Load-balanced and Energy-efficient Coverage of Dispersed Events Using Mobile Sensor/Actuator Nodes

Wassila Lalouani*, Mohamed Younis**, Mohamed El-Amine CHERGUI*** and Nadjib Badache****

*High National School of Computing Science, Algiers, Algeria. Email: w_lalouani@esi.dz

**Dep.t of Computer Science and Elect. Eng., University of Maryland Baltimore County. Email:younis@cs.umbc.edu

***USTHB, Faculty of Mathematics, Laboratory RECITS, Algiers, Algeria. mchergui@usthb.dz

****Department of Theories and Computer Engineering, CERIST, Algiers, Algeria. Email: badache@mail.cerist.dz

Abstract—**We consider networks where mobile sensor/actor nodes move to specific locations in order to conduct data collection or deliver a response to an event. The challenge is to find the best tour for the mobile nodes in order to visit the given set of locations. In this paper, the objective of the optimization is to extend the node lifetime by emphasizing both path efficiency and balanced energy consumption when identifying and assigning tours to mobile nodes. Compared to existing schemes in the literature, we consider the initial position of mobile sensors when determining the tours. We formulate the optimization as a balanced multi-salesman travel problem and propose a solution based on a two-step approach. First, we determine the shortest tour that includes all event locations by forming the Hamiltonian cycle. Then, we formulate the optimal partitioning of such a cycle as a linear program (LP) where the objective is to reduce the tour length while minimizing the maximum tour a node has to make. For scalability and to expedite convergence, we propose a method for solving the LP formulation based on Branch & Price algorithm. The simulation results confirm the effectiveness of our optimization formulation and the advantage of our solution compared to competing schemes.**

Keywords: **Sensor networks, sensor dispatch, mobile sensor/actors, tour optimization.**

I. INTRODUCTION

Wireless sensor network (WSNs) can be beneficial in many applications like environment monitoring, battlefield surveillance, and target tracking. Traditionally, statics sensors are deployed in the WSN area to carry out the application tasks. Recent advances for Microelectromechanical systems have allowed the emergence of mobile nodes with enhanced sensing and computing capabilities. These nodes may move on demand in order to support statics sensors or to fulfill specific application requirements. Depending on the application, mobile nodes may be expected to accomplish diverse missions such as collecting on-site data, replacing broken nodes, and even responding to a serious or suspicious events. Thus, introducing the mobility to a sensor network improves its capabilities and reduces the deployment and maintenance costs [1][2]. One of the most important problems in the context of hybrid WSNs is how the mobile nodes fulfill their missions in an efficient manner. In fact, the increased movement of these nodes diminishes their lifespan which has a direct impact on the quality of service for the application. The objective of the mobile node dispatch optimization is to find the best tour to visit a given set of locations [3].

Published node dispatching schemes have focused on reducing the total distance to be travelled by the mobile nodes as a means for gauging the energy consumption overhead and assessing the efficiency of the approach. However, these schemes have not considered balancing the overhead among nodes, which is important for maximizing the robustness of the network to unpredictable tour patterns and increased future requests. The importance of energy balancing among mobile devices is due to the fact that some nodes may travel longer distances and deplete their energy supply at a high rate, which negatively affects the overall lifetime of the WSN [1][2]. Yet, focusing on load balancing alone will increase the mean distance travelled by the nodes and increase the overall energy consumed in the WSN to achieve some desired application service. Therefore, in this paper our objective is to emphasize both path efficiency and balanced energy consumption by formulating the dispatch problem as min-max optimization of the distance travelled by the individual mobile nodes.

When the number of mobile nodes matches the event locations, the dispatch optimization becomes an event-to-node assignment problem based on the proximity of the event location to the initial (current) position of the individual nodes. However, if the number of mobile nodes is insufficient, some (or all) nodes will have to tour multiple event locations and the problem becomes more complex. In that case, existing solutions usually group the event locations into clusters and then assign a node to serve each of them [2][3]. While these solutions have considered a variety of optimization objectives, they did not take into consideration the initial position of nodes when forming the clusters and defining the tours, which may significantly increase the overall travel distance and consequently shorten the network lifetime. We overcome such shortcoming by formalizing the dispatch optimization as a Balanced Multiple Traveling Salesman Problem (bmTSP) which considers both node and event locations to determine the assignment. The mTSP is a variant of the TSP problem and strives to find tours for n salesmen, all of them are to start and end at a depot, such that each location is visited exactly once and the total cost of visiting all locations is minimized. Meanwhile for the bmTSP, (1) the solution should strive to equalize the length of the path travelled by the salesmen, and (2) a mobile node should start from its initial position and does not have to return back to that position since it will serve other event locations in the future. Thus, the bmTSP is a more complicated variant of the mTSP problem, which is known to be NP-hard.

Given the complexity of the bmTSP problem, the optimization would involve a large number of variables, reflecting all possible tours of event locations and combinations of node assignment to these tours. To tackle this challenge, we pursue a two-step approach. First, we determine the shortest tour that includes all event locations by forming the Hamiltonian cycle. Then, we formulate the optimal partitioning of such a cycle as a linear program (LP) such that every mobile node serves a partition with the least possible tour length and while minimizing the maximum tour a node

978-1-4799-5952-5/15/$31.00 ©2015 IEEE

has to make. Although, such an approach considerably reduces the complexity of the problem, existing LP solvers use the branch and bound method which requires the consideration of all possible routes in order to determine the exact solution. Since most the variables corresponding to tour combinations will be equal to zero in the solution, we employ the column generation with constrained branching, i.e., Branch & Price algorithm, to enable scalability for large setups and expedite the convergence. Figure 1 shows an articulation of a scenario involving three nodes. Our approach is validated through simulation and is shown to outperform a competing scheme.

