

Parallelized Simulated Annealing for Model Updating in Ad-Hoc Wireless Sensing Networks

Andrew T. Zimmerman, Jerome P. Lynch
*Department of Civil & Environmental Engineering
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI USA
atzimmer@umich.edu, jerlynch@umich.edu*

Abstract

The engineering community has recently begun to adopt wireless sensing technologies for use in many sensing applications. These low-cost sensors provide an optimal setting for dense sensing networks, and can make large amounts of sensor data available. Also, the computational power embedded within each sensing node allows a wireless network to interrogate data within the network and in real-time. In a monitoring situation, these capabilities can be leveraged to detect and locate changes in system properties that could be an early indication of malfunction within a complex physical system. In this paper, a distributed model updating technique is embedded within the computational core of a wireless sensing network. Using a novel distributed implementation of the simulated annealing method, an ad-hoc network of wireless sensing units can determine updated system properties by iteratively matching data derived from an analytical model of the system with collected sensor data. By comparing updated system properties with those obtained in a baseline state, a wireless sensing network can detect and locate changes in the system. Experimental verification of this technique is provided on a network of wireless sensor prototypes using simulated results from a Navy ship-board chilled water system monitored with a wireless sensor network.

1. Introduction

Wireless sensing networks offer an extensive list of advantages to the engineering community. Primarily, they are attractive because they are significantly less expensive than traditional tethered systems, which have installation costs of a few thousand dollars per sensing channel when installed in large systems (e.g. buildings) [1]. In addition to these cost savings,

wireless technology also provides a framework for computational capabilities to be embedded within each individual sensing node. As a result, a wide variety of commercial and academic wireless sensing platforms have been developed and validated over the past decade [2].

It is widely anticipated that ad-hoc wireless sensor networks consisting of thousands of nodes will soon be deployed in practice, with each device in a network being capable of independently performing sensing, computation, and communication tasks simultaneously. Because of the sensor density associated with wireless networks, raw sensor data will be made available in much greater quantities than previously possible in a tethered setting. These large data sets can provide a powerful tool for the validation of analytical models predicting system performance, and may even allow for the early detection of changes within a monitored system. However, these benefits can only be obtained if the data gathered by a wireless network can be analyzed quickly and efficiently.

In this study, a distributed model updating framework is developed for an ad-hoc wireless sensing network. Using a novel distributed implementation of the widely used simulated annealing technique, an ad-hoc group of wireless sensors is programmed to determine current system properties by updating an analytical model to replicate sensor data. The network then looks for changes in system properties that may serve as an indicator of damage within the monitored system. Applications envisioned for this updating framework include the monitoring of failure-prone machinery and large-scale civil infrastructure, among many others.

Experimental validation of this technique is provided on a network of wireless sensor prototypes. In this study, simulated results from a Navy ship-board chilled water system instrumented with a dense wireless sensor network are used to update system

parameters in order to detect and localize damage within the system. Several advantages of the model updating method, including ease of a parallel implementation, are highlighted using this validation testbed.

2. Background on model updating

In an engineered system requiring a network of sensors, the ability to detect system damage is often very desirable. While there are several methods of translating raw sensor data into an estimate of damage, one common technique involves comparing system properties in an unknown state of health to those in a known, undamaged state [3,4]. In recent years, as computing resources have become more powerful and less expensive, it has become feasible to iteratively adjust system parameters in an analytical model such that the analytical system produces response data that matches results obtained experimentally. This technique is referred to as “model updating”. Using this method, damage can be detected in a system by periodically searching for changes in model parameters that can be linked directly to suboptimal system performance. Because these parameters describe properties local to one section of a system, damage can not only be detected, but localized through direct association with a particular system element.

A wide variety of model updating techniques have been developed over the years [5]. One common approach is to define an objective function, E , which relates the difference between analytical and experimental data. This function can be repeatedly evaluated with varying values of the analytical model parameters until the difference between the analytical and experimental response is minimized [6].

