# Predicting and Explaining Success and Task Duration in the Phoenix Planner

David M. Hart and Paul R. Cohen

Computer Science Technical Report 92-19

Experimental Knowledge Systems Laboratory
Department of Computer Science
University of Massachusetts
Amherst, Massachusetts 01003

**Abstract**

Phoenix is a multi-agent planning system that fights simulated forest fires. In this paper we describe an experiment with Phoenix in which we uncover factors that affect the planner's behavior and test predictions about the planner's robustness against variations in some of these factors. We also introduce a technique – path analysis – for constructing and testing causal explanations of the planner's behavior.

This research is supported by DARPA Contract F49620-89-C-00113; by an Intelligent Real-Time Problem Solving Initiative, Contract AFOSR-91-0067; by NSF under an Issues in Real-Time Computing grant, CDA-8922572; and by Texas Instruments Corporation.

## 1 INTRODUCTION

It is difficult to predict or even explain the behavior of any but the simplest AI programs. A program will solve one problem readily, but make a complete hash of an apparently similar problem. For example, our Phoenix planner, which fights simulated forest fires, will contain one fire in a matter of hours but fail to contain another under very similar conditions. We therefore hesitate to claim that the Phoenix planner "works." The claim would not be very informative, anyway: we would much rather be able to predict and explain Phoenix's behavior in a wide range of conditions (Cohen 1991). In this paper we describe an experiment with Phoenix in which we uncover factors that affect the planner's behavior and test predictions about the planner's robustness against variations in some factors. We also introduce a technique—path analysis—for constructing and testing causal explanations of the planner's behavior. Our results are specific to the Phoenix planner and will not necessarily generalize to other planners or environments, but our techniques are general and should enable others to derive comparable results for themselves.

In overview, Section 2 introduces the Phoenix planner; Section 3 describes an experiment in which we identify factors that probably influence the planner's behavior; and Section 4 discusses results and one sense in which the planner works "as designed." But these results leave much unexplained: although Section 4 identifies some factors that affect the success and the duration of fire-fighting episodes, it does not explain how these factors interact. Section 5 shows how correlations among the factors that affect behavior can be decomposed to test causal models that include these factors.

## 2 PHOENIX OVERVIEW

Phoenix is a multi-agent planning system that fights simulated forest-fires. The simulation uses terrain, elevation, and feature data from Yellowstone National Park and a model of fire spread from the National Wildlife Coordinating Group Fireline Handbook (National Wildlife Coordinating Group 1985). The spread of fires is influenced by wind and moisture conditions, changes in elevation and ground cover, and is impeded by natural and man-made boundaries such as rivers, roads, and fireline. The Fireline Handbook also prescribes many of the characteristics of our firefighting agents, such as rates of movement and effectiveness of various firefighting techniques. For example, the rate at which bulldozers dig fireline varies with the terrain. Phoenix is a real-time simulation environment—Phoenix agents must think and act as the fire spreads. Thus, if it takes too long to decide on a course of action, or if the environment changes while a decision is being made, a plan is likely to fail.

One Phoenix agent, the Fireboss, coordinates the firefighting activities of all field agents, such as bulldozers and watchtowers. The Fireboss is essentially a thinking agent,[1] using reports from field agents to form and maintain a global assessment of the world. Based on these reports (e.g., fire sightings, position updates, task progress), it selects and instantiates fire-fighting plans and directs field agents in the execution of plan subtasks.

A new fire is typically spotted by a watchtower, which reports observed fire size and location to the Fireboss. With this information, the Fireboss selects an appropriate fire-fighting plan from its plan library. Typically these plans dispatch bulldozer agents to the fire to dig fireline. An important first step in each of the three plans in the experiment described below is to decide where fireline should be dug. The Fireboss projects the spread of the fire based on prevailing weather conditions, then considers the number of available bulldozers and the proximity of natural boundaries. It projects a bounding polygon of fireline to be dug and assigns segments to bulldozers based on a periodically updated assessment of which segments will be reached by the spreading fire soonest. Because there are usually many more segments than bulldozers, each bulldozer digs multiple segments. The Fireboss assigns segments to bulldozers one at a time, then waits for each bulldozer to report that it has completed its segment before assigning another. This ensures that segment assignment incorporates the most up-to-date information about overall progress and changes in the prevailing conditions.

Once a plan is set into motion, any number of problems might arise that require the Fireboss's intervention. The types of problems and mechanisms for handling them are described in Howe & Cohen 1990, but one is of particular interest here: As bulldozers build fireline, the Fireboss

[1] Though it has the same architecture as other agents, it has few sensors or effectors and is immobile. For a detailed description of the Phoenix agent architecture and planning mechanisms see Cohen et al. 1989.

compares their progress to expected progress.[2] If their actual progress falls too far below expectations, a plan failure occurs, and (under the experiment scenario described here) a new plan is generated. The new plan uses the same bulldozers to fight the fire and exploits any fireline that has already been dug. We call this error recovery method *replanning*. Phoenix is built to be an adaptable planning system that can recover from plan failures (Howe & Cohen 1990). Although it has many failure-recovery methods, replanning is the focus of the experiment described in the next section.