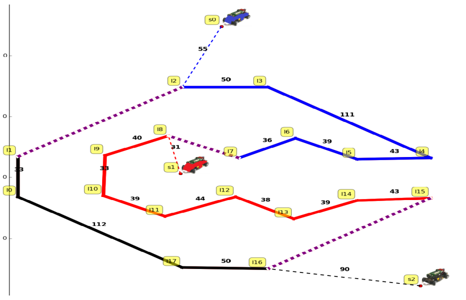

Figure 1: An example scenario where three nodes are to optimally tour 17 event locations

The paper is organized as follows. The next section summarizes related work in the literature. Section III discusses the system model, states the assumptions and analyzes the problem. The optimization formulation is provided in Section IV and the solution is discussed in Section V. Section VI presents the simulation results. Finally section VII concludes the paper and highlights our planned extension.

II. Related Work

Mobility for improved operation: Published techniques that take advantage of mobile nodes can be classified based on the node mix in the WSN. When all nods can move, such mobility has been exploited as a means for improving the quality of the service that a WSN offers or deal with operational challenges. Examples include improving data fidelity, increasing connectivity, filling coverage holes or approximate the event distribution [4]-[7]. However, these approaches do not consider the travel path optimization when multiple nodes are involved. Although the node dispatch problem is considered in [8], the number of event locations is assumed not to exceed the available node count and thus the problem becomes node assignment rather than path optimization. On the other hand, in hybrid WSNs the stationary nodes probe the environment while the mobile nodes provide a complementary role such as collecting more data to improve fidelity and event tracking [9], tolerating loss of coverage and connectivity [10][11], and performing application-specific tasks [12]. Although the travel distance overhead is factored in, the main focus is on the application aspect. In addition, nodes often do not tour part of the area and the dispatch problem does not arise.

Touring services: The use of mobile nodes to tour various location in an area of interest has been pursued in quite a few publications. A touring node can serve in WSNs as a *mobile relay node* (MRN) [13] [21][23][24] which relay data between nodes or network segments, as a *mobile data collector* (MDC) [14][22], which visits the individual segments and carries data to the sink, or as a *mobile base-station* (MBS) [15][17][18][23]. While a key objective of published algorithms in this category is to find the shortest tour path along which the node visits a set of locations, delay and buffer size are the main concerns in quite a few approaches. This particularly applies to work on MDC and significantly affects the solution mechanism. To illustrate, if the mobile node cannot store more than a certain amount of data, it will be forced to go to the destination(s) to offload the data payload [22]. That makes the optimization derived by the volume of data rather the travel distance. In addition, the delay of data delivery makes the provision of rendezvous points or with data ferry an important issue [16][23].

On the other hand, mobile nodes have been used for collecting data in an energy-efficient and reliable manner when routes between different nodes of a network are not available. For example, in [17] the authors have proposed energy efficient data collection protocols in single-hop WSNs by employing *data MULEs* which are capable of short-range wireless communication and move in an uncoordinated manner to provide connectivity in sparse WSNs. In addition, Alsalih et al. [18] have proposes data collection schemes which place MBSs such as autonomous unmanned vehicles (AUVs) in order to prolong the network lifetime. However, the tour length is not considered as an optimization metric in most of these schemes. Finally, grouping stops, either evert location or data sources, based on proximity has been pursued in [13][16][19][20] as a means to minimize the tour of mobile nodes. However, the positions of the nodes have not been factored in the solution.

Dispatch Problem formulation: As the mTSP is the core of vehicle routing problems (VRPs), most of the mathematical formulations of VRPs are variations and/or extensions of the mTSP problem. Different types of integer programming formulations have been proposed for the mTSP problem including assignment based formulations, a tree-based formulation and a three-index flow-based formulation [25]. With the exception of the latter, these formulations cannot be used in our case, since the mobile nodes have different initial positions. The three-index flow-based formulation uses very large number of variables even for moderate sized mTSP, which makes the optimal solution of the model impractical. Published solutions for the mTSP problem can be classed as: exact, heuristics and transformation to TSP. Examples of existing exact solutions include integer linear programming formulations [26], and Lagrangean relaxation [27]. These exact solutions can handle only problems with small sizes.

Some of the published approaches transform the mTSP to the standard TSP. As reported in [25], those transformation are not efficient, since the resulting TSP is highly degenerate, especially with increased number of salesmen. Some work has considered graph based transformations [28]. Unlike published schemes, this paper proposes more simplified transformation based on two steps, (i) a Hamiltonian cycle is formed for all locations, and then (ii) optimal partitioning of such a cycle is sought. We further exploit the well-known branch and price algorithm to solve the cycle partitioning formulation. The branch and price (B&P) has been used to solve similar problems in the literature, e.g., the Multiple Tour Maximum Collection Problem [29]. We use the B&P to solve the bmTSP problem. Finally, the approach of [3] and MAM [30] tackle the same of problem we address in the paper. The former, which we use as a baseline for comparison in our simulation, pursues clustering the event locations, and does not factor the node positions in the optimization. MAM separates the tour formation and the location assignment problems and does not optimize the node dispatch in an integrated manner.

