The Cloud Hunter's Problem

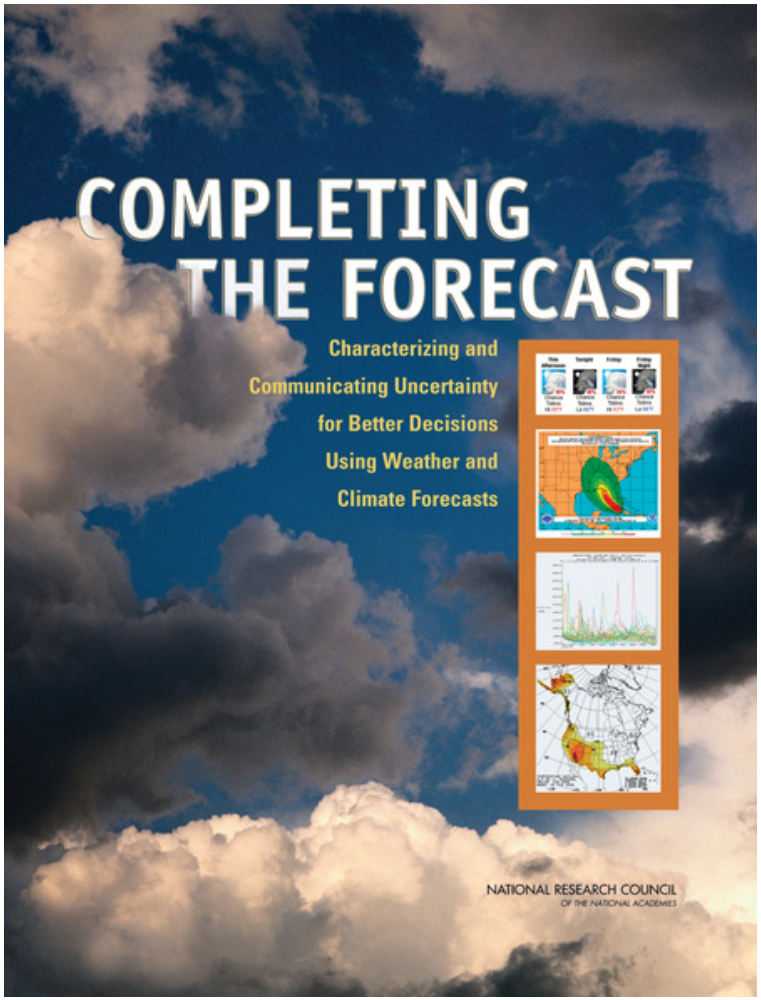
An Automated Decision Algorithm to Improve the Productivity of Scientific Data Collection in Stochastic Environments

Arthur Small

based on work with JASON B. STEFIK, JOHANNES VERLINDE, NATHANIEL C. JOHNSON

Decision Analysis

Beginnings: Probabilistic forecasting



The Forum

57° 33°

50 CENTS THURSDAY, APRIL 24, 1997 FARGO-MOORHEAD

Flood of '97

Finger-pointing begins in Forks

Mayor says poor forecasting doomed city; weather service says it gave its best effort

FIGURE 4.2 Headline from *The Forum* newspaper, April 24, 1997. SOURCE: Forum Communications Company.