# 3 IDENTIFYING THE FACTORS THAT AFFECT PERFORMANCE

We designed an experiment with two purposes. A *confirmatory* purpose was to test predictions that the planner's performance is sensitive to some environmental conditions but not others.[3] In particular, we expected performance to degrade when we change a fundamental relationship between the planner and its environment—the amount of time the planner is allowed to think relative to the rate at which the environment changes—and not be sensitive to common dynamics in the environment such as weather, and particularly, wind speed. We tested two specific predictions: 1) that performance would not degrade or would degrade gracefully as wind speed increased; and 2) that the planner would not be robust to changes in the Fireboss's thinking speed due to a bottleneck problem described below. An *exploratory* purpose of the experiment was to identify the factors in the Fireboss architecture and Phoenix environment that most affected the planner's behavior, leading to the causal model developed in Section 5.

The Fireboss must select plans, instantiate them, dispatch agents and monitor their progress, and respond to plan failures as the fire burns. The rate at which the Fireboss thinks is determined by a parameter called the *Real Time Knob*. By adjusting the Real Time Knob we allow more or less simulation time to elapse per unit CPU time, effectively adjusting the speed at which the Fireboss thinks relative to the rate at which the environment changes.

The Fireboss services bulldozer requests for assignments, providing each bulldozer with a task directive for each new fireline segment it builds. The Fireboss can become a bottleneck when the arrival rate of bulldozer task requests is high or when its thinking speed is slowed by adjusting the Real Time Knob. This bottleneck sometimes causes the overall digging rate to fall below that required to complete the fireline polygon before the fire reaches it, which causes replanning (see Section 2). In the worst case, a Fireboss bottleneck can cause a thrashing effect in which plan failures occur repeatedly because the Fireboss can't assign bulldozers during replanning fast enough to keep the overall digging rate at effective levels. We designed our experiment to explore the effects of this bottleneck on system performance and to confirm our prediction that performance would vary in proportion to the manipulation of thinking speed. Because the current design of the Fireboss is not sensitive to changes in thinking speed, we expect it to take longer to fight fires and to fail more often to contain them as thinking speed slows.

In contrast, we expect Phoenix to be able to fight fires at different wind speeds. It might take longer and sacrifice more area burned at high wind speeds, but we expect this effect to be proportional as wind speed increases and we expect Phoenix to succeed equally often at a range of wind speeds, since it was designed to do so.

## 3.1 EXPERIMENT DESIGN

We created a straightforward fire fighting scenario that controlled for many of the variables known to affect the planner's performance. In each trial, one fire of a known initial size was set at the same location (an area with no natural boundaries) at the same time (relative to the start of the simulation). Four bulldozers were used to fight it. The wind's speed and direction were set initially and not varied during the trial. Thus, in each trial, the Fireboss receives the same fire report, chooses a fire-fighting plan, and dispatches the bulldozers to implement it. A trial ends when the bulldozers have successfully surrounded the fire or after 120 hours without success.

The experiment's first dependent variable then is Success, which is true if the fire is contained, and false otherwise. A second dependent variable is shutdown time (SD), the time at which the trial was stopped. For successful trials,

[2] Expectations about progress are stored in *envelopes*. Envelopes represent the range of acceptable progress, given the knowledge used to construct the plan. If actual progress falls outside this range, envelope *violation* occurs, invoking error recovery mechanisms (Cohen, St. Amant & Hart 1992, Hart, Anderson & Cohen 1990).

[3] The term "planner" here refers collectively to all Phoenix agents, as distinct from the Fireboss agent.

shutdown time tells us how long it took to contain the fire.[4]

Two independent variables were wind speed (WS) and the setting of the Fireboss's Real Time Knob (RTK). A third variable, the first plan chosen by the Fireboss in a trial (FPLAN), varied randomly between trials. It was not expected to influence performance, but because it did, we treat it here as an independent variable.

WS: The settings of WS in the experiment were 3, 6, and 9 kilometers per hour. As wind speed increases, fire spreads more quickly in all directions, and most quickly downwind. The Fireboss compensates for higher values of wind speed by directing bulldozers to build fireline further from the fire.

RTK: The default setting of RTK for Phoenix agents allows them to execute 1 CPU second of Lisp code for every 5 minutes that elapses in the simulation. We varied the Fireboss's RTK setting in different trials (leaving the settings for all other agents at the default). We started at a ratio of 1 simulation-minute/cpu-second, a thinking speed 5 times as fast as the default, and varied the setting over values of 1, 3, 5, 7, 9, 11, and 15 simulation-minutes/cpu-second. These values range from 5 times the normal speed at a setting of 1 down to one-third the normal speed at 15. The values of RTK reported here are rescaled. The normal thinking speed (5) has been set to RTK=1, and the other settings are relative to normal. The scaled values (in order of increasing thinking speed ) are .33, .45, .56, .71, 1, 1.67, and 5. RTK was set at the start of each trial and held constant throughout.

FPLAN: The Fireboss randomly selects one of three plans as its first plan in each trial. The plans differ mainly in the way they project fire spread and decide where to dig fireline. SHELL is aggressive, assuming an optimistic combination of low fire spread and fast progress on the part of bulldozers. MODEL is conservative in its expectations, assuming a high rate of spread and a lower rate of progress. The third, MBIA, generally makes an assessment intermediate with respect to the others.[5]

When replanning is necessary, the Fireboss again chooses randomly from among the same three plans.[6]

We adopted a basic factorial design, systematically varying the values of WS and RTK. Because we had not anticipated a significant effect of FPLAN, we allowed it to vary randomly.