III. System Model and Problem Formualtion

A. System and Problem Models

We consider a hybrid WSN that is composed of a set of stationary sensors and a smaller set of mobile nodes. The mobile nodes are assumed to have similar capabilities, i.e., the same initial energy and motion speed. Compared to the stationary sensors, the mobile nodes are equipped with advanced sensing and actuation devices and are employed to verify the accuracy of situational assessment and perform tasks in response to certain events [2]. In other words, the mobile nodes are randomly deployed to support the stationary sensors in fulfilling the application requirements. Thus, the locations to be visited by mobile nodes depend on the events that the stationary nodes detect. Since the event locations are unpredictable and change over time, we divide the time into multiple rounds where in each round the stationary sensors identify a set of points of interest in the monitored area. Each of these points is to be visited once by a mobile node. The problem addressed in this paper can then be stated as follow: Given a set m of locations $L = \{l_1, l_2, ..., l_m\}$, we aim to optimally assign those locations to appropriate mobile nodes $S = \{s_1, s_2, ..., s_n\}$. We make no assumption about the event locations and the initial position of the mobile nodes.

B. Optimization Objective

Since the mobile nodes are energy-constrained, the optimization problem is geared for minimizing the travel distance in order to extend the network lifetime. One possible objective function that is considered in [3], is to reduce the total distance travelled by all nodes. However, balancing the load among the mobile nodes is very important, especially since the points of interest are not uniformly distributed in the area and the load on the individual mobile nodes would vary significantly. Through a detailed example, it was shown in [2] that more events could be served when the energy consumption is balanced among the individual mobile nodes. However, load balancing alone could increase the mean travel distance. Different from the existing work, our objective is to minimize the total energy cost due to the node movements while equalizing the travel load among the mobile nodes.

When $|S| = |L|$, the node assignment (dispatch) problem becomes the least cost assignment of nodes to points of interest, where the cost corresponds to the distance a node travels from its initial position. However, when $|S| < |L|$, nodes should start from their initial positions and then visit a set of locations to ensure that all locations are visited. In order to deal with such a situation, published approaches tend to group events locations into clusters then assign each of these clusters to an appropriate mobile node without considering the initial positions of nodes. Such an approach does not lead to finding the optimal travel distance. We aim at overcoming this shortcoming.

C. Problem Analysis

We model the points of interest and the initial positions of mobile nodes as vertices $V = \{v_0, v_1, ..., v_{n+m}\}$. The problem could be then formalized using a directed graph $G = (V, E)$, where E is the set of edges denoting the direct links between vertices, i.e., between pairs of events locations and between mobile nodes and these locations. To measure the cost of travelling, we define the matrix $D = \{d_{ij}, | v_i, vj \in (S \cup L)\}$, where d_{ij} is the weight of an edge in G. To prevent a mobile node s_i from returning to its initial location and creating a cycle, the cost of a link to a vertex that corresponds to s_i initial position is set to ∞. Similarly, the links between a pair of mobile nodes is set to ∞. We want to partition G in order to form the shortest n non-overlapping and cycle-free tours, each containing one mobile nodes such that the tours include all event locations while being close in length. Such an optimization is a variant of the Multiple Traveling Salesman (mTSP) problem. More precisely, we try to solve a Balanced Tours Multi-Traveling Salesman Problem (bmTSP). However, the mTSP is known to be NP-hard [26].

IV. Tour Length Optimization Formulation

As pointed out, we are concerned with the problem of optimal mobile nodes assignment to the event locations such that the network lifetime is increased. Specifically, given n nodes and m locations, the objective is to find n paths with minimum balanced total travel, whereby all the event locations are visited exactly once and each node visits at least one location. In the previous section we have shown that such optimization is in fact a bmTSP problem. In this section, we propose a formulation for solving such a bmTSP problem to enable efficient handling of large m and n values.

Let R_k be the set of paths for node s_k. We define the binary variable y_{kr} to indicate whether $r \in R_k$. Thus, the tour constitutes a route for one mobile node that visits a subset of locations points, i.e., while factoring in the distance to be travelled by the node to the assigned locations. We associate a cost with each route matching the sum of the weights of all links on the route. Given a set of m event locations, we have 2^m routes which correspond to all possible subsets of event locations. As y_{kr} associates a route r to a node s_k, we have $C_n^{2^m}$ possible variables. Obviously, the complexity of the problem exponentially grows with an increased number of visited locations.

We make the following two key observations that will enable the simplification of the solution: (1) the nodes should visit their subset of locations in an optimized manner to reduce the travelled distance. The travel path of a node s_k is modeled as a set of consecutive directed links. A link exists if s_k traverses l_j just after visiting l_i. Thus the route that does not reduce the travel path for s_k should not be considered; (2) since the objective is to reduce the total distance travelled by all nodes, we may construct a TSP path for all locations and then partition it. This not only meets the objective but also reduces the number of variables in the formulation. As shown in Figure 1, determining the shortest tour that visits all locations exactly once and returns to the starting point will cut the number of considered links and sure will reduce the number of variables. In the figure, once being at l_0 it is more appropriate for s_i to visit l_i because coming from another location may increase the travelled distance. Such a path is referred to in the literature as the Hamiltonian cycle.

