RC24539 (W0804-089) April 23, 2008
Mathematics

IBM Research Report

A Comparison of Bayesian Network and Goal Programming Approaches for National Analysis Module (Fire Program Analysis System - Phase 2)

Gyana R. Parija
IBM India Research Laboratory
4, Block - C, Institutional Area
Vasant Kunj, New Delhi - 110 070, India

Tarun Kumar, Yan Liu
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

Steve Carty
IBM Global Busines Services
6300 Diagonal Highway
Boulder, CO 80301-6108

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Fire Program Analysis System - Phase 2

FPA-2

A comparison of Bayesian Network and Goal Programming approaches for National Analysis Module

Technical Problem Definition

By

Gyana R Parija

Tarun Kumar

Yan Liu

Steven Carty

Table of Contents

1 Objective

As part of identifying the best approach to be used for National Analysis, we performed a detailed comparison of Bayesian Belief Network Modeling and Goal Programming approach. A reference problem was identified and then cast as a BBN model and Goal Programming. Results from the analysis were gathered and compiled into a set of recommendations. Since validation is a key to a successful project, we outline some ideas on validating the proposed approach.

2 Business Problem

To enable the broader team to relate to the outcome of this analysis, it was decided to keep the business problem relevant to fire domain. We have used two program components: Initial Attack and Fuels and two effectiveness-efficiency performance metrics: IA Success and WUI @Risk. Also, we have used GeNIe for building National Analysis Bayesian Network Model.

3 Bayesian Network (BN) Modeling Approach

Bayes networks are becoming an increasingly important tool for modeling and analyzing the data in many applications, such as inferring gene regulatory networks from the mRNA microarray data, deep space and knowledge acquisition in astronomic space by NASA, and social network study in social science.

3.1 National Analysis Reference Model: Bayesian Network Approach

3.1.1 Overview

Figure1 shows our initial design of the Bayesian networks

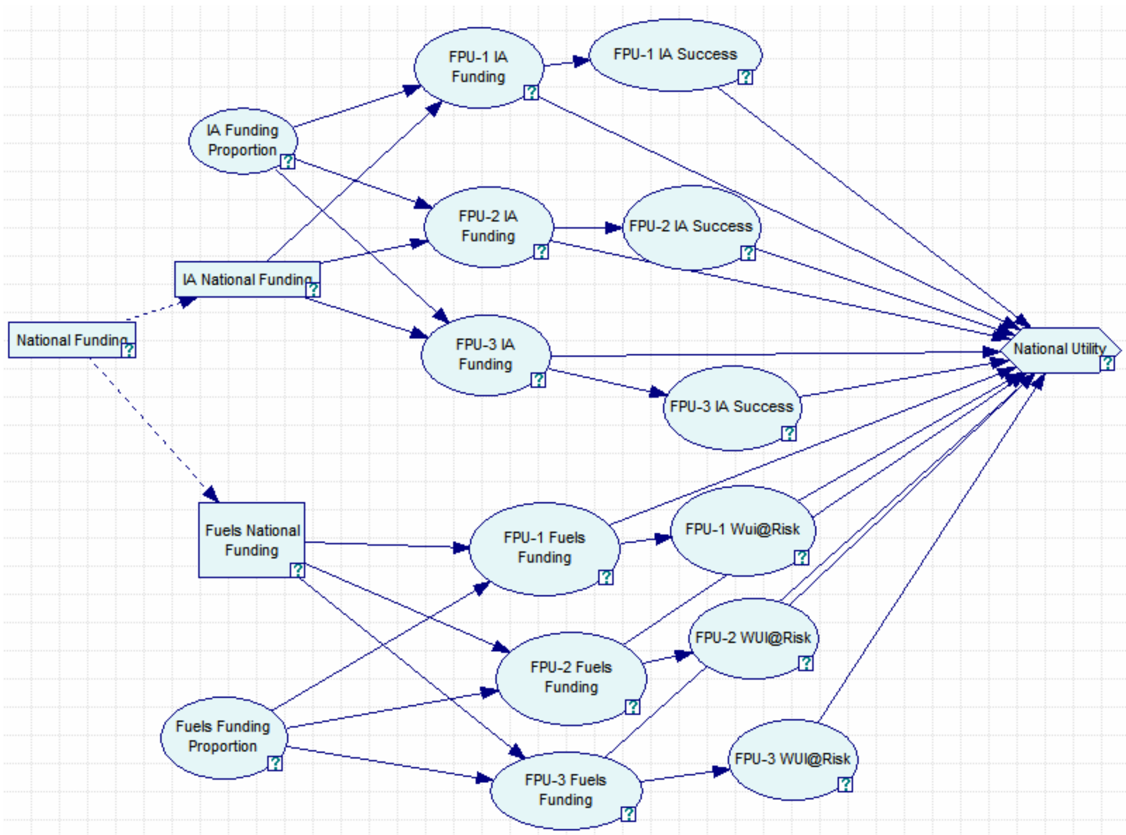


Figure1: Initial design of Bayesian networks for fire management

3.1.2 Descriptions of the model

To demonstrate the basic idea of our Bayesian network for the fire management task. We show a subgraph with only two FPU as in Figure2.

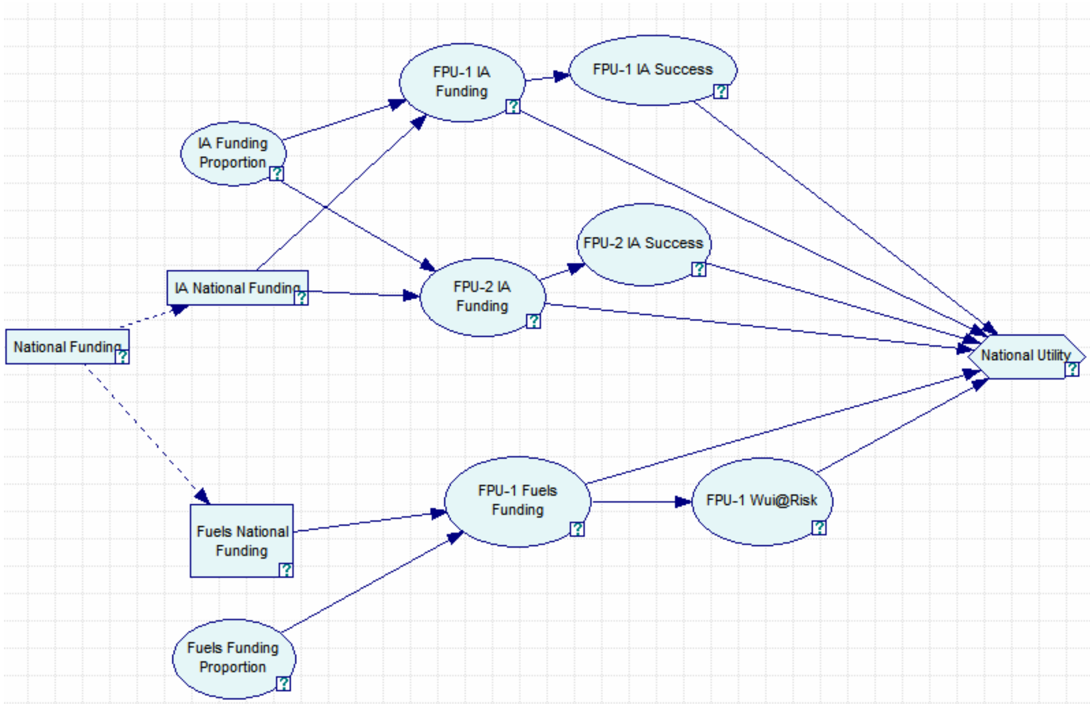


Figure2: A simplified example of Bayesian networks with two FPU

Nodes definition: In the example, we have 10 types of nodes in the graph, including:

National funding: the total budget for nation-wide fire management. Ideally, the variable should be continuous (value range as [0, budget limit]) denoting the amount of money for national fire management. In our model, it is a discrete variable with three possible values: high, medium, low

IA National funding: the total budget for nation-wide initial attack (IA) management. Ideally, the variable should be continuous (value range as [0, national funding value]). In our model, it is a discrete variable with two possible values: high, low

Fuel National funding : the total budget for nation-wide fuel management. Ideally, the variable should be continuous (value range as [0, national funding value]). Notice that the sum of Fuel National funding and IA funding cannot exceed the national funding. In our model, it is a discrete variable with two possible values: high, low.

IA funding proportion: the proportion of IA funding that can exceed the budget. Ideally, the variable should be continuous (value range as [0.95, 1.05]). In our model, it is a discrete variable with three possible values: between_95_100, eq_100, between_100_105.

Fuel funding proportion: the proportion of fuel funding that can exceed the budget. Ideally, the variable should be continous (value range as [0.95, 1.05]). In our model, it is a discrete variable with three possible values: between_95_100, eq_100, between_100_105.