# 4 RESULTS FOR SUCCESS RATE AND SHUTDOWN TIME

We collected data for 343 trials, of which 215 succeeded and 128 failed, for an overall success rate of 63%. Tables 1a-c break down successes and failures for each setting of the independent variables RTK, WS, and FPLAN. Column S in these tables is the number of Successes, F is the number of Failures, and Tot is the total number of trials. Certain trends emerge in these data that confirm our earlier predictions. For example, in Table 1a, the success rate improves steadily as the thinking speed of the Fireboss increases. However, other patterns are less clear, such as the differences for each setting of WS in Table 1b. How do we know if these values are significantly different? For a *categorical* dependent variable such as Success (which has only two possible values), a chi-square test ($\chi^2$) will determine whether the observed pattern is statistically significant.

Figures 1a-c show the success rates for each setting of each independent variable. The table categories Success and Failure are broken down further into those trials which did not replan and those that did.

[4] Several other dependent variables were measured, notably Area Burned. However, using Area Burned to assess performance requires stricter experimental controls over such factors as choice of fire-fighting plan than were used here.

[5] The first plan of this variety developed in Phoenix was called Multiple-Bulldozer-Indirect-Attack, or MBIA, which signified a coordination of bulldozers working at some distance from the fire on fireline segments determined by the Fireboss's projections. SHELL is a variant of MBIA that builds a tighter shell of fireline, thus reducing the cost of forest burned. MODEL is another variant of MBIA that applies an analytical model of fire projection (Cohen 1990). It makes conservative projections at the default parameters used in this experiment.

[6] The same high-level plans can be used in the initial attack on a fire and on subsequent tries. When used in replanning, a plan is adapted to take advantage of any fireline that has already been dug near the fire. It is also based on updated conditions such as the current size and shape of the fire.

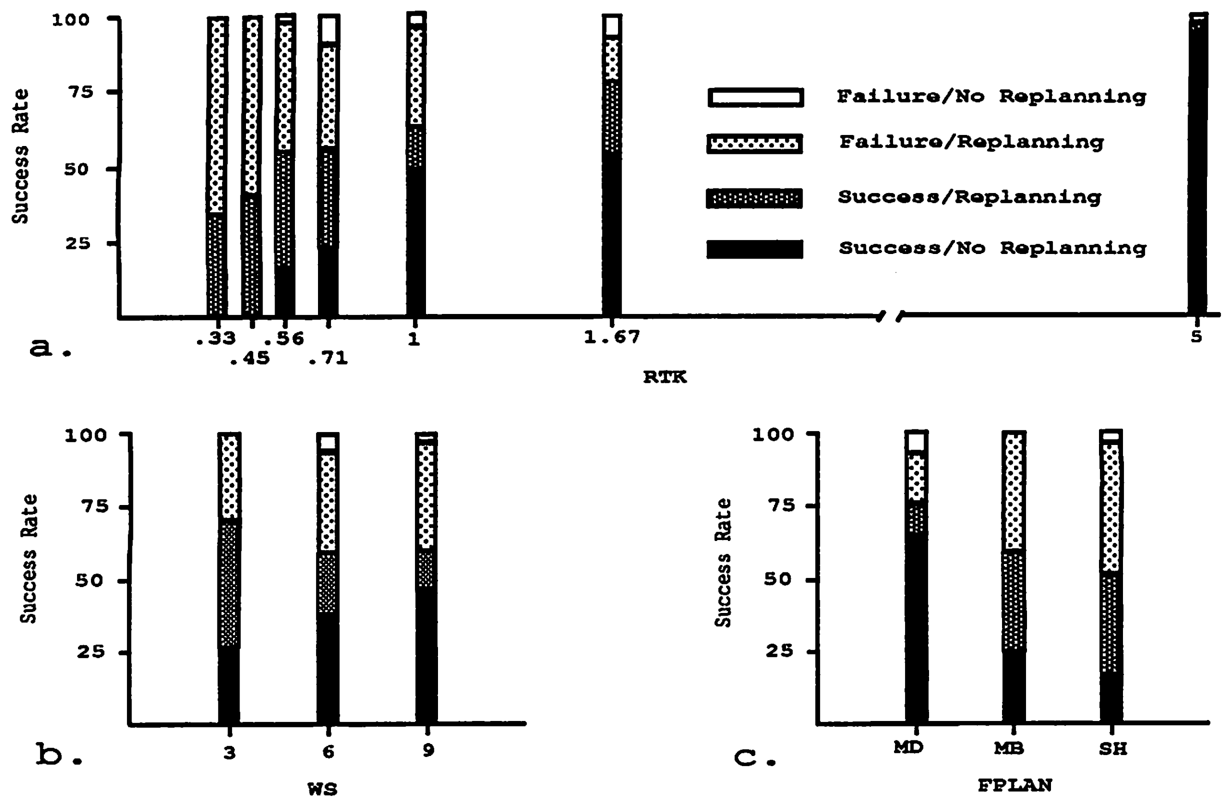


**Figure 1:** Successes by a) Real Time Knob, b) Wind Speed, and c) First Plan Tried

## 4.1 EFFECT OF INDEPENDENT VARIABLES ON SUCCESS

Table 1a shows successes by the independent variable RTK. A chi-square test on the Success-Failure x RTK contingency table in Table 1a is highly significant ($\chi^2(6) = 49.081$, $p < 0.001$), indicating that RTK strongly influences the relative frequency of successes and failures. At the fastest thinking speed for the Fireboss, RTK=5, the success rate is 98%, but at the slowest rate, RTK=.33, the success rate is only 33%. Figure 1a shows graphically that as RTK goes down (i.e., thinking speed decreases) the success rate declines. At RTK=1, the default setting, 63% of the trials were successful. Note how rapidly the *success of the initial plan* decreases—for RTK $\leq$ .45, no trial succeeds without replanning. However, the overall success rate declines more slowly as replanning is used to recover from the bottleneck effect described in Section 3. If we compare the rate of success without replanning to that with replanning in Figure 1a, we see that replanning buffers the Phoenix planner, allowing it to absorb the effect of changes in Fireboss RTK without failing. This effect is statistically highly significant.