Thus, in order to simplify the solution, a two-step approach is pursued: (i) determine the Hamiltonian cycle for all event locations in order to reduce the overall distance, and consequently energy consumption; (ii) formulate the optimization problem as a partitioning problem of the Hamiltonian cycle. The Hamiltonian cycle limits the number of edges in G that ought to be considered. Each event-related vertex in the graph has exactly 2 links rather than (m-1) links.

The set of all event locations is denoted according to the cycle $0, 1, \ldots, m$. The set of arcs related to event location l_i within the cycle are denoted as l_{i-1} and l_{i+1}. There is also a link connecting each node-vertex to each location-vertex in the cycle. There are no arcs ending at the node initial locations. Defining T as the total travelled distance in the optimal node assignment and $c_r = \sum_{i,j\in r} d_{ij}$ as the travel distance of the route r, the objective function can be formulated as:

$$\text{Min Max}\left(\left|\sum_{r\in R_k} y_{kr*}\, c_r - \sum_{r\in R_{\acute{k}}} y_{\acute{k}r*}\, c_r\right|,\ \left(T - \sum_j \sum_{r\in R_k} y_{kr*}\, c_r\right)\right)$$

$\forall k \neq \acute{k}$, with the following constraints:

$$\sum_{k\in S}\sum_{r\in R_k} y_{kr} * P_{pr} = 1 \quad \forall p \in L\,, \tag{1}$$

Where $P_{pr} = 1$ if location p is a part of route "r", and 0 otherwise.

$$\sum_{r\in R} y_{kr} = 1 \quad \forall k \tag{2}$$

$$\sum_{r\in R}\sum_{p\in L} y_{kr} * B_{pr} * P_{pr} = 1 \forall k \tag{3}$$

Where $B_{pr} = 1$, if location p is the nearest to the beginning of route r, and 0 otherwise.

$$y_{kr} \in \{0,1\} \forall k, r \tag{4}$$

Where the objective is to minimize the maximum difference between the distances travelled by each pair of mobile nodes while reducing the total distance for all nodes; constraint (1) enforces that all visits are to be completed once; constraint (2) ensures that every node participates; constraint (3) excludes the links that are inconsistent with the optimization objective since it does not make sense to assign a non-TSP, i.e., not shortest, route to a node. Constraint (3) opts to reduce the search space, probably at the expense of optimality because a feasible route should correspond to the TSP path. In other words, the y_{rk} is a feasible variable if there is no location l_p within the Hamiltonian cycle relating l_i and l_j and is nearer to s_k than l_i. where the route r includes the sub path relating location l_i and l_j. In the next section, we develop two key simplifications to enable the handling of large WSN setups.

V. Optimization Approach

While there are several efficient algorithms available to solve linear programs, those algorithms require that all variables to be considered at the initialization phase. Although, our formulation reduces the complexity of the problem, it still involves many variables and risk scalability for large setups. In fact, there are m^n possible solutions to this problem. As most variables will be equal to zero in the optimal solution, we are interested in selecting only a subset of variables that need to be considered. The column generation method exploits such an idea in order to solve a variety of NP-hard problems such as crew scheduling and vehicle routing [31]. The appealing feature of column generation is to consider the optimization problem only for a sufficiently meaningful subset of variables. More variables are added only when needed. The column generation method includes two steps; first it considers the optimization of a restricted version of the problem, referred to as restricted master problem (RMP). RMP maintains the same objective function of the optimization and considers dual multipliers for each constraint. Given a non-negative vector of dual variables, the second step (pricing or sub-problem) progressively adds new potentially good columns, i.e., variables, to reduce cost until an optimality criterion is attained or there is no variable to add. If there is a variable that has improved the solution, i.e., progress has been made, the process is repeated. In each iteration the RMP is optimized using popular LP solvers, e.g., simplex algorithm, and we look for a variable to add to the considered subset. More details about the column generation method can be found in [31].

In order to solve our mathematical model, we decompose the problem into master problem and the pricing. As the column generation approach does not automatically guarantee optimal solutions, the branch and bound (B&B) algorithm often is used with the column generation method to enhance its performance [31]. In the rest of this section, we show how we solve our optimization formulation for large setup.

A. Column Generation

To adopt the column generation method, we first decompose the optimization formulation into master and pricing problems. The master problem is an integer program whose solution cannot be obtained directly, so its LP version based on Lagrangean relaxation is solved instead. Careful consideration of the optimization model in the previous section reveals that only constraint (1), i.e., the assignment constraints, involves all mobile nodes while the remaining constraints are dealing with each mobile node separately. In such a situation, the Lagrangean relaxation or decomposition could be applied to break up the overall problem into a master problem and a pricing problem for each node. This approach is promoted in [31] as the most successful decomposition for similar optimization problems. Thus, the master problem is defined through constraint (1) and the objective function. In other words, the master problem specifies the assignment of nodes to exactly one location and the binary requirement on the variables. The rest of the constraints are part of the pricing problem which has a modified objective function. Since the nodes have a different initial positions and energy, we have n independent pricing problems.