Simulated annealing (SA) is one of the most common algorithms for stochastically searching for the global minimum of such an objective function. This method has been used frequently in model-based damage detection techniques [6]. SA is modeled after the annealing process of material physics. In this process, a substance is slowly cooled from a molten state such that it assumes the global minimum energy state amidst a nearly infinite number of possible configurations. In 1953, Metropolis *et al.* developed an algorithm which simulates this process for model updating and other pattern classification applications [7]. The Metropolis criterion expresses the probability of a new system state being accepted at a given system temperature, and can be stated as: accept the new state if and only if

$$E_{new} \leq E_{old} - T \cdot \ln(U) \quad (1)$$

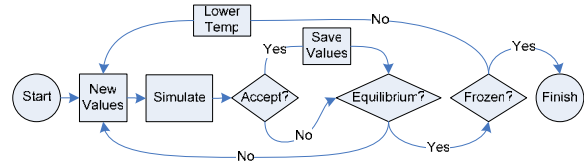


Figure 1. Standard simulated annealing cooling schedule

where E is the value of the objective function for a given energy state, U is a uniformly distributed random variable between 0 and 1, and T is the temperature of the system. The addition of the $T \cdot \ln(U)$ term allows the system to accept an invalid state in hopes of avoiding premature convergence to a local minima. Thus, state rejection is more common near the end of the SA process when T is low.

A standard SA algorithm (shown in Figure 1) begins the optimization process by assigning an initial temperature T_0 and letting the Metropolis algorithm run for N iterations. During each iteration, certain analytical model parameters are reassigned in a pseudorandom fashion, and the objective difference between the experimental and analytical output is determined. This newly created state is then either accepted or rejected based on the Metropolis criterion (Equation 1). After N iterations, the temperature of the system is reduced by a factor ρ , such that $T_{new} = \rho \cdot T_{old}$. This process continues until very few new states are accepted and the system has, in essence, frozen.

3. Distributed simulated annealing on a WSN

Because the memory requirements are negligible for most updating-based damage detection techniques, this type of analysis is attractive for implementation on a wireless sensing network, where storage capacity is very small. Over the years, many parallel SA techniques have been developed and successfully implemented [8]. Unfortunately, most of these methods rely on communication between processing nodes taking place before and after each generated state in the search tree. This creates a constant demand for communication and limits the effectiveness of these methods in the wireless domain. However, by taking advantage of the fact that the annealing process typically rejects more energy states than it accepts, (especially as the algorithm converges toward a solution), the SA procedure can be parallelized to take advantage of the computational resources distributed across a wireless sensing network.

Specifically, this is done by breaking up the traditionally serial SA search tree (which is continuous across all temperature steps) into a set of smaller search trees, each of which corresponds to a given temperature step and begins with the global minimum

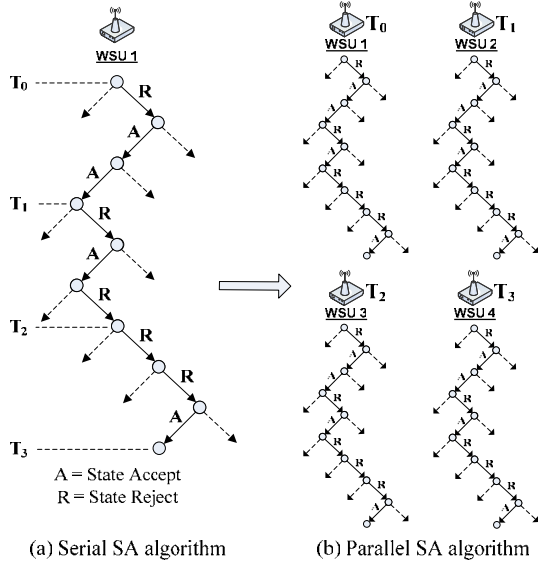


Figure 2. Parallelized simulated annealing algorithm

values for the preceding step. Each of these smaller trees can be assigned to any available node on the network, and thus can run concurrently. Figure 2 illustrates this principle.

