'\}$
 6:     $\mathcal{J}_i^+ = \mathcal{J}_i^+ \cup \mathcal{Z}$
 7: end while
 8: $\mathcal{J}_i^+ = \mathcal{J}_i^+ \cup \emptyset$
 9: return $\mathcal{J}_i^+$
```

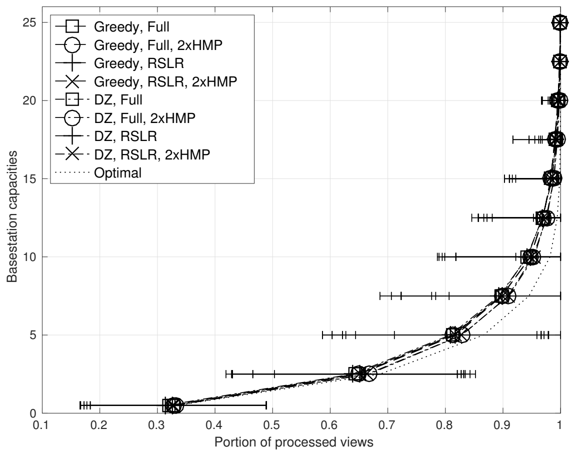

Figure 1. Required capacities as a function of the portion of views that can be processed for the various algorithms. Bars show 95% confidence intervals.

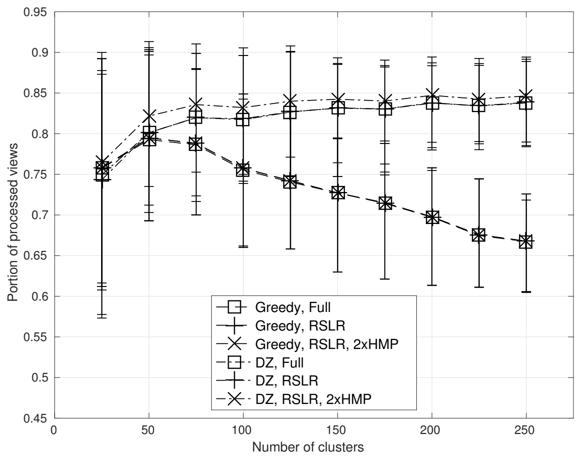

Figure 2. Portion of views that are processed as a function of the number of clusters in the system. The portion of views being processed by the greedy algorithm decreases while the DZ based algorithm maintains its performance. The available capacity of each basestation is directly proportional to the number of sensors. Bars show 95% confidence intervals.

VI. Numerical results

We performed simulations to study the performance of the proposed algorithm. We placed base stations in a uniform grid within a square area, and clusters of sensors uniformly at random within the square area. Each cluster consisted of sensors deployed evenly on a circle around the center of the cluster. The number of sensors in a cluster were uniformly distributed within 50% of the average cluster size. The sensors of a cluster were connected in sub-graphs according to an Erdős-Rényi random graph model [15], with the additional constraint that all sensors must have at least one edge to another sensor. We used the weighting parameter of the Erdős-Rényi random graph model to control the total number of edges in the *view-graph*. We ran each simulation 100 times with different random seeds, and the presented results are the averages with 95% confidence intervals.

Figure 1 shows the required capacity as a function of the share of views that can be processed, computed using greedy or Dyer-Zemel for solving the MCKP and considering all items in a class (Full) or using the RSLR algorithm, and with the HMP algorithm run either once or twice (2xHMP). To be able to compute the optimal solution to MVAP, we used 4 basestations, and 16 clusters totaling 96 sensors, with a weighting parameter of 0.6 for the *view-graph*. The figure shows how the required capacities of the basestations must be increased in order to process a given portion of the multi-views. We observe that for these small scenarios all algorithms are similar in terms of processed views, and all algorithms perform well when compared to the optimal solution, with no algorithm ever being more than 10% worse than the optimal solution. We will forgo the optimal solution for the rest of the section in order to be able to consider larger systems.

Figure 2 shows the portion of processed views as a function of the number of clusters, for 9 basestations, clusters containing on average 6 sensors, and a weighting parameter of 0.6. The capacities of the basestations are scaled to be direct proportional to the number of sensors in the system. The figure shows that despite the increasing capacity the portion of views that can be processed when using the greedy algorithm decreases, while the DZ based algorithm shows no decrease in the portion of processed views as the number of sensors increases.

