Allocation of Resources in CB Defense: Optimization and Ranking

by

NMSU: J. Cowie, H. Dang, B. Li
Hung T. Nguyen
UNM: F. Gilfeather

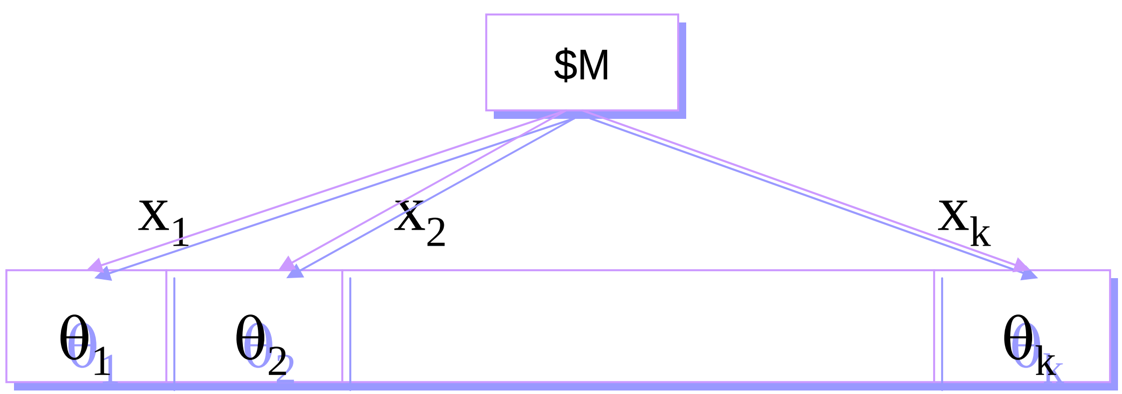
Oct 26, 2005

Outline

- Problem Formulation
- Architecture of Decision System
- Optimization Methods
- Ranking Procedures
- Conclusions and Future Work

Problem Formulation

Given $M, let $\theta=(\theta_1,\ldots,\theta_k)$ be the mitigating variable of some asset, find "<u>optimal</u>" allocations $(x_1,\ldots,x_k)$ to $(\theta_1,\ldots,\theta_k)$ to minimize consequences $c=(c_1,\ldots,c_k)$ of an CB attack to the asset, and rank these allocations according to various possible "preferences" of decision-makers.



$$\sum_{i=1}^{k} x_i = \$M$$

Problem Formulation

Example:

Asset: An airbase

Money: $M=1million

Defense Measures:

θ_1: chemical agent detector

θ_2: biological agent detector

θ_3: perimeter protection

θ_4: trained onsite personnel

θ_5: chemical prophylaxis

θ_6: biological prophylaxis

θ_7: medical treatment

Consequences:

c_1: number of casualties

c_2:cost of remediation

c_3:number of days of operation disruption

c_4:negative geo-political impacts

Problem Formulation

- *Optimal Allocations*

We need to formulate an “objective function”.

$\varphi : \Omega \to \Psi$, where

$$\Omega = \{(x_1, \cdots, x_k) : \sum_{i=1}^{k} x_i = M\} \subseteq R^k$$

$$\Psi = \{c = (c_1, \cdots, c_m)\} \subseteq R^m$$

Ω is the space of allocations,

Ψ is the space of consequences.

The optimization problem is **Min $\varphi(x_1,\ldots,x_k)$** subject to $(x_1, \cdots, x_k) \in \Omega$, which is an optimization problem with multiple objectives. In principle, the problem can be solved by standard techniques using decision-makers’ preferences and value trade offs.

Problem Formulation

- *How to obtain the objective function* $\varphi : R^k \to R^m$?

$$X=(x_1,\cdots,x_k)\to\Delta\theta=(\Delta\theta_1,\cdots,\Delta\theta_k)\to c(\Delta\theta)=(c_1(\Delta\theta),\cdots,c_m(\Delta\theta))$$

The relation between $X=(x_1,\ldots,x_k)$ and the consequence $\varphi(X)$ is:

a) $c(\theta)$: the consequence c is a function of θ.