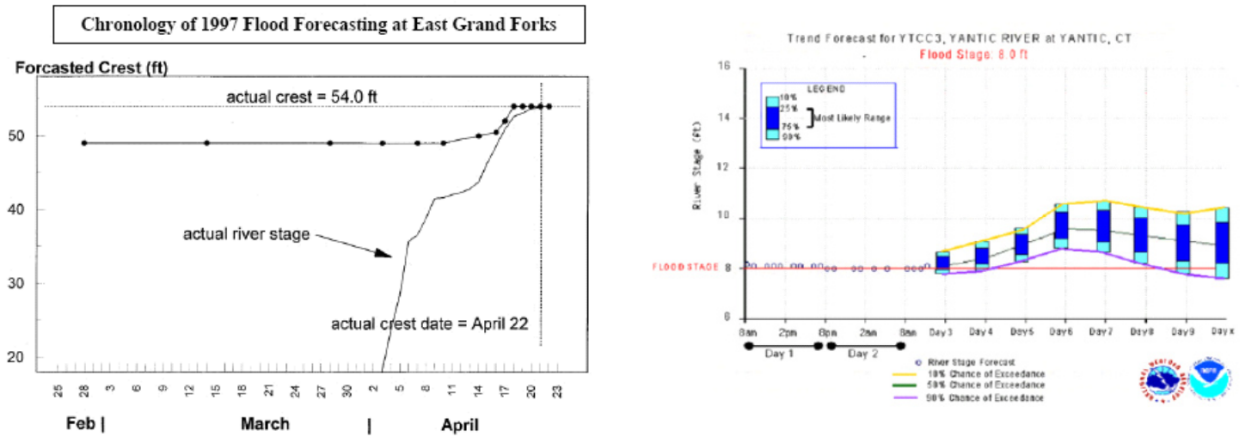


FIGURE 4.3 Left: Deterministic forecasts issued by NWS prior to the Red River flood of 1997. Right: Probabilistic river stage forecast from AHPS. SOURCE: NWS.

FIGURE 1: Communication of forecast information during the Red River Flood of 1997 in Grand Forks, North Dakota

The promise: data-driven decision making

Data science hype: "Better information $\rightarrow$ Better decisions!"

Claim: That's wrong.

Data-driven decision-making also requires complementary analytics and products:

- *Prediction* : calibration of error, $\rightarrow$ *probabilistic* forecasts
- *Optimization* : application-specific models and tools, attention to user objectives
- *Visualization/communication* : intelligence *actionable* in terms of user's decision problem

Realizing the promise of data-driven decision-making

On prediction, data science gallops along: machine learning, classification, prediction, inference...

Claim: The promise of data science to empower data-driven decision-making lags its potential, because of insufficient focus on *complementary* analytics:

- Rigorous quantification of *uncertainty* and *error* in outputs of predictive models
- Accurate formal representation of decision problems – including decision-maker's objectives and tolerance for risk
- Appropriate use of optimization techniques
- Visualizations or other products customized to decision contexts

Goal: Identify good *actions*, conditional on predictions made with error

The Cloud Hunter's Problem

Meet the Cloud Hunter

The Cloud Hunter is an atmospheric scientist.

The Cloud Hunter wants to collect data from inside *liquid boundary layer clouds*.