We are confronted with two issues: (i) the pricing problem looks for an additional variable to nominate for inclusion based on a cost/reward value. However, our linear model does not associate a cost to variables. In fact, contrarily to the objective function that tends to reduce the total travel distance where the cost of each variable is the distance traveled by a node, in our multi-objective function there is no specific cost associated with each variables; (ii) the pricing problem exploits the dual variables associated with each constraints, while the variable in constraint (1) are associated to given event locations. Obviously, within the pricing problem of node s_k, we are interested in testing the attractiveness of each sub-path within the Hamiltonian cycle for s_k. Therefore, the constraint should be reformulated in such a way that it reflects a given route (sub-path) rather than a given location. To overcome these two issues, we introduce a new notation that allows us to express the cost (reward) associated for each route. We define the cost of a path r to node s_k as $Price_{kr}$, which represents the attractiveness of route r. In order to calculate it, we fix the assignment of route r to a node s_k within the current solution (based on the recent iteration). The difference between the objective function obtained with and before the assignment constitute the $Price_{kr}$.. upon calculating the price, we should consider the assignment that enhance the objective function. Obviously the most attractive variable is the one with the most positive reward since it increases the value of the objective function. The restricted master problem can thus be stated as:

$$\text{Max}\left(\sum_r \sum_k Price_{kr} * x_{kr} * y_{kr}\right)$$

Subject to:

$$\sum_{r_i \in L} \sum_{\acute{r} \in R} \sum_{k \in S} x_{k\acute{r}} * y_{k\acute{r}} * p_{kr_i} = |r| \quad \forall r, \tag{5}$$

$$\sum_r y_{kr} = 1 \quad \forall k \tag{6}$$

$$x_{kr} \geq y_{kr} \quad \forall k, r \tag{7}$$

$$y_{kr} \in \{0,1\} \forall k, r \tag{8}$$

In this model, a binary value x_{kr} indicates whether the variable y_{kr} (belong to the set of selected column) is used or not. Constraint (5) ensures that each location is visited at most once. Constraint (5) is a reformulation of constraint (1) and is associated with the route r to guarantee that all tasks are to be completed. It also enables finding the dual variables u_{rk} for the constraint to be used in the pricing problem for identifying new variables y_{kr}. Constraint (6) limits the number of routes of a node to *1*. Constraint (7) ensures that the selected route for each node belongs to the feasible solution.

In the column generation methodology, the set of columns in the linear master problem is limited to only those that have already been generated, hence the term restricted master problem. Thus, we should find a set of initial variables for the master problem. We apply Munkres algorithm to assign one location for each node based on the cost matrix that represents the distance between the locations and nodes. The matrix is then updated with the new position of nodes. The procedure is repeated to assign the remaining locations in the same way. Since the obtained assignment would not obey constraint (2), a node may not follow a continuous travel path within the Hamiltonian cycle; therefore we assign to a node the longest consecutive sequences of locations within the cycle. The sequences are then linked to meet the constraint (1).

In addition to the prime variable y_{kr}, the duals variables associated to the constraints are also obtained. We use u_{rk} to denote the dual variable associated with constraint (4) for route r, the pricing problem determines whether some variable X_{kr} has a positive reduced cost. This condition can easily be stated as follows. Note that X is different from x that is used in the master problem.

$$\text{Max} \left(\sum_r (Price_{kr} - u_{rk}) \times X_{kr} \right.$$

Subject to:

$$\sum_r X_{kr} = 1 \ \forall k \tag{9}$$

$$X_{kr} \in \{0,1\} \forall k, r \tag{10}$$

B. Branch and Bound (B&B)

While the column generation employs a good subset of the possible variables, it does not automatically guarantee optimal solutions. When the column generation algorithm cannot generate new variables to incorporate to the RMP, we exploit B&B branch in order to progressively explore the overall search space. B&B organizes the solution space of the relaxed problem into a branching tree. Thus, B&B deals with the size of the problem by successively partitioning the solution space into branches until the optimal solution is found. A branch contains a set of feasible solutions that meet the problem constraint and allow us to exclude options that cannot enhance objective function. To apply B&B, we o take advantage of the price already calculated. and rank variables in increasing order according to their price. While variables with positive price should be included as a column to solve the RMP, the variables of the worst price should be considered to determine the pair of routes that deteriorate the objective function. In fact, the negative reward of variables is either due to the fact that the overall travel distance is not optimal or the difference in the length of nodes' travel paths is not optimal. Upon calculating the price, the solutions that have a non-optimal overall path should be discarded. At the same time, as our objective is to balance the nodes' travel distance, we determine the pair of routes (variables) that deteriorate the objective function when being in the same feasible solution. Thus, instead of branching on a single variable, we branch on route pairs by exploiting the relation between variables. More specifically we separate the search space based on the pair of variables $(y_{ku}\, y_{kv})$ that deteriorate the balancing value (or objective function) into four branches. A branch implies that the values of y_{ku} and y_{kv} are considered at the same time. Solutions within this branch should be pruned by bound (they exceed the objective function value). The others branches imposes feasible solutions within the branch cannot include both of them. Compared to single Variables branching, branching on pair of variables allow us to eliminate a several unpromising solutions. Furthermore, to reduce the computation time, we separate feasible solutions within a branch according to the constraint (5). Intuitively it makes sense, since progressively we will find only variables that improve the balanced optimal travel path by retrieving the worst solutions. A pseudo code summary of the various optimization steps is provided in Figure 2.