(Data/Scenarios from experts).

b) $\theta(X)$: is a function of X. (Cost model).

So,

$$\varphi(X)=c\circ\theta(X)$$

Problem Formulation

Example 1: An Example of Objective Function

$$c_i(X) = \alpha \sum_{s=1}^{p} \{ c_i^0(s) L(s) \prod_{j=1}^{k} [1 - e(s, \theta_j(X))] \}$$

Where:

*α is a normalization constant,
*p is the number of scenarios,
*$c^0{}_i(s)$ initial consequence before improvements for the s^{th} scenario,
*L(s) is a probability (sums to 1),
*$\theta_j(X)$ is the number of detectors of the i^{th} kind, it is a function of X.
*e(s, $\theta_j(X)$) is the effectivity of the i^{th} kind of defense measure on the scenario s.

Problem Formulation

Minimize $f(x_1, x_2, \cdots, x_6) = \sum_{s=1}^{5} 10 \prod_{j=1}^{6} [1 - \frac{s}{5} \sin(\frac{x_j}{x}\pi)]$

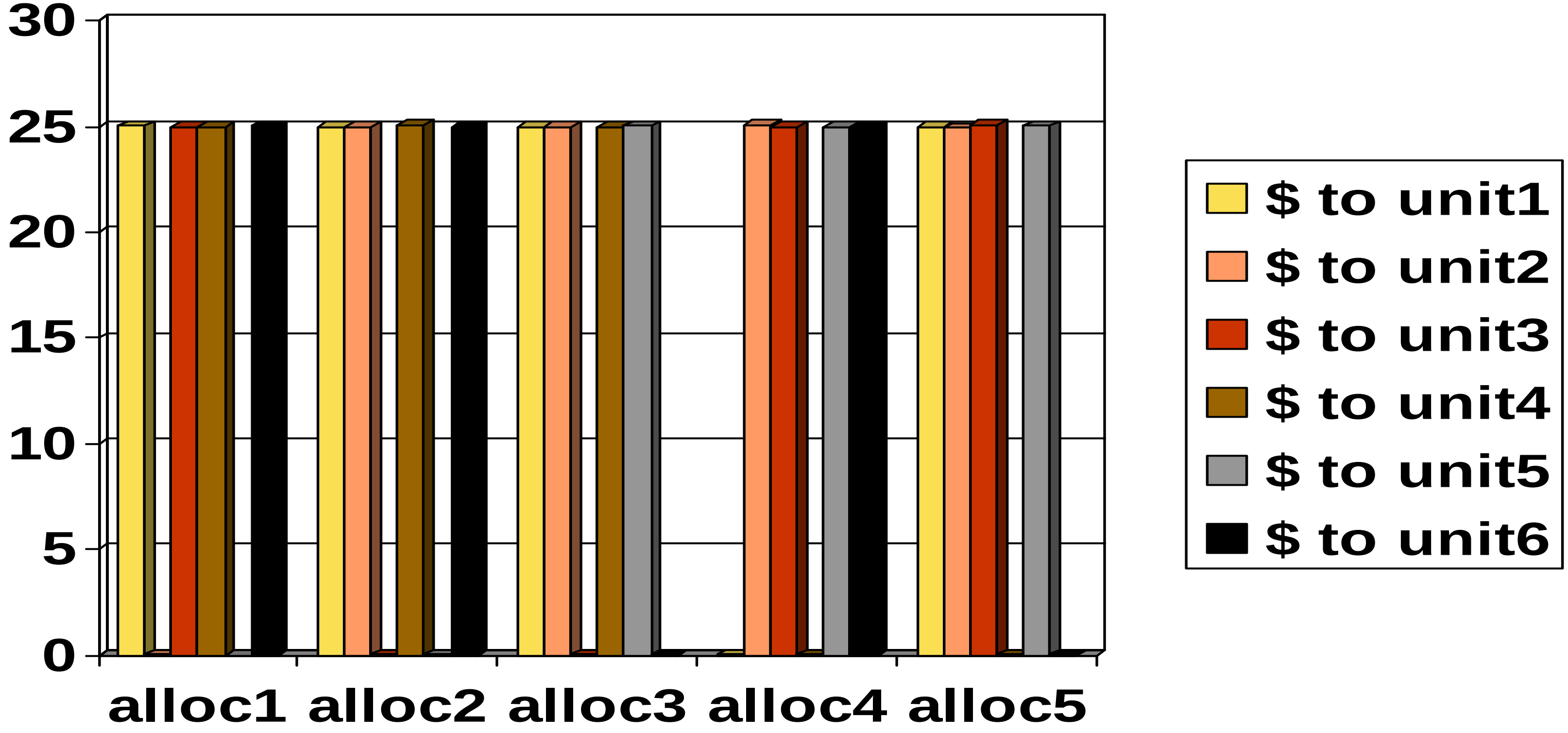
subject to $\sum_{i=1}^{6} x_i = 100 \;, \; x_i \geq 0 \,.$