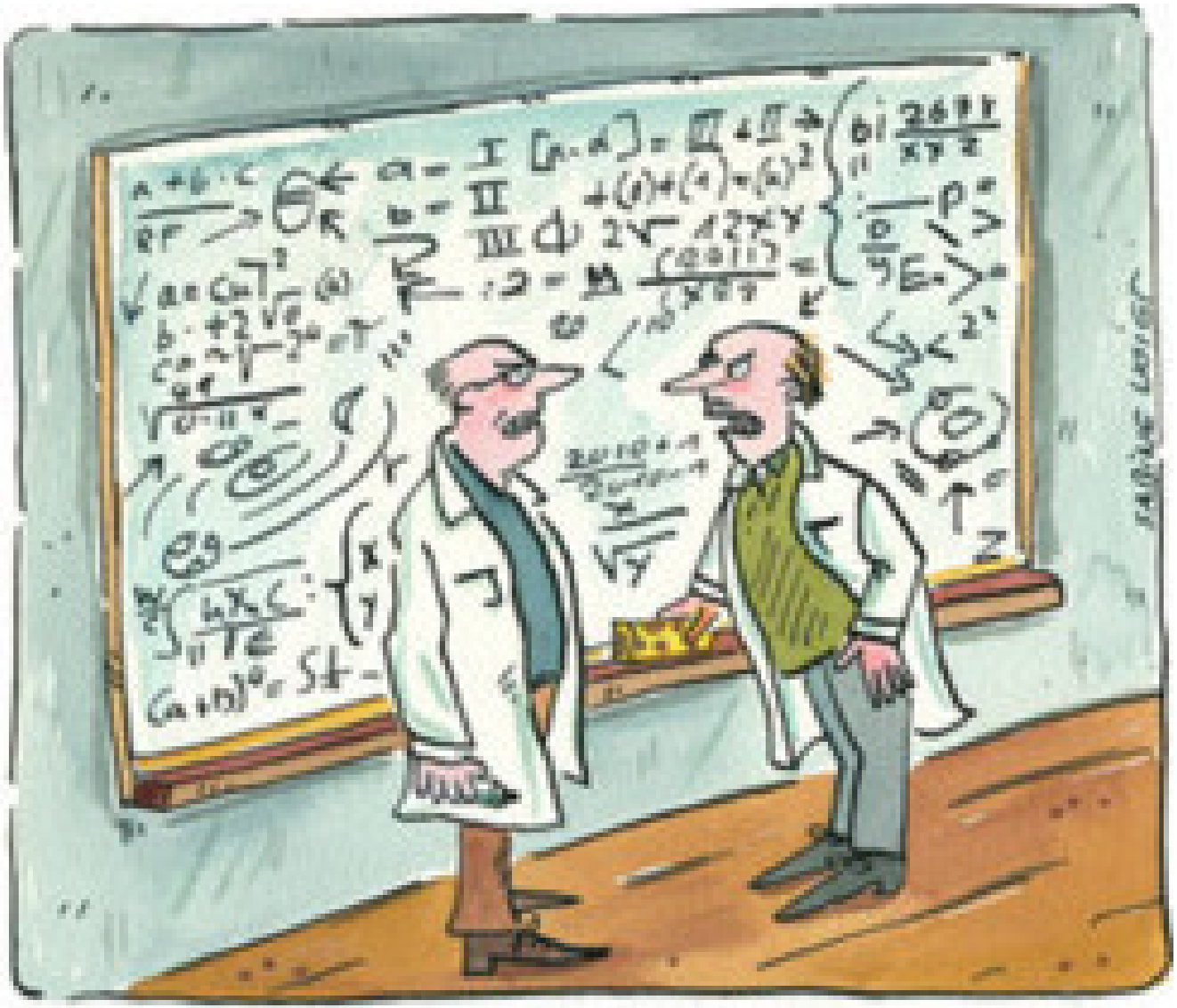
That takes aircraft. Which are expensive.

The Cloud Hunter's Decision Problem

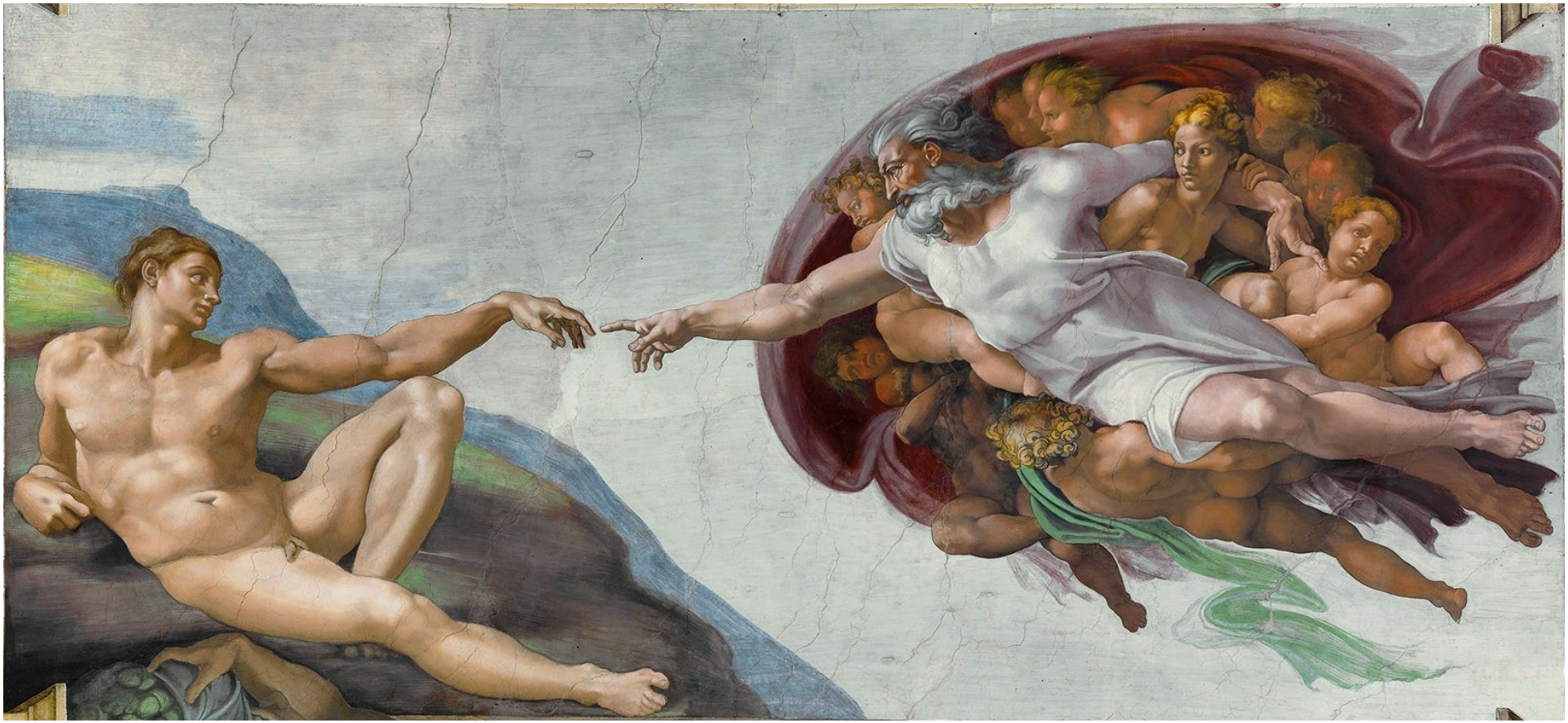
The Cloud Hunter's Problem concerns how to allocate a fixed budget of flight hours between dates over the course of a field season.