FPU IA funding: the budget of IA funding for FPU1. Ideally, the variable should be continuous (value range [0, IA_Funding_Limit]). In our model, it is a discrete variable with two possible values: high and low

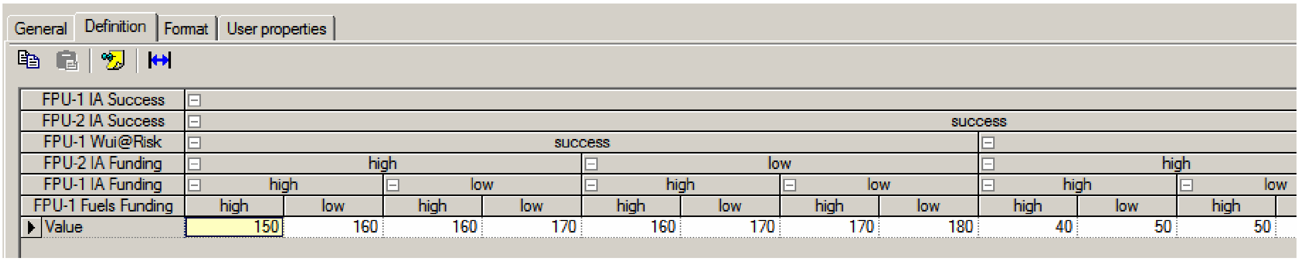
FPU fuel funding: the budget of IA funding for FPU1. Ideally, the variable should be continuous (value range [0, IA_Funding_Limit]). In our model, it is a discrete variable with two possible values: high and low

FPU IA success: whether the IA control is successful or not. It is a discrete variable with two possible values: 0 or 1

WUI @risk: whether the fuel control is successful or not. It is a discrete variable with two possible values: 0 or 1

National utility: the utility function that evaluates the effectiveness of fire management. It is a continuous value.

Edge Definition: most part of the graph in Figure 2 is self-explained except the one for “national utility”. There is an edge from the node “FPU IA funding” to “national utility” because the utility is not only determined by the IA success rate, but also has to be normalized by the funding that we input for IA control. For example, the value of the “national utility” should be high if the “IA funding” is low while the “IA success” is 1; on the other hand, the value of “national utility” should be medium if the IA funding is high and the IA success is 1. Figure 3 shows an example of the PDF we defined in the model



FPU-1 IA Success											
FPU-2 IA Success	success										
FPU-1 Wui@Risk	success										
FPU-2 IA Funding	high				low				high		
FPU-1 IA Funding	high		low		high		low		high		low
FPU-1 Fuels Funding	high	low	high	low	high	low	high	low	high	low	high
Value	150	160	160	170	160	170	170	180	40	50	50

Figure3: An example of PDF of National utility

3.1.3 Inference

In using GeNIe for developing Bayesian networks for decision support and for solving these models, we used the Bayesian networks inference algorithms, including the clustering and sampling algorithms. Specifically, we analyzed the National model using two algorithms: finding best policy and policy evaluation.

3.1.3.1 Find Best Policy

This influence diagram algorithm computes the optimal decision only. It does not compute the expected utility values. Using this algorithm in analyzing the National Analysis model provides us with the optimal value of the National Funding decision.

Among the three decision options of National Funding viz: Low / Medium / High, the algorithm computes the optimal value as “High”. This is the only outcome of the model.

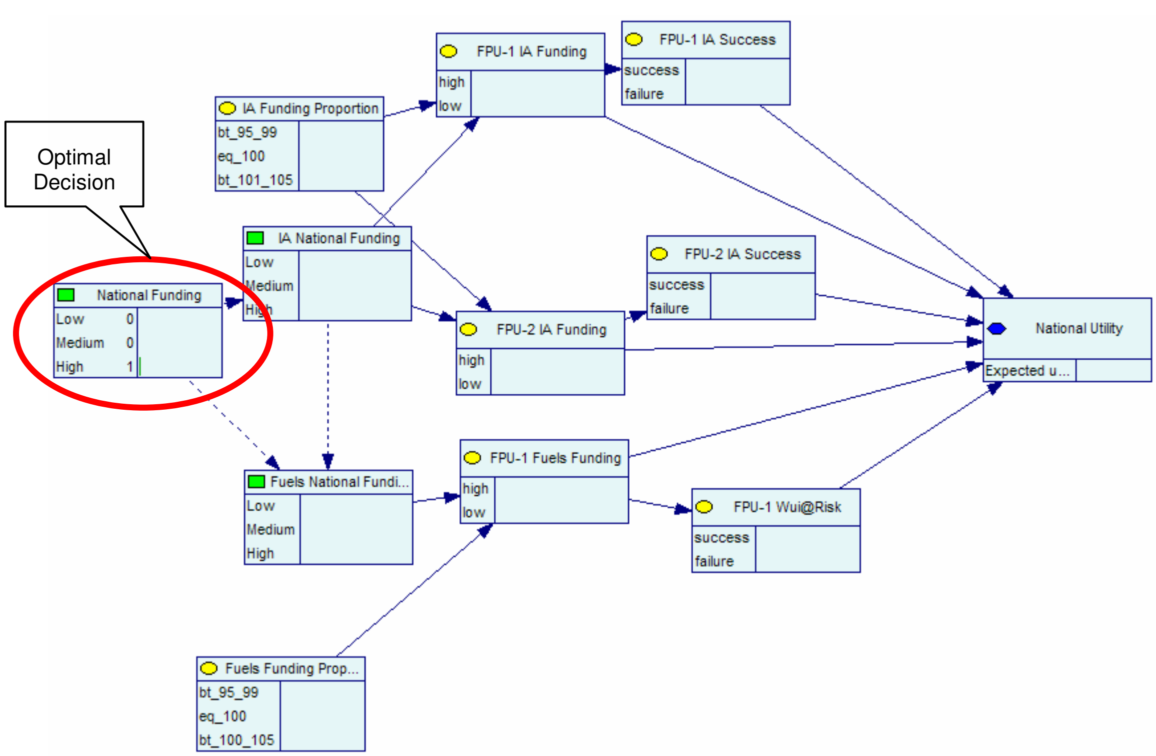


Figure4: Using Find Best Policy algorithm to analyze National Model

3.1.3.2 Policy Evaluation

This algorithm computes the expected utility of each of the decision alternatives. It does this by performing repeated inferences in the network.

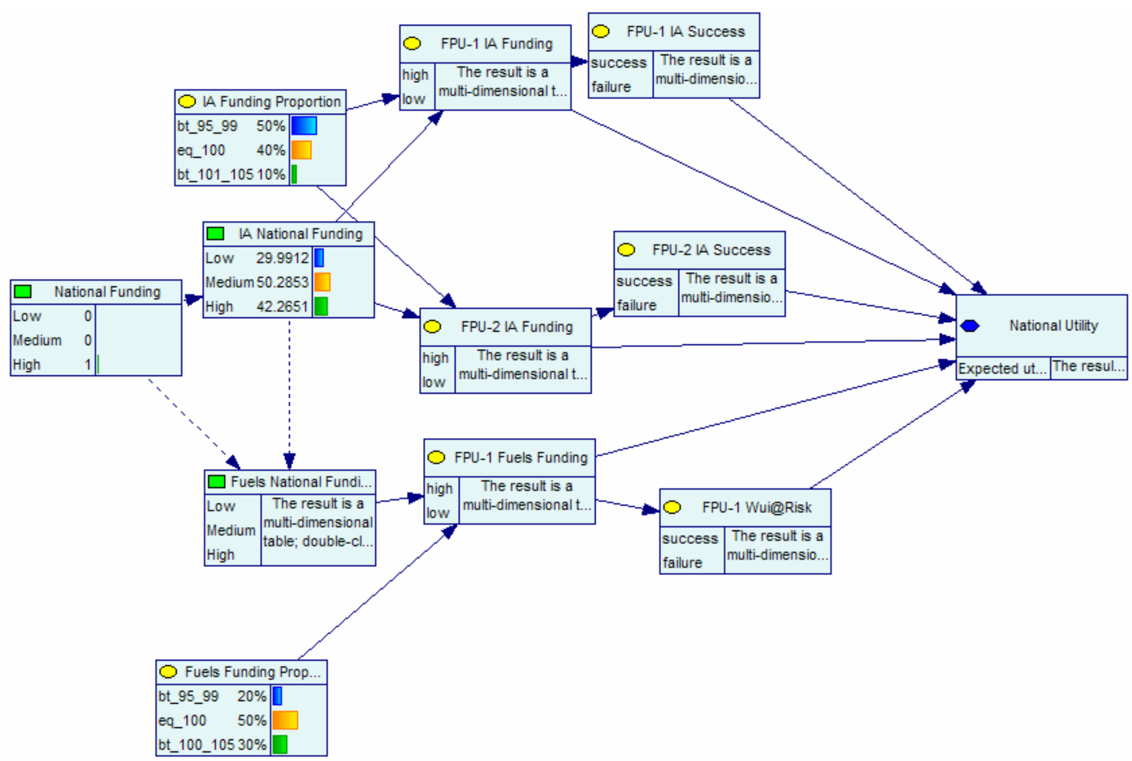


Figure5: Using Policy Evaluation algorithm to analyze National Model

As a result, it computes the expected utility values for the National Analysis Model corresponding to all combinations of IA National Funding and Fuels National Funding.