Using simulated annealing, we get:

$(x_1, x_2, \cdots, x_6)$	Minimum
(25.0077 0.0053 24.9808 24.9981 0.0004 25.0076)	9.6074
(24.9952 24.9950 0.0072 25.0397 0.0052 24.9578)	9.6074
(24.9629 24.9837 0.0278 24.9988 25.0181 0.0087)	9.6076
(0.0309 25.0083 24.9321 25.0111 25.0168 0.0007)	9.6075
(0.0366 25.0442 24.9996 0.0079 24.9497 24.9620)	9.6076
(24.9827 24.9416 25.0395 0.0117 25.0086 0.0160)	9.6075
(24.9924 25.0519 0.0210 0.0217 24.9539 24.9592)	9.6076

Problem Formulation

Histogram for optimal allocations (X=$100).



Problem Formulation

Component-wise optimization is not a proper solution for our multiple objective problem.

Example 2: Minimize $\varphi(x_1,\ldots,x_k)$ over a constrained set A with predetermined acceptable consequences levels.

$$A = \{(x_1,\cdots,x_k) : \sum_{i=1}^{k} x_i = M \text{ and } \varphi_j(x_1,\cdots,x_k) \le \hat{c}_j\}$$

Architecture

INPUT

- Data from scenario tree: mitigating variables θ's, likelihood, consequences …
- Total budget $X
- Acceptable consequences $\widehat{C}$
- Cost model

ANALYTIC TOOLBOX

- Utility function C=φ(X)
- Human interface: questionnaire

OPTIMIZATION
Minimizes utility function C=φ(x) subject to:

$$\sum x \leq X$$

and

$$C \leq \widehat{C}$$

RANKING
Ranks allocations using Choquet Integral, AHP, Multi Objective Programming

OUTPUT

$$\begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \\ \vdots \\ x_k^* \end{bmatrix} \begin{bmatrix} c_1^* \\ c_2^* \\ \vdots \\ c_m^* \end{bmatrix}$$

Optimization

1. Optimization of the problem in Example 2.

Minimize total cost $M = \sum_{i=1}^{k} w_i \Delta\theta_i$ subject to $c_j - \sum_{i=1}^{k} e_{ij} \Delta\theta_i \le \hat{c}_j$ for j=1,…,m, where

*k is the number of defense measures (mitigating variables)
*w_i is the cost of improving θ_i by one unit
*$\Delta\theta_i$ is the improvement in defense measure facilities θ_i
*c_j is the j^{th} consequence component in the initial variant (data)
*e_{ij} is the decrease in the result c_j of a scenario attack if we increase θ_i by one
*m is the number of components of a consequence vector after an attack

Optimization

Let $[w_1, \ldots, w_5] = [3,5,17,27,6]$, $c = [1000, 100, 60, 20]$,

$\hat{c} = [200, 50, 29, 13]$, and

$$(e_{ij}) = \begin{bmatrix} 40 & 4 & 3 & 1 \\ 48 & 3 & 2 & 2 \\ 55 & 6 & 3 & .4 \\ 80 & 5 & 4 & 3 \\ 66 & 4 & 4 & 2 \end{bmatrix} \qquad \Delta\theta = \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta\theta_4 \\ \Delta\theta_5 \end{bmatrix}$$

We want to find $\Delta\theta_i$'s such that $M = \sum_{i=1}^{k} w_i \Delta\theta_i$ is minimized.

