Simplifying solar harvesting model-development in situated agents using pre-deployment learning and information sharing

Huzaifa Zafar
Department of Computer Science
University of Massachusetts, Amherst
hzafar@cs.umass.edu

Daniel Corkill
Department of Computer Science
University of Massachusetts, Amherst
corkill@cs.umass.edu

ABSTRACT

Agents deployed in real-world applications typically need to determine various aspects of their local environments in order to make appropriate decisions. This must be done quickly, as performance can suffer until each agent develops a model of its environment. We use a strategy of factoring this model development (that would typically be done using multi-agent learning) into two phases: pre-deployment (site independent, individual agent) learning and post-deployment (site dependent, multi-agent) model completion. By performing as much learning as possible prior to deployment, we simplify what needs to be determined on-site by each agent. Furthermore, we use collective sharing of local-observation information, in conjunction with temporal and spatial constraints in relating information, to reduce the number of observations needed to perform each agent's model-completion activities. In this paper, we apply this two-phase strategy in developing prediction models for solar energy to be harvested by each agent in a power-aware wireless sensor network. In all but the most unlikely of environmental conditions, this strategy allows individual-agent harvesting models to be completed using only the first and second day's observations.

1. ENVIRONMENTAL MODEL DEVELOPMENT

Multi-agent system (MAS) applications, such as wireless sensor networks (WSNs), suffer from performance degradation while agents develop models of their local environments. In developing these models, agents can either use observation information from other agents in the network (typically done using multi-agent learning (MAL)) or learn the models individually (using traditional machine learning (ML) techniques). Agents using either MAL or ML require many observations before they are able to develop a reasonable local model.

In order to develop these models faster, we introduce a strategy that factors model development into two phases; a pre-deployment learning phase, and a post-deployment model completion phase. The pre-deployment phase uses traditional ML techniques to develop a partial, parametrized, model of the environment that captures the agent-specific, site-independent aspects of the complete model. The site-specific factors are then maintained as parameters in this model. By shifting all the site-independent learning to pre-deployment, an agent can dramatically reduce the information and time required to complete the model once situated.

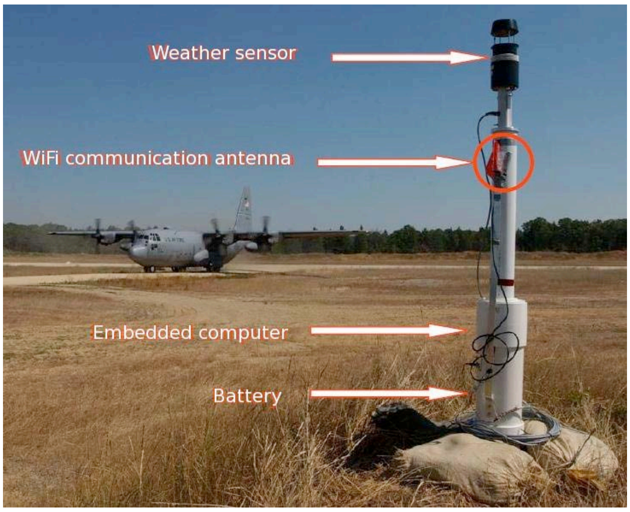

Figure 1: A TACMET-augmented CNAS sensor agent

Post-deployment model completion, can be aided further by sharing information with other agents in the network. By using constraints defined on the parameters, an agent can use the shared information to constrain the site specific parameters in its local model. In fact, we show that in all but the most unlikely of environmental conditions an agent is able to develop an accurate solar harvesting model after a single day of observations; but only if they use shared information from other agents in the network.

Similar separation strategies have been used in developing parametrized models with pre-deployment functional dependencies and then using those dependencies along with information sharing to improve the model post-deployment. Determining costs in network level routing algorithms is one such example [5, 7, 12]. However, the two-phase strategy presented in this paper is novel in shifting a majority (if not all) learning to the pre-deployment phase, and using post-deployment information sharing (along with con-

This work is supported by the AFRL "Advanced Computing Architecture" program, under contract FA8750-05-1-0039. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views contained in this paper are the authors.'

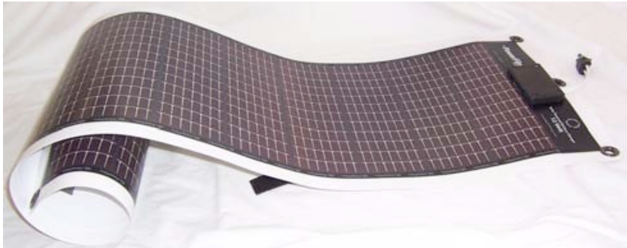

Figure 2: The Rollable Solar Panel that we are using

straints defined on parameter values) to complete a uniquely parametrized model with minimal local observations.

