Submitted to *IEEE/RSJ International Conference on Intelligent Robots and Systems*
Aug 2005, Edmonton, Canada

Task Allocation for Event-Aware Spatiotemporal Sampling of Environmental Variables*

Maxim A. Batalin and Gaurav S. Sukhatme
Robotic Embedded Systems Laboratory
Center for Robotics and Embedded Systems
Computer Science Department
University of Southern California
Los Angeles, CA 90089, USA
maxim@robotics.usc.edu, gaurav@usc.edu

Yan Yu[1], Richard Pon[2], Jason Gordon[2], M. H. Rahimi[1], William J. Kaiser[2], Gregory J. Pottie[2], and Deborah E. Estrin[1]
Center for Embedded Networked Sensing
[1]*Department of Computer Science*
[2]*Department of Electrical Engineering*
University California, Los Angeles
Los Angeles, CA 90095
kaiser@ee.ucla.edu

***Abstract*— Monitoring of environmental phenomena with embedded networked sensing confronts the challenges of both unpredictable variability in the spatial distribution of phenomena, coupled with demands for a high spatial sampling rate in three dimensions. For example, low distortion mapping of critical solar radiation properties in forest environments may require two-dimensional spatial sampling rates of greater than 10 $samples/m^2$ over transects exceeding 1000 m^2. Clearly, adequate sampling coverage of such a transect requires an impractically large number of sensing nodes. A new approach, Networked Infomechanical System (NIMS), has been introduced to combine autonomous-articulated and static sensor nodes enabling sufficient spatiotemporal sampling density over large transects to meet a general set of environmental mapping demands.**

This paper describes our work on a critical parts of NIMS, the Task Allocation module. We present our methodologies and the two basic greedy Task Allocation policies - based on time of the task arrival (*Time* policy) and distance from the robot to the task (*Distance* policy). We present results from NIMS deployed in a forest reserve and from a lab testbed. The results show that both policies are adequate for the task of spatiotemporal sampling, but also complement each other. Finally, we suggest the future direction of research that would both help us better quantify the performance of our system and create more complex policies combining time, distance, information gain, etc.

I. Introduction

A wide range of critical environmental monitoring objectives in resource management, environmental protection, and public health, all require distributed sensing capability for investigation of large and complex three dimensional spaces [1]. Mapping of spatiotemporally distributed phenomena generally requires that distributed sensors be immersed in the environment under study [2]. Important examples include 1. the measurement of spatiotemporally distributed solar radiation fields that directly determines the distribution of plant growth [3] and 2. direct measurement of atmospheric microclimate variables and the flux of water and carbon dioxide at the interface between the atmosphere and ecosystems. The fundamental spatial heterogeneity and temporal evolution of environments requires that measurements be distributed and raises many challenges for high fidelity characterization of environmental field variables.

The above applications of distributed sensing all require **high spatiotemporal fidelity measurements**. Since the scale of phenomena (for example, consider solar radiation) match the scale of the environmental structure (for example, plant structure) then the spatial mapping resolution may require centimeter precision. At the same time, the spatial extent of characterization must match the scale of the environment. For example, forest ecosystems require mapping over characteristic vertical extent from the forest canopy top to floor in height and many 10s of meters of lateral extent.

The conventional approach to study spatiotemporal variable phenomena, has been to deploy static distributed sensors. However, using static sensor networks alone may lead to measurement distortion; often as a result of improper spatial-sampling distribution (specifically due to mismatch between the spatial structure of the phenomena and sensor node placement). This may be inevitable because the rapid time evolution of environmental phenomena may be unpredictable at design time. Further, increasing sensor deployment density may induce excessive disturbance to the sensed environment and may be economically infeasible since every node in the network must then be equipped with a sensor capable of characterizing the studied phenomenon.

Since environmental dynamics drive unpredictable and variable sensor coverage requirements, sensor node *mobility* may be exploited to ensure adequate coverage. Mobile sensor nodes (in effect, robots) may respond to, adapt and optimize sensing fidelity according to spatiotemporally variable requirements. Mobility, therefore, permits high fidelity spatiotemporal sampling, while allowing the number of robots to be small (in relation to the number of static sensors that may otherwise achieve similar coverage). However, despite the benefits of a mobile robotic sensor approach, measurement distortion may still occur. Since navigation between sampling points takes time, unassisted robotic sensor sampling may occur at a rate insufficient to sample rapidly varying

*This material is based upon work supported by the National Science Foundation (NSF) under Grant No. ANI-00331481. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