**Table 1a:** Trials Partitioned by Real Time Knob.

| RTK | S | F | Tot |
|---|---|---|---|
| .33 | 10 | 20 | 30 |
| .45 | 14 | 19 | 33 |
| .56 | 22 | 18 | 40 |
| .71 | 54 | 42 | 96 |
| 1 | 27 | 16 | 43 |
| 1.67 | 38 | 11 | 49 |
| 5 | 50 | 2 | 52 |

Table 1b shows successes by wind speed. The small differences in success are marginal ($\chi^2(2) = 5.354$, $p < 0.069$), as we predicted in Section 3. Figure 1b shows a curious trend—as WS increases, the success rate for the first plan goes up, while the success rate in trials involving replanning diminishes. The increase in success rate for the first plan occurs because as WS increases, Phoenix overestimates the growth of the fire and plans a more conservative containing fireline.

**Table 1b:** Trials Partitioned by Wind Speed.

| WS | S | F | Tot |
|---|---|---|---|
| 3 | 85 | 35 | 120 |
| 6 | 67 | 50 | 117 |
| 9 | 63 | 43 | 106 |

Table 1c shows successes by first plan tried. Differences in success are highly significant ($\chi^2(2) = 16.183$, $p < 0.001$), which we had not expected when designing the experiment. As shown in Figure 1c, SHELL has a very low success rate without replanning, reflecting its aggressive character, while the conservative MODEL has an initial success rate of 65%. MBIA's initial success rate is slightly better than SHELL's (though the difference is not statistically significant).

**Table 1c: Trials Partitioned by First Plan Tried.**

| FPLAN | S | F | Tot |
|---|---|---|---|
| shell | 69 | 62 | 131 |
| mbia | 48 | 35 | 83 |
| model | 98 | 31 | 129 |

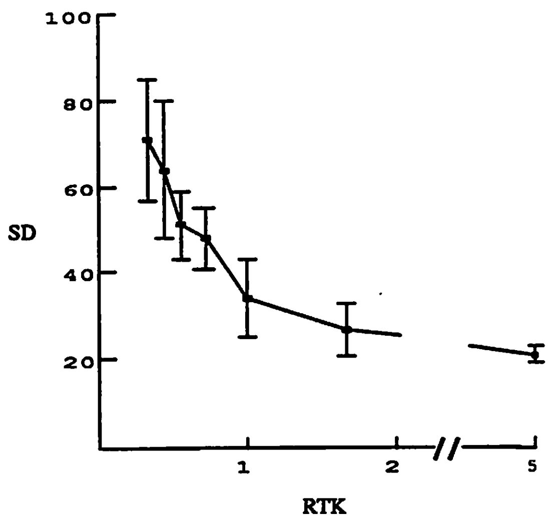


**Figure 2:** Mean Shutdown Time (in Hours) by Real Time Knob. Error Bars Show 95% Confidence Intervals.

### 4.2 EFFECT OF RTK ON SHUTDOWN TIME

Figure 2 shows the effect of RTK on the dependent variable Shutdown time (SD). The interesting aspect of this behavior is the transition at RTK=1. SD increases gradually between RTK=5 and 1, and the 95% confidence intervals around the mean values overlap. Below 1, however, the slope changes markedly and the confidence intervals are almost disjoint from those for values above 1. This shift in slope and value range for SD suggests a threshold effect in Phoenix as the Fireboss's thinking speed is reduced below the normal setting of RTK. The cost of resources in Phoenix is proportional to the time spent fighting fires, so a threshold effect such as this represents a significant discontinuity in the cost function for resources used. For this reason we pursued the cause(s) of this discontinuity by modeling the effects of the independent variables on several key endogenous variables,[7] and through them on SD, with the intent of building a causal model of the influences on SD.

## 5 INFLUENCE OF ENDOGENOUS VARIABLES ON SHUTDOWN TIME

We measured about 40 endogenous variables in the experiment described above, but three are of particular interest in this analysis: the amount of fireline built by the bulldozers (FB), the number of fire-fighting plans tried by the Fireboss for a given trial (#PLANS), and the overall utilization of the Fireboss's thinking resources (OVUT).

FB: The value of this variable is the amount of fireline actually built at the end of the trial. FB sets a lower limit on SD, because bulldozers have a maximum rate at which they can dig. Thus, when the Fireboss is thinking at the fastest speed and servicing bulldozers with little wait time, SD will be primarily determined by how much fireline must be built.

#PLANS: When a trial ran to completion without replanning, #PLANS was set to 1. Each time the Fireboss replanned, #PLANS was incremented. #PLANS is an important indicator of the level of difficulty the planner has fighting a particular fire. It also directly affects FB. As described in Section 2, replanning involves projecting a new polygon for the bulldozers to dig. Typically the new polygon is larger than the previous one, because the fire has now spread to a point where the old one is too close to the fire. Thus, the amount of fireline to be dug tends to increase with the number of replanning episodes.