2. SOLAR HARVESTING MODELS

In this paper we apply our two phase strategy to the development of solar-harvesting prediction models in the Collaborative Network for Atmospheric Sensing (CNAS) sensor network [3]. This model is used to predict the solar energy to be harvested by an agent given a cloudiness forecast for each time period in the following solar day. A TACMET-augmented CNAS sensor agent is shown in Figure 1 includes capabilities for weather sensing and WiFi based communication. Each agent is powered by a un-managed 12V battery (labeled in the figure). In order to extend the lifetime of the network, a rollable (thin film on plastic) solar panel (see Figure 2) is added to each agent. This allows the battery reserves to grow depending on the attenuation due to clouds, shade, and panel angle experienced by the agent during the course of the day. Moreover a requirement of CNAS is that the positioning of sensor agents cannot be optimized for either data collection or solar harvesting. Eventually sensor agents may be airdropped (rather than hastily deployed by hand). In either case, the exact placement of sensor cannot be regulated, which requires the shade and tilt attenuation to be determined once the agents are situated. Therefore, each agent in the network needs to develop the following environmental model:

$$E(t, l, \alpha) = E_{max}(t) * (1 - (f(C(t)) + g(S(t, l)) + h(T(t, \alpha))))$$

Where:

$t =$ *time of day*

$l =$ *geographic location of the agent (and its solar panel)*

$\alpha =$ *angle of the solar panel*

$C(t) =$ *function to determine % cloud cover given time*

$S(t, l) =$ *function to determine % shade cover given time and location*

$T(t, \alpha) =$ *function to determine % angle of solar tilt with respect to panel angle given time*

Figure 3 shows an example set of curves that illustrate the environmental model. In developing the solar harvesting model, we use an hour long time period (using the averaged solar energy observed during the hour as our "observation"). However, any other time period can also be used.

The observation by an agent ($E(t, l, \alpha)$) depends on the time period of the observation, the location of the agent, and the angle of its solar panel relative to the sun. This can

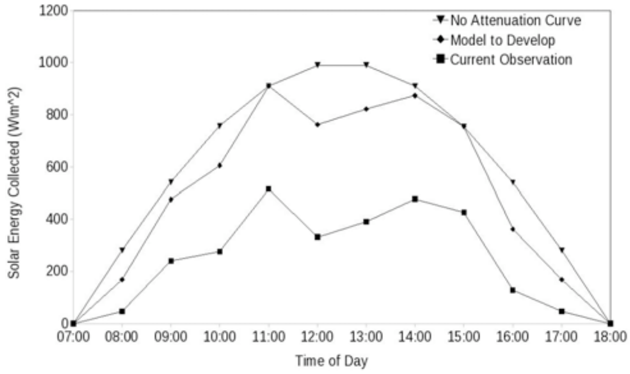

Figure 3: Example cumulative solar energy observed, along with a curve representing the local harvesting model of an agent

be given by the unattenuated energy ($E_{max}(t)$) at time t for this agent's solar panel, reduced by the attenuation from clouds ($C(t)$), shade ($S(t, l)$) and the tilt ($T(t, \alpha)$)[1]. The model also accommodates a non-linear effect of attenuation on the maximum energy observed. These relationships are given by functions f, g and h in the model.

We assume l and α to be constant for the lifetime of an agent (CNAS sensor agents are not mobile), and can be combined into a single parameter e that denotes the environment of the agent. The shade and tilt functions can be modified to $S(t, e)$ and $T(t, e)$. This makes both shade and tilt attenuation depend on the same parameters, allowing us to combine shading and tilting into a single attenuation called site attenuation ($S(t, e)$). Moreover, we can combine functions g and h, into a single function we call k, applied to $S(t, e)$.

3. PRE-DEPLOYMENT LEARNING

We start with the following environmental model (from the previous section) to be developed by each agent:

$$E(t, e) = E_{max}(t) * (1 - (f(C(t)) + k(S(t, e)))) \quad (1)$$

Once an agent develops a complete model, $C(t)$ will be provided for, and the agent is expected to use the rest of the parameters from equation 1 in predicting an observation. Therefore the post-learning model is as follows:

$$E(t, e, C) = E_{max}(t) * (1 - (f(C) + k(S(t, e)))) \quad (2)$$

In moving from equation 1 to equation 2, the agent needs to retain functions E_{max}, f, k and $S(t)$.

The function E_{max} is dependent on the time of observation but not the site of the agent. Therefore, we can learn $E_{max}(t)$ pre-deployment. An example curve for $E_{max}(t)$ learned by an agent (pre-deployment) is shown in Figure 3 as "No attenuation curve (E(t))".

[1] The accurate solar-harvesting model is more complicated than this, since the shade function will be applied to the residual energy after clouds have taken their effect. Similarly tilt applies after clouds and shade have taken their effect. However, we use the simple linear version to ease illustration. The reasoning remains the same for the more complicated model.

Similarly, $C(t)$ cannot be determined pre-deployment, since it depends on the cloud cover experienced by the agent at time t. However, again the function f can be learned pre-deployment in the same way as E_{max}. An example curve is given in Figure 4.