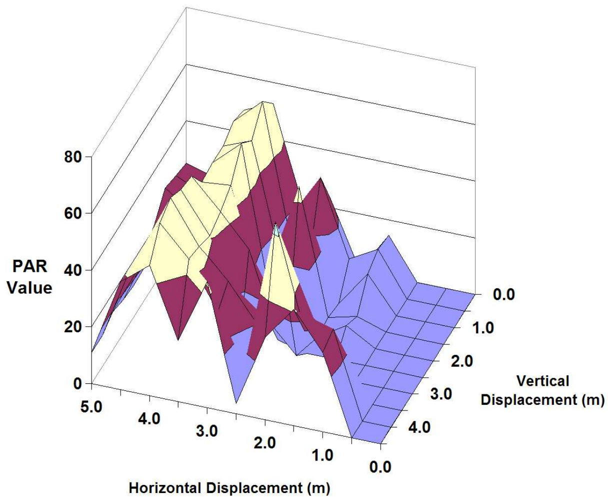


Fig. 1. Map of solar radiation intensity obtained in a forest ecosystem by a NIMS system transporting a light intensity sensor within the canopy. Solar radiation intensity is indicated in contours as it varies spatially according to horizontal and vertical sensor displacement.

phenomena. In the event that the robotic sensors may have no prior information about the precise location of time-varying phenomenon, then, in the worst case, robots may be required to sample the entire environment to achieve adequate coverage. This may delay data acquisition and induce a potential loss of sampling fidelity.

A. Motivating Application

The requirement for a sustainable and precise mobile sensor network monitoring capability for environments has inspired the development of Networked Infomechanical Systems (NIMS). NIMS [4] introduces infrastructure-supported mobility with actuated sensors that may autonomously explore a three-dimensional volume. The NIMS approach provides an unprecedented spatial extent and motion resolution, combined with sustainable operation. The robotic sensing methods reported in this paper reports are directed to the application of high fidelity mapping of solar radiation flux distribution in the forest environment. Solar radiation reaching the forest floor surface determines the growth of plant structures at this level. This radiation is spatially filtered by the complex ecosystem structure [3]. The Photosynthetically Active Radiation (PAR) spectral region of solar light illumination contributes to this growth and is directly measured in the investigation reported here. Figure 1 shows the results for the first high resolution mapping of PAR field variables in a forest ecosystem. This map represents the intensity of PAR measured in a vertical plane within the mixed conifer forest of the James San Jacinto Mountain Reserve [5] where NIMS is permanently deployed (Figure 2) and continuously operating. Note that the characteristic scale of variability for solar radiation is less than 1m and persists throughout the ecosystem.

B. General Approach: Combining Static and Mobile Sensing

In previous work [6] a new Networked Infomechanical Systems (NIMS) architecture that combined *both static and mobile* sensor nodes was introduced. This architecture achieved a spatiotemporal environment coverage that was dramatically advanced over that of either system alone. It was shown that mobility allows the networked sensor system to always seek the spatiotemporal sampling distribution to achieve a *specified* fidelity of environmental variable reconstruction. It was also demonstrated that, mobility permits the NIMS system to respond to initially unpredictable and variable environmental evolution.

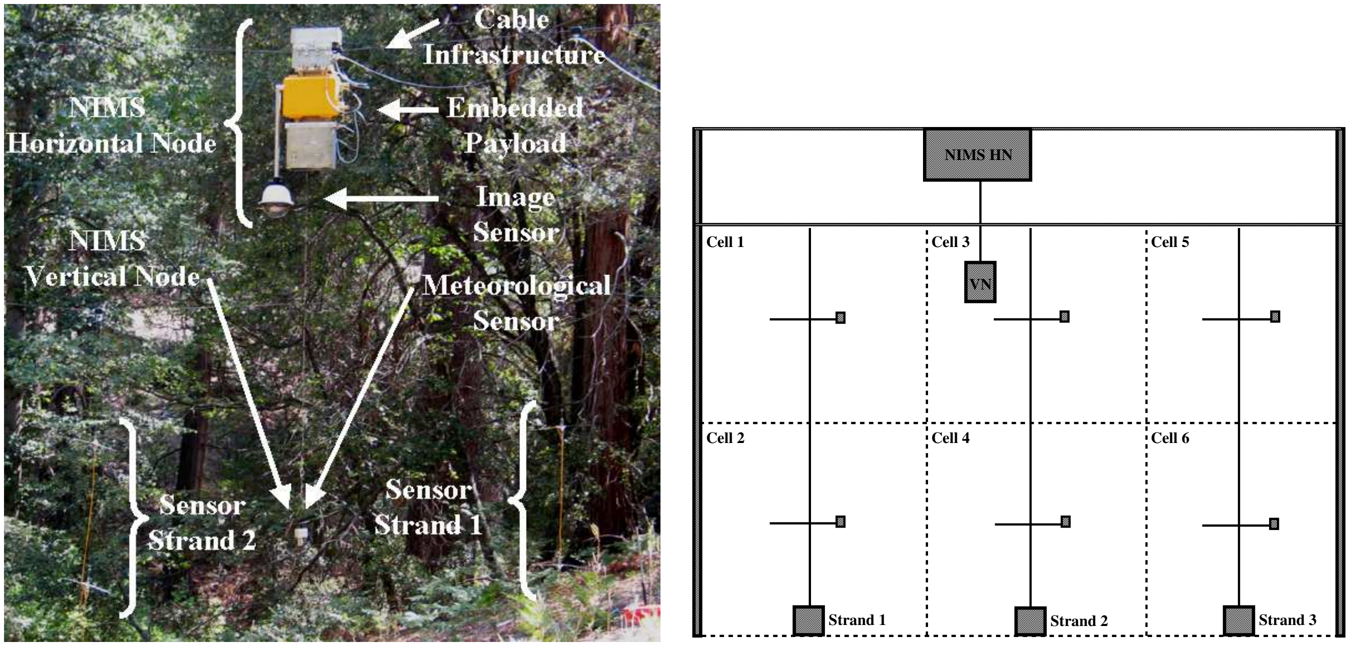
In [6] static sensor nodes are positioned in the volume surrounding a transect in which the mobile node operates. Every sensor node is responsible for reporting a phenomenon occurring in the vicinity. The mobile node then uses simple task allocation to determine which node has higher priority and then utilizes sampling algorithms (either raster or adaptive sampling) to sample only the vicinity of the node that detected the phenomenon. [6] experimentally shows the benefits of such a stratification of the sampling transect.