OVUT: This variable, overall utilization, is the ratio of the time the Fireboss spends thinking to the total duration of a trial. Thinking activities include monitoring the environment and agents' activities, deciding where fireline should be dug, and coordinating agents' tasks (Cohen et al. 1989). The Fireboss is sometimes idle, having done everything on its agenda, and so it waits until a message arrives from a field agent or enough time passes that another action becomes eligible. We expected to see OVUT increase as RTK decreases; that is, as the Fireboss's thinking speed slows down, it requires a greater and greater proportion of the time available to do the cognitive work required by the scenario. Replanning only adds to the Fireboss's cognitive workload.

[7] A variable is called "endogenous" if it is influenced by independent variables and influences, perhaps indirectly through other endogenous variables, dependent variables.

### 5.1 REGRESSION ANALYSIS

Having identified these variables, we set about quantifying their effects using multiple regression.[8] We regressed SD on WS, RTK, FPLAN, OVUT, #PLANS and FB. These factors accounted for 76% of the variance in SD. Standardized beta coefficients are often cited as measures of the relative influence of factors; in Table 2a they tell us that FB has the largest influence on SD (beta = .759), with RTK and OVUT following close behind. But if the beta's represent the strength of influence, they are surprising. OVUT has a negative influence on SD, which is counterintuitive and appears to contradict the positive correlation (.42) between them in Table 2b. WS and #PLANS have virtually no influence on SD, even though #PLANS is strongly correlated with SD (.718). And although WS is essentially uncorrelated with SD (-.053), it is correlated with FB (.363), which in turn is strongly correlated with SD (.755). Finally, WS and RTK are correlated in Table 2b (.282), which seems impossible given that they were varied systematically. In short, the regression analysis and the correlation matrix contain counterintuitive entries. We will see this is because regression is based on an implicit model, one that almost certainly does not correspond to the structure of Phoenix.

Table 2a: Regression For Y: SD on X's: WS, RTK, FPLAN, OVUT, #PLANS, FB

| | B | Beta | t statistic of B |
|---|---|---|---|
| WS | -2.564 | -0.261 | -5.334 $p < .001$ |
| RTK | -8.057 | -0.580 | -6.503 $p < .001$ |
| FPLAN | .968 | .035 | .827 $p < .283$ |
| OVUT | -.347 | -.438 | -4.879 $p < .001$ |
| #PLANS | 3.411 | .115 | 1.742 $p < .088$ |
| FB | .002 | .759 | 11.641 $p < .001$ |

Table 2b: Correlation Coefficients

| | WS | RTK | FPLAN | OVUT | #PLNS | FB |
|---|---|---|---|---|---|---|
| WS | 1.000 | | | | | |
| RTK | .282 | 1.000 | | | | |
| FPLAN | .117 | .151 | 1.000 | | | |
| OVUT | -.257 | -.913 | -.016 | 1.000 | | |
| #PLNS | -.183 | -.409 | -.432 | .379 | 1.000 | |
| FB | .363 | -.249 | -.088 | .288 | .658 | 1.000 |
| SD | -.053 | -.484 | -.193 | .420 | .718 | .755 |

[8] Multiple regression builds a linear model of the effects of any number of X variables on a *continuous* variable Y, which in this case is SD. It fits a hyperplane to the data in an n-dimensional space using the least-squares method, where n = the number of X variables + 1. One of the measures produced by multiple regression is $R^2$, which is the percentage of variance accounted for by the linear model.

### 5.2 PATH ANALYSIS

A technique called *path analysis* (Asher 1983, Li 1975) lets us view correlation coefficients of the variables in Table 2b as sums of hypothesized influences among factors. Consider the surprising result that wind speed (WS) is essentially uncorrelated with shut-down time (SD). We expected WS to have two possible effects on SD:

Effect 1. If WS increases then the fire burns faster, and this means more fireline must be built (i.e., FB increases), which will take longer. Therefore increasing WS should increase SD.

Effect 2. For high wind speeds, if a fire isn't contained relatively quickly, then it might not be contained at all. For example, if a fire has been burning for 60 hours or more, and WS = 3, then the probability of the fire being eventually contained is .375. But if WS = 6, the probability of eventually containing an old fire is only .2, and if WS = 9, the probability drops to .13. We measured SD for successful trials only, because, by definition, an unsuccessful trial is one that exceeds a specified SD without containing the fires. But successful containment of old fires is relatively unlikely at higher wind speeds, so as WS increases, we see fewer older fires contained, thus fewer high values of SD. This leads us to expect a negative correlation between WS and SD. Note that this correlation represents an effect of missing data, not a true negative causal relationship between WS and SD.

Path analysis enables us to test a model in which the correlation $r_{WS\,SD}$ is composed of Effect 1 and Effect 2, which cancel each other out. Consider, for example, the path diagram in Figure 3. It shows WS positively influencing the amount of fireline that gets built (FB), and FB positively influencing SD (we will shortly describe how the numbers are derived). This path, WS→FB→SD, corresponds to Effect 1, above, and is called an *indirect* effect of WS on SD, mediated by FB. At the same time, WS *directly* and negatively influences SD on the path WS→SD, corresponding to Effect 2. Figure 3 shows the strength of WS→SD is -.377. The rules of path analysis dictate that the strength of WS→FB→SD is the product of the strengths of the constituent links, WS→FB and FB→SD, that is, (.363)(.892) = .328. The estimate of the correlation between WS and SD, $\hat{r}_{WS\,SD}$, is obtained by summing the direct and indirect effects, that is, .328 – .377 = -0.53. This is the sum of all legal ways for WS to influence SD given the structure in Figure 3. For the model in Figure 3, $\hat{r}_{WS\,SD} = r_{WS\,SD}$, but this doesn't happen in general.