VI. PEFROMANCE VALIDATION

The effectiveness of our approach is validated through simulation. This section discusses the simulation setup, performance metrics and results.

A. Simualtion Environment and Performance Metrics

We have validated our approach through extensive simulation experiments. The simulation environment is developed in Python. In the simulation, nodes are deployed using a uniform random distribution in a rectangular area of 450 × 300 m^2. The

```
create master problem model.
generate the initial feasible pattern.
compute the price and objective.
find the unmatched variables pairs.
while (feasible pattern pool is not empty):
  relaxed =True
  While (Relaxed)
     solve master_problem(pattern, Objective, price)
     find the duals variables for each route.
      for each sensor k:
        solve the pricing problem
     find new variable.
     IF newVariable:
        insert the variable to the pattern
        compute the price.
        add unmatched variables pairs.
     else:
         relaxed=False
     End IF
   update the feasible pattern pool
 End
End
```

Figure 2: Pseudo code for the tour optimization solution

event locations are randomly picked within the area as well. The number of nodes and event locations are varied in the experiments. Only energy consumption due to motion is accounted for, assuming it is linearly proportional to the travelled distance. We compare the performance of our solution to that of [3] and with three variants of the optimizations model, representing the least total energy (*Energy*), i.e., distance, balanced travel load (*Balance*), and balanced remaining energy (*Rmg Blc*), respectively. We use branch and price (*B&P*) to refer to the simplified optimization model in Section IV and "*Travel Balance*" to denote the model in Section III, which is solved in our implementation using the Simplex algorithm; both of our models strive to minimize the total distance while balancing the load. The approach of [3] opts to reduce the energy consumption of all mobile nodes in every round and at the same time tries to balance the travel load. When the number of nodes is sufficient to cover the locations, the problem is transformed into a maximum cardinality matching problem in a weighted bipartite graph. In the case of insufficient node count, the locations are grouped using balanced clustering heuristic where each cluster is served by a distinct mobile node.

B. Number of Survived Mobile Nodes

The principal objective of the dispatch solution is to increase the network lifetime. In other words, we should extend the lifespan of mobile nodes. Since the event locations change over time, we have considered the number of alive (survived) nodes in each rounds, and measured the network lifetime by the number of rounds of dispatch. We fix the number of mobile sensors to 20 and the number of event locations to 100. The energy consumed by a mobile nodes is set to 0.21 J (joule) per inch [2]. Figure 3 shows that the balanced remaining energy (*Rmg Blc*) and the balanced travel distance (*Balance*) models yield the shortest lifetime due to the fact that they do not consider the reduction of the overall energy consumption. In fact, the energy balancing model does not consider the travel load, causing some nodes to move long distances. Although, the least total distance (*Energy*) model optimizes the total energy consumed in the dispatch, the load is unbalanced and some nodes exhaust their energy early. The baseline algorithm yields a longer system lifetime, because they adjust the clustering parameters to care for the total energy and to balance the load. However, our approach, namely *B&P* and "*travel Balance* yield the best results and significantly outperform the baseline solution. Such performance advantage is attributed to the consideration of the initial node positions and to the rigorous mathematical foundation of our optimization models. The model (Travel Balance) that considers multi-objective offer a better lifetime compared to the B&P due to the relaxed constraints, which could have potentially eliminated optimal tours from the solution set. Naturally the selection between our two optimization models will have be subject to computational complexity and optimality trade-off; nonetheless, both yield better results than the state-of-the art solutions in the literature.

C. Travel Overhead

We evaluate the overall distance travelled by the mobile nodes under different dispatch optimization algorithms. Because of the NP-hardness of the mTSP problem, we have first managed to run the optimal model (*Travel Balance*) for small networks due to the computation complexity (Figure 4). Then, in Figure 5, we compare the baseline solution with the B&P algorithm to address the scalability issue. In other words, we consider two scenarios, small and large setups. In the first scenario, we vary the number of event locations in each round and fix the node count to 20. In the second scenario, we consider 50 mobile nodes instead. Figure 4 shows the results for the small network. As seen in the figure the distance for all solutions increase with the number of event locations as there is more spots to visit. Obviously, the model that minimizes the total travel distance leads the way since it does not consider load balancing. Our two solutions come next and close the gap as the number of events increases. The other approaches that care for load balancing as a primary objective, i.e., *Balance*, and *Rmg Blc*, perform the worst in Figure 4, which is expected given their focus. A key observation is how our solutions outperform the baseline, which factors in both total travel overhead and load balancing. Figure 5 compares our *B&P* solution with the baseline approach for high event count that are spread in a larger 1000×1000 m^2 area. The figure demonstrates the performance advantage of our approach especially as the number of event grows.