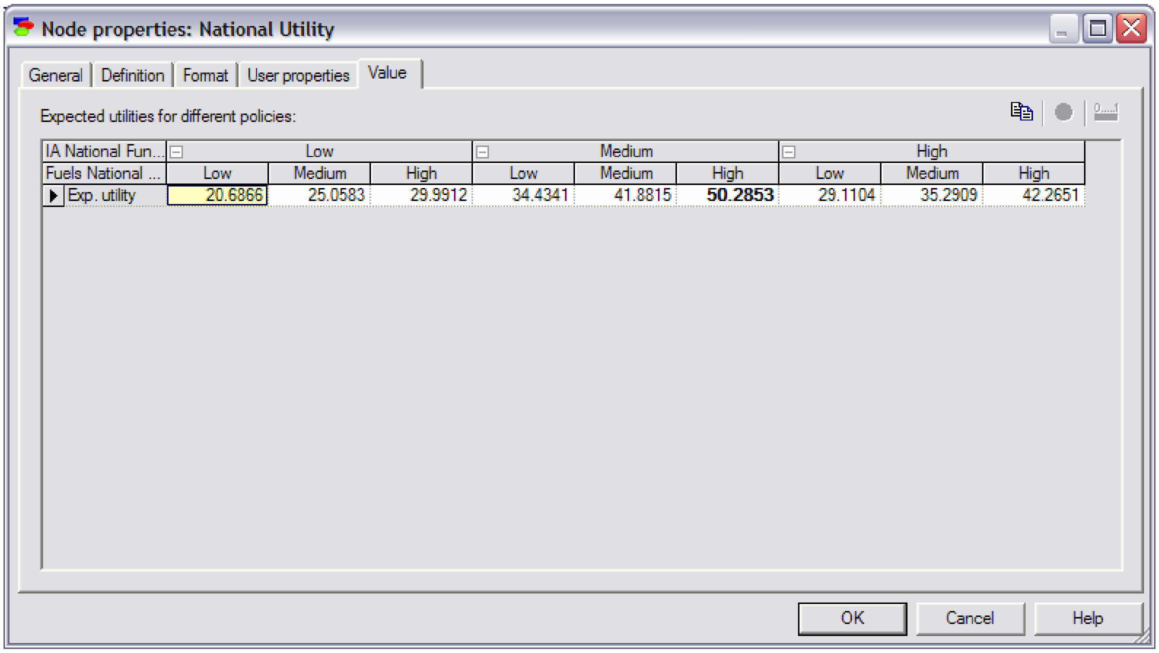


Figure6: Expected Utility values for National Analysis Model

3.1.4 An Simple Example of Discretized Model

In the initial design of national analysis model described above, we make one simplification about the funding value by using discrete values as “high” or “low”. In the real applications, we need the funding assessment to be at much finer granularity so that the results of the analysis model are meaningful and useful. A straightforward solution to the problem is setting all the nodes involving the funding values as continuous variable. However, this is well beyond the capability of the Bayesian network models, which are designed to only handle the discrete variables.

To solve the problem, we design a discretized version of the desired model with less complication. Figure 7 shows a simple example, in which there are only two nodes, “funding” and “success rate”. The possible value of the “funding” node is 1, 2, 3…, and 100, representing the amount of funding as $1M, $2M, …, and etc. The possible value of “success rate” is binary, i.e. 1 or 0. To complete the definition, we need to define the probability distribution P(Funding) and P(Success Rate| Funding). In this example, we set the distribution as follows:

P(Funding) ~ Poisson(60), P(Success Rate|Funding) ~ Burnolli(min(Funding/75, 1)). In the real applications, we can estimate the probability density from the data.

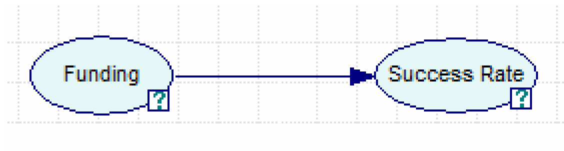


Figure7: A simple example of discretized model

With the definition above, we are able to make some simple inference over the model. For example, given the success rate as 1, we can answer the question of what is the probability that the Funding amount is $54M (see Figure 8).

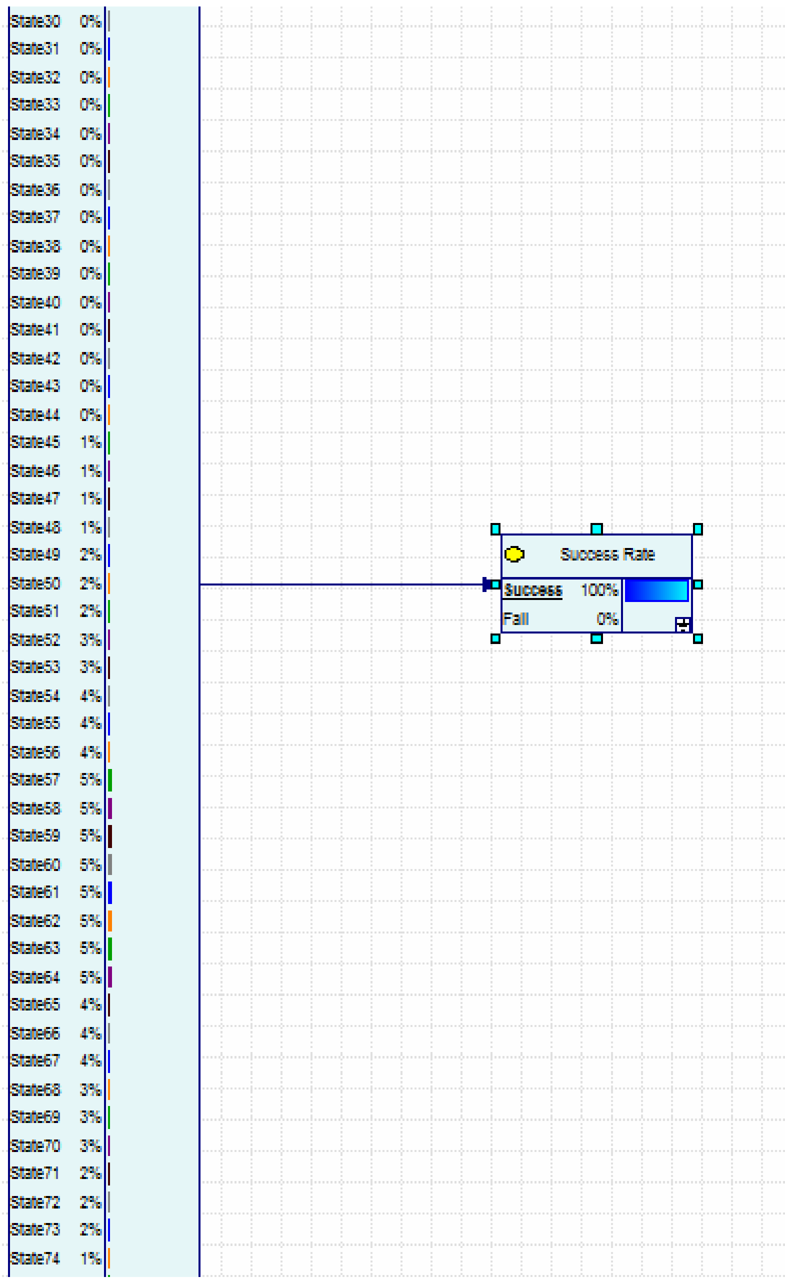


Figure8: Inference on the discretized model: set the evidence of “success rate” as 1, infer the value of funding

4 Transforming the BBN Outputs for Input to Goal Program

Fpu level Bayesian Network analysis provides as output at each budget level, the (discrete) probability distributions for the EEPs. At a given National Funding level f, f_{klj} represents the funding for at jth \$ level, lth program component, for kth fpu corresponding to the program effectiveness measures (EEPs) S_{kl} .

The table below describes the output probability distribution data for one Fpu obtained from the Fpu level Bayesian network model. p_{klji} represents the probability value for kth Fpu, jth funding level, lth program component, and ith state.

Probabilities of S_{kl} states →	S_{kl1}	…	S_{kli}	…	S_{klm}
f_{klj} levels ⬇					
f_{kl1}	p_{kl11}				p_{kl1m}
f_{k2}	p_{kl21}				p_{kl2m}
…	…				
…					
f_{klj}			p_{klji}		
…					
…					
…					
…					
$f_{k\ln}$					$p_{k\ln m}$

Figure9: Output probability distribution data from Fpu level Bayesian Network Model