Figure 2 also shows how using the reduced classes of the RSLR algorithm and running the HMP algorithm twice affects the portion of processed views for the greedy and DZ algorithms. We can see that there is practically no difference between algorithms using the RSLR algorithm and those using the complete classes, showing that this reduction in complexity comes at a very low cost. While the second round of assignment has no effect when using the greedy algorithm, it does increase the portion of processed views for the DZ based algorithm. The reason is that the greedy algorithm will always assign as many items as the capacities allow to each basestation, leaving no usable capacity, while the DZ based algorithm often results in some unused capacity after the first round of assignment. This unused capacity is then used to assign additional sensors that were overlooked in the first round.

Figure 3 shows the same scenario as Figure 2, but with the number of clusters fixed and varying the average size of the clusters instead. Again, the capacities of the basestations scale with the number of sensors in the system. The figure shows that the DZ based algorithm performs well, even when the cluster sizes are large, that is, when the multi-views are highly dependent.

Figure 4 shows the cumulative distribution function (CDF) of the weights of all sensors assigned to 9 basestations when average cluster size is 6 sensors and the weighting parameter is 0.6, and explains why the greedy algorithm perform worse

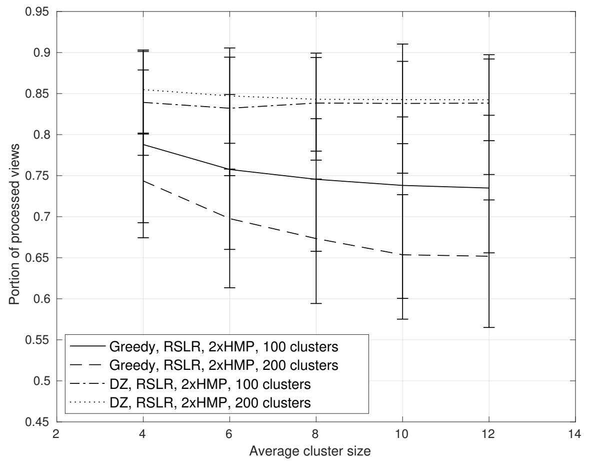

Figure 3. Portion of views that are processed as a function of the average cluster size. The increased number of sensors has no effect on DZ algorithm, while the number of views processed by the greedy algorithm decreases as the size of clusters increases. Bars show 95% confidence intervals.

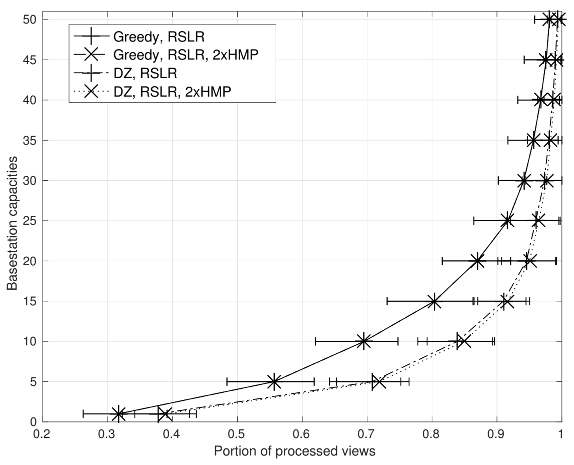

Figure 5. Required capacities as a function of the portion of views that can be processed for the various algorithms. The DZ algorithm requires significantly less capacity to schedule (close to) all sensors. Bars show 95% confidence intervals.

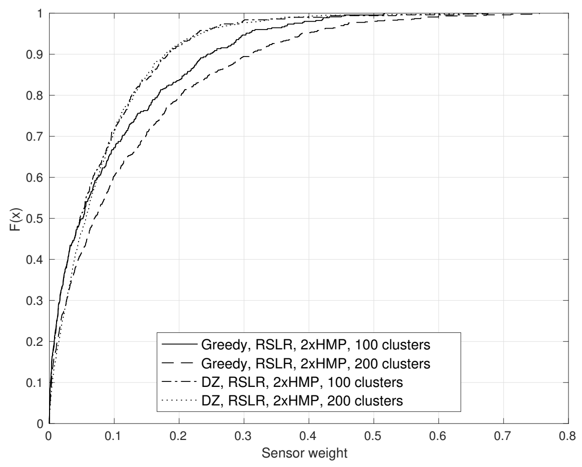

Figure 4. CDF of the weights of processed sensors. As greedy does not consider the weights when assigning sensors, heavier items tend to be selected compared to the DZ based algorithm.