D. Example Application Scenario

To highlight the key features on our approach, we compare the tours it forms for an example application scenario to those resulting when the baseline is employed. The tours are shown in Figure 6. As shown in Figure 6(a), the baseline solution forms three clusters $c_1 \leftarrow \{l2,l14,l5,l3,l10,l8\}$, $c_2 \leftarrow \{l9,l17,l16,l13,l0,l1\}$, $c_3 \leftarrow \{l11,l7,l6,l15,l4,l12\}$ with respective cost equal to 193, 196, 206 and then uses a maximum matching algorithm to assign clusters to nodes. Our

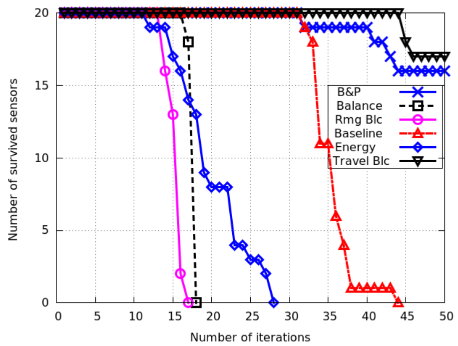

Figure 3: Number of live nodes as a function of rounds (an indication of network lifetime).

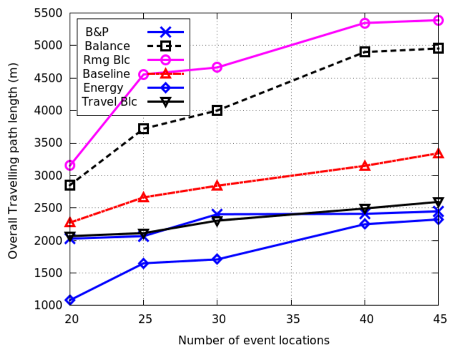

Figure 4: Total travel distance as a function of the number of event locations.

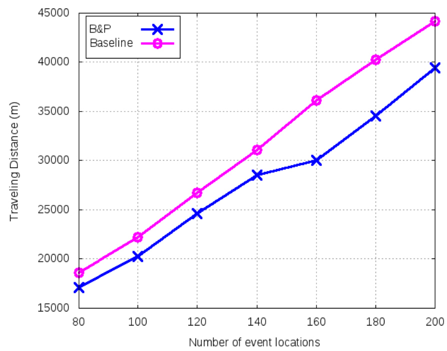

Figure 5: Comparing the travel overhead to baseline as the number of events grows.

approach first forms the Hamiltonian cycle and then partitions the cycle into three tours (Figure 6(b)). The bold dotted (purple) lines reflect the edges in the cycles that are not part of the tours. Although in the baseline, the tours for the clusters are balanced in length, such a feature is lost when factoring in the distance a node travels from its initial position to its assigned cluster. This is because the baseline solution separates the tour formation and the node assignment. The optimal solution in term of total travel distance has T=914, with the following assignment: $s_1 \leftarrow \{l_4,l_5,l_6,l_7,l_8,l_9,l_{10},l_{11}\}$, $s_0 \leftarrow \{l_{12},l_{13},l_{14},l_{15}\}$, and $s_2 \leftarrow \{l_{16},l_{17},l_0,l_1,l_2,l_3\}$. Our approach is obviously closer to the optimal total distance, yet while

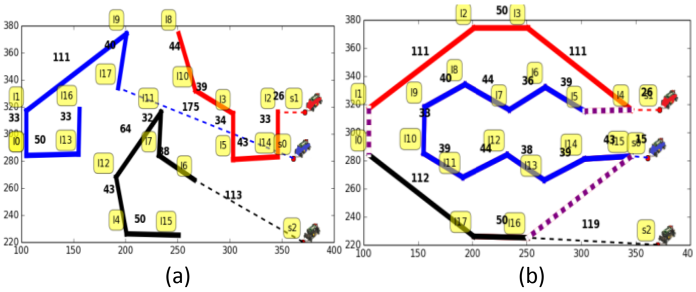

Figure 6: Comparing the tours of (a) the baseline (s_0=442, s_1=219 s_2=340, T=1001, balanced Travel =223), and (b) our approach (s_0= 410, s_1=298, s_2= 281, T=989, balanced Travel =129), for an example scenario of 18 event locations and 3 nodes,

minimizing the maximum tour where s_0 travels a shorter distance than in the baseline.

VII. CONCLUSION

In this paper, we have presented an energy efficient and balanced multi-node dispatch approach for covering dispersed event locations in WSN. We have formulated the dispatch optimization as a bmTSP problem and proposed a novel transformation of the problem into two-step optimization formulation. In the first we determine the minimum length cycle of all locations, and find an optimal partitioning of this cycle in the second step. We have further focused on the second step and developed an optimization model. Since such optimization model may not scale for large setups, we have employed B&P. Our approach can extend the system lifetime by reducing and balancing the energy consumption of mobile nodes. The simulation results have demonstrated the effectiveness of our approach and its performance advantage over competing schemes and variant models of interest. Our future work includes extending the approach to handle events regions in the presence of obstacles and optimizing delay. We envision also extending the solution to optimize the communication energy of the static sensors.

REFERENCES

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless Sensor and actor networks: Research Challenges," *Elsevier Ad hoc Network Journal*, Vol. 2, pp. 351-367, 2004.

[2] Y.-C. Wang, "Mobile sensor networks: system hardware and dispatch software," *ACM Computing Surveys*, 47(1), pp. 12:1–12:36, Jul. 2014.

[3] Y.-C. Wang, W.-C. Peng, and Y.-C. Tseng, "Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor network," *IEEE Trans. on Parallel and Distributed Systems*, 21(12), pp. 1836-1850, Dec. 2010.

[4] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Trans. Embedded Computing Systems*, 3(1), pp. 61–91, 2004.

[5] N. Heo and P.K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Trans. Systems, Man and Cybernetics—Part A: Systems and Humans*, 35(1), pp. 78–92, 2005.