The rapid time rate of change of field variable value leads to a need for even higher performance in the response of the mobile sensing system to environmental events. In this paper we introduce two new Task Allocation algorithms used by NIMS and present the results of performance analysis obtained directly in the field and operating in the forest reserve [5]. In some cases, additional performance analysis is obtained from measurements on a physical testbed duplicate of NIMS that includes sensor data records obtained from the field. This permits reproducible, repeated measurements of the same events for a probe of system performance.

II. NETWORKED INFOMECHANICAL SYSTEMS

Figure 2 shows a NIMS system deployed in the forest reserve and continuously operating. This system includes a supporting cable infrastructure, a horizontally mobile embedded computing platform payload, image sensing systems, and a vertically mobile meteorological sensor package carrying humidity, temperature, and PAR sensing nodes. Wireless networking is incorporated to link static sensor nodes (suspended) with the vertically and horizontally mobile elements. The NIMS infrastructure is elevated in the environment within the forest and thus lies above environmental obstacles to solar radiation. The NIMS system is deployed in a planar transect of length 70m and average height of 15m with a total area of over 1,000 m^2.

The introduction of static sensor nodes in a deployment of sensors must also be compliant to the complex forest environment. This paper reports a NIMS architecture developed to enable distributed Task Allocation incorporating vertically suspended static sensors that are referred to as *sensor strands*. Sensor strands, also exploiting infrastructure, are suspended in the plane parallel to the NIMS sampling transect (Figure 2a). Every sensor strand includes two sensors separated by 1-2 meters. Data from strand sensor elements is sampled by



(a) Two Sensor Strands and NIMS Node

(b) Schematic representation of the deployed NIMS.

Fig. 2. NIMS deployed at the James San Jacinto Mountain Reserve (http://www.jamesreserve.edu). This image shows the NIMS cable infrastructure, horizontal transport node (carrying an embedded computing platform, image sensor, and vertical transport control, and vertically mobile meteorological sensor node.

an embedded sensor node based on the Intel $Stargate^{TM}$ platform. Communication between strand nodes and the mobile, horizontal node occurs over an IEEE 802.11b wireless interface. Figure 2b schematically shows the experimental architecture at the James San Jacinto Mountain Reserve. As shown in Figure 2b, there are three sensor strands with six PAR sensors deployed in this transect. Note that each sensor samples PAR values that are then considered to be representative of the light intensity in the vicinity of this sensor. Hence, the sampling transect is discretized into six cells. The sampling scheduling (determining where and when to sample) is guided by the Task Allocation system hosted on the NIMS node.