Optimization

10 options of $(\Delta\theta_1,\ldots,\Delta\theta_5)$ with their corresponding amount of money.

$(\Delta\theta_1,\ldots,\Delta\theta_5)=(14,1,1,1,1)$, with $M=97$;

$(\Delta\theta_1,\ldots,\Delta\theta_5)=(11,2,1,1,2)$, with $M=99$;

$(\Delta\theta_1,\ldots,\Delta\theta_5)=(13,2,1,1,1)$, with $M=99$;

$(\Delta\theta_1,\ldots,\Delta\theta_5)=(9,1,1,1,4)$, with $M=100$;

$(\Delta\theta_1,\ldots,\Delta\theta_5)=(11,1,1,1,3)$, with $M=100$;

$(\Delta\theta_1,\ldots,\Delta\theta_5)=(13,1,1,1,2)$, with $M=100$;

$(\Delta\theta_1,\ldots,\Delta\theta_5)=(15,1,1,1,1)$, with $M=100$;

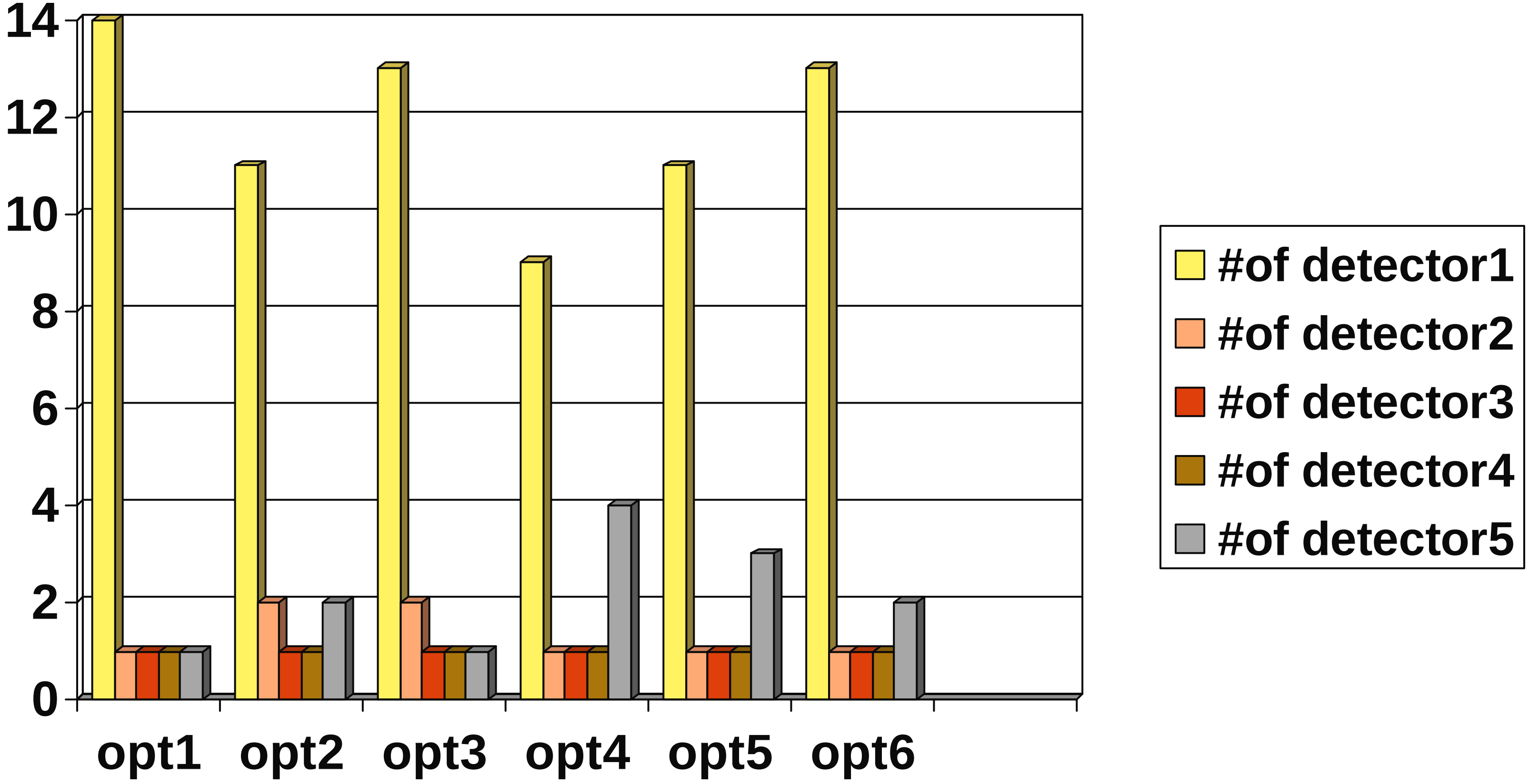
$(\Delta\theta_1,\ldots,\Delta\theta_5)=(10,3,1,1,2)$, with $M=101$;