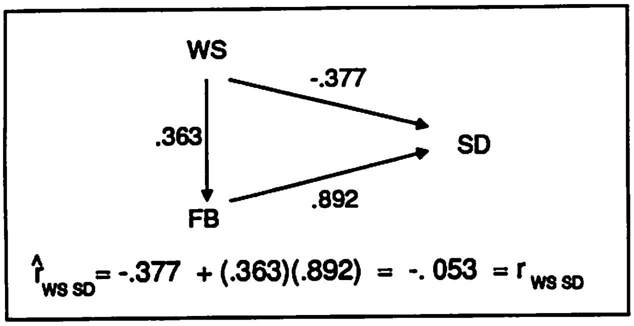


**Figure 3:** A Simple Path Diagram Showing Three Variables and Their Influences.

Thus we decompose the correlation $r_{WS\,SD}$ into two additive effects: WS increases FB as expected and decreases SD (spuriously, as noted above) as expected, and these effects cancel.

Path analysis involves three steps:

1) Propose a *path model* (such as the one in Figure 3). The model represents causal influences with directed arrows (e.g., FB→SD) and correlations with undirected links (see Figure 4a).
2) Derive *path coefficients* (such as -.377, .363 and .892). The magnitude of a path coefficient is interpreted as a measure of causal influence.
3) Estimate the strength of the relationship between two factors (such as WS and SD) by multiplying path coefficients along paths between the factors and summing the products over all legal paths between the factors.

Step 3 is entirely algorithmic given some simple rules (described below) that define legal paths. Step 2 involves some judgment because some models allow multiple ways to derive one or more path coefficients. A model is a concise statement of hypothesized causal influences among factors, and the space of models grows combinatorially with the number of factors, so step 1, proposing a model, is apt to benefit from knowledge about the system we are modeling.[9]

All three steps will be clearer if we briefly describe the relationship between multiple linear regression and path analysis. They are basically the same thing: both derive path coefficients for a model. The difference is simply that one particular model is implicit in multiple regression. Consider an elaboration of Figure 3, in which we add the RTK as an additional causal influence on SD. Figure 4a shows the implicit model fit by multiple regression, and Figure 4b shows a model that we think is a better representation of what is actually going on in Phoenix.

The regression model assumes that all predictor variables (WS, FB, RTK) are correlated, and assumes all directly influence the criterion variable (SD). Correlated variables are linked by undirected paths, which are labeled with the correlations. Table 2b presents the correlation matrix derived from our experiment. Multiple regression generates standard partial regression (beta) coefficients for each direct path between the predictor and criterion variables. These are -.291, .81 and -.2 in Figure 4a. Each represents a standardized measure of the influence of one predictor variable on the criterion variable with the effects of the other predictor variables held constant. The resulting regression equation in standard format is $\hat{SD} = .81\,FB - .29\,WS - .2\,RTK$. Because the regression coefficients are standardized they can be compared: a unit change in FB produces .81 units change in SD, whereas a unit change in WS produces -.29 units change in SD. FB is the stronger influence.

Figure 4a represents a decomposition of the correlations between SD and the other variables. The correlations can be reconstituted by summing the influences along paths just as we did in Figure 3. Path analysis has three rules for identifying paths:

1) No more than one undirected link can be part of a path (e.g., FB→RTK→SD is legal, but WS→FB→RTK→SD isn't)
2) A path cannot go through a node twice.
3) A path can go backward on a directed link, but not after it has gone forward on another link (e.g. FB←WS→SD in Figure 4b is legal but #PLANS→FB←WS in Figure 5 isn't).

The strength of each multilink path is just the product of its constituent coefficients, so the strength of the path FB→RTK→SD in Figure 4a is (-.249)(-.2) = .0498. The estimated correlation between a predictor and a criterion variable is the sum of the strengths of the paths that connect them. Thus

$$\begin{aligned}\hat{r}_{FB\,SD} = .755 = {} & .81 && \textit{direct FB→SD path}\\ & + (.363)(-.291) && \textit{FB→WS→SD}\\ & + (-.249)(-.2) && \textit{FB→RTK→SD}\end{aligned}$$

So multiple regression follows the three steps of path analysis. First, propose a model, specifically, a model in

---

[9] Pearl and Verma are developing efficient algorithms, related to path analysis, for causal induction (Pearl & Verma 1991).

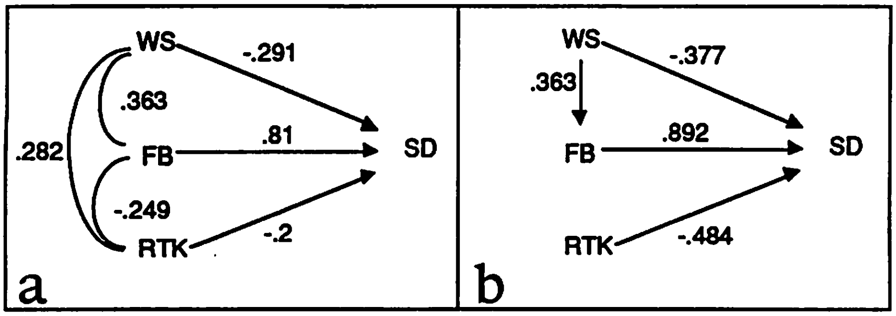


**Figure 4:** A Shows the Path Model Implicit in Multiple Regression. The Path Model in B Better Captures the Relationships Among These Variables in Phoenix.

which all predictor variables are correlated and directly linked to the criterion. Second, estimate path coefficients, specifically, calculate standard partial regression coefficients for the direct paths between predictor and criterion variables, and label the undirected links with the appropriate correlations. Third, estimate the correlations between each predictor and criterion variable by identifying legal paths between them, calculating the strength of each path, and summing the path strengths. In multiple regression, the estimated correlations are always identical to the actual correlations.