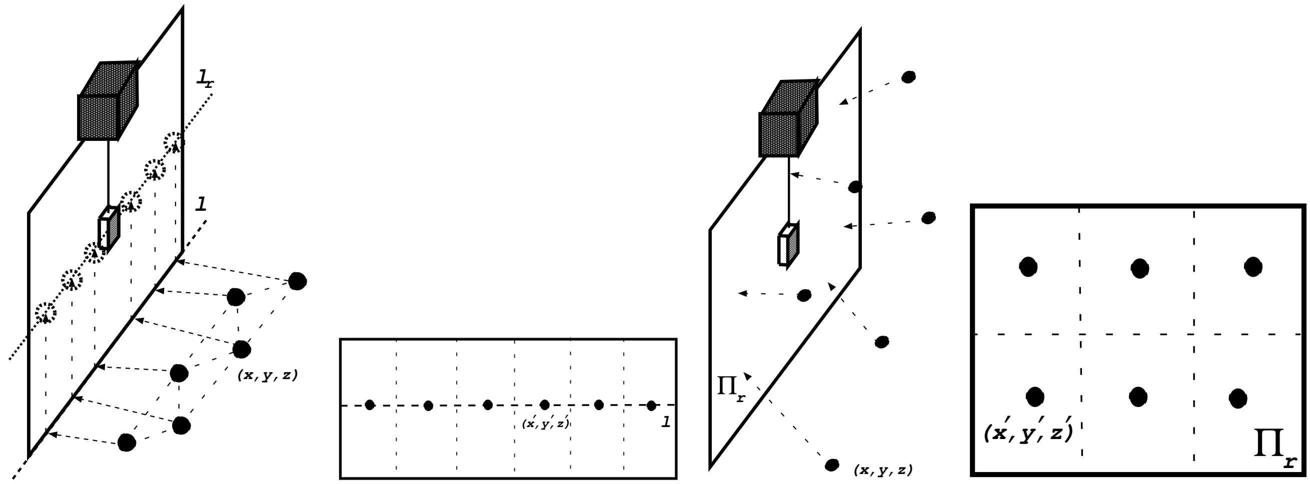
III. Task Allocation Algorithms

Task Allocation (TA) is the problem of assigning available resources to tasks. For a comprehensive overview of TA formulations see [7]. There are two major subdivisions: offline and online. Offline TA is the problem of assigning resources to tasks if certain information (e.g. the distribution of task arrival times, relative task priority) is known *a priori*. The assignment process proceeds offline. The offline TA problem, in its most general form, is equivalent to the conjunctive planning problem [8] which is NP-Complete.

Our focus here is on online task allocation. In online TA, all information about the tasks becomes available only upon task arrival. The assignment of resources to tasks must be computed in real time, one at a time. It has been shown [7] that greedy algorithms for an online TA problem, that is considered in this paper, are 3-competitive to an optimal offline solution. It has also been shown [7] that without domain knowledge about the problem there is no solution that is better than the one provided by greedy algorithms. Following the model in [9], we think of task assignment occurring in *decision epochs*. A *decision epoch* is a period of time during which only the tasks which have arrived since the end of the previous epoch are considered for assignment. Increasing the *decision epoch* to infinity converts the online TA into the offline TA problem. We model the NIMS system as an online TA problem, since it is designed for real-life autonomous field applications in dynamic environments.

Our work is related to the body of work on the problem of online multi-robot task allocation (MRTA). For an overview and comparison of the key MRTA architectures see [10], which subdivides MRTA architectures into behavior-based and auction-based. For example, ALLIANCE [11] is a behavior-based architecture that considers all tasks for (re)assignment at every iteration based on the robots' utility. Utility is computed by measures of acquiescence and impatience. Auction-based approaches include the M+ system [12] and Murdoch [13]. Both systems rely on the Contract Net Protocol (CNP) that puts available tasks for auction, and candidate robots make 'bids' that are their task-specific utility estimates. The highest bidder (i.e., the best-fit robot) wins a contract for the task and proceeds to execute it.

The proposed TA algorithm differs from the above MRTA



(a) The 2D-case

(b) The 2D-case projection onto the transect

(c) The 3D-case

(d) The 3D-case projection onto the transect

Fig. 3. Different SN topologies and corresponding projections onto the transect.

approaches. It relies on a static *network*, and communication, sensing, and computation are distributed. The motivation for the TA algorithm derives from the need to efficiently sample the phenomena instead of the entire environmental space. As has been discussed in the introduction, it is impractical to deploy sufficient numbers of fixed sensors to achieve required sensing fidelity. As will be shown, the system combining fixed and mobile nodes enables efficient sampling. TA becomes the primary driver of efficient data collection in this system, since it allows the user to select a subset of the environment for sampling, as opposed to sampling the entire environment. In addition, TA manages system resources, so that resources are not consumed unless assigned most effectively.

A. Methodology

The general online TA system functions in the following way. Suppose at a given decision epoch the system maintains a set of resources $R = \{r_1, ..., r_n\}$ and tasks $T = \{t_1, ..., t_k\}$. Tasks are prioritized based on a criterion C. C is an application dependent function and can combine such parameters as task arrival time, task importance, etc. A set of assignments $A = (l = min(n, k) : \{a_1, ..., a_l\})$ is computed as follows.

$$\forall_{a \in A} a = argmax_{j=(1,...,|R|)}(U(r_j, t)) \qquad (1)$$

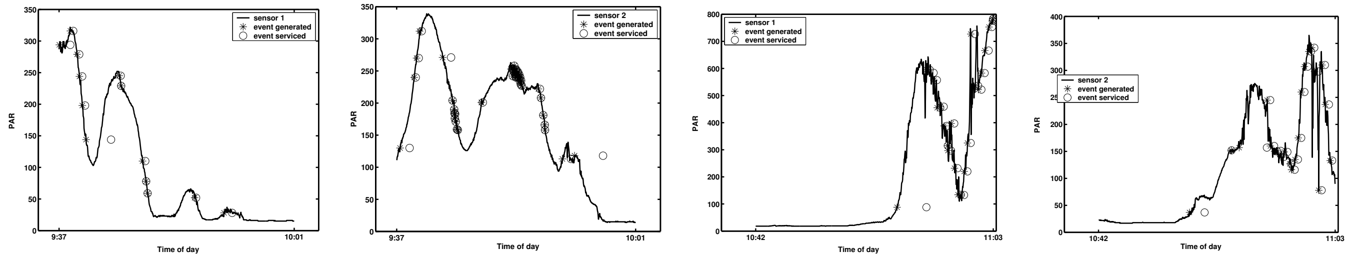
where t is the next unassigned task according to C and $U(r_j, t)$ is the j-th resource utility value for accomplishing t. The assigned resource and corresponding task are removed from R and T respectively, before the next assignment. The utility function is chosen to be application and resource dependent. In our model, once assigned, resources cannot be reallocated. After a resource has completed its task it becomes available for a new assignment. In the terminology of [14] we adopt a *commitment* as opposed to an *opportunism* strategy.

