

Procedure for improving wildfire simulations using observations

Problem Presenter: JOHN BENOIT

USDA Forest Service, Riverside Fire Laboratory, Riverside, CA, USA

Authors:

ANNA BELKINE, Simon Fraser University, Vancouver, Canada

EUN HEUI KIM, California State University, Long Beach, CA, USA

FLORENCE J. LIN, University of Southern California, Los Angeles, CA, USA

SEEMA NANDA, Tata Institute of Fundamental Research, Bangalore, India

PEI-ZHE SHI, University of Washington, Seattle, WA, USA

BRIAN STOCK, Harvey Mudd College, Claremont, CA, USA

CARISSA TUDOR, Claremont McKenna College, Claremont, CA, USA

September 29, 2009

Abstract

This report suggests a variational update method for improving wildfire simulations using observations as feedback to update information. We first assume a one-dimensional fire model for simplicity and present numerical simulations obtained in this case. As possible alternative approaches, we also discuss two other update methods: a particle filter method and an optimal control method.

1 Introduction

Two problems were suggested by John Benoit for the one-week workshop at Harvey Mudd College. Both of these were based on fire modeling research by Francis M. Fujioka *et al.* [3, 2] at the USDA Forest Service, Riverside Fire Laboratory:

Problem 1. Develop spatial and temporal probability models to simulate human-caused ignitions.

Problem 2. Suggest improvements in the prediction accuracy of an existing fire spread modeling system.

We decided to pursue Problem 2 for various reasons. Modeling human-caused fire would require understanding and information on variables and parameters that are not readily available. There is information in the literature providing us with a starting point for the second problem, allowing us to make a contribution which is of more practical significance.

1.1 Background

In an effort to develop improved fire management strategies, the USDA Forest Service employs probabilistic methods to determine likely fire spread behavior. FARSITE, a simulator (developed by Mark A. Finney [1, 6]) is currently used by the Forest Service to predict the spread of fire. The exact model or the assumptions used in FARSITE are not readily available to us. However we do know that it assumes that fire spreads elliptically and that the propagation of fire in FARSITE uses Huygens Principle which describes the formation of the leading edge of the fire boundary as a wavefront, (as opposed to other models which use level set modeling) [1]. The FARSITE software receives inputs which include weather conditions, terrain, vegetation specifications and ignition points, at a given time step t_k . The output of FARSITE is an image of the predicted fire area at time t_{k+1} .

The USDA Forest Service is interested to develop methods that would improve the prediction of fire spread behavior without changing the FARSITE program directly. In the absence of details about FARSITE we treat it like a “black box” that we cannot modify. We assume that error in the prediction of the burned forest area is due to imperfect input information and not due to significant model errors by FARSITE itself. Thus we focus our attention on achieving better results by working with the inputs.

There are at least two systemic ways in which input parameters are imperfect or incomplete. First, the measurements of the inputs include a degree of error. For example, wind is measured at a specific point. The point measurements cannot fully take into account terrain or other conditions that vary between the point of measurement and the location of the fire. While adjustments are made to accommodate these inconsistencies, the wind value (as entered into FARSITE) is an approximation of the actual wind at the fire site. The second possible way in which information about the input is incomplete is when different inputs are collected at different frequencies. For example, it could be that wind speed is updated every hour by the weather station, but information about humidity comes from a different location and is updated every 10 minutes. In this case one would run the simulation every 10 minutes to use all available information; however, one would have to repeatedly produce a new wind input for 6 simulations until a new wind measurement is received.

We employ a recursive updating technique to change the input measurements and thus improve the prediction of fire spread behavior. We do this on the second input problem described above. Further we work with only one parameter, namely wind to begin with. But our method is easily extended to address the first problem of input information and multiple parameter inputs.

We simplify the problem to a one dimensional *Naive Fire 1* model and propose a two dimensional *Naive Fire 2* to predict a moving fire *boundary*. The *Naive Fire 1* model was

used to run the recursive updating method.

Myriad recursive updating techniques exist and could be applied to this problem. We consider two different approaches to update missing input information. The first approach uses a variational method to update inputs based on minimizing the difference between the predicted fire spread and the actual fire spread. The second approach employs an algorithm developed by Feng Gu and Xiaolin Hu [4] for a particle filter designed specifically for the purpose of updating parameters in a fire prediction system.