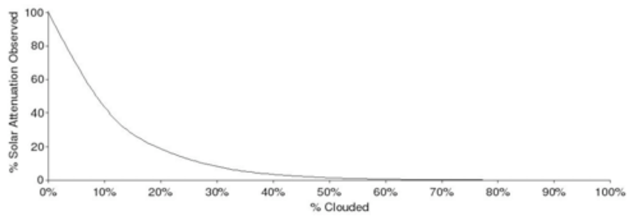

Figure 4: Relationship between cloud levels and solar energy harvested at noon, given no site attenuation

Finally, we can learn k by synthesizing various tilt and shade environments pre-deployment, leaving only the $S(t, e)$ function to determine on-site.

Constraints

We make use of the following constraints on variables in our model;

1. For any given time in a day, the shading and tilting attenuation for an agent is constant across multiple days[2].
2. We assume the cloud cover is spatially consistent enough, that every agent experiences the same degree of cloud attenuation for any energy measurement period.
3. We assume that the effect of shading and cloud attenuation is such that unless an agent experiences 100% cloud and site attenuation, it observes some solar energy value (possibly very small) that can be used to determine the degree of shading and clouding. Such non-zero observations are termed *meaningful*

Defining Convergence

An agent's solar harvesting model is said to converge when, the agent can determine the value returned by the function $Site(t, e)$, given the constant e of the agent.

We define two levels of convergence. The first level is *hour*-convergence, where an agent converges (or is able to determine the value of S) for a given hour. The second level is *day*-convergence, where the agent has obtained hour-convergence for every hour of the solar day.

4. SITUATED MODEL COMPLETION WITHOUT INFORMATION SHARING

We illustrate our strategy for completing the parametrized solar harvesting model (learned by each agent during the first phase) by looking at various examples, increasing in complexity. In the following three scenarios, we look at what a non-sharing agent can determine using only its local observations.

Scenario A

Description: No information sharing. No site or cloud attenuation.

Every observation the agent makes is at 100% of the expected maximum energy that can be collected for the hour. The agent can easily determine that the environmental attenuation is at 0% (that $E(t, e) = E_{max}(t)$). Once the agent determines its environmental attenuation, it achieves hour-convergence for that hour. Once the agent achieves hour-convergence for each hour in the solar day, it achieves day-convergence.

Scenario B

Description: No information sharing. No site attenuation. Varying cloud attenuation.

Since the observations the agent makes is less than its $E_{max}(t)$ the agent is unable to determine its environmental attenuation for the hour. For example, if an agent observes 50% energy loss, it can draw one of the three possible conclusions; 1) it lost 50% of its energy due to cloud cover, 2) it lost 50% of its energy due to site attenuation and 3) it lost 50% of its energy due to some combination of cloud and site attenuation. Without additional information, the agent must assume anywhere between 0% to 50% of the energy is lost due to site attenuation. Since site and cloud attenuation are not necessarily constant across hours of the day, it has to make similar observations for all hours of the day. On subsequent days, if the agent makes observations greater than 50%, it can reduce the range of expected loss due to site attenuation. However, the only way an agent will achieve hour-convergence is if it makes a 100% observation (meaning no site attenuation). Moreover, the agent will have to make such an observation for every hour of the day in order to achieve day-convergence.

In a later section, we will prove that an agent cannot determine the apportioning of site and cloud attenuation based on its own observations and requires observations from multiple agents within the network.[3]

Scenario C

Description: No information sharing. Varying site and cloud attenuation.

We assume, site attenuation is the same at the same time each day. However, site attenuation does vary across hours. This prevents an agent from achieving day-convergence by converging for a single hour. The scenario is similar to scenario B, and we can draw some of the same conclusions from it. If an agent makes a 50% observation, it will conclude site shading is responsible for 0% to 50% of the solar attenuation for the hour. However, since the there exist some site attenuation, an agent will never make a 100% observation, and hence can never hour-converge on its own.[4]

5. SITUATED MODEL COMPLETION WITH INFORMATION SHARING

In the remaining 4 scenarios, we look at how information sharing improves convergence.

[2]We assume that the solar angle changes so little from day to day that it can be considered fixed. The model-completion phase of our strategy is fast enough that it can be repeated every few weeks to adjust for seasonal sun-angle change.

[3]This apportioning could be learned over many days given an expected cloudiness distribution. However, the number of days this would require precludes this approach.

[4]without some form of learning (see previous footnote)

Scenario D

Description: Observations from at least one hour-converged agent are received every hour. No site attenuation. Varying cloud attenuation.

We assume the hour-converged agent in this scenario can provide a meaningful observation. In the rest of the paper, as we talk about a day-converged (or hour-converged) agent sharing meaningful observations with other agents, we assume they can provide the degree of cloud attenuation experienced by themselves or other agents in the network.