Fly/No-fly decisions must be made 1 day ahead, based on imperfect day-ahead forecasts of whether conditions are good or bad for collecting required data.

Current decision-making process:



Proposed decision-making process:



Formal model of the Cloud Hunter's decision problem

Model setup: Conditions for data collection over the course of the field season

D : length of the field season in days

$d = D, \ldots, 1$: index of dates

X_d : quality of conditions for data collection:

- $X_d = 1$ if conditions on date d are good, 0 otherwise
- Each X_d a binary random state variable, i.e., a Bernoulli trial

A field season is a particular realization $x_D, \ldots, x_1$ of the stochastic process $X_D, \ldots, X_1$.

Will assume the X_d are independent and identically distributed (i.i.d.).

- Assumption not actually required, but keeps things simpler and clearer.

Vector notation: $\mathbf{X} =< X_D, \ldots, X_1 >$ denotes the stochastic process; $\mathbf{x} =< x_D, \ldots, x_1 >$ denotes a particular realization.

Model setup: Decisions, resource constraints

$F \leq D$: number of flights in the Cloud Hunter's budget.

$f = F, \ldots, 1$: index of flights remaining in budget

a_d : binary control variables ("actions")

- $a_d = 1$ iff they opt to fly on date d, 0 otherwise

$\mathbf{a} =< a_D, \ldots, a_1 >$: actions chosen on dates $d = D, \ldots, 1$

Resource constraint: $\sum_d a_d \leq F$.

(Assume flights left over at the end of the season have no residual value.)

Payoffs and objectives

Payoffs: For a given sequence of choices $\mathbf{a}$ and realizations $\mathbf{x}$, the realized amount of data collected U is given by

$$U = \mathbf{a} \cdot \mathbf{x} = \sum_d a_d x_d$$

.

Decision-maker's objective: Choose a fly/no-fly decision rule to maximize data collected in expection, subject to the resource constraint on total allowable flights:

Choose $\mathbf{a}$ to $max_{\mathbf{a}} = E[\mathbf{a} \cdot \mathbf{X}]$, subject to $\sum_d a_d \leq F$.

Important: This is a *substantive assumption* about the decision-maker's goals.

- Models a decision-maker with a high tolerance for *risk*.

Forecasts

Decision taken on basis of a day-ahead forecast.

Before taking each decision, decision-maker receives a forecast signal $s_d \in \mathbb{S}$.