The system consists of a mobile node suspended on a cable and a static sensor network. We assume that the network is predeployed where each node knows its location in a global coordinate system. The network monitors the environment for events of interest (motion, change in light intensity, etc). The problem then is to prioritize the events, and drive the mobile node to a vantage point from which a particular event is better observed. Once the node arrives at the event location, the local phenomenon is measured. In TA terminology, a robot is a *resource* and a detection by a sensor node of an event requiring perusal by a robot is a *task*.

Figure 3 shows two network topologies that we define - positioned on the ground (the 2D-case) and more generally, in the volume surrounding the transect (the 3D-case). In order for the TA system to plan the node's motion the goal points should lie in the transect plane. Hence, we project the nodes locations onto the transect plane. As a result we get a set of points on a line l (2D-case, Figure 3a) or a plane Π_r (3D-case, Figure 3c), both of which lie in the transect plane. In the 2D-case, l is the line where the transect plane intersects the ground plane. Since, the mobile node cannot move along that line, we translate l to a parallel line l_r on the transect. We define the projection function in the 2D-case $PROJ_{l_r}$ and 3D-case $PROJ_{\Pi_r}$.

Based on tasks projected locations TA divides the transect into *slices* (2D-case, Figure 3b), or generally *cells* (3D-case, Figure 3d). With every projected node k we associate a cell C_n.

Note that a 2D system is sometimes preferred because it is easier to setup in the field and for some applications a 2D perspective is enough. As an example, consider studying



(a) Time policy. Data from sensor 1

(b) Time policy. Data from sensor 2

(c) Distance policy. Data from sensor 1

(d) Distance policy. Data from sensor 2

Fig. 4. PAR data acquired by the first sensor strand during one of the field experiments. Events generated and serviced are shown for Time and Distance policies. Note that events are rendered time of occurrence vs. the PAR value of the event.

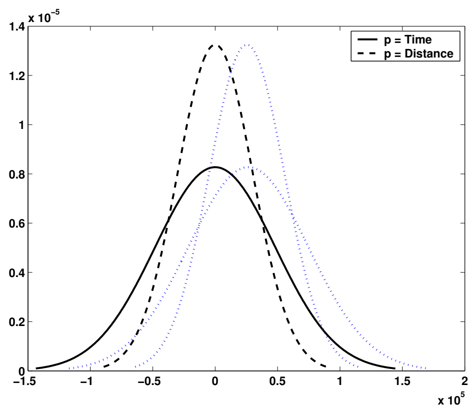


Fig. 5. OnTime in a form of a zero-mean Gaussian distributions for Time and Distance policies. The OnTime of events generated by all sensors is considered. Dotted (blue or lighter) graphs show the distributions at original mean.

sunlight intensity shining through a forest canopy. In this case a sensor network with illumination detectors can be placed on the ground. Suppose node k discovered an interesting reading (say an abnormal light value). The TA system then would guide the robot towards the goal point on l_r computed by $PROJ_{l_r}$. The mobile node then can study appropriate *slice* C_k. The general 3D-case system is investigated in this paper.

B. The Task Allocation System

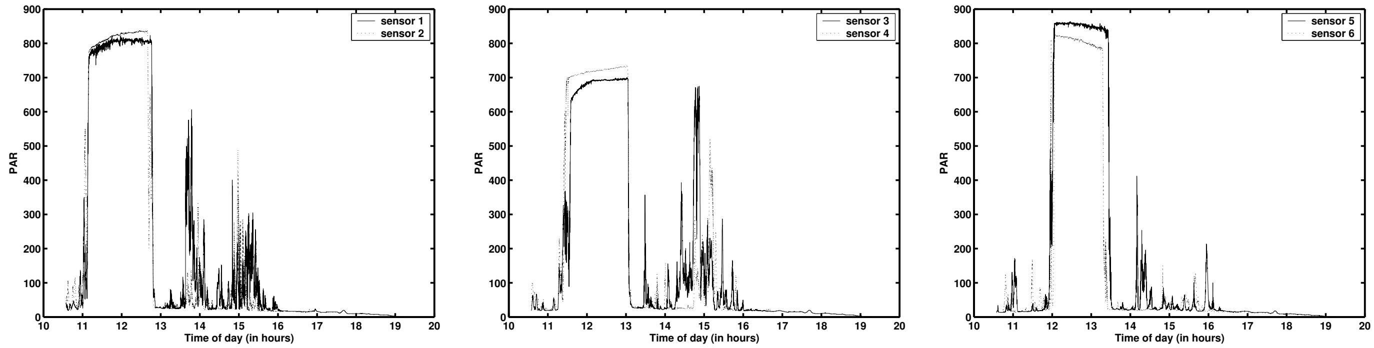
Our system is a special case of the TA methodology described above - with only one resource (mobile node) for task assignment. We consider the problem of assigning tasks one at a time. In this case the greedy assignment is obviously optimal. Consider task assignment Equation 1. Since there is only one mobile node, the next task with highest priority (according to criterion C) is assigned to the mobile node, no matter what the mobile node's utility function might be. In this paper we consider two basic greedy policies, one based on a task's arrival time (we term it the *time* policy) and another based on the distance to the task (we term it the *distance* policy). In our system these policies essentially define the task prioritization criterion C.