Assume agent A makes a 50% observation, and goes through the same decision process as in Scenario B. Agent B also makes a 50% observation (since we assume identical cloud attenuation across agents). Since agent B is hour-converged, it can share with agent A, exactly how clouded it is for this hour. This allows agent A to determine conclusion 1 from scenario B and hence hour-converge, too.

If a single agent exists that has already hour-converged, all other agents in the network will hour-converge as soon they obtain a report from the converged agent. This is because the hour-converged agent is able to determine the degree of cloudiness experienced during the hour. As it shares this information with other agents in the network, agents are able to separate individual site and cloud attenuation, and are able to determine site attenuation for that hour immediately, leading to their own hour-convergence.

Scenario E

Description: Information sharing among agents. No site attenuation. Varying cloud attenuation. No hour-converged agents.

Since cloud attenuation is constant at the same time across agents, all agents will make exactly the same observation regarding the effects of attenuation on their observations. Here sharing does not help, as neighboring agents are unable to provide any additional information, as they cannot distinguish cloud and site attenuation. Similar to previous no-information-sharing scenarios, agents will have to experience a 100% observation to be able to hour-converge.

As shown in this scenario, if all agents make the exact same observation, no additional information is gained from sharing. Similarly, an agent can not learn anything new from hourly observations on corresponding days unless the observation has a higher energy value than a previous day's observation at the same hour. The analysis of this scenario highlights the convergence benefit of having site-attenuation variance among agents.

Scenario F

Description: Observations from at least one hour-converged agent are received every hour. Varying site and cloud attenuation.

This scenario is similar to scenario D. Since we have a hour-converged agent in our network, that agent is able to separate the site and cloud attenuation in its current observation. This allows it to determine the exact level of cloud attenuation and share this information with un-converged agents. Each un-converged agent uses the information to separate the two attenuation components from its own observation and hour-converge.

Scenario G

Description: Multiple information sharing agents. Varying cloud and site attenuation.

If all agents are site attenuated in exactly the same way (highly unlikely), this scenario becomes equivalent to Scenario E (as all agents make the same observation at each hour). Otherwise, there is at least one agent that makes a different observation, and all agents can gain information from this difference. For example, agent A observes 70% of its maximum energy. For the same hour, agent B observes 80%. Agent A can then conclude that too should lose at most 80% of its energy due to cloud attenuation. This implies the remaining loss must be due to site attenuation, reducing the site attenuation range from 0%-30% to 10%-30%. Sharing therefore allows us to tighten the lower bound on the range, while the observations allow us to tighten the upper bound. Also, if agent C makes a 100% observation, agent A can immediately hour-converge by saying its losing 30% of its energy due to site attenuation. Finally if agent D makes a 50% observation, agent A can gain no additional information from agent D since all 50% could be lost due to site attenuation. This shows us sharing is limited in the same way as hourly observations when assisting agents in developing a bound on the site attenuation.

A second benefit of sharing is hour-convergence without having to see a 100% observation. In fact, with sharing an agent can hour-converge, for any given hour, with 2 different observations for that hour, and 2 correspondingly different observations for the same hour by a different agent. The set of equations solved by the two agents over the two days is as shown below. For ease of expression, we assume $f(Cloud)$ and $k(S)$ are linear.

Agent A:

Day 1:

$$1 - (Cloud(t_1) + Site(t_1, e_1)) = O(t_1, e_1) \tag{3}$$

Day 2:

$$1 - (Cloud(t_2) + Site(t_2, e_1)) = O(t_2, e_1) \tag{4}$$

Where $O(t, e) = \frac{E(t,e)}{E_{max}(t)}$. Also, constraints define $t_1 = t_2$ for the *Site* function, since the the hour of the day is the same for both days.

Agent B:

Day 1:

$$1 - (Cloud(t_1) + Site(t_1, e_2)) = O(t_1, e_2) \tag{5}$$

Day 2:

$$1 - (Cloud(t_2) + Site(t_2, e_2)) = O(t_2, e_2) \tag{6}$$

As long as $O(t_1, e_1) \neq O(t_2, e_1) \neq O(t_1, e_2) \neq O(t_2, e_2)$, the above equations are solvable, since we have 4 variables, and 4 equations. Therefore, except under very unlikely environmental conditions, agents are able to hour-converge for each hour throughout day 2. We have already shown that all other agents hour-converge if 1 agent hour-converges. Note domain constraints play a big part in arriving at this conclusion. Each agent can make use of the constraint that its site attenuation is the same at the same hour each day and that all agents have the same cloud attenuation at the same time. This allows any single agent to solve the 4 equations from observation reports received from other agents in the network.

Note that, a single agent cannot hour-converge even if it makes 4 observations over a period of 4 days. This is because each additional observation adds one additional variable as illustrated below:

Day 1:

$$1 - (Cloud(t_1) + Site(t_1, e_1)) = O(t_1, e_1) \quad (7)$$

Day 2:

$$1 - (Cloud(t_2) + Site(t_2, e_1)) = O(t_2, e_1) \quad (8)$$

Day 3:

$$1 - (Cloud(t_3) + Site(t_3, e_1)) = O(t_3, e_1) \quad (9)$$

Day 4:

$$1 - (Cloud(t_4) + Site(t_4, e_1)) = O(t_4, e_1) \quad (10)$$

Here we have 4 equations and 5 variables, which is unsolvable. Similarly, we cannot take 4 differing observations from 4 separate agents for the same hour as we would again end up with 5 variables.