Calibration: map this signal to a probability of good conditions:

$$p(s) = \Pr\{X_d = 1 | s_d = s\}$$

.

(Will assume stationarity.)

Distribution of forecast signals

More than one day ahead, don't know which forecast signals $s \in \mathbb{S}$ will be received.

But, *do* know the the likelihood of receiving different signals.

$\pi(s)$: probability that forecasting system will generate signal s.

$\pi(\cdot)$ defines a probability distribution over the set $\mathbb{S}$ of possible forecast signals.

The task of the decision analyst

Given this set-up, the job of the decision analyst is to devise an *optimal decision rule* $a^* = a(d, f|p)$ that delivers a recommended action—fly or no-fly—as a function of

- d the number of days left in the field season,
- f the number of flights left in the budget, and
- $p(s)$ the forecast probability that a flight today would be successful.

A decision rule $a(\cdot)$ is deemed optimal if its consistent application maximizes in expectation the yield of successful flights realized from a given budget F.

Comment

This is a challenge of *intertemporal optimization*: each choice a_d alters the expected payoffs for subsequent decisions.

Need to use tools of optimization that account for this intertemporal structure.

Dynamic programming turns out to be the right tool for the job.

Optimization via dynamic programming

Intertemporal optimization via dynamic programming

Basic idea: break the decision problem into two pieces: (i) the next day's decision, and (ii) the rest of the field season after that.

Assume you've got the right answer (!). Given that this answer is optimal, derive the properties that solution must have.

Solve via backward induction.

Suppose an optimal decision rule $a(d, f|s)$ has been found.

Let $V(d, f)$ denote the expected number of successes that would be realized from repeated application of this rule, starting from initial conditions $< d, f >$.

By construction, $V(\cdot)$ equals the maximand of the decision-maker's objective function, beginning from these initial conditions, under the substitution $a = a(d, f|s)$:

$$V(d, f) = E^{\pi}\left[\sum_{i=d,\ldots,1} a(i, f_i|s_i) \cdot X_i\right]$$

where the expectation is taken over the probability distribution of all possible sequences of forecast signals (determined by π), and where f_i denotes the number of flights remaining on date i when the optimal program is followed.

$V(d, f)$ called the *value function* for this problem.

Because $a(\cdot)$ is assumed to be optimal, $V(\cdot)$ must satisfy a particular recursive relationship:

$$V(d, f) = E^{\pi}\left[a_d^* \cdot X_d + V(d-1, f - a_d^*)\right]$$

where $a_d^* = a(d, f|s_d)$ is the optimal decision.

Once the forecast signal s_d is received, a^* will be chosen optimally:

- If $a^* = 1$, then $V(d, f|s_d) = E[X_d|s_d] + V(d-1, f-1)$.
- If $a^* = 0$, then $V(d, f|s_d) = V(d-1, f)$.

Since $V(\cdot)$ is by construction a maximum, we must have:

$$V(d, f|s_d) = \max\{E[X_d|s_d] + V(d-1, f-1), V(d-1, f)\}$$

$$= \max\{p(s_d) + V(d-1, f-1), V(d-1, f)\}$$

This formula implies the optimal decision rule: $a^* = 1$ ("fly") only if

$$p(s_d) \geq V(d-1, f) - V(d-1, f-1)$$

i.e., when the expected payoff of the current day's opportunity exceeds the loss in marginal cost associated with arriving tomorrow with one fewer flight in the bank.

Call $p(s_d)$ the *hurdle probability*.