We present detailed discussion of the variational update approach along with preliminary results in section 2 and an outline of a particle filter approach, in section 3. We also describe an outline of an alternative formulation of the variational update approach as an optimal control approach in section 4.

2 A variational update approach with preliminary results using a one-dimensional fire spread model

We attempt to answer the following question in the affirmative: Using an observation of an actual fire, can one improve a prediction of the resulting boundaries without modifying the FARSITE program?

2.1 Variational update of the wind in a one-dimensional *Naive Fire 1* model of a moving fire boundary

The goal in this update method is to use given initial wind observation to find a corrected wind input that minimizes the sum of the squares of the differences between the predicted and observed boundaries.

In a one-dimensional model of a fire, that is, a fire that can only spread along a line in either direction, there are two boundary points x_L and x_R , representing left and right boundaries, respectively. Let the rates of change dx_L/dt and dx_R/dt of the boundaries satisfy:

$$\frac{dx_L}{dt} = -r + w(t) \tag{1}$$

$$\frac{dx_R}{dt} = r + w(t) \quad , \tag{2}$$

where r is the constant rate of the fire spread to the right in the absence of wind, $-r$ is the constant rate of the fire spread to the left in the absence of wind, and $w(t)$ is the time-dependent wind velocity.

Let $\mathbf{x}_0 = (x_L(t_0), x_R(t_0))$ be the boundary at initial time t_0 . Let the initial wind at time t_0 be w_0 . Let $\mathbf{x}_1 = (x_L(t_1), x_R(t_1))$ be the boundary at time t_1 generated by a simulator S (e.g., from a one-dimensional model, from FARSITE, or from any other simulator), i.e.,

$$\mathbf{x}_1 = S(\mathbf{x}_0, w_0). \quad (3)$$

Using the observed boundary \mathbf{X}_1 of an actual fire at time t_1 , the error in the simulated boundary \mathbf{x}_1 at time t_1 , denoted by e_1 , is given by

$$e_1 = \|\mathbf{x}_1 - \mathbf{X}_1\| \quad (4)$$

where $\|\mathbf{x}_1 - \mathbf{X}_1\|$ is the magnitude of the vector $\mathbf{x}_1 - \mathbf{X}_1$. We apply a linear correction to the wind w_0 at time t_0 . Let the corrected wind be y_0 given by

$$y_0 = a_0 w_0 + b_0. \quad (5)$$

Here a_0 and b_0 are parameters to be determined by solving the following minimization problem

$$\min_{a_0, b_0} \|S(\mathbf{x}_0, y_0) - \mathbf{X}_1\|. \quad (6)$$

This is the error between the boundary determined by simulator and the actual observed boundary X_1 . Note the simulator boundary uses the parameterized wind y_0 as the input. We iterate this process till we reach a specified error tolerance. Thus, using the observed boundary \mathbf{X}_i of an actual fire at time t_i , the error e_i at the i th iteration will be given by

$$e_i = \|\mathbf{x}_i - \mathbf{X}_i\|. \quad (7)$$

At each time t_i use a linear correction y_i for the wind w_i :

$$y_i = a_i w_i + b_i, \quad a_i, b_i \in \mathbf{R}. \quad (8)$$

At each time step a_i and b_i are parameters to be determined by minimizing the error between the simulated boundary determined by the parameterized wind y_i and the actual observed boundary \mathbf{X}_{i+1} at the next time step

$$\min_{a_i, b_i} \|S(\mathbf{x}_i, y_i) - \mathbf{X}_{i+1}\|. \quad (9)$$

There are several minimization routines readily available. We resort to Matlab to solve this, using its *fminsearch* routine.

2.2 Examples of results for the variational method for one-dimensional *Naive Fire 1*

The variational method applied to the one-dimensional *Naive Fire 1* uses a one-step linear correction

$$y_t = a w_t + b.$$

To test the effectiveness of the method, we generate a series of artificial winds, each one being a function of time, and compare the result of variational method with the actual fire boundaries. As a control, we also show the result from uncorrected model, i.e., take y_t as the wind w_t at time t directly to predict the burning region in the future. In each test, at time t_k , all the historical data up to time t_k is accessible to the algorithm, while the prediction of time t_{k+1} is compared to the actual burning region.

In Figures 1-5, the red diamonds are the predicted burning boundaries using the linear “correction” $aw_t + b$ for the wind. The black stars are the predicted burning boundaries using the “uncorrected” model w_t for the wind on a coarse time grid. The blue solid lines are the “actual” burning boundaries for the model on a fine time grid, assuming no observation error.