Two Fpu National Model Example

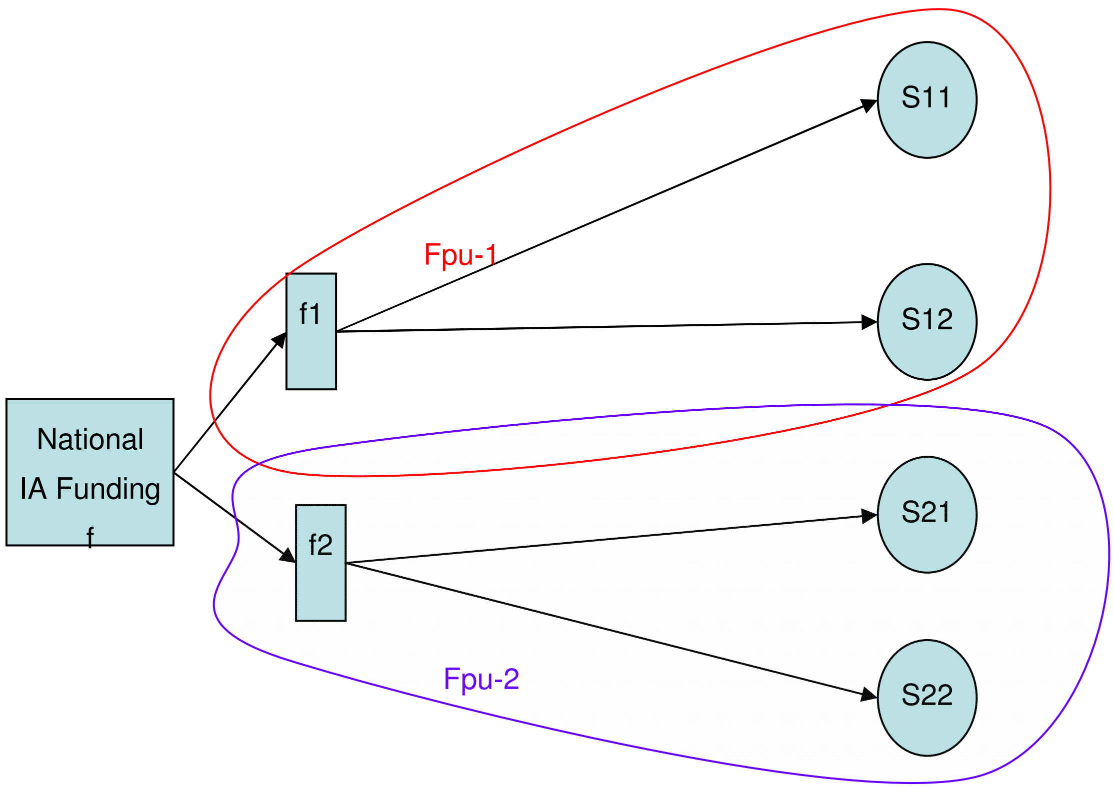


Figure10: A two Fpu National Model representation

Here, we are interested in the following probability distribution in a parametric form from the output of the Bayesian Network:

$$p_{kli}(f) = P\{S_{kl} = s_i \mid f_{kl} = f\} = (a_{kli} \times f + b_{kli}), \forall i$$

$$\sum_i p_{kli}(f) = 1, \forall f$$

To obtain such a distribution computationally, we propose a mathematical programming formulation as follows:

Decision variables

a_{kli} First coefficient for the linear term of the linear expression

b_{kli} Second coefficient for the constant term of the linear expression

z_{klji} Residual variable for funding level j

Parameters

p_{klji} Probability of success at funding level j

f_{klj} $ value at funding level j

j Summation index over the funding levels {1,..n}

l Summation index over the states {1…m}

The regression problem for each FPU-k, and Program Component-l, may be defined as:

$$Min \sum_{i,j=1}^{m,n} z_{klji} \qquad (1)$$

Subject to:

$$z_{klji} \geq p_{klji} - (a_{kli} \times f_{klj} + b_{kli}) \qquad \forall i, j \qquad (2)$$

$$z_{klji} \leq (a_{kli} \times f_{klj} + b_{kli}) - p_{klij} \qquad \forall i, j \qquad (3)$$

$$(a_{kli} \times f_{klj} + b_{kli}) \geq 0, \forall i, j \qquad \forall i, j \qquad (4)$$

$$\sum_{i=1}^{m} (a_{kli} \times f_{klj} + b_{kli}) = 1 \qquad \forall j \qquad (5)$$

Formulation Description

The formulation described above seeks to find the best possible linear fit. To achieve this, the objective term in (1) minimizes the sum of absolute deviations. Constraints (2) and (3) linearize the absolute deviation term: $z_{klji} = \left| p_{klji} - (a_{kli} \times f_{klj} + b_{kli}) \right|$. Constraint (4) ensures that the probability values are non-negative, and constraint (5) enforces that at a given funding level, the sum of the probabilities over the state space is 1. Also, it is worth noting here that one may alternatively set up the regression problem by using a convex quadratic obejctive (square of the deviations) as an appropriate distance metric.

The data needed for solving such a math program instance would come from the underlying FPU-level BN output data table with (m x n) probability data entries.

5 Goal Programming (GP) Modeling Approach

Goal Programming is an important area of multiple criteria optimization. The idea of goal programming is to establish a goal level of achievement for each criterion. GP is ideal for criteria with respect to which target (or threshold) values of achievement are of significance. GP involves conceptualization of objectives as Goals. Each objective is assigned weights and / or priorities for achievement of goals. The presence of deviational variables d_i^+ and d_i^- measure overachievement and underachievement from target (or threshold) levels t_i . The objective involves minimization of weighted-sums of deviational variables to find solutions that best satisfy the goals. Usually, a point that satisfies all the goals is not feasible. Thus, we try to find a feasible point that achieves the goal "as closely as possible". The way in which such points are found using priority and / or weighting structures defines goal programming.

Goal Programming methodology deals with multiple objectives (goals) in two different ways, namely Preemptive Goal Programming, and Non-preemptive Goal Programming. The next section illustrates the difference between these two ways of applying Goal Programming methodology from the perspectives of varying levels of complexity and richness of solution.

5.1 Preemptive vs Non-preemptive Goal Programming

In goal programming there are two basic models: Non-preemptive model (Archimedean model) and the preemptive model. To define each approach we first define some basic terminology

5.1.1 Non-Preemptive Goal Programming

In non-preemptive goal programming, we generate candidate solutions by computing points in S whose criterion vectors are closest, in weighted sense, to the utopian set in criterion space. Consider the GP

$$goal\{c^1 x = z_1\} \qquad z_1 \geq t_1$$

$$goal\{c^2 x = z_2\} \qquad z_2 = t_2$$

$$goal\{c^3 x = z_3\} \qquad z_3 \in [t_3^l, t_3^u]$$

$$s.t. \quad x \in S$$

The non-preemptive formulation of the above GP is

$$\min[w_1^- d_1^- + w_2^+ d_2^+ + w_2^- d_2^- + w_3^- d_3^- + w_3^+ d_3^+]$$

$$\text{s.t.} \qquad c^1 x + d_1^- \geq t_1$$

$$c^2 x - d_2^+ + d_2^- = t_2$$

$$c^3 x + d_3^- \geq t_3^l$$

$$c^3 x - d_3^+ \leq t_3^\mu$$

all $d's \geq 0$

We observe that:

1. The w's in the objective function are positive penalty weights.
2. Each goal gives rise to a goal constraint, except range goals that give rise to two.
3. Only deviational variables associated with undesirable deviations need be employed in the formulation.
4. The non-preemptive objective function is a weighted-sum of the undesirable deviational variables.
5. Non-preemptive GP's can be solved using conventional LP software.

The goal constraints are soft constraints in that they do not restrict the original feasible region S. In effect, they augment the feasible region by casting S into higher dimensional space, thereby creating the augmented GP feasible region.

The w's allow us to penalize undesirable deviations from goal with different degrees of severity. In fact, more elaborate means of penalizing deviations, from goal can be employed using piecewise linear programming.

5.1.2 Preemptive Goal Programming

In preemptive goal programming, the goals are grouped according to priorities. The goals at the highest priority level are considered to be infinitely more important that goals at the second priority level, and the goals at the second priority level are considered to be infinitely more important than goals at the third priority level, and so forth. Let us consider a preemptive GP:

$$goal\{c^1 x = z_1\} \qquad z_1 \leq t_1$$

$$goal\{c^2 x = z_2\} \qquad z_2 \geq t_2$$

$$goal\{c^3 x = z_3\} \qquad z_3 = t_3$$

s.t. $x \in S$

We define the preemptive GP in the following lexicographic format:

$$lex \quad \min\{d_1^+, d_2^-(d_3^+ + d_3^-)\}$$

$$c^1 x - d_1^- \leq t_1$$

$$c^2 x + d_2^- \geq t_2$$

$$c^2 x - d_3^+ + d_3^- \geq t_3$$

$x \in S$

all $d's \geq 0$

To solve the above *lex min* problem using conventional LP software, as many as three optimization stages may be required. In the first stage, we solve

$$\min\{d_1^+\}$$

$$\text{s.t. } c^1 x - d_1^- \leq t_1$$

$$x \in S$$

$$d_1^+ \geq 0$$

If this problem has alternative optima, we form and then solve the second stage problem

$$\min\{d_2^-\}$$

$$\text{s.t. } c^1 x \leq t_1 + (d_1^+)^*$$

$$c^2 x + d_2^- \leq t_2$$

$$x \in S$$

$$d_2^- \geq 0$$

where $(d_1^+)^*$ is the optimal value of d_1^+ from stage one. If the second-stage problem has alternative optima, we form and then solve the third stage problem

$$\min\{(d_3^+ + d_3^-)\}$$

$$\text{s.t. } c^1 x \leq t_1 + (d_1^+)^*$$

$$c^2 x \geq t_2 - (d_2^-)^*$$

$$c^2 x - d_3^+ + d_3^- \geq t_3$$

$$x \in S$$

$$d_3^+, d_3^- \geq 0$$

where $(d_2^-)^*$ is the optimal value of d_2^- from stage two. Any solution to the third stage lexicographically minimizes the preemptive GP.

In general, we may not have to solve as many optimization stages as there are priority levels. We can cease our progression through the optimization stages as the optimization stage is encountered that has a unique solution. Thus, a less than desirable consequence of th preemptive approach is that lower order goals may not get a chance to influence the GP generated solution.

5.1.3 Sensitivity Issues

In practice, goal programs typically have large numbers of deviational variables. With possibly different units of measure, assigning weights is often very difficult. Consequently, in non-preemptive GP, we initially try to apply a reasonable set of penalty weights with plans to do sensitivity experiments with other sets of weights to see if better solutions can be located. Many times the strategy works satisfactorily. Many times it does not. When it does not, this is what might happen. The initial set of weights produces an unsatisfactory solution. Then, some of the weights are shifted and the problem resolved. Sometimes, the new solution will be the same as the old because both sets of weights pertain to the same augmented GP flexible region vertex. Sometimes, even though we may fell that we have made intelligent changes in the weights, a poorer solution results. Sometimes, despite the fact that we may have made only slight changes in the weights, a drastically different solution results because we may have jumped to a vertex on the opposite side of a large goal-efficient facet. The net effect is that often a user will come away from a non-preemptive GP experience feeling frustrated about this or her ability to control the movement of the solution as the weights are varied.