The Task Allocation system consists of two algorithms, one running on the static sensor nodes and the other on the mobile node. The algorithm of static nodes is simple - retrieve data from the sensors, process it, and deliver to the mobile node via a wireless link.

The algorithm running on the mobile node is as follows. Suppose the mobile node receives the sensor data from the static node i. This data is analyzed and if there is a difference greater than a threshold in the current sensor data with respect to the previously stored value, a sampling task for the sensor node i is created. The task for the robot is then to travel to the location of the node that generated the task (after that a sampling policy can be applied to the vicinity of the static node, but this is not our focus here). Next, if the task generated by node i is not stored in a set of currently active tasks T_a, it is added to this set. If the mobile robot is available for the next task and $T_a \neq 0$, the next task is extracted from T_a according to the criterion C. We implemented two policies for the criterion C - *time* policy (tasks with smaller time stamp get priority) and *distance* policy (tasks closer to the robot get priority). Note that since the system does not have any prior knowledge about the spatiotemporal variation of event arrival, simple greedy scheduling (*time* and *distance*) is appropriate. In our future work, as we will learn more about the nature of the phenomenon, we plan to incorporate that knowledge into the task allocation process. Next, based on the task information the mobile node needs to compute a goal point. If the task's position is p then the goal position will be $PROJ_{l_r}(p)$ in 2D-case and $PROJ_{\Pi_r}(p)$ in 3D-case (see Figure 3). The robot then moves to the computed location of the task. After the robot completes its last task it becomes available for reassignment.

IV. Experimental Results

We performed a set of experiments using our task allocation system and compared two policies - *Time* and *Distance*. The difference between the two policies is the priority in which the tasks are ordered prior to assignment to robot occurs. First, a set of experiments was conducted on a NIMS



(a) Strand 1 (b) Strand 2 (c) Strand 3

Fig. 6. PAR data acquired by the sensor strands on August 21st 2004, from 10:33 till 20:00.

deployed in The James San Jacinto Mountain Reserve (as shown in Figure 2). Note that because of space limitations, only representative graphs are presented. Figure 4 shows the representative PAR data from sensor 1 and 2 collected during the operation of the *Time* policy (Figure 4ab) and the *Distance* policy (Figure 4cd). Figure 4 also shows points in time when events were generated and serviced by both policies for 2 sensors. Note that events are generated in response to fluctuations in PAR. As shown in Figure 4, events are generated proportionally to the density of the 'spikes' in PAR data and cover all significant 'spikes' of PAR data.

As an evaluation metric of the task allocation policy performance we use the task (event) *OnTime*. A task *OnTime* is the difference between the time the task (event) was generated (or registered) by the system and the time it was serviced by the robot. Figure 5 shows the comparison between the cumulative event *OnTime* of the *Time* policy and the *Distance* policy. For visualization purposes, in Figure 5 event's *OnTime* is presented in a form of a zero-mean Gaussian distribution. It follows that the *Distance* policy has smaller average *OnTime* with smaller deviation.

These experiments show that the presented task allocation system achieves spatiotemporal sampling with both policies. However, in order to compare the performance of the two policies we need to run each on the *same* and longer set of data.

A. TA Policies: Time vs. Distance

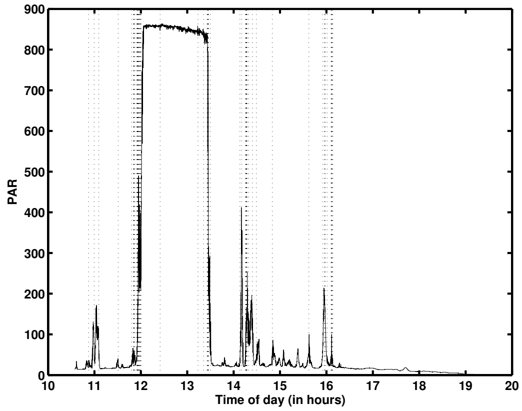
In order to compare the performance and characteristics of the two policies we have recorded a set of sensor strand data (approximately 9 hours and 30 minutes) spanning one day. Then we can *replay* the strand data through the same interface as in the field system in our lab testbed system. The testbed system is computationally identical to the NIMS system (at the James Reserve) and has the same suite of physical devices, such as motors and sensors. Hence, the behavior of the testbed is virtually identical to the NIMS at the James Reserve.