than DZ based algorithm. As the greedy algorithm does not consider the weight of items when assigning sensors, heavier items will tend to be selected when compared to the DZ based algorithm, especially when the number of sensors is larger. We see that the greedy algorithm has a much higher portion of heavy items compared to the DZ based algorithm, e.g., for 200 clusters the average weight of sensors is 0.12 for the Greedy while 0.08 for the DZ based assignment.

Figure 5 shows the required capacities of basestations in order to process a given portion of the views. Compared to Figure 1, the figure shows results for larger systems of 9 basestations and 200 clusters with average cluster size of 6, maintaining an average degree of 0.6. We see that the DZ based algorithm requires significantly less capacity in order to process the same portion of views as the greedy algorithm. The advantage of the DZ based algorithm increases further with a second round of assignment, with up to 30% more views processed compared to the greedy algorithm.

VII. Conclusion

We considered the problem of assigning sensors to computing resources located at or near base stations in such a way that the total number of multi-views processed within a time frame is maximized. We showed that the problem is NP-hard and proposed an approximation algorithm by transforming the problem to a generalized assignment problem with dependent profits and by considering dominating sets of items for clusters of sensors. Our numerical results show that using the approximate dominating sets of items provides good performance at low computational complexity, and combined with the DZ algorithm it can provide significant bandwidth savings compared to the greedy solution. Our work could be extended to consider constrained processing resources, and to the case of mobile sensors with interesting application in vehicular safety and urban UAV networks.

References

[1] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3 – 19, 2013.

[2] S. O. Ba and J.-M. Odobez, *Probabilistic Head Pose Tracking Evaluation in Single and Multiple Camera Setups*. Springer Berlin Heidelberg, 2008, pp. 276–286.

[3] M. Du, A. C. Sankaranarayanan, and R. Chellappa, "Robust face recognition from multi-view videos," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1105–1117, March 2014.

[4] L. Baroffio, A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, G. Dán, E. Eriksson, V. Fodor, J. Ascenso, and P. Monteiro, "Enabling visual analysis in wireless sensor networks," in *Proc. of IEEE Intl. Conf. on Image Procesing (ICIP), Show and Tell*, October 2014.

[5] E. Eriksson, G. Dán, and V. Fodor, "Predictive distributed visual analysis for video in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1743–1756, 2016.

[6] ——, “Algorithms for distributed feature extraction in multi-camera visual sensor networks,” in *Proc. of IFIP/TC6 Networking*, 2015.

[7] D. B. Shmoys and Éva Tardos, “An approximation algorithm for the general assignment problem,” *Mathematical programming*, vol. 62, pp. 461–474, 1993.

[8] R. Cohen, L. Katzir, and D. Raz, “An efficient approximation for the generalized assignment problem,” *Information Processing Letters*, vol. 100, no. 4, pp. 162–166, 2006.

[9] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. New York, NY, USA: Springer-Verlag Berlin Heidelberg, 2004.

[10] U. Feige, D. Peleg, and G. Kortsarz, “The dense k-subgraph problem,” *Algorithmica*, vol. 29, no. 3, pp. 410–421, 2001.

[11] M. E. Dyer, “An O(n) algorithm for the multiple-choice knapsack linear program,” *Mathematical Programming*, vol. 29, no. 1, pp. 57–63. [Online]. Available: http://dx.doi.org/10.1007/BF02591729

[12] E. Zemel, “An O(n) algorithm for the linear multiple choice knapsack problem and related problems,” *Information processing letters*, vol. 18, no. 3, pp. 123–128.

[13] R. Bar-Yehuda, K. Bendel, A. Freund, and D. Rawitz, “Local ratio: A unified framework for approximation algorithms. in memoriam: Shimon even 1935-2004,” *ACM Comput. Surv.*, vol. 36, no. 4, pp. 422–463, Dec. 2004. [Online]. Available: http://doi.acm.org/10.1145/1041680.1041683

[14] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley and Sons, 1990.

[15] P. Erdős and A. Rényi, “On random graphs I,” *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.