In preemptive GP, users frequently address sensitivity concerns by rotating priorities. If there are r priority levels, there are r! different ways of rotating the priorities. Normally, a user will select a small number of these possibilities and the resolve the problem for each of them. The result is usually a group of some of the most different goal-efficient points in S.

5.2 Goal Programming formulation of National Analysis Model

In FPA, the **objectives** would include the EEPs. **Decision variables** would be fire program component **funding levels** for each FPU. **Constraints** would include various **cost limits (budget appropriation)**, and other **interagency program guidelines** to be followed.

We propose the technical problem underlying the national analysis to be multi-criteria (e.g. multiple EEP's but with a known priority ordering) decision problem defined on BN representations of the cause-and-effect relationships available for the FPU's in which:

- There are multiple program components such as IA, Fuels, Prevention etc competing for available funding at the national level $(c \in C)$
- There are multiple effectiveness-efficiency-performance metrics (e.g. IA success, WUI@Risk etc) that need to be managed in an equitable manner $(m \in M)$, with a target T_m, overachievement R_m and underachievement P_m
- There are finitely many states $(s \in S_m)$ available for each effectiveness metric considered
- The conditional probability distributions, $F^m(\{x_c \in C\})$ (parameterized on the funding amount allocated to each program component, x_c) of the state random variables at chance nodes are known *a priori* through prior detailed FPU level analysis – *this is an important assumption as these parameterized probability distributions help model various effectiveness metrics dynamically*

to address two key features of fire management: risk mitigation and contingency planning

The technical problem then is defined as:

For a given priority ordering for

$(s \in S_m) EEP's \{EEP^{(1)} > EEP^2 > \ldots > EEP^k\}, k \in M_m = \{1,2,\ldots\}, find(d_{kf}^{+*}, d_{kf}^{+*})$ by solving the k-th iteration preemptive goal program:

Minimize (weighted deviation of k-th EEP): $D_k = \sum_{f \in FPU} (R_{kf} d_{kf}^+ + P_{kf} d_{kf}^-)$

<u>Subject to:</u>

(goals): $EEP(F^m(\{x_{cf}, c \in C\}, r(x_{cf})) + d_{mf}^+ - d_{mf}^- \geq T_{mf} \quad \forall m \in M, f \in FPU$

(budget appropriation): $\sum_{cf} x_{cf} \leq B$

(interagency guidelines): …

(preemptions): $\sum_{f \in FPU} (R_{kf} d_{kf}^+ + P_{kf} d_{kf}^-) = D_k^* \quad \forall m \in \{1,2,\ldots,k-1\}, f \in FPU$

$D_k = \sum_{f \in FPU} (R_{kf} d_{kf}^+ + P_{kf} d_{kf}^-)$

After all |M| iterations are completed, the resulting solution $(\{x_{cf}, c \in C\}, r(x_{cf})$, $\forall f \in FPU$ is a complete listing of the funding allocations across all program components for each FPU and the associated national fire resource organization at the requested national budget level.

Note that this solution is a result of the priority order input to the goal program. By enumerating other dominant priority orders for the EEP's (these enumerations will be small in number (120 for 5 EEPs), and solving the GP for each priority order, one can compute the solution "basis" for further analysis.

5.3 National Analysis Model: Goal Programming Approach

For building a goal program model for National Analysis, we assume two Fpu's : Fpu-1 and Fpu-2. The national funding f needs to be split between fpu-1 funding f1 and fpu-2 funding f2 such that $f_1 + f_2 = F$. For each fpu-k, we use two EEP measures: S_{k1} (IA Success) and S_{k2} (WUI@Risk). In this section, we assume that the parameterized (closed-form) probability distributions for each EEP measure take on the form of an exponential distribution. Section 4 describes in detail the approach to generate parameterized (closed-form) probability distribution functions for each EEP measure from the output of the Fpu level Bayesian Network Model. The goal program for this case can be formulated as:

Decision variables

e_1 Underachievement of IA Success for Fpu's 1 & 2

e_2 Overachievement of IA Success for Fpu's 1 & 2

k_1 Underachievement of WUI@Risk for Fpu's 1 & 2

k_2 Overachievement of WUI@Risk for Fpu's 1 & 2

f_1 Funding level for Fpu 1

f_2 Funding level for Fpu 2

Parameters

w_1 penalty weight for Fpu 1

w_2 penalty weight for Fpu 2

F Funding at National Level

p_1 Underachievement Penalty for IA Success

p_2 Overachievement Penalty for IA Success

q_1 Underachievement Penalty for WUI @ Risk

q_2 Overachievement Penalty for WUI @ Risk

j Summation index over states of EEP metrics

k The index identifying the k-th Fpu

Before we are ready to define the goal program, we first derive the following expressions from the underlying BN data for FPU-k. The conditional probability and expectations have been computed using the distribution functions (linear in "f") derived in Section 4. The resulting conditional probability and expectation is then defined as:

$$P\{S_{k1} \geq s_{k1j} \mid f_k = f\} = \sum_{i=j} P\{S_{k1} = s_{k2i} \mid f_k = f\} = (a_{k1i} \times f + b_{k1i})$$

$$P\{S_{k2} \geq s_{k2j} \mid f_k = f\} = \sum_{i=j} P\{S_{k2} = s_{k2i} \mid f_k = f\} = (a_{k2i} \times f + b_{k2i})$$

$$E[S_{k1} \mid f_k = f] = \sum_{i=1} s_{k1i}(a_{k1i} \times f + b_{k1i})$$

$$E[S_{k2} \mid f_k = f] = \sum_{i=1} s_{k2i}(a_{k2i} \times f + b_{k2i})$$

We define the resulting national analysis goal program (FPA/NA/GP) as:

$$Maximize \left(\sum_{i=1} s_{k1i}(a_{k1i} \times f + b_{k1i}) + \sum_{i=1} s_{k2i}(a_{k2i} \times f + b_{k2i}) \right)$$

$$>>> Minimize \left(\sum_{i=j} (a_{k1i} \times f + b_{k1i}) + \sum_{i=j} (a_{k2i} \times f + b_{k2i}) \right)$$

$$s.t.\ f_1 + f_2 = F$$

$$f_1, f_2 \geq 0$$

We formulate the above problem as Non-Preemptive and Preemptive goal program.

5.3.1 Non-Preemptive Goal Program (FPA/NA/NGP)

$$Maximize \quad \begin{array}{l} w_1 * \left(\sum_{i=1} s_{k1i}(a_{k1i} \times f + b_{k1i}) + \sum_{i=1} s_{k2i}(a_{k2i} \times f + b_{k2i}) \right) - \\ w2 * \left(\sum_{i=j} (a_{k1i} \times f + b_{k1i}) + \sum_{i=j} (a_{k2i} \times f + b_{k2i}) \right) \end{array}$$

$$s.t.\ f_1 + f_2 = F$$

$$f_1, f_2 \geq 0$$

The above goal program will provide optimal allocations $f^*{}_1$ and $f^*{}_2$

5.3.2 Preemptive Goal Program (FPA/NA/PGP)

First Stage problem:

$$Minimize\ D_1 = p_1 * e_1 + p_2 * e_2$$

$$s.t.$$

$$\left(\sum_{i=1} s_{k1i}\left(a_{k1i} \times f + b_{k1i}\right) + \sum_{i=1} s_{k2i}\left(a_{k2i} \times f + b_{k2i}\right)\right) + e_1 - e_2 = T_1$$

$$\left(\sum_{i=j}\left(a_{k1i} \times f + b_{k1i}\right) + \sum_{i=j}\left(a_{k2i} \times f + b_{k2i}\right)\right) + k_1 - k_2 = T_2$$

$$f_1 + f_2 = F$$

$$f_1, f_2, e_1, e_2, k_1, k_2 \geq 0$$

The first stage problem will provide the optimal value $D^*{}_1$

Second Stage problem:

$$Minimize\ D_2 = q_1 * k_1 + q_2 * k_2$$

$$s.t.$$

$$\left(\sum_{i=1} s_{k1i}\left(a_{k1i} \times f + b_{k1i}\right) + \sum_{i=1} s_{k2i}\left(a_{k2i} \times f + b_{k2i}\right)\right) + e_1 - e_2 = T_1$$

$$\left(\sum_{i=j}\left(a_{k1i} \times f + b_{k1i}\right) + \sum_{i=j}\left(a_{k2i} \times f + b_{k2i}\right)\right) + k_1 - k_2 = T_2$$