[6] G. Wang, G. Cao, and T.F.L. Porta, "Movement-assisted sensor deployment," *IEEE Trans. Mobile Computing*, 5(6), pp. 640–652, 2006.

[7] P. Basu and J. Redi, "Movement control algorithms for realization of fault-tolerant ad hoc robot networks," *IEEE Net*, 18(4), pp. 36–44, 2004.

[8] Y.C. Wang, C.C. Hu, and Y.C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Trans. Mobile Computing*, 7(2), pp. 262–274, 2008.

[9] Y. Zou and K. Chakrabarty, "Distributed Mobility Management for Target Tracking in Mobile Sensor Networks," *IEEE Trans. Mobile Computing*, 6(8), pp. 872-887, Aug. 2007.

[10] G. Wang, G. Cao, P. Berman, and T.F.L. Porta, "Bidding protocols for deploying mobile sensors," *IEEE Trans. Mobile Computing*, 6(5), pp. 563–576, 2007.

[11] Z. Shen, Y. Chang, H. Jiang, Y. Wang, and Z. Yan, "A generic framework for optimal mobile sensor redeployment," *IEEE Trans. Vehicular Technology*, 59(8), pp. 4043–4057, 2010.

[12] M.B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-Based Multi-robot Coordination: A Survey and Analysis," Proc. IEEE, vol. 94(, no. 7), pp. 1257-1270, July 2006.

[13] F. Senel and M. Younis, "Optimized Interconnection of Disjoint Wireless Sensor Network Segments Using *K* Mobile Data Collectors," *Proc. of IEEE Int'l Conf. on Comm. (ICC'12)*, Ottawa, Canada, 2012.

[14] H. Almasaeid, and A. E. Kamal, "Data Delivery in Fragmented Wireless Sensor Netowrks Using Mobile Agents," *Proc. the 10th ACM/IEEE Int'l Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM),* Chania, Greece, October 2007.

[15] H. Almasaeid, and A. E. Kamal, "Modeling Mobility-Assisted Data Collection in Wireless Sensor Networks," *Proc. of the IEEE Global Telecomm. Conf. (GLOBECOM'08),* New Orleans, LA, Dec 2008.

[16] B. Kalyanasundaram, and M. Younis, "Using Mobile Data Collectors to Federate Clusters of Disjoint Sensor Network Segments," *Proc. of the IEEE Int'l Conf. on Comm. (ICC 2013)*, Budapest, Hunger, June 2013.

[17] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *J. of Mobile Networks and App.*, Vol. 11, pp. 327-339, 2006.

[18] W. Alsalih, H. S. Hassanein, S. G. Akl, "Placement of multiple mobile data collectors in wireless sensor networks*," Journal of Ad Hoc Networks*, 8(4), pp. 378-390, 2010.

[19] J. L.V.M. Stanislaus, M. Younis, "Delay-conscious Federation of Multiple Wireless Sensor Network Segments using Mobile Relays," *Proc. of the 76th IEEE Vehicular Technology Conference (VTC2012-Fall)*, Québec City, Canada, September 2012.

[20] A. Abbas and M. Younis, "Establishing Connectivity among Disjoint Terminals Using a Mix of Stationary and Mobile Relays Computer Communications," *Computer Comm*., Vol. 36, pp. 1411–1421, 2013.

[21] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," *Proc. ACM MobiHoc*, 2004.

[22] D. Jea, A.A. Somasundara, and M.B. Srivastava, "Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks," *Proc. First IEEE/ACM Int'l Conf. Distributed Computing in Sensor Systems (DCOSS '05)*, June 2005.

[23] R. Moazzez-Estanjini, J. Wang, and I.C. Paschalidis, "Scheduling mobile nodes for cooperative data transport in sensor networks," *IEEE/ACM Trans. Networking*, 21(3), pp. 974–989, 2013.

[24] F. El-Moukaddem, E. Torng, and G. Xing, "Mobile relay configuration in data-intensive wireless sensor networks," *IEEE Trans. Mobile Computing*, 12(2), pp. 261–273, 2013.

[25] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures" *Omega*, 34(3), pp. 209-219, 2006.

[26] R. V. Kulkarni, and P. R. Bhave, "Integer programming formulations of vehicle routing problems," *European Journal of Operational Research* 20, pp. 58–67, 1985.

[27] N. Christofides, A. Mingozzi, P. Toth, "Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations," *Mathematical Programming*, Vol. 20, pp.255–82, 1981.

[28] R. Jonker and T. Volgenant, "An improved transformation of the symmetric multiple traveling salesman problem," *Operations Research* 36(1), pp.163–7, 1988.

[29] S. E. Butt, and D. M. Ryan, "An Optimal Solution Procedure for the Multiple Tour Maximum Collection Problem using Column Generation," *Computers & Operations Research*, 26(4), 427-441, 1999.

[30] Y.-C. Wang, "A two-phase dispatch heuristic to schedule the movement of multi-attribute mobile sensors in a hybrid wireless sensor network," *IEEE Trans. on Mobile Computing*, 13(4), pp. 709–722, Apr. 2014.

[31] G. Desaulniers, J. Desrosiers, and M. M. Solomon, *Column Generation*, GERAD 25th Anniversary Series. (eds) Springer, New York.