Multiple regression is a fine way to decompose correlations into their component influences *if you believe that multiple regression's implicit causal model represents your system.* Multiple regression is just path analysis on this implicit model, so if you don't believe the model you can propose another and run path analysis on it. This is what we did in Figure 4b. We know that WS and RTK are independent because our experiment varied them independently in a factorial design. (The reason they are correlated is the sampling bias identified as effect 2, above.) So we want to test a model in which WS influences SD directly and through FB, and RTK influences SD directly. The only question is how to estimate the path coefficients. The basic rules, which yield the coefficients in Figure 4b, are:

1) If W and X are uncorrelated causes of the criterion variable Y, then the path coefficients $\rho_{YX}$ and $\rho_{YW}$ are just the correlation coefficients $r_{YX}$ and $r_{YW}$, respectively.
2) If W and X are correlated causes of the criterion variable Y, then the path coefficients $\rho_{YX}$ and $\rho_{YW}$ are the standard partial regression coefficients $b'_{YX \cdot W}$ and $b'_{YW \cdot X}$, respectively, obtained from the regression of Y on X and W.

Is Figure 4b a better model than Figure 4a? We can answer the question in two ways. The statistical answer is that no model fits the data better, in terms of accounting for variance in the criterion variable, than the regression model. But this is hardly surprising when you consider that the regression model assumes everything influences everything else. The system analyst's answer is that we don't want models in which everything influences everything else: we want models in which some links are left out, in which causal influences are localized, not dissipated through a network of correlations. Let's ask, then, what it means for one such model to be better than another. Again, the judgment depends on how well each accounts for the variance in the criterion variable and how accurately each estimates the correlations between variables, and, how well each represents what we surmise to be the causal structure of our system. Clearly, these criteria interact. We can imagine a model that fits the data well but cannot represent what we know to be the causal structure, but often we explore different plausible causal structures by seeing how well each fits the data.

The structure in Figure 5 represents one of our first guesses at the causal structure that relates WS, FPLAN and RTK to SD. We expected WS and FPLAN to each directly influence both #PLANS and FB, but neither to directly influence SD. We also expected RTK to influence #PLANS and SD directly. We thought #PLANS might influence FB and SD. We made these guesses based on regression analyses, the correlation matrix in Table 2b, some of the graphs shown earlier, and our general knowledge about how the Phoenix planner works.

After estimating the path coefficients as shown in Figure 5, we estimated the correlations $\hat{r}_{SDi}$ between SD and each variable i. The estimates and the actual correlations are as follows:

| | WS | FPLAN | RTK | #PLANS | FB |
|---|---|---|---|---|---|
| $\hat{r}_{SDi}$ | .118 | -.197 | -.533 | .719 | .778 |
| $r_{SDi}$ | -.053 | -.193 | -.484 | .718 | .755 |

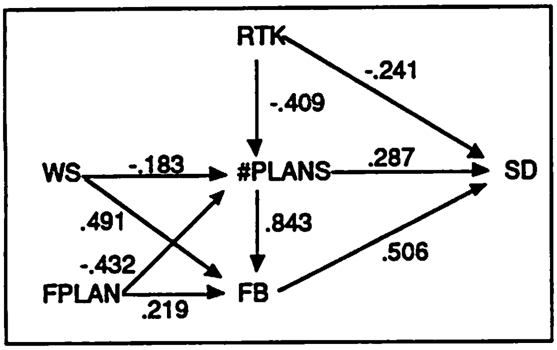


**Figure 5:** Path Model Relating Variables Influencing Shutdown Time.

Except for the disparity between the estimated and actual correlations between WS and SD, this model accounts pretty well for the actual correlations. At this point, we wanted to explain the influence of RTK on #PLANS. Why should decreasing RTK (slowing the Fireboss's thinking speed) increase the number of plans? One explanation is something like thrashing: There is always the possibility that the environment will change in such a way that a plan is no longer appropriate, but this is much more likely when the environment changes rapidly relative to planning effort (i.e., when RTK is decreased). Thus, decreasing RTK means the Fireboss will have to throw away plans before they make much progress, resulting in an increase in #PLANS. To test this we introduced another variable, OVUT, which measures the percentage of time in a trial that the Fireboss spends planning. We expected OVUT to decrease with RTK, supporting the thrashing explanation. Figure 6 shows a modification of Figure 5, with the path RTK→OVUT→#PLANS instead of RTK→#PLANS.