As the parallelized search progresses, updated global state information is disseminated downwards through the network, allowing each node to maximize the effectiveness of its search at a given temperature. Specifically, when a node detects a new global minimum energy state, it will propagate a message downwards through the tree until all nodes at lower temperatures have restarted their search based on the new information. This may seem wasteful at high temperatures, but as the search algorithm converges on a solution it becomes decreasingly likely that a new global minimum will be found at a given temperature step. Thus, especially as the search nears completion, the number of nodes in the network weighs heavily on the amount of time saved by parallelization.

Another important benefit of using a wireless sensing network to perform distributed data analysis is that by taking advantage of the ad-hoc nature of certain wireless network topographies, computational tasks can be distributed intelligently to all available nodes in a given network. Within the parallel model updating methodology, each node in the search tree can independently beacon the network, looking for idle units. In this way, an ad-hoc assignment of tasks can propagate through the network, with each node making itself newly available when its computations are complete. This creates a very scalable framework for embedded tasks such as distributed model updating, as will be illustrated in greater depth in Section 4.

An additional modification to the traditional SA search algorithm is also presented as a tool to increase

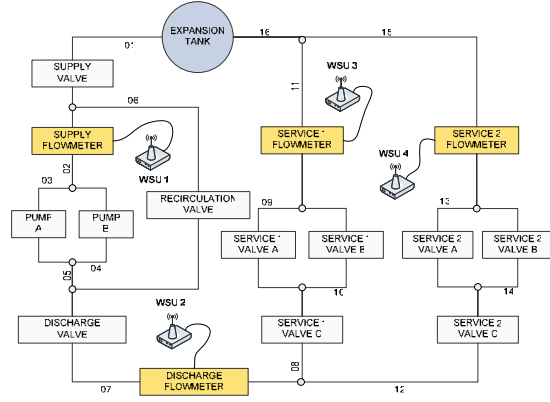


Figure 3. Navy chilled water distribution system

the speed at which a converged solution can be found in a distributed search. In addition to decreasing the annealing temperature after each temperature step, a second parameter defining the percentage of total search space to be made available for state transitions (or “search radius”) is also decreased. This allows the algorithm to limit the randomness of its search as it converges on a solution.

4. Experimental validation

To validate the performance of the proposed distributed SA process, a set of experiments were run on simulated data from a Navy shipboard chilled water system (shown in Figure 3). The pipe network in this system contains sixteen pipes of equal diameter but varying length. The system is driven by two identical pumps, located along pipes three and four, and there are four flowmeters measuring flow within the network. Each of these flowmeters is monitored by a single wireless sensing unit.

For this study, the Narada wireless sensing prototype developed at the University of Michigan is adopted [9]. This unit has at its core an Atmel Atmega128 microprocessor with 128kB of external SRAM for computation and data storage. Its communication interface consists of the Chipcon CC2420 IEEE 802.15.4 compliant wireless radio. It has a four channel, 16-bit Texas Instruments ADS8341 ADC for data acquisition, and a two channel, 12-bit Texas Instruments DAC7612 DAC for actuation. Given that the hardware design of the Narada wireless sensor shares many components with the Berkeley Mote platform [10], it is conceivable that the SA algorithm proposed in this study could be easily embedded on a network of Motes.

An analytical model of the pipe network described above is designed and embedded within each sensing node. Given a particular system configuration, this software can determine the expected flowrate and head