$$p_1 * e_1 + p_2 * e_2 = D_1^*$$

$$f_1 + f_2 = F$$

$$f_1, f_2, e_1, e_2, k_1, k_2 \geq 0$$

The second stage problem will provide optimal values $D^*{}_2, f_1^*, f_2^*$

6 Recommended BN and GP Modeling Approaches in Solving the FPA Technical Problem

6.1 Technical Architecture

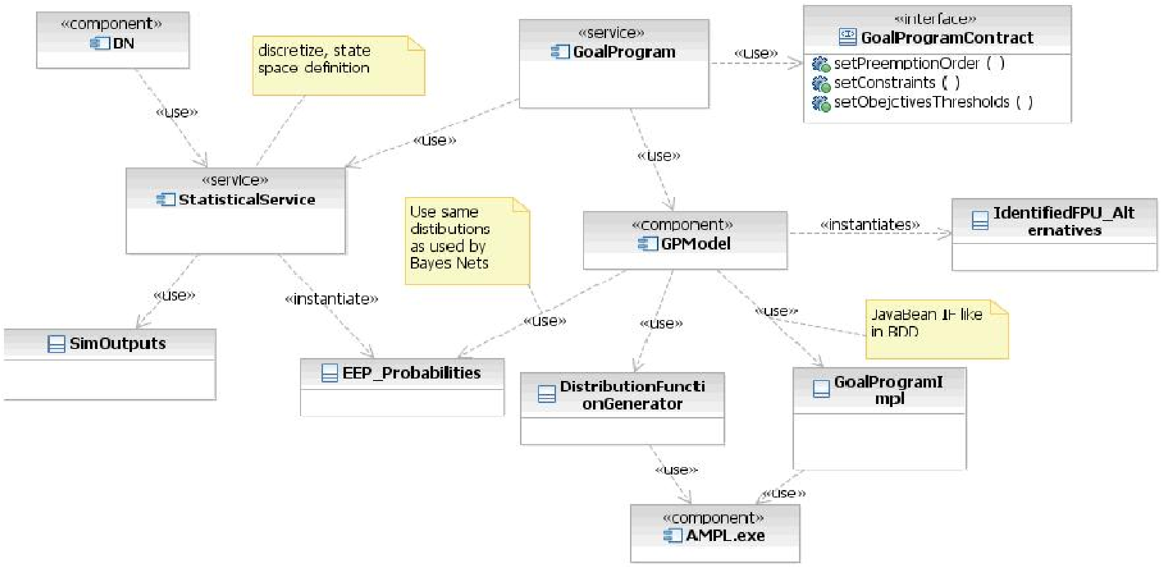


Figure11: Technical Architecture of Fpa

Figure 11 represents the high level technical architecture of the Fpa system. The results of the Fpu level simulation run will be processed by the Statistical Service Component (SSC). The SSC will analyze the simulation outputs to create optimal decritized bins. The output of the SSC will form input to the Bayesian Network. These outputs will also be used by the Distribution Function Generator (DFG) to generate a closed form expression for the EEPs. The EEP expressions will then be used in the Goal Program to optimize over a set of funding alternatives using a priority ordered set of EEPs for all Fpus.

6.2 FPU Analysis Reference Model: Bayesian Network Approach

BN will be used to represent and understand variability and associated risks for each FPU. When an FPU's simulations are complete, there will be dozens of combinations of alternatives, and dozens of simulated fire seasons, meaning thousands of modeled fire events. The system will process the statistics on these results and present the distributions in a Bayesian network. BNs allow a very rapid and rigorous analysis of the risk associated with different management choices. They also allow a variety of sensitivity analyses that can reveal where management can have the most influence, or alternatively, where lack of knowledge most degrades confidence in predictions.

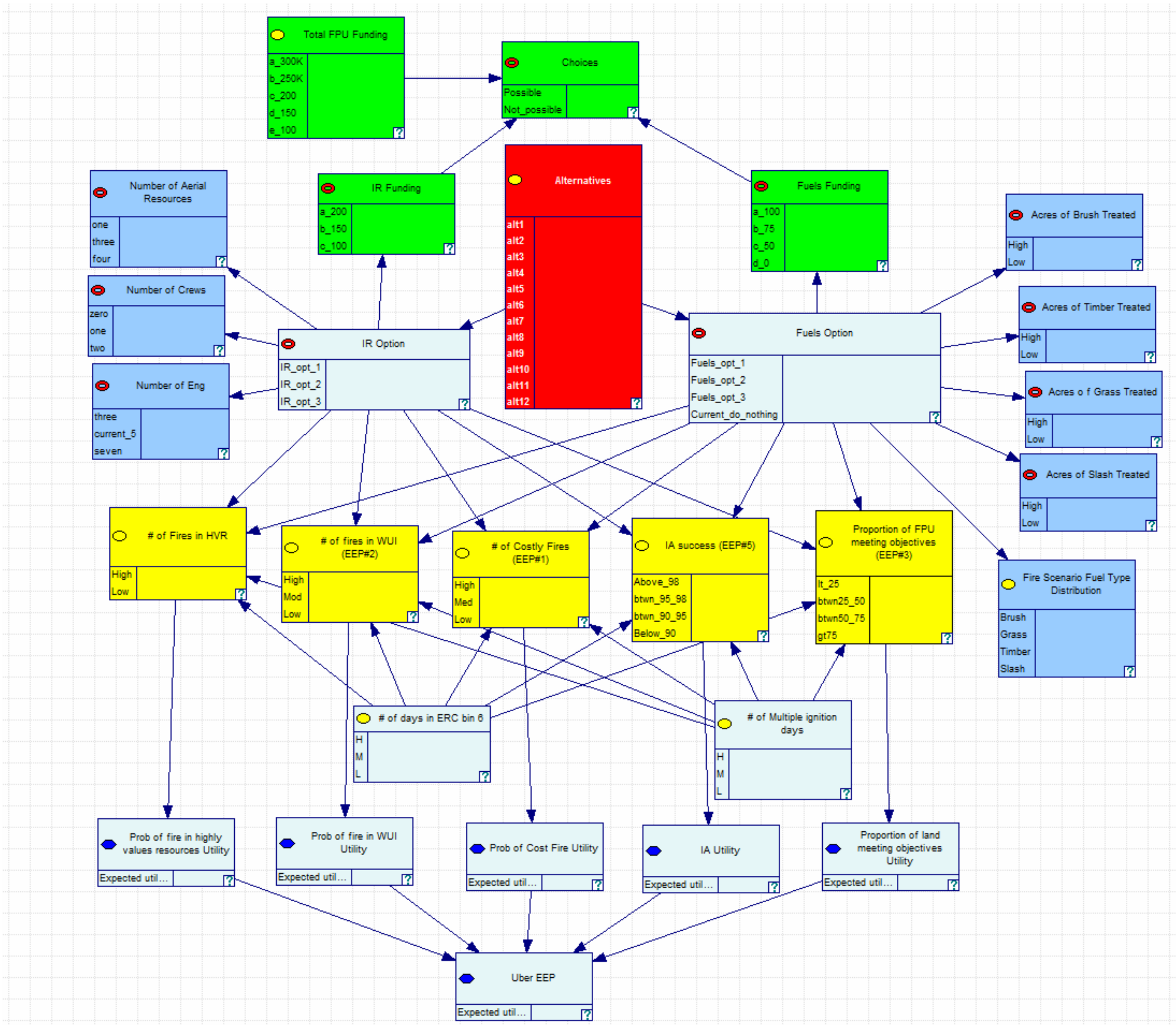


Figure12: Bayesian network for FPU level fire management analysis

Figure 9 shows the Bayes network for the FPU level analysis. In the graph, there are five types of nodes, including (1) funding related nodes (green) (2) combination alternative node (red) (3) EPPs nodes (yellow) (4) resource nodes (dark blue) and (5) utility nodes (blue)

The simulation data will be used to estimate the conditional probability of the yellow nodes given their parents (blue nodes).

6.3 Transforming the BBN Outputs for Input to Goal Program

The detailed FPU-level Bayesian Network analysis provides as output the (discrete) probability distributions for the EEPs, for each Initial Response (IR) and Fuels (FL) treatment alternative. ${f^{IR}}_{kj}$, ${f^{FL}}_{kj}$ represent the Initial Response and Fuels (IR/FL) costs associated with the j[th] IR/FL treatment alternative for the k[th] fpu.

The table below describes the output probability distribution data for the l-th effectiveness measure (EEP) S_{kl} for the k-th Fpu. p_{kjli} represents the probability of the EEP measure S_{kl} being in the i[th] state while at the j[th] IR/FL treatment alternative.

EEP-1: Average annual number of costly fires (S_{kl})

Annual # costly fires (i)	% yrs in this range (p_{kjli})
0	0.33
1	0
2	0.66
>=3	0

Decision variables

a_{kl}^{FL} Coefficient for the k-th Fpu, l-th EEP for Fuels

a_{kl}^{IR} Coefficient for the k-th Fpu, l-th EEP for IR

c_{kli} Residual variable k-th Fpu, l-th EEP and i-th State

Parameters

p_{kjli} Probability of success at treatment alternative j

${f^{IR}}_{kj}$, ${f^{FL}}_{kj}$ IR and FL costs at treatment alternative j

j index for the program alternatives

l Index for the states

k Fpu

Formulation Description

One of the objectives is to optimize the EEP^{kjl} :

$$E[EEP^{kjl}] = \sum_i m_i * p_{kjli}$$

To be able to evaluate these objectives numerically over a wide range of funding alternatives, it is desirable to obtain a function of the form:

$$p_{kli}(f^{FL}, f^{IR}) = a_{kl}^{FL} f^{FL} + a_{kl}^{IR} f^{IR} + c_{kli}, \forall kli$$

in which, we parameterize the program alternative by using the funding components f^{FL} and f^{IR} separately.