1. Constant wind (Figure 1)

We test the algorithm in the trivial case: constant wind $w_t = w_0$. Under constant wind, the wind information in the future is predicted accurately; no correction is needed. Here the predictions of both the corrected and uncorrected models are exact. See Figure 1.

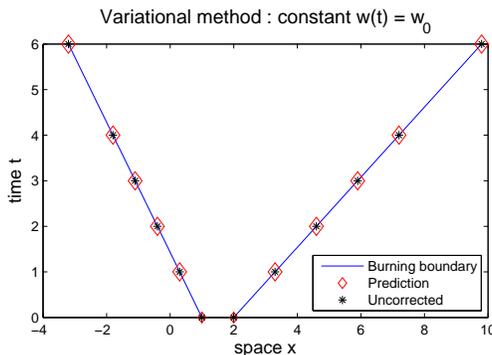


Figure 1: Variational method for a constant wind.

2. Linear wind (Figure 2)

We next suppose the wind is changing linearly, i.e., $w_t = w_0 + r_w t$, where r_w is the time rate of change of the wind. The linear wind is important because if the interval between two time points is small, or if the wind is not changing drastically, linear wind is generally a good approximation.

Figure 2 shows that the prediction $aw_t + b$ using a variational method is better than the uncorrected model prediction w_t . However, even for a linear wind, the linear correction cannot make an exact prediction, because a linear wind results in a quadratic motion of the burning boundary due to the effect of time integration.

3. Constant wind with a periodic oscillation (Figure 3)

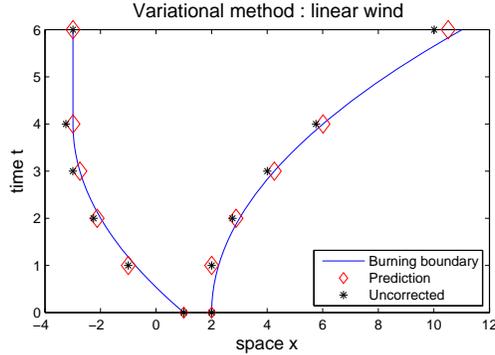


Figure 2: Variational method for a linear wind.

To mimic a real wind, we add a periodic oscillation to a constant wind, i.e., $w_t = w_0 + \alpha_w \sin(\omega_w t)$, where α_w and ω_w are the amplitude and the frequency, respectively, of the wind's oscillation. The period of oscillation is assumed to be much shorter than the time interval between time points. This wind profile represents a long lasting constant blowing wind with small fluctuation, which is very likely to be the case in reality. Figure 3 shows that a variational method using $aw_t + b$ can capture the dominant trend of the wind better than uncorrected method using w_t , where the prediction of the latter is greatly affected by instantaneous fluctuation.

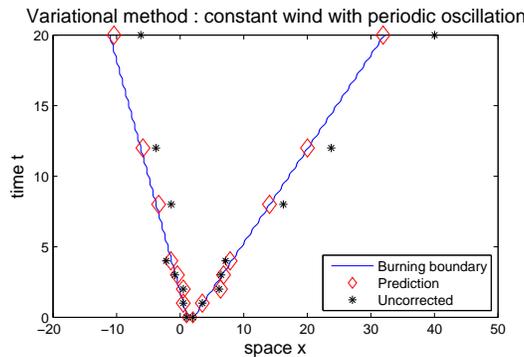


Figure 3: Variational method for a constant wind with periodic oscillation.

4. Linear wind with a periodic oscillation (Figure 4)

Then we test the variational method on a linear wind with a periodic oscillation, i.e., $w_t = w_0 + r_w t + \alpha_w \sin(\omega_w t)$. The performance is pretty good.