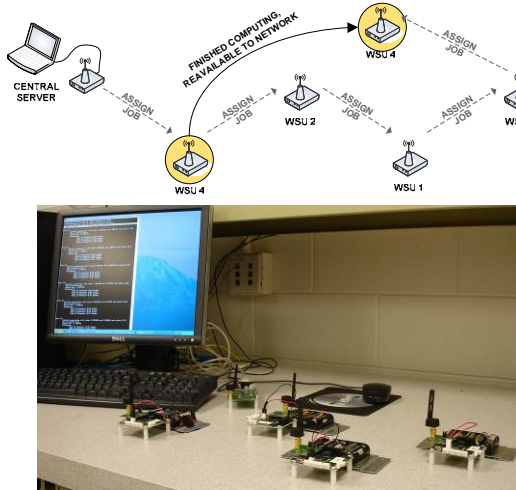


Figure 4. Ad-hoc computational parallelization

loss in each pipe. By conserving mass at each pipe junction and using the linearized Darcy-Weisback equation to relate head loss in loops, a set of linear equations describing the piping network can be automatically formulated [11]. These equations can then be solved using Gauss-Jordan elimination, and values for flowrate can be found. However, because head losses in this environment are dependent on flowrates and flowrates are dependent on head losses, this process must iterate until a solution is converged upon. Once embedded within the Narada's Atmega128 processor, each execution of the Gauss-Jordan process is timed and found to take roughly 0.73 seconds.

This type of system is an ideal testbed for distributed model updating. Because flowrate data is constantly available at several locations throughout the pipe network, an objective function can easily be created to compare experimental observations with analytical model projections:

$$E = \sum_i \sqrt{(Q_{anal}^i - Q_{sense}^i)^2} \quad (2)$$

where E is the objective difference between analytical and experimental results and Q_{anal}^i and Q_{sense}^i are the analytical and sensed flowrates, respectively, in pipe i .

Using the SA method, this objective function can be repeatedly evaluated given different values for a set of model properties until an "updated" configuration of parameters is found that minimizes the objective difference between expected and observed flowrates. Because changes in this objective function can be interpreted as a direct indicator of damage, the choice of updating parameters is extremely important in terms of the ability to detect and localize system bifurcations. In the case of a chilled water system, it is advantageous to choose updating parameters that strongly influence

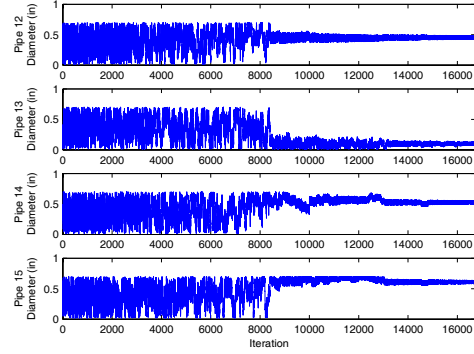


Figure 5. SA pipe diameter convergence

flowrate, such as pipe diameter. By doing so, it becomes possible to look for changes in pipe properties as an indication of suboptimal performance.

For the sake of simplicity, it is assumed that all damage in this system is local to sector three, which contains pipes 12,13,14, and 15. In order to account for a reduction in flow rate that may result from a blocked or leaking pipe, pipe diameters for each of these four pipes are designated as the updating parameters for this problem. All pipes in the system are initialized with a diameter of 0.5 inches. In order to introduce simulated damage to the system, sensor results are simulated with pipe 13 having an 80% reduction in pipe diameter.

Having obtained sensor readings for the damaged state, a central server sets up the model updating problem and chooses one sensing unit to serve as the initial network coordinator. This unit then searches for an available node to assign additional tasks to, and the SA process continues in an ad-hoc fashion until a set of updated parameters are converged upon. This ad-hoc process can be seen in Figure 4, and a time series representation of pipe diameter convergence can be seen in Figure 5.

In order to evaluate the effectiveness of the parallelized updating algorithm, an entire model update is completed several times with varying numbers of Narada nodes made available for computation. It can be seen from Figure 6 that this type of parallelization can drastically decrease the time required to find

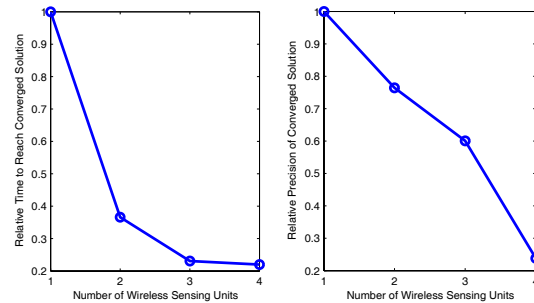


Figure 6. Speed and accuracy improvements using parallel SA

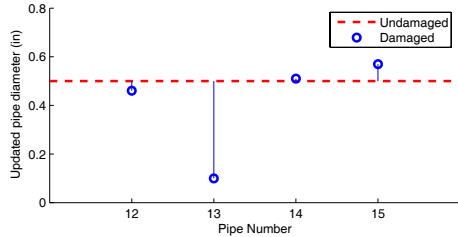


Figure 7. Updated results for pipes in sector three

updated model properties and detect damage within a system. It can also be seen from this figure that increasing the number of computing sites also improves the accuracy of the final solution.