Formulation

Assuming a linear form, we set up the following regression problem to compute the coefficients:

$$\text{Min} \sum_{kjli} \left\| a_{kl}^{FL} f_{kj}^{FL} + a_{kl}^{IR} f_{kj}^{IR} + c_{kli} - p_{klij} \right\|^2$$

Subject to:

$$\sum_i (a_{kl}^{FL} f_{kj}^{FL} + a_{kl}^{IR} f_{kj}^{IR} + c_{kli}) \geq 0, \forall kl$$

$$\sum_i (a_{kl}^{FL} f_{kj}^{FL} + a_{kl}^{IR} f_{kj}^{IR} + c_{kli}) = 1, \forall kl$$

However, we will resort to rigorous statistical analysis to learn the functional form of the desired probability distribution function which will be used in a subsequent decision problem to optimize an objective that is expressed in terms of this distribution function.

6.4 National Analysis Model: Goal Programming Formulation

In National Analysis, we are dealing with multiple objectives and hence the decision problem is cast as a goal programming (non-preemptive) problem formulated as follows:

$$\max \sum_{kj} \sum_{l \in MAX} w_l E\left[EEP^{kjl}\right] - \sum_{l' \in MIN} w_{l'} E[EEP^{kjl'}]$$

$$a_{kl}^{FL} f_{kj}^{FL} + a_{kl}^{IR} f_{kj}^{IR} + c_{kli} \geq 0 \qquad \forall k, j, l, i$$

$$\sum_i a_{kl}^{FL} f_{kj}^{FL} + a_{kl}^{IR} f_{kj}^{IR} + c_{kli} = 1 \qquad \forall k, j, l$$

$$\sum_{kj} f_{kj}^{FL} + f_{kj}^{IR} \leq B$$

$$\sum_{kj} f_{kja}^{FL} + f_{kja}^{IR} \leq B_a \qquad \forall a \in Agencies$$

$$f_{kj}^{FL} = \sum_{a} f_{kja}^{FL}$$

$$f_{kj}^{IR} = \sum_{a} f_{kja}^{IR}$$

$\{f_{kja}, j \in J\}$ is a SOS-1 for $\forall ka$ - Only one alternative of $\{j \in J\}$ needs to be selected

$0 \le \sum_{kja} f_{kja}^{IR} \le B^{IR_{UB}}$ - Maximum budget level for IR program

$0 \le \sum_{kja} f_{kja}^{FL} \le B^{FL_{UB}}$ - Maximum budget level for FL program

6.5 Goal Programming solution approach

The proposed Goal Programming solution approach can optimize over different funding alternatives for each priority ordered set of EEP's without complete enumeration of the state-space (e.g., 5 EEPs with 4 states each, 9 FPU funding alternatives across 2 program components with 3 funding alternatives for each, and about 150 FPU's => 843750 states!) Introduction of agency level funding decisions within an FPU introduces additional dimensionality to the decision space.

7 Conclusions

We analyzed the Bayesian Network Modeling and Goal Programming approach by defining and building a representative National Model by using two program components: Initial Attack and Fuels and two effectiveness-efficiency performance metrics : IA Success and WUI @Risk. In the BBN model, we defined utility nodes and used inference algorithms implemented by GeNIe to access the relative utility of the decisions. We also compared the non-preemptive and preemptive goal programming approaches and uncovered weighting and sensitivity issues. Finally, we proposed a hybrid approach that utilizes the best of BBN and Goal programming for building a National Model.

7.1 Validation Approach

One of the cornerstones of any successful model is its thorough validation. The validation of the proposed approach involves validating each sub system. Due to the component nature of the proposed approach, each subsystem can be validated by using a combination of expert system, real time system measurements and analytical validation model to ensure that the models are good representation of the real world scenario.

8 Appendix

8.1 Bayesian Approach to Probability and Statistics

A Bayesian network (or a belief network) is a directed acyclic graph which represents independencies embodied in a given joint probability distribution over a set of variables V.

8.1.1 Definition of Bayesian networks

A Bayesian networks is a directed graph, together with an associated set of probability tables. The graph consists of nodes and edges. The nodes represent the concerned variables, which can be either discrete or continuous. For example, the node “fire alarm” is discrete having values “true” or “false”, whereas the node 'system safety' might be continuous (such as the probability of failure on demand). The edges represent “causal” or directly influential relationships between variables. For example, the fire alarm is influenced by the system safety values and amount of smoke in the lobby.

The graph representation is very intuitive; however, there are several common pitfalls which demand special attention. First, the absence of an edge between two nodes A and B does not suggest that A is independent of B. As shown in Figure 9, the values of A will affect that of C, which be passed on and used to determine the value of B. Therefore A and B are not independent; however, A is independent of B given C.

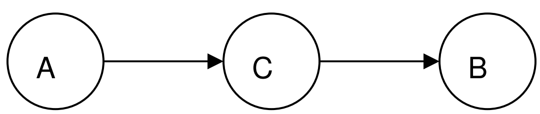


Figure13: An example of conditional independence in Bayesian networks

Second, there is no one-to-one mapping from the graphs to the independencies embodied in the data. For example, the factor graph (which is an "undirected" graph version of the BN with the same set of dependencies) is fully connected can represent all types of dependencies since all the variables are dependent of each other. In the real applications, we seek a graph that is able to include all the dependencies within the data while retain simplicity as much as possible.

8.1.2 Learning

Given a Bayesian network G = <V, E>, we can define the joint probability of all variables in the graph (represented by set V) as a joint product of the probability of each variable given its parents in graph G, i.e.

$$P(V) = \prod P(V_i \mid parents(V_i)).$$

Then our next question is how to define the conditional probability distributions for each variable. If we are given a set of data with the observations for all the variables, then $P(V_i| parents(V_i))$ can be calculated easily using the Bayes rule, i.e.

$$P(V_i \mid parents\ (V_i)) = P(V_i, parents\ (V_i))/ P(parents\ (V_i)),$$

where $P(V_i$, parents$(V_i))$ and P(parents $(V_i))$ can be estimated using the maximal likelihood estimation of the data. For example

$P(V_i= v_i$, parents (V_i)= parents(v_i)) =

# of examples with $V_i = v_i$ and parents(V_i) = parents(v_i) / total # of examples.

However, it becomes much more complex if we can observe only the partial data, that is, there are some hidden variables we want to model in the graph. Then an EM approach or other iterative learning has to be applied [Heckerman, 1996]. We omit the detailed discussion.

Notice that modeling the independencies between variables represented by the graph is an essential concept in the Bayesian network approach. Otherwise, we need to define a joint probability with a complexity that grows exponentially with the number of nodes.

8.1.3 Inference

After building a Bayesian network, our ultimate goal is to answer questions: given the observation of some variables, what is the probability of other variables. For example, given the fire alarm is on, what is the probability that there is a fire happening? These types of questions are called inference in Bayesian network terminology. There are many inference algorithms, including exact inference and approximate inference.

The elimination algorithm is the basic method for exact inference. The main idea is to efficiently marginalize out all the irrelevant variables using factored representation of the joint probability distribution. Consider the graph in Fig.1, the probability P(C) can be computed by

$P(B) = \sum_A \sum_C P(C|A)P(B|C)P(A)$

$= \sum_A (\sum_C P(C|A)P(B|C))P(A)$

$= \sum_A m1(A, B) P(A)$

$= m2(B)$

The intermediate factors m1 and m2 can be seen as messages passing from the variables that have been integrated. When we want to compute several marginals at the same time, a dynamic programming can be applied to reuse some messages in the elimination algorithm. If the underlying graph is a tree, we can use sum-of-product, or belief propagation, which is a generalization of the forward-backward algorithm in HMMs [Rabiner, 1989]. For a general graph, it has to be converted to into a clique tree by moralization and triangulation. After that, a local message passing algorithm can be applied, which could be either the sum-of-product algorithm or the junction tree algorithm, a variation designed for undirected models.

The computational complexity of the exact inference algorithms is exponential in the size of the largest cliques in the induced graph. For many cases, such as grids or factor graph, it is intractable to make exact inferences and therefore approximate algorithms, such as sampling, variational methods or loopy belief propagation, have to be applied. Sampling is a well-studied field in statistics and various sampling algorithms have been proposed. A very efficient approach for high dimensional data is Markov Chain Monte Carlo (MCMC), which includes Gibbs sampling and Metropolis-Hastings sampling as special cases.

8.2 Goals and Utopian Sets

A multiple objective problem may have four types of goal criteria, as portrayed in the Fig 10

1. Greater than or equal to.
2. Less then or equal to
3. Equality
4. Range

The t_i are target values (a) on or above which, (b) on or below which, (c) at which, or (d) between which we wish to reside.