6. EXPERIMENTAL RESULTS

In the last few sections, we analyzed the benefit of our two phased model-development strategy using a number of scenario cases. We call this strategy PLASMA (Pre-deployment Learning And Situated Model-development in Agents). In this section, we evaluate PLASMA operationally in a simulated solar harvesting environment.

All experiments were performed using a simulator developed in GBBopen. For our first experiment, we use a network of 6 agents. The simulation spans 3 days. Cloud attenuation decreases from 50% on day 1 to 30% on day 3, with 10% decrements at the end of the day. Site attenuation varies from 30% for agent 1 to 80% for agent 6 with 10% increments with each agent. We show a similar curve as our previous experiment for agent 2 that experiences 40% site attenuation in Figure 5. In the figure we show 4 curves.

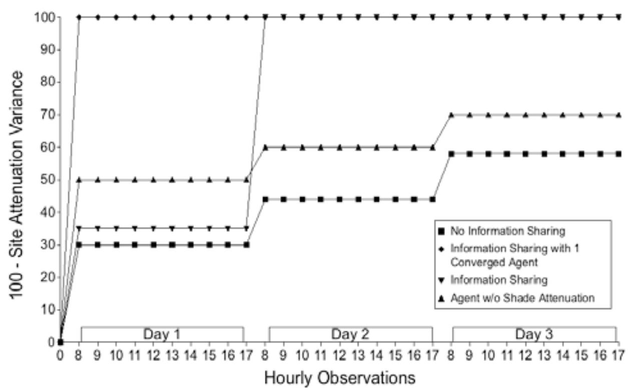

Figure 5: As the number of hourly observations increase, the range of the site attenuation determined by various scenarios in the model development strategy is explored

The first curve is a "No information sharing curve", wherein agents in our network do not share any information with one another. On day 1, An agent experiences 50% cloud attenuation and 40% site attenuation, for a combined 70% attenuation. Since it is not sharing any information, it determines its site attenuation to be anywhere in the 0%-70% range, the size of which is 70%. Similar calculation is done for days 2 and 3. The second curve is "Information Sharing with 1 converged agent", wherein the agent shares information with another agent in the network that has already day-converged. The day-converged agent does not have 100% site attenuation and is able to predict the level of cloud attenuation for the hour, allowing the agent depicted in the graph to day-converge. The third curve is the "Information sharing" curve. With information sharing our agent is able to perform better on day 1, since it has another agent in its network which sees lower attenuation than itself, and is able to lower the range. However, on day 2, the agent is able to form 4 equations using observations from itself and its neighbor agents and day-converge. Finally for the fourth curve, "Agent w/o Site Attenuation", we remove site attenuation from the single agent. Also, no information sharing is allowed. This way, as the agent sees improving observations from one day to the next, its able to improve its site attenuation range, but can do no better without seeing 100% sun.

For our second experiment, we show the $E(t, l)$ function developed at the end of day 2, given 0 cloud attenuation. Here we have an network with 2 new agents. We showcase the curve learned by one of the two agents. The agent experiences clouds for the first 6 observations (first 6 hours of the day) and site attenuation for the remaining 4 hours. By using the 4 equations developed above, the agent is able to figure out the degree of site attenuation experienced, and project the observations from the clouded periods to the expected maximum values. Figure 6 shows the corresponding model learned. In the figure, we showcase the benefit of both

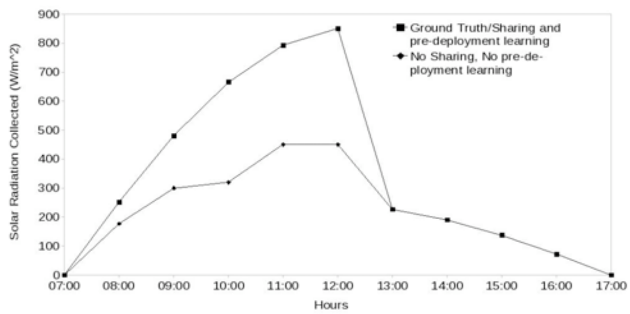

Figure 6: Model developed during the simulated run

sharing and pre-deployment learning. The ground truth is labeled as "Accurate Model", and represents what the agent should see given no cloud attenuation. Given observations with 100% sun for the first 5 hours, the agent is able to both learn the approximate attenuation due to clouds, and is able to use pre-deployed learning to project the current observation to the expected no cloud attenuation observation.

We are thus able to show, in our simulated environment, the benefit of both pre-deployment learning and collective information sharing demonstrated mathematically in previous sections.