Figure 6 shows prerecorded sensor strand PAR measurements for all three strands (six sensors). We use this data for all of the following experiments. We conducted experiments for two policies (*Time* and *Distance*), for three different thresholds (10, 25, 50). As discussed in Section III-B, a threshold is used by our system to determine when to trigger an event (a task) and is a measure in units of PAR.

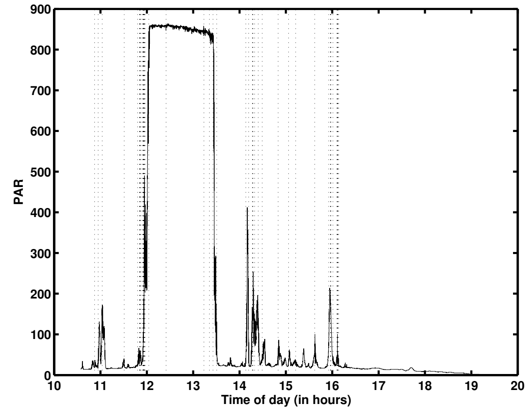
Figure 7 shows data from Sensor 5 (third strand) and generated events by *Distance* policy (Figure 7a) and *Time* policy (Figure 7b). Note that both policies generate events so that the spikes in the PAR data are covered, which in turn means that each of those spikes can be sampled by the system. Figure 7 also shows a good spatiotemporal phenomenon 'coverage' by both policies.

Figure 8 shows a magnified view of part of the data from sensor 3 (second strand). Generated and serviced events are drawn on this figure, for *Distance* policy (Figure 8a) and *Time* policy (Figure 8b). Note that at certain points in time, due to inherent differences between the two policies we consider, some events are generated by one policy and not generated by the other. As a result, the *OnTime* for same generated events is different.

Finally, Figure 9 shows a comparison of the performance of both policies as a measure of cumulative event *OnTime*. Figure 9a shows the change in average *OnTime* for different values of the threshold. It follows that *OnTime* becomes smaller (the system responds to events faster) with bigger threshold values. This result is expected, however - the smaller the threshold, more events are generated and hence the system spends more time to service all events. Figure 9a also shows that there is no significant difference in performance between the two policies. If we consider Figure 9a representing the comparison of average values of the policies, or the means, then Figure 9b shows the comparison in deviation of both policies from the mean. Figure 9b also does not show significant difference between the two policies.

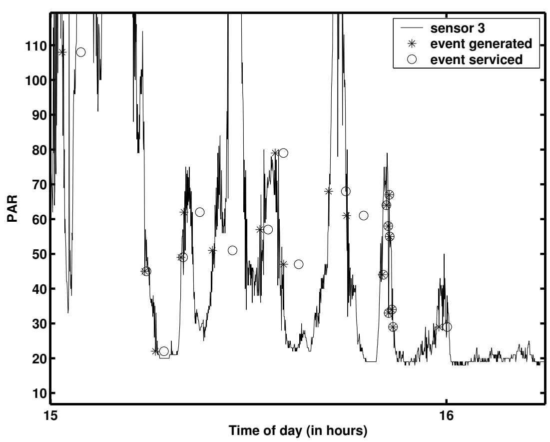


(a) Distance policy. Sensor 5

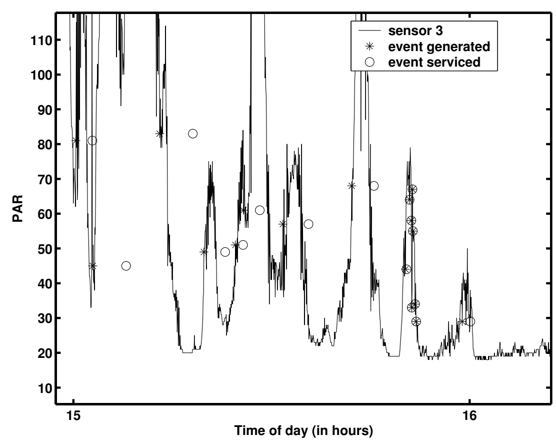


(b) Time policy. Sensor 5

Fig. 7. Estimation of PAR fluctuations with event generation densities by both policies.



(a) Distance policy. Magnified view of data from Sensor 3



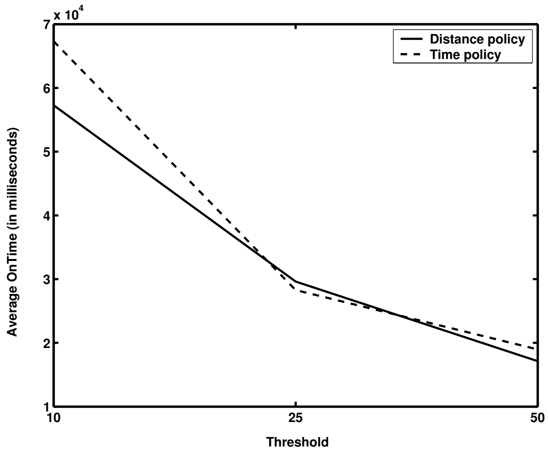
(b) Time policy. Magnified view of data from Sensor 3

Fig. 8. Event generation and servicing by both policies. Note that in some cases events are generated at different times and the OnTime of some events varies depending on a policy.