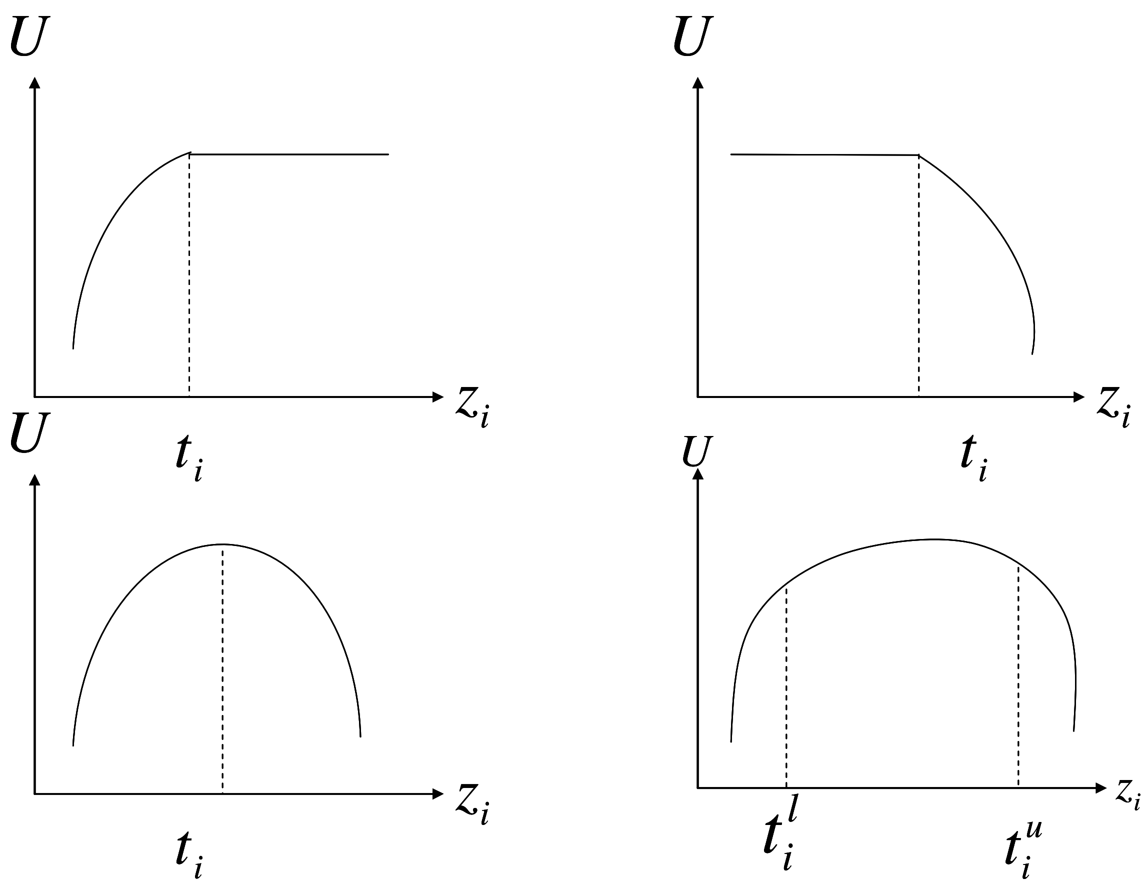


Figure14: Goal Criteria Types

A goal programming problem with, for instance, one of each type of goal criteria is expressed as

$$goal\{c^1 x = z_1\} \qquad z_1 \geq t_1$$

$$goal\{c^2 x = z_2\} \qquad z_2 \leq t_2$$

$$goal\{c^3 x = z_3\} \qquad z_3 = t_3$$

$$goal\{c^4 x = z_4\} \qquad z_4 \in [t_4^l, t_4^u]$$

$$s.t. \quad x \in S$$

The information in parentheses on the left specifies the value of z_i to be achieved (if possible) in relation to stipulated t_i target values.

Consider the GP

$$goal\{c^1 x = z_1\} \qquad z_1 \geq t_1$$

$$goal\{c^2 x = z_2\} \qquad z_2 \in [t_2^l, t_2^u]$$

$$s.t. \quad x \in S$$

whose decision space representation is given in figure 11. In this figure c^1 = (1, 1/2), c^2 c2 = (1/2, 1), x^1 = (4,1) and x^2 = (0,5)., the cross-hatched area is the utopian set in decision space. This is the set of points in R_n at which all goals are simultaneously satisfied. The criterion space representation of the GP is given in figure 12 where z_1 = (4 ½, 3) and z_2 = (2 ½, 5), the cross-hatched area is the utopian set in criterion space. This is the set of criterion vectors in R_k that simultaneously satisfy all goals. Since there are no points in figure 11 and 12 that feasibly satisfy all goals simultaneously, our goal programming endeavor is to find the point in S whose criterion vector "best" compares with the utopian set in criterion space.

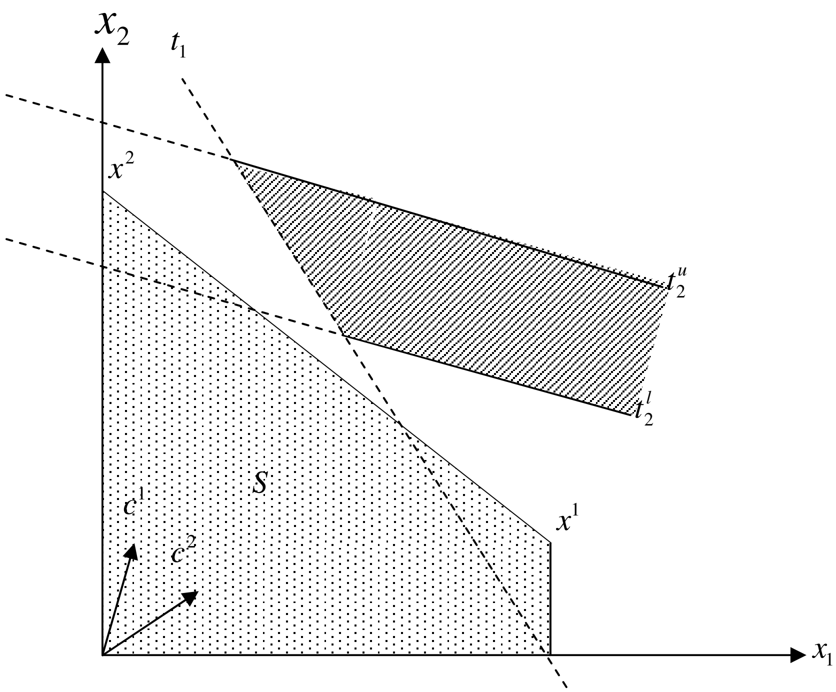


Figure15: Decision Space representation

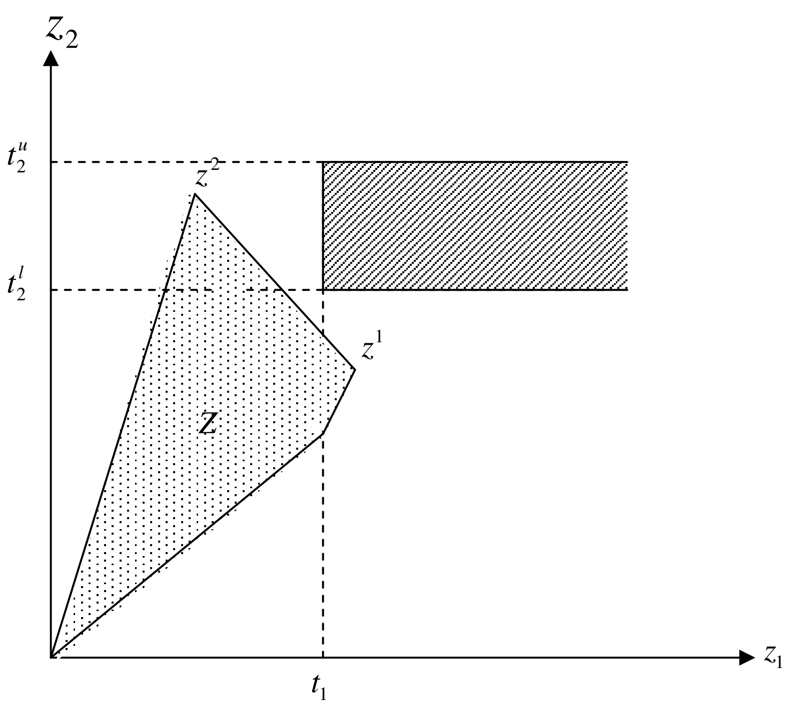


Figure16: Criteria Space representation

9 Acknowledgements

This work was supported under the U.S. Department of Agriculture contract No. 43-82X9-3-5073. We would also like to acknowledge Andy Kirsch of National Park Service and the Fire Program Analysis project team for their valuable inputs and support.

References

- Steven Carty, "*Fpa II Conceptual Architecture*", January 16, 2007.
- Andy Kirsch, "National Model: National_complete_eeps_per FPU_wfu_color_10_27_06.xdsl" (2006)
- LR Rabiner, "A tutorial on hidden Markov models and selected applications *in speech recognition*", Proceedings of the IEEE, 1989.
- Judea Pearl, "*Causality: Models, Reasoning and Inference*", 2000, Cambridge University Press.
- Spirtes, Glymour and Scheines, "*Causation, Prediction and Search*", (2nd edition), MIT Press, 2001.
- F. R. Kschischang, B. J. Frey and H.-A. Loeliger, "*Factor graphs and the sum-product algorithm IEEE Transactions on Information Theory*", February, 2001.
- D. Heckerman "*A tutorial on learning with Bayesian networks*", Microsoft Research tech. report, MSR-TR-95-06, 1996.
- F. V. Jensen. "*Bayesian Networks and Decision Graphs*", Springer, 2001.
- M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. "*An introduction to variational methods for graphical models*", 1997.
- D. MacKay. "*An introduction to Monte Carlo methods*", 1998.

- G. L. Nemhauser and L. A. Wolsey, "*Integer and Combinatorial Optimization, John Wiley & Sons Inc.*", Hoboken, New Jersey (1988).
- Ralph E. Steuer, "Multiple Criteria Optimization: Theory, Computation and Application", 1996.