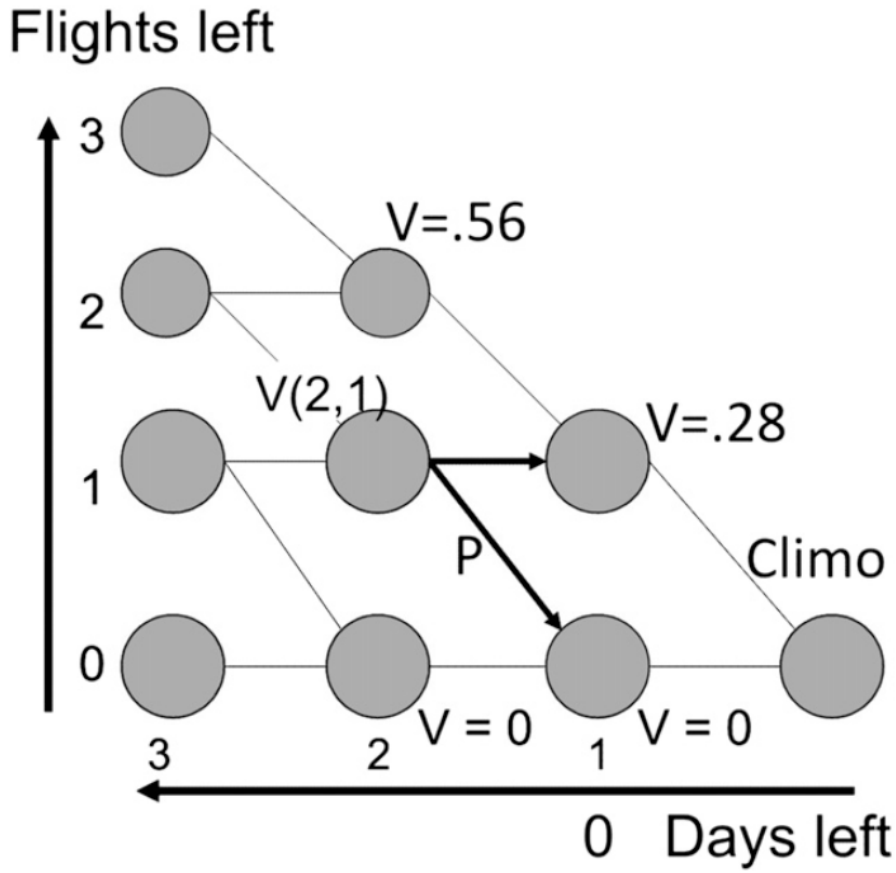


Figure 2: Graphical representation of the decision algorithm.

Probabilistic forecasting of favorable conditions using self-organizing maps

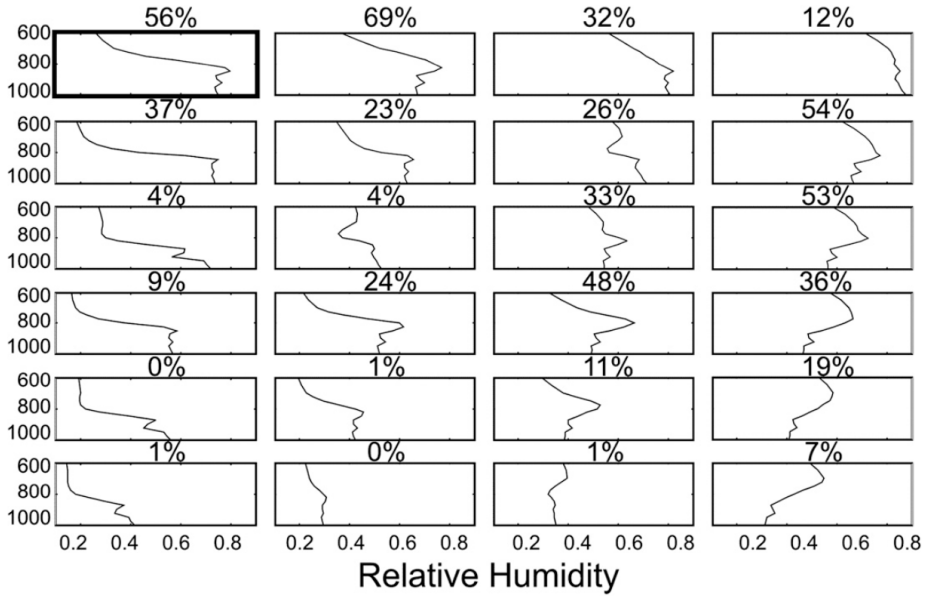


FIGURE 3: 6 X 4 SOM grid for relative humidity profiles

13.3%	20.6%	4.4%	3.7%
1.5%	8.1%	1.5%	4.4%
0%	3.7%	4.4%	11.8%
2.9%	3.7%	7.4%	6.6%
0%	0%	0.7%	1.5%
0%	0%	0%	0%