Figure 7 shows the ability of this algorithm to accurately detect and locate system bifurcations. It can be seen that the SA method rightly identified a significant diameter reduction in pipe 13, while pipes 12, 14, and 15 only observe minor variations.

5. Conclusions

With recent advancements in wireless technology, it has become feasible to consider the large-scale deployment of wireless sensing networks in a wide variety of engineering applications. Because of the high density with which wireless sensors can be deployed within a given network, a large amount of in-network response data can be made available. Thus, it has become extremely desirable to develop automated techniques which utilize the computational capabilities of individual nodes within a network to determine system-wide properties.

In this paper, a novel distributed implementation of the simulated annealing method is developed which utilizes the ad-hoc properties of a wireless sensing network to perform model updating tasks in a distributed setting. This study expands on previous work in parallel simulated annealing by creating a framework with relatively low communication demands. This approach is extremely scalable, making it appropriate for use in a large wireless sensing network. It can also be easily ported for use with virtually any existing wireless sensing device.

For validation purposes, a testbed was devised using simulated data from a Navy shipboard chilled water system. Software was written that could estimate system flowrates based on pipe quality information and was embedded within the computational core of a network of Narada wireless sensing prototypes. It was found that this model updating method could accurately detect system damage by investigating changes in system properties, and that the parallel implementation of the SA process led to significant time savings and improved solution accuracy.

6. Acknowledgements

The authors would like to express their gratitude to the US Office of Naval Research for sponsoring the research presented (Young Investigator Award granted to Prof. J.P. Lynch). Additional support has been provided to Andrew Zimmerman by the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program.

7. References

- [1] M. Celebi, "Seismic Instrumentation of Buildings (With Emphasis on Federal Buildings)," *TR No. 0-7460-68170*, United States Geological Survey, Menlo Park, CA, 2002.
- [2] J. P. Lynch and K. J. Loh, "A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring," *Shock Vib. Dig.*, Vol. 38(2), 2006, pp. 91-128.
- [3] S.W. Doebling, C.R. Farrar and M.B. Prime, "A summary review of vibration-based damage identification methods," *Shock Vib. Dig.*, Vol. 30(2), 1998, pp. 91-105.
- [4] Anne Teughels, Johan Maeck, Guido De Roeck, "Damage Assessment by FE Model Updating using Damage Functions," *Computers and Structures*, Vol. 80(25), 2002, pp. 1869-1879.
- [5] J.E. Mottershead and M.I. Friswell, "Model Updating in Structural Dynamics: A Survey," *Journal of Sound and Vibration*, Vol. 167(2), 1993, pp. 347-375.
- [6] R.I. Levin and N.A.J. Lieven, "Dynamic Finite Element Model Updating using Simulated Annealing and Genetic Algorithms," *Mechanical Systems and Signal Processing*, Vol. 12(1), 1998, pp. 91-120.
- [7] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller, "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, Vol. 21(6), 1953, pp. 1087-1092.
- [8] Daniel R. Greening, "Parallel simulated annealing techniques," *Physica D*, Vol. 42, 1990, pp. 293-306.
- [9] Andrew Swartz, Deokwoo Jung, Jerome Lynch, Yang Wang, Dan Shi, and Michael Flynn, "Design of a wireless sensor for scalable distributed in-network computation in a structural health monitoring system," *Proc. Int. Workshop on Structural Health Monitoring*, Stanford, CA, 2005.
- [10] S. Hollar, *COTS Dust*. MS Thesis. University of California at Berkeley, 2000.
- [11] Roland W. Jeppson, *Analysis of Flow in Pipe Networks*, Ann Arbor Science, 1976.