5. Random wind (Figure 5)

Out of curiosity, we tested the variational method on a random wind, which undergoes a Brownian motion B_t in one dimensional space, i.e., $w_t = a_w B_t$ with amplitude a_w . Not surprisingly, the predicted boundary is nowhere close to the actual burning boundary

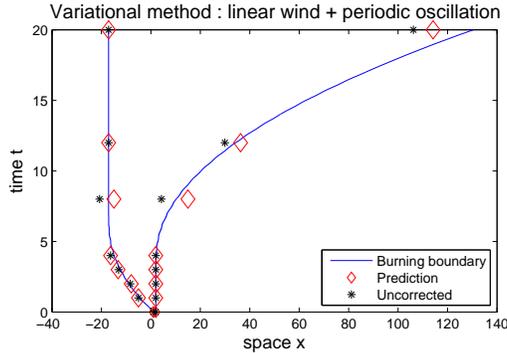


Figure 4: Variational method for a linear wind with periodic oscillation.

in Figure 5. This is the only case where the uncorrected model w_t performs better than corrected model $aw_t + b$. The reason is that the uncorrected model uses the wind speed w_t at time t , which is the expectation of the wind speed in the future, i.e., the best guess for a Brownian motion. However, the “corrected” model $aw_t + b$ tries to make an assumption regarding the linearity of the moving burning boundaries; this assumption is totally wrong for a random wind.

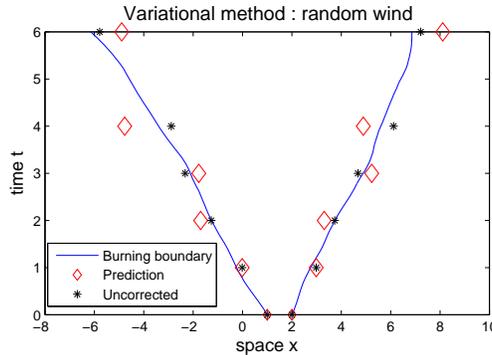


Figure 5: Variational method for a random wind.

3 A particle filter approach

3.1 Particle filter update of the wind for a FAUX-SITE model of a moving burned area

In this section we summarize the particle filtering method developed by Gu and Hu [4]. Unfortunately because of insufficient data we had during the workshop, an implementation of this method was not feasible.

Using an initial wind observation $wind_k$ and the observed burned area A_k at time t_k , the goal is to find a corrected input $wind_{k+1}$ at time t_{k+1} that will minimize the sum of the squares

of the differences between the predicted burned area $\widetilde{FS}(wind_k, A_k)$ and the observed burned area B_k at time t_{k+1} . Notice that the minimization is now on the differences between the predicted and the observed burned areas rather than the differences between the predicted and observed boundaries of these areas. The notation B_k is the observed burned area at time t_k , *not* to be confused with a Brownian motion.

Consider the wind to be modeled by a 2-dimensional vector that varies with time but *not* with space. The simulated boundary P_{k+1} at time t_{k+1} is

$$P_{k+1} = FS(wind_k, P_k) \quad , \quad (10)$$

where the output from FS is the *boundary* of the burned area from a proposed FAUX-SITE simulator of the burned area or some other simulator.

Using the particle filter method of Gu and Hu [4], the wind velocity steps with time according to

$$wind_{k+1} = f(wind_k) + \nu_k \quad , \quad (11)$$

where $f(wind_k)$ produces the wind $wind_{k+1}$ at a time t_{k+1} and the noise ν_k is the model error at time t_k .

Let A_k and A_{k+1} be the observed burned areas at times t_k and t_{k+1} , respectively. Then the newly burned area B_k between time t_k and t_{k+1} is

$$B_k = A_{k+1} - A_k \quad . \quad (12)$$

The measurement error ω_k at time t_k between the observed newly burned area B_k and the simulated *newly burned area* \widetilde{FS} from FAUX-SITE is

$$\omega_k = B_k - \widetilde{FS}(wind_k, A_k) \quad . \quad (13)$$

The model error ν_k in the wind over time t_k is normally distributed $N(0, \sigma_\nu^2)$ with mean 0 and variance σ_ν^2 . The measurement error ω_k in the newly burned area over time t_k is normally distributed $N(0, \sigma_\omega^2)$ with mean 0 and variance σ_ω^2 .

Let B_0 be the burned area at time t_0 . Beginning with the burned area B_0 at time t_0 and the wind w_0 at time t_0 , let $\widetilde{FS}(wind_0, A_0)$ be the burned area at time t_1 generated by the simulator FS (i.e., from FAUX-SITE). Using the observed burned area B_1 of an actual fire at time t_1 , the error e_1 in the simulated burned area $\widetilde{FS}(wind_0, A_0)$ at time t_1 will be given by

$$e_1 = \|\widetilde{FS}(wind_0, A_0) - B_0\| \quad . \quad (14)$$

At each time t_{k+1} , let a corrected input $f(wind_k)$ for the wind $wind_k$ at time t_k take the form

$$wind_{k+1} = f(wind_k) \quad . \quad (15)$$

Here $wind_{k+1}$ is the output of the particle filter method (instead of a result determined by minimizing the error between the simulated boundary determined by the parameterized wind y_i and the actual observed boundary \mathbf{X}_{i+1}). At each time step, using the observed burned area B_k of an actual fire from time t_k to t_{k+1} , the error e_{k+1} in the simulated burned area $\widetilde{FS}(wind_k, A_k)$ at time t_{k+1} will be given by

$$e_{k+1} = \|\widetilde{FS}(wind_k, A_k) - B_k\| \quad . \quad (16)$$

Continue to repeat to determine the simulated burned area at successive times.