Figure 4: Conditional probability distribution of SOM state realizations following a forecast of SOM state 1

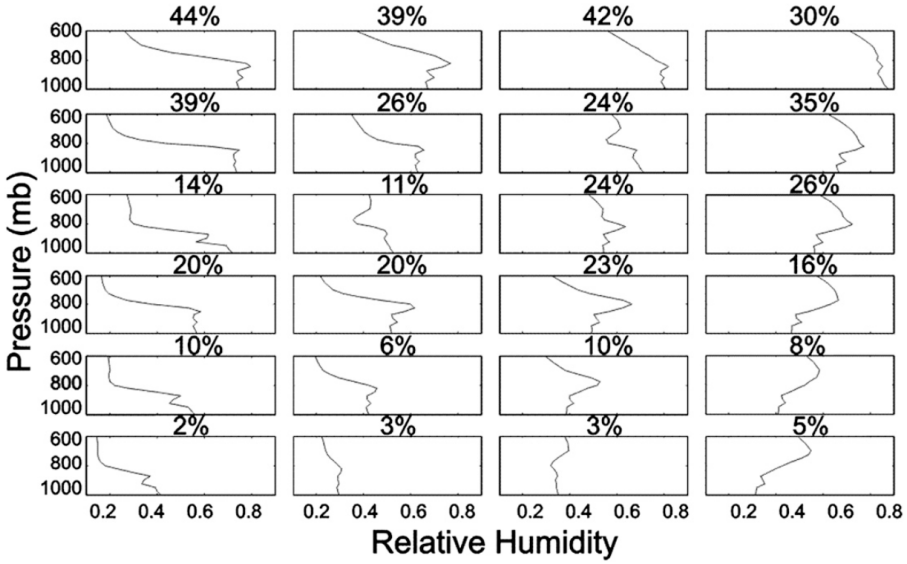


FIGURE 5: Estimated probabilities of good conditions for data collection, as a function of day-ahead SOM forecast