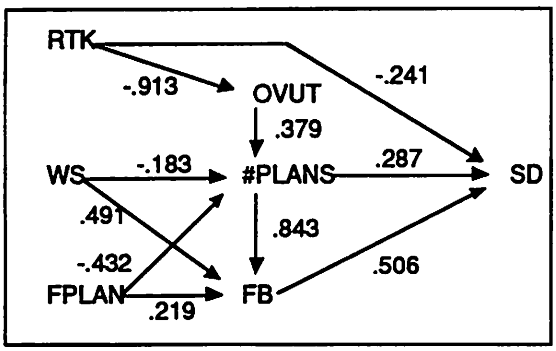


**Figure 6:** Adding the Endogenous Variable OVUT.

For this model, estimated correlations between SD and all the other variables are not appreciably different than they were for the model in Figure 5. But it appears that the variable OVUT does not add much to our understanding of thrashing, because it is completely determined by RTK. Consider what happens when we derive path coefficients for a slightly different model (Figure 7). In this case, OVUT has almost no influence ($r_{OVUT\#PLANS}$ = .032) on #PLANS. Recall, however, that this path coefficient is the standardized partial regression coefficient $b'_{OVUT\ \#PLANS \bullet RTK}$; that is, the effect of OVUT on #PLANS with RTK held constant. The fact that this number is nearly zero means that OVUT has no effect on #PLANS *when RTK is held constant*; in other words, the effect of OVUT on #PLANS is due entirely to RTK.

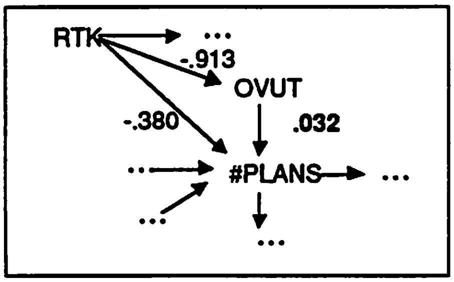


**Figure 7:** Showing the Effect of OVUT on #PLANS is Due Entirely to RTK.

# 6 CONCLUSION

We have presented results of an experiment with the Phoenix planner that confirm our predictions that its performance would be sensitive to some environmental conditions but not others. We have shown that the planner is not sensitive to variation in initial wind speed, a common environmental dynamic it faces. On the other hand, our results show that performance degrades as we change a fundamental relationship between the planner and its environment–the rate at which the Fireboss agent thinks. As we slowed the Fireboss's thinking speed in the experiment by decreasing RTK, performance degraded to the point where no plan succeeded on the first try. However, the planner was still able to succeed in many cases by replanning. While the success rate using replanning also degrades, replanning acts as a buffer, preventing the planner from failing catastrophically when it can't think fast enough to keep up with the environment. The data also show that replanning exerts a large influence on SD. We have presented a causal model, developed using path analysis, of the effects on SD of the various independent and endogenous variables we measured.

Replanning occurs when the environment doesn't match the Fireboss's expectations. In the current experiment, the rate at which the expectations became invalid was set by RTK. But the effect was indirect: Low RTK ensured that the Fireboss would be swamped (OVUT), which meant that bulldozers had to wait for instructions, which, in turn, increased the probability that they would not be able to carry out their instructions by their deadlines. This is what caused plans to fail. Environmental changes were only the instrument of the problem; RTK initiated it. But expectations, and thus plans, can also fail if the environment itself changes. We have yet to study whether replanning makes Phoenix robust against these changes, though our results with RTK suggest it does.

### Acknowledgements

This research is supported by DARPA under contract #F49620-89-C-00113; by AFOSR under the Intelligent Real-time Problem Solving Initiative, contract #AFOSR-91-0067; by NSF under an Issues in Real-Time Computing grant, #CDA-8922572; and by Texas Instruments Corporation. Thanks go to David Westbrook for invaluable design and programming help, Mike Sutherland for his statistical expertise, Scott Anderson for insightful comments on an early draft, and the many members of EKSL who have contributed to Phoenix. We also wish to thank an anonymous reviewer for a thorough and constructive reading. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

### References

Asher, H.B. 1983. *Causal Modeling*. Sage Publications.

Cohen, P.R., St. Amant, R. & Hart, D.M. 1992. Early warnings of plan failure, false positives, and envelopes: Experiments and a model. Dept of Computer Science, Technical Report #92-20, University of Massachusetts, Amherst.

Cohen, P.R. 1991. A survey of the Eighth National Conference on Artificial Intelligence: Pulling together or pulling apart? *AI Magazine* 12(1): 16-41.

Cohen, P.R. 1990. Designing and analyzing strategies for Phoenix from models. *Proc. of the Workshop on Innovative Approaches to Planning, Scheduling and Control*. Pp. 9-21. Morgan Kaufmann.

Cohen, P.R., Greenberg, M.L., Hart, D.M. & Howe, A.E. 1989. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine* 10(3): 32-48.

Hart, D.M., Anderson, S.D. & Cohen, P.R. 1990. Envelopes as a vehicle for improving the efficiency of plan execution. *Proc. of the Workshop on Innovative Approaches to Planning, Scheduling, and Control*. Pp. 71-76. Morgan Kaufmann.

Howe, A.E. & Cohen P.R., 1990. Responding to environmental change. *Proc. of the Workshop on Innovative Approaches to Planning, Scheduling, and Control*. Pp. 85-92. Morgan Kaufmann.

Li, C.C. 1975. *Path Analysis–A Primer*. Boxwood Press.

National Wildlife Coordinating Group, Boise, Idaho. *NWCG Fireline Handbook*. November, 1985.

Pearl, J. & Verma, T.S. 1991. A theory of inferred causation. *Principles of Knowledge Representation and Reasoning: Proc. of the Second International Conference*, J. Allen, R. Fikes, & E. Sandewall (Eds.). Morgan Kaufman. Pp. 441-452.