3.2 Algorithm: Particle filtering method

We now present the particle filtering algorithm developed by Gu and Hu [4].

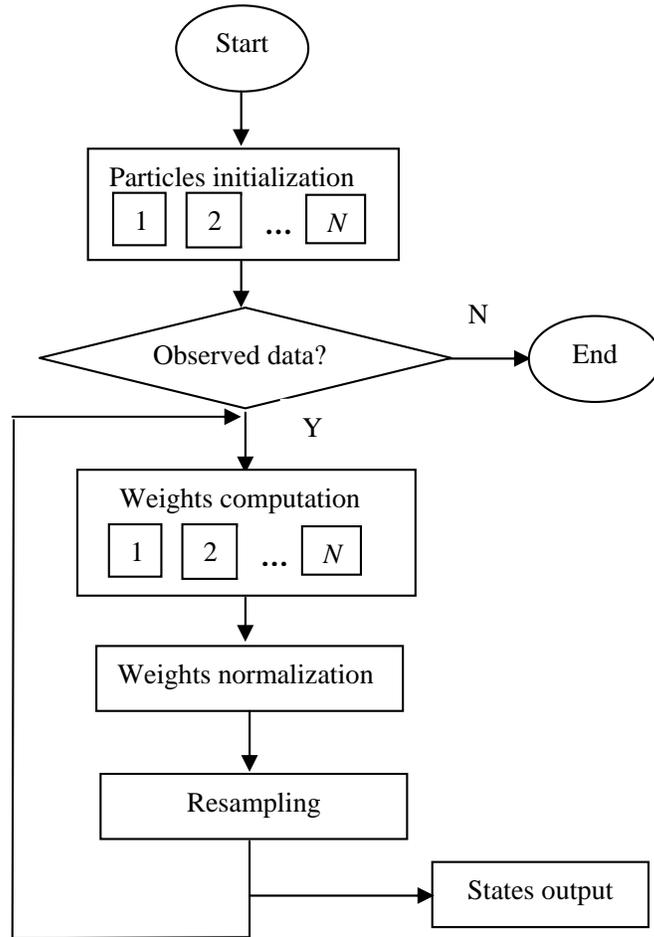


Figure 6: From Gu and Hu [4]

We first let $wind_k = [wsp_k, wdir_k]$, where wsp_k is the wind speed at time t_k and $wdir_k$ is the wind direction at time t_k . Recall that we have

$$\begin{aligned} wind_{k+1} &= f(wind_k) + \nu_k, \\ B_k &= A_{k+1} - A_k, \\ \omega_k &= B_k - \widetilde{FS}(wind_k, A_k), \end{aligned}$$

where $\nu_k \sim N(0, \sigma_\nu^2)$ and $\omega_k \sim N(0, \sigma_\omega^2)$.

The particle filtering algorithm is then as follows.

1. Particles initialization:

For $i = 0$ to $N - 1$,

Randomly generate $vwsp(i, 0) \sim N(0, \sigma_\nu^2)$ and $vmdir(i, 0) \sim N(0, \sigma_\omega^2)$;
 $wsp(i, 0) = wsp_0 + vwsp(i, 0)$;
 $wdir(i, 0) = wdir_0 + vmdir(i, 0)$;

2. Weights computation:

For $i = 0$ to $N - 1$,

Randomly generate $vwsp(i, k) \sim N(0, \sigma_\nu^2)$ and $vmdir(i, k) \sim N(0, \sigma_\omega^2)$;
 $wsp(i, k) = f_{wsp}(wsp(i, k - 1)) + vwsp(i, k)$;
 $wdir(i, k) = f_{wdir}(wdir(i, k - 1)) + vmdir(i, k)$;
 $wind(i, k) = [wsp(i, k), wdir(i, k)]$;
Randomly generate $w(i, k) \sim N(0, \sigma_\omega^2)$;
 $B(i, k) = \widetilde{FS}(wind(i, k)) + \omega(i, k)$;
 $weights(i, k) = obA(k + 1) - obA(k) - B(i, k)$;
 $weights(i, k) = \frac{1}{\sigma_\omega \sqrt{2\pi}} \exp(-\frac{weights(i, k)^2}{2\sigma_\omega^2})$;

3. Weights normalization:

Set $s_{wts} = s_{wts} + weights(i, k)$.