Results

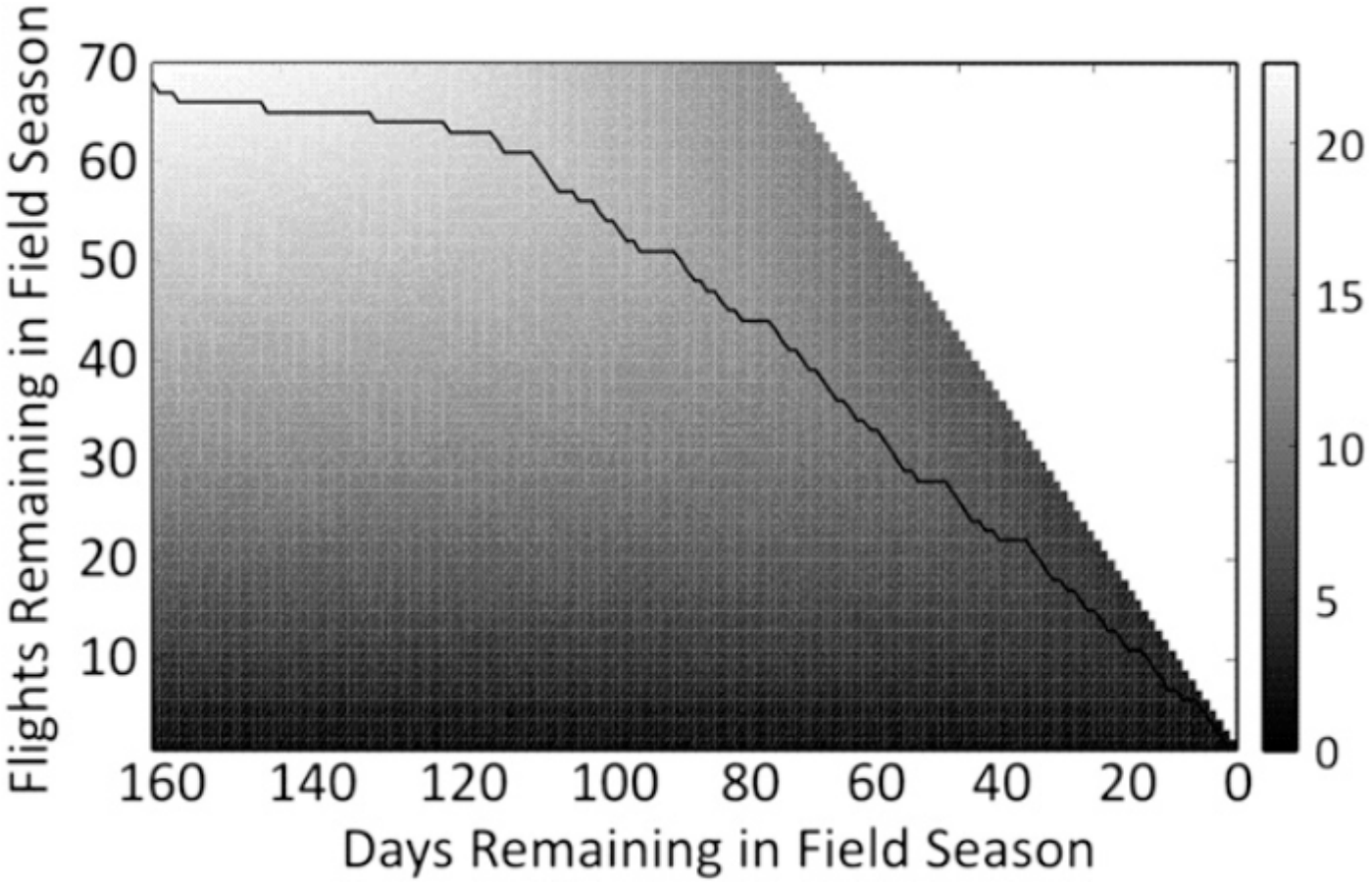


Figure 6: Computed values for the value function $V(d, f)$

Dark line : simulated sequence of flight decisions

- Diagonal movements = fly dates, horizontal movements = no-fly dates

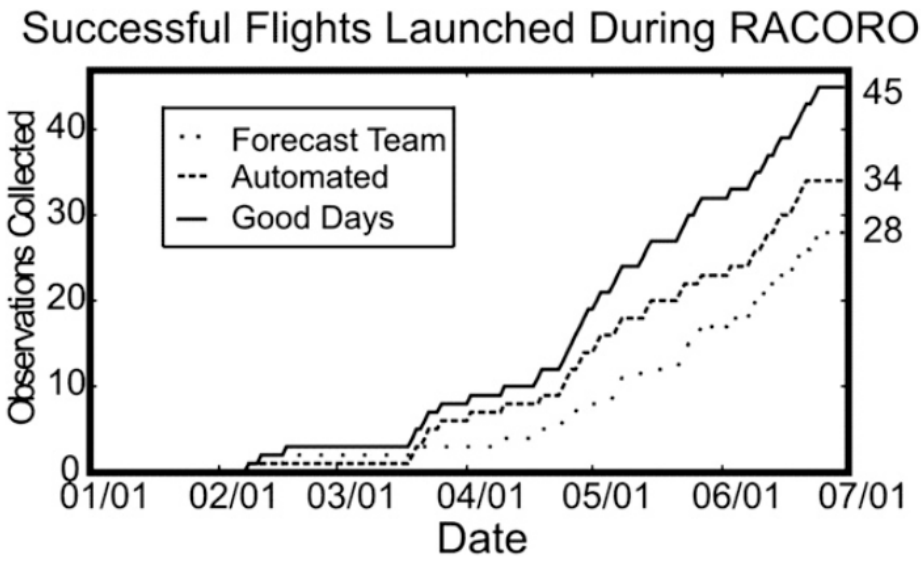


Figure 7: Results: the algorithm's simulated performance during 2009 field season, compared with realized performance of heuristic decision procedure.

The algorithm achieves a 21% increase in the number of successful flights.

	Heuristic procedure	Automated algorithm
Flights launched	56	66
Successes	28	34
Type I errors	28	32
Type II errors	17	11

Figure 8: Summary of outcomes

Successes are flights launched on days with good conditions.

Type I errors are decisions to fly only to find no clouds.

Type II errors are decisions to stand down only to find that the desired conditions existed.