7. POWER MANAGEMENT EXPERIMENTS

In this section we look at load assignment based on energy harvested by a group of agents in a wireless sensor network.

The problem was looked at previously by Kansal et. al [10]. We extend our simulation environment as follows; each agent can undertake a certain task load. To accomplish this load, the agent requires a certain amount of energy. As the load increases, energy requirements also increase linearly. The agent earns a certain utility if a task is completed. Utility-load relation is the same as that used by Kansal et. al and is shown in Figure 7.

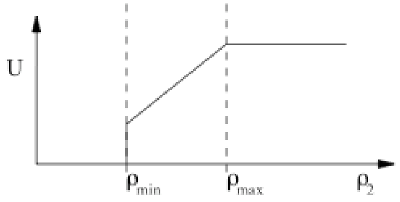

Figure 7: ρ is the size of the task. Really small tasks have no utility. As the size of the task gets bigger than ρ_{min}, the agent starts increasingly gaining utility, which maxes out at ρ_{max}

We implemented the duty cycle adaptation algorithm defined in Kansal et. al [10]. In order to be able to use their adaptation algorithm with the model developed by PLASMA, we modified the adaptation algorithm to make decisions solely on energy levels predicted for a given time period. Such a reactive algorithm could also be extended to take into account predictions of other agents within the system, which allows for better load balancing across multiple agents. However, this extension was not implemented so as to be able to compare results directly with Kansal et. al. We also implemented their Energy Prediction and Optimization model and compared it (using the original adaptation algorithm) with the local model developed by each situated agent in PLASMA (using the modified adaptation algorithm). For the experiments, we use a network of 10 agents. For each agent, we picked a random period of time during the day (of random size) to be shaded. Note, once the shading attenuations are randomly assigned to each agent, they remain fixed for all the remaining simulations in the section and are shown in Table 1. The simulation now spans

Agent	Time period of shading	% Shade Attenuation
1	9:30 - 12:00	27%
2	15:30 - 16:30	80%
3	15:00 - 18:00	14%
4	10:00 - 11:30	32%
5	13:00 - 18:00	15%
6	13:15 - 14:00	66%
7	11:45 - 13:45	51%
8	10:30 - 13:00	42%
9	14:15 - 16:30	21%
10	9:00 - 10:00	37%

Table 1: Shade attenuation for agents in the simulation

20 days, with random cloud attenuation for every time period. A random cloud-attenuation pattern was established for the 20 simulated days and used in all simulation experiments. Mean cloud attenuation is around 25%, with about 20% variance. Each agent is asked to predict the expected solar energy for every time period, given expected cloud attenuation. The variance from the observation is calculated for each time period, for each agent, and averaged for each of the 20 days. Figure 8 compares the average variance calculated. As we can see from the figure, PLASMA is able to

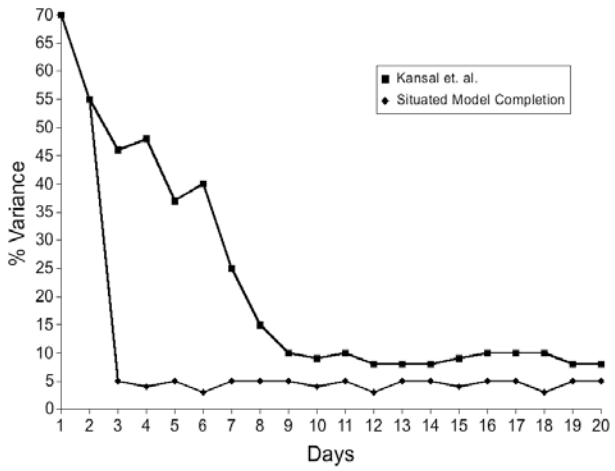

Figure 8: Comparing our strategy with the learning algorithm described by Kansal et. al.

develop a more accurate model much faster than the Energy Prediction model defined by Kansal et. al. (the small variance is primarily due to the error in the cloud attenuation prediction provided to the agents, where mean error is about 5% with a variance of about 2%, modeled into the simulator. This is to model the real world, where its impossible to accurately predict cloud attenuation for any given time period). Also, since Kansal et. al. combine their learning algorithm with their prediction model (which does not take into account shade attenuation), their algorithm is unable to converge to PLASMA.