For $i = 0$ to $N - 1$,

$s_{wts} = s_{wts} + weights(i, k)$;

For $i = 0$ to $N - 1$,

$n_{wts} = \frac{weights(i, k)}{s_{wts}}$;

4. Resampling:

Set $q(0) = n_{wts(0, k)}$;

For $i = 0$ to $N - 1$,

$q(i) = q(i - 1) + n_{wts}(i, k);$
 Uniformly generate N numbers between 0 and 1 and sort them as an array u ;
 Set $Count = 1$;
 For $j = 0$ to $N - 1$,
 While ($q(count) < u(j)$);
 $count = count + 1$;
 $temp(j) = wind(count, k)$;
 For $l = 0$ to $N - 1$,
 $wind(l, k) = temp(l)$;

5. Output:

$os(k) = 0$;
 For $i = 0$ to $N - 1$,
 $os(k) = os(k) + wind(i, k) * n_{wts}(i, k)$;

4 An optimal control approach

In this section, we present an application of the optimal control method suggested by Kang [5].

4.1 One-dimensional *Naive Fire 1* model of a moving fire boundary

Using an initial wind observation $w_0 (= w(t_0))$, find a corrected input $w(t)$ that will minimize the sum of the squares of the differences between the predicted boundaries $\mathbf{x}(t)$ and observed boundaries $\mathbf{X}(t)$ over time t .

In a one-dimensional model of a fire with boundaries x_L and x_R , let the rates of change dx_L/dt and dx_R/dt of the boundaries satisfy

$$\frac{dx_L}{dt} = -r + w(t) \tag{17}$$

$$\frac{dx_R}{dt} = r + w(t) \quad , \tag{18}$$

where r is the constant rate of the fire spread to the right in the absence of wind, $-r$ is the constant rate of the fire spread to the left in the absence of wind, and w is the wind velocity.

Let $\mathbf{x}_0 = (x_L(t_0), x_R(t_0))$ be the boundary at time t_0 . Beginning with the boundary \mathbf{x}_0 at time t_0 and the wind w_0 at time t_0 , let $\mathbf{x}_1 = (x_L(t_1), x_R(t_1))$ be the boundary at time

t_1 generated by a simulator S (e.g., from a one-dimensional model, from FARSITE, from FAUX-SITE, or from some other simulator), i.e.,

$$\mathbf{x}_1 = S(\mathbf{x}_0, w_0) \quad . \quad (19)$$

Using the observed boundary \mathbf{X}_1 of an actual fire at time t_1 , the error e_1 in the simulated boundary \mathbf{x}_1 at time t_1 will be given by

$$e_1 = \|\mathbf{x}_1 - \mathbf{X}_1\| \quad . \quad (20)$$

Here the wind function $w(t)$ over the time interval $[t_0, t_1]$ is to be determined by minimizing the integral over time of the error between the simulated boundary determined by the wind $w(t)$ and the actual observed boundary

$$\min_{w(t)} \int_{t_0}^{t_1} \|S(\mathbf{x}_0, w_0) - \mathbf{X}_1\| dt \quad (21)$$

subject to the dynamics of the boundaries governed by equations (17) and (18).

Repeat at successive times t_i . Using the observed boundary \mathbf{X}_i of an actual fire at time t_i , the error e_i in the simulated boundary \mathbf{x}_i at time t_i will be given by

$$e_i = \|\mathbf{x}_i - \mathbf{X}_i\| \quad . \quad (22)$$

Here, the wind function $w(t)$ over the time interval $[t_0, t_N]$ is determined by minimizing the sum of the integrals over time of the errors between the simulated boundary determined by the wind $w(t)$ and the actual observed boundary \mathbf{X}_{i+1} at the next time step t_{i+1}

$$\min_{w(t)} \sum_{i=0}^{N-1} \left(\int_{t_i}^{t_{i+1}} \|S(\mathbf{x}_i, w_i) - \mathbf{X}_{i+1}\| dt \right) \quad (23)$$

subject to the dynamics of the boundaries governed by equations (17) and (18).