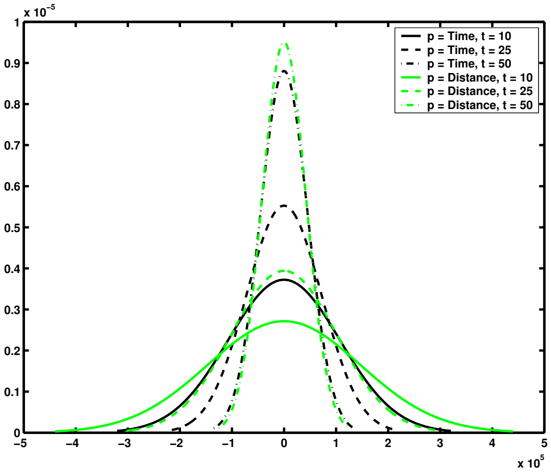
V. Conclusions and Future Work

Virtually all environmental monitoring applications require a high fidelity characterization capability for environmental variables. This implies a high spatiotemporal sampling rate. Networked Infomechanical Systems (NIMS) combining both fixed and mobile nodes was introduced for addressing this problem. In this paper we presented the Task Allocation component of NIMS. Specifically, we introduced a system in which static nodes act as triggers for the sensor sampling events to occur by the NIMS node. We described the two basic task allocation policies that we have used - the *time* policy (tasks with more recent time stamps get priority) and the *distance* policy (tasks closer to the robot get priority).

We performed extensive experiments of the two policies on the NIMS deployed at James Reserve and on our lab testbed. The main conclusion that we can draw is that it appears that both policies are adequate for the problem of spatiotemporal sampling. However, from the presented experimental results follows that the *time* policy performs better than the *distance* policy in some cases and vice versa. In general, however, the performance of both policies based on greedy algorithm is similar. In our future work, we plan to analyze the data gathered by the proposed task allocation system and construct a model of the studied phenomenon. Next, we plan to improve the performance of the TA system with domain knowledge obtained from the model, which, in turn, would provide better data for the model improvement. We also plan to investigate the performance of our TA system from the perspective of



(a) Comparison of average OnTime



(b) Comparison of the OnTime standard deviation

Fig. 9. Comparison of an average OnTime and OnTime in a form of a zero-mean Gaussian distributions for Time and Distance policies (p). Three different thresholds (t) used. The OnTime of events generated by all sensors is considered.

information gain. That is, given a particular set of events generated and serviced what is the information gain of the system for two policies?

References

[1] D. Estrin, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *ICASSP 2001*, Salt Lake City, USA, 2001.

[2] C. M. P. Ozanne, D. Anhuf, S. L. Boulter, M. Keller, R. L. Kitching, C. Korner, F. C. Meinzer, and A. W. Mitchell, "Biodiversity meets the atmosphere: A global view of forest canopies," *Science*, vol. 301, pp. 183–186, 2003.

[3] R. W. Pearcy, *Radiation and Light Measurements*, ser. Physiological Ecology - Field Methods and Instrumentation. Chapman & Hall, London, U.K, ch. 6, pp. 97–116.

[4] NIMS, "http://www.cens.ucla.edu/portal/nims.html."

[5] J. S. J. M. Reserve, "http://www.jamesreserve.edu."

[6] M. Batalin, M. Rahimi, Y. Yu, S. Liu, A. Kansal, G. Sukhatme, W. Kaiser, M. Hansen, G. Pottie, M. Srivastava, and D. Estrin, "Call and response: Experiments in sampling the environment," in *Proc. ACM SenSys*, 2004.

[7] B. Gerkey, "On multi-robot task allocation," University of Southern California, Tech. Rep. Technical Report CRES-03-012, 2003.

[8] D. Chapman, "Planning for conjunctive goals," *Artificial Intelligence*, vol. 32, pp. 333–377, 1987.

[9] M. Batalin and G. Sukhatme, "Using a sensor network for distributed multi-robot task allocation," in *Proc. IEEE International Conference on Robotics and Automation*, New Orleans, Louisiana, April 2004, pp. 158–164.

[10] B. Gerkey and M. J. Mataric, "Multi-robot task allocation: Analyzing the complexity and optimality of key architectures," in *To appear in Proceedings of the IEEE International Conference on Robotics an Automation (ICRA03)*, Taipei, Taiwan, 2003.

[11] L. E. Parker, "Alliance: An architecture for fault-tolerant multi-robot cooperation." in *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, 1998, p. 220240.

[12] S. Botelho and R. Alami, "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement." in *Proc. of IEEE International Conferenceon Robotics and Automation (ICRA)*, 2000, p. 293298.

[13] B. P. Gerkey and M. J. Mataric, "Sold!: Auction methods for multi-robot coordination." in *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, 2002, p. 758768.

[14] E. H. Ostergard, M. J. Mataric, and G. S. Sukhatme, "Distributed multi-robot task allocation for emergency handling." in *In Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2001, pp. 821–826.