We compared utilities generated by the two algorithms in our 10 agent simulation. For the first experiment, each agent is provided with no accompanying battery, which forces each agent to make decisions based completely on the expected solar energy for each time period. We set ρ_{min} to 2 and ρ_{max} to 5. Also, agents can earn a utility between 0 and 5 every hour, depending on the amount of solar energy it can harvest for that hour and the task load acquired for the hour. Agents are penalized 1 utility point per tasks they cannot complete. Figure 9 shows the results of the experiment. As we can see from the figure, PLASMA is fully informed starting on day 3, except for days when the energy harvested is less than the energy required to perform all the tasks. However, since Kansal's algorithm depends on some battery for their reactive algorithm, we provide each agent with a battery of infinite size to see how our strategy compares. Figure 10 shows the results. Here we see the Energy Prediction Model is able to do as well as our strategy once it converges to the correct learned values. The variance in learned values depicted in Figure 8 is covered by the infinite battery capacity, which allows Kansal et. al. to use all the energy harvested during the day. However, a lot of utility is wasted while the agent is trying to learn

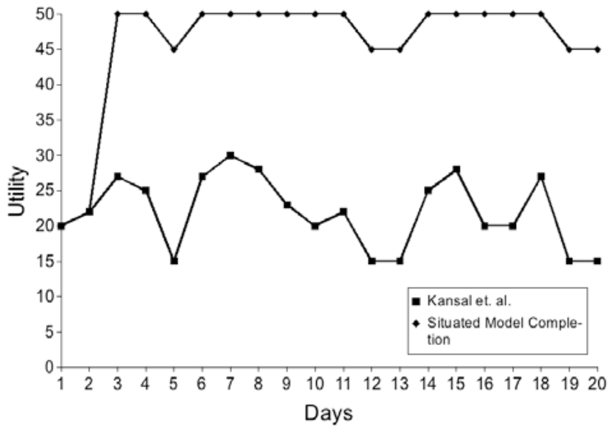

Figure 9: Utility is calculated by averaging the total utility earned by each of the 10 agents per day. This experiment assumes each agent has no battery attached.

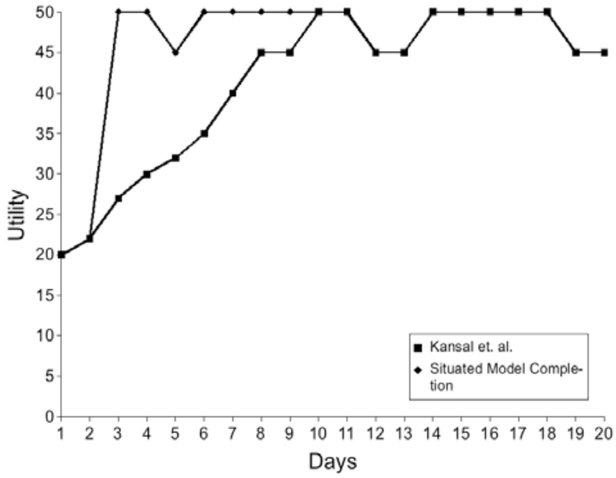

Figure 10: Utility is calculated by averaging the total utility earned by each of the 10 agents per day. This experiment assumes each agent has a battery of infinite size.

during the first 10 days. Also, beyond the first 10 days, how well the Kansal agent actually performs given a limited battery capacity, will vary somewhere between performance given no spare battery (Figure 9) and performance given an infinite battery (Figure 10), depending on how much of the difference between the prediction and actual solar energy harvested is stored by the battery.

From the above three simulations, we can see the benefit of PLASMA over a similar strategy both in the speed of convergence and lower battery buffer requirements.

8. APPLICATION TO OTHER DOMAINS

In order to be able to apply PLASMA to a model development problem, the domain must satisfy the following conditions:

1. **Pre-deployment and Post-deployment Phase**. A partial local model that captures the agent-specific, site independent model is developed pre-deployment. Factors that are site-dependent remain as parameters within the parametrized model. Parts of the parametrized model that are dependent on the environment of the agent can be developed post-deployment. In the solar harvesting domain, the maximum energy curve of the solar panel, and dependencies between the maximum energy and cloud and site attenuation were developed pre-deployment. Individual site attenuation experienced was determined post-deployment to complete the model.

2. **Cooperative Information Sharing**. Agents exchange useful site-specific information to collectively improve local model completion. In the solar harvesting domain, agents share observations as well as cloud cover estimations from developed models in order to help other agents develop their own local models.

3. **Constraints**. Agents require constrains to relate observations as well as observations made over time as well as observations made at the same time by other agents. In the solar harvesting domain, the parametrized model had two environmental factors, cloud and site attenuation. In order to converge, we require two constraints and 4 observations that relate those factors using both constraints. Furthermore, if we increase the number of factors to three, we require 3 constraints and 9 observations to converge.

9. RELATED WORK

Multi-agent learning (MAL) is traditionally subdivided into multi-agent reinforcement learning (MARL) and multi-agent inductive learning (MAIL). MARL extends traditional reinforcement learning to multi-agent systems [2, 6, 13]. The disadvantage of MARL is the large observation set required for convergence using reinforcement learning techniques. MAIL involves learning models by interacting with other agents in the network. Work in cooperative MAIL [1, 4, 9] involves determining what information needs to be shared between agents in the network, and how to learn the entire model post-deployment. The disadvantages are similar to MARL in that many observations are required to develop a usable on-site model.