Optimal control software, such as DIDO [5, 7, 8], may be used to determine the optimal wind function $w(t)$, playing the role of the control parameter.

4.2 The wind $w(t)$ as an optimal control

In the variational update method, the following previously assumed forms for the time-dependence of the wind will be replaced by a function $w(t)$ to be determined:

Case 1 (Constant wind): $w(t) = w_0$.

Case 2 (Linear wind): $w(t) = r_w t + w_0$.

Case 3 (Constant wind with a periodic oscillation): $w(t) = w_0 + \alpha_w \sin(\omega_w t)$.

Case 4 (Linear wind with a periodic oscillation): $w(t) = r_w t + w_0 + \alpha_w \sin(\omega_w t)$.

Case 5 (Random wind): $w(t) = a_w B_t$.

In the optimal control approach, the wind function $w(t)$ is determined as the solution of an optimal control problem. The objective function may be taken to be the time integral of the magnitude of the difference between the predicted and the observed fire boundaries. The velocity $w(t)$ of the wind is the control to be optimized subject by minimizing an objective function subject to the equations governing the dynamics of the fire boundaries.

5 Concluding remarks

The first suggestion is to (1) develop a variational update method that would start with an initial observed wind parameter and use an updated wind model to determine the fire boundary at future time steps. Preliminary calculations apply this approach to a one-dimensional *Naive Fire 1*. This could be extended to a two-dimensional *Naive Fire 2*.

Further, there are two alternative suggestions for potential developments:

(2) Apply a particle filter method to determine an estimated wind to be used to determine the fire boundary at the next time step.

(3) Apply an optimal control approach to determine the wind as a function of time up to time t based on minimizing the difference between the observed fire boundary and the model fire boundary up to time t . Then use the past wind information to determine the fire boundary at the next time step.

6 Acknowledgements

We thank the USDA Forest Service, Riverside Fire Laboratory, Riverside, CA, for suggesting this problem and John Benoit for presenting it and for discussions. We wish to acknowledge Chris Farmer and Cameron Hall from Oxford University for their suggestions and help with obtaining relevant references for this project. Thanks to Wei Kang of the Naval Postgraduate School, Monterey, CA, for bringing the DIDO optimal control software (cf. [5, 7]) to our attention, and thanks also to Talithia Williams of Harvey Mudd College for presenting a tutorial on Kalman filters. And finally a big thanks to Ellis Cumberbatch for putting together a wonderful workshop which made it possible for us to work on this very interesting problem.

References

- [1] Mark A. Finney, FARSITE: Fire Area Simulator–Model Development and Evaluation, United States Department of Agriculture, Forest Service, Rocky Mountain Research Station, February 2004.
- [2] Francis M. Fujioka, A New Method for the Analysis of Fire Spread Modeling Errors, *International Journal of Wildland Fire* **11**, 193-203 (2002).
- [3] Francis M. Fujioka, Charles Jones and Phillip J. Riggan, Fire Modeling and Remote Sensing: Converging Technologies, *Proceedings of Fifth NRIFD Symposium*, November 30 - December 2, 2005.
- [4] Feng Gu and Xiaolin Hu, Towards Applications of Particle Filters in Wildlife Spread Simulation, in S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W.

Fowler, eds., *Proceedings of the 2008 Winter Simulation Conference*, IEEE, pp. 2852-2860 (2008).

- [5] Wei Kang and Naz Bedrossian, Pseudospectral Optimal Control Theory Makes Debut Flight, Saves NASA \$1 Million in Under Three Hours, *SIAM News*, **40** (7), September 2007.
- [6] Tonja Opperman, Jim Gould, Mark Finney, and Cordy Tymstra, Applying Fire Spread Simulators in New Zealand and Australia: Results from an International Seminar, *USDA Forest Service Proceedings RMRS-P-41*, pp. 201-212 (2006).
- [7] I. Michael Ross, *User's Manual for DIDO: A Matlab Application Package for Solving Optimal Control Problems*, NPS Tech. Rep. MAE-03-005, September 2003, Naval Post-graduate School, Monterey, CA.
- [8] I. Michael Ross, A Roadmap for Optimal Control: The Right Way to Commute, *Annals of the New York Academy of Sciences* **1065**, 210-231 (2006).