We applied our two-phase multi-agent model development strategy to solar harvesting models in WSNs. Sensor networks now incorporate agents that can harvest energy from their environment [8, 10, 11]. Energy usage protocols have been developed that depend on energy harvested by individual agents in the network, including those for routing [16], duty cycling [15], and sleep wake cycling [14] among various others. In each of these protocols, there is an added benefit in having each agent estimate the amount of energy it can harvest at any particular time based on cloud forecast and sharing this information with other agents in the network.

10. CONCLUSIONS

We presented a two-phase development strategy called PLASMA (Pre-deployment Learning And Situated Model-development in Agents) for MAS environmental models that separates into a pre-deployment (agent-specific, site-independent) and post-deployment (multi-agent, site-dependent) phase.

By developing a majority of the model pre-deployment, we are able to simplify what needs to be accomplished once the agent is situated. Furthermore, we reduce the number of observations required by an agent post-deployment, by sharing information among agents. Finally, in order to relate observations from other agents to the local model, we take advantage of domain-specific constraints between the various environmental parameters in our model. The number of constraints and number of cooperative agents required to converge is related to the number of environmental parameters. Thus by developing a majority of the model pre-deployment we can converge in a very short time.

In building its solar harvesting model, each agent is able to obtain day-convergence given observations from day one and two (except under extremely unlikely environmental conditions) when sharing information among agents. hour-convergence requires: 1) another hour-converged agent, that is able to provide a meaningful report, or 2) four observations, over two days with different cloud attenuation across days, and different shade attenuation across agents.

We also showed: 1) if a single agent within the multi-agent system day-converges, all other agents will also day-converge as soon as the day-converged agent shares its information with the rest of the network and the other agents have made observations throughout the day. 2) a non-interacting agent requires a single observation with no site or cloud attenuation to be able to hour-converge.

11. REFERENCES

[1] S. Benson. Inductive learning of reactive action models. In *International Conference on Machine Learning*, pages 47–54, 1995.

[2] P. Brazdil, M. Gams, S. Sian, L. Torgo, and W. van de Velde. Learning in distributed systems and multi-agent environments. In *EWSL-91: Proceedings of the European working session on learning on Machine learning*, pages 412–423, New York, NY, USA, 1991. Springer-Verlag New York, Inc.

[3] D. Corkill, D. Holzhauer, and W. Koziarz. Turn Off Your Radios! Environmental Monitoring Using Power-Constrained Sensor Agents. *First International Workshop on Agent Technology for Sensor Networks (ATSN-07)*, 2007.

[4] W. Davies and P. Edwards. The communication of inductive inferences. In G. Weiß, editor, *ECAI '96: Selected papers from the Workshop on Distributed Artificial Intelligence Meets Machine Learning, Learning in Multi-Agent Environments*, pages 223–241. Springer-Verlag, 1997.

[5] Y. T. Hou, Y. Shi, and H. D. Sherali. Rate allocation in wireless sensor networks with network lifetime requirement. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 67–77, New York, NY, USA, 2004. ACM.

[6] J. Hu and M. P. Wellman. Multiagent reinforcement learning: theoretical framework and an algorithm. In *Proc. 15th International Conf. on Machine Learning*, pages 242–250. Morgan Kaufmann, San Francisco, CA, 1998.

[7] Z. Hu and B. Li. On the fundamental capacity and lifetime limits of energy-constrained wireless sensor networks. *Real-Time and Embedded Technology and Applications Symposium. Proceedings. RTAS 2004. 10th IEEE*, pages 2–9, 25-28 May 2004.

[8] X. Jiang, J. Polastre, and D. Culler. Perpetual environmentally powered sensor networks. In *IEEE Information Processing in Sensor Networks*, pages 463–468, 2005.

[9] R. A. Jr., W. L. Johnson, J. Rickel, and A. Scholer. Learning domain knowledge for teaching procedural skills. In *AAMAS'02*, 2002.

[10] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. In *ACM Transactions on Embedded Computing Systems*, 2006.

[11] A. Kansal and M. Srivastava. An environmental energy harvesting framework for sensor networks. In *ACM Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2003.

[12] L. L. L, N. Shroff, and R. Srikant. Asymptotically optimal power-aware routing for multihop wireless networks with renewable energy sources. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 2:1262–1272 vol. 2, 13-17 March 2005.

[13] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, pages 157–163, New Brunswick, NJ, 1994. Morgan Kaufmann.

[14] D. Niyato, E. Hossain, and A. Fallahi. Sleep and Wakeup Strategies in Solar-Powered Wireless Sensor/Mesh Networks: Performance Analysis and Optimization. In *Transactions on Mobile Computing*, pages 221–236, Feb 2007.

[15] C. Vigorito, D. Ganesan, and A. Barto. Adaptive Control for Duty-Cycling in Energy Harvesting-based Wireless Sensor Networks. In *Proceedings of the Fourth Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 2007)*, San Diego, CA, June 2007.

[16] T. Voigt, H. Ritter, and J. Schiller. Utilizing solar power in wireless sensor networks. In *28th Annual IEEE International Conference on Local Computer Networks*, pages 416–422, October 2003.