$(\Delta\theta_1,\ldots,\Delta\theta_5)=(12,3,1,1,1)$, with $M=101$;

$(\Delta\theta_1,\ldots,\Delta\theta_5)=(6,2,1,1,5)$, with $M=102$;

Optimization

Histogram for the options



Optimization

2. Optimization of vector-valued "utility function" using decision-makers' preferences.

To optimize $\varphi(X)=(c_1(X),\ldots,c_m(X))^T$ where $X=(x_1,\ldots,x_k)$, we can use the common method of weighted linear combination of the components c_j (X), i.e., optimize

$$\sum_{j=1}^{m} w_j c_j(x)$$

where w_j is the scaling weight for c_j.

Optimization

How to obtain the w_j's?

Example: Let $c_1(X)$=repair cost ($), $c_2(X)$=human casualties. The overall utility function is expressed in $. So $w_1=1$. Decision makers will be asked to express their preferences among consequences leading to the identification of w_2 (Keeney and Raiffa, 1993).

The weighting method

The Weighting method

This method addresses the difficulty of comparing two multi-component consequence vectors by finding appropriate weights that can be used to convert all the components to the same unit (dollars in this demo)

Consequence vector C1

Cost of repair ($)

7000 [+] [-]

Casualties

30 [+] [-]

C1 ($) = 7000 + w * 30

Consequence vector C2

Cost of repair ($)

5000 [+] [-]

Casualties

35 [+] [-]

C2 ($) = 5000 + w * 35

Make adjustments until C1 and C2 are considered "equal" and click Done to calculate the weight w

[Done]

Weight calculation

7000 + w * 30 = 5000 + w * 35

w = 400 ($/casualty)

Optimization

3. Some optimization methods

a) Genetic algorithms

This "evolutionary" type of optimization method is appropriate for non-smooth objective functions. The method is inspired from the reproduction process in biology. This method is designed to optimize an objective function f for which we do not know its analytic expression but, given input $\theta = (\theta_1, \theta_2, \ldots, \theta_k)$, the value $f(\theta)$ can be found.

Optimization

b). Stochastic approximation (Robbins and Monro, 1951)

This method is designed to optimize an unknown function f(θ) when , for specified θ, the value f(θ) can be provided. This can be done by asking experts from MIIS.

Problem: Find a minimum point, $\theta^* \in R^p$, of a real-valued function f(θ), called the "loss function," that is observed in the presence of noise.

Optimization

(1) Finite Difference: The iterative procedure is

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k)$$

$$\hat{g}_{ki}(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k e_i) - y(\hat{\theta}_k - c_k e_i)}{2c_k}$$

a_k: goes to 0 at a rate neither too fast nor too slow,
$y(\theta)$: the observation of $f(\theta)$,
e_i: a vector with a 1 in the ith place, and 0 elsewhere,
c_k: goes to 0 at a rate neither too fast nor too slow.
Initialize $\hat{\theta}_1$, calculate $\hat{g}_1(\hat{\theta}_1)$ and $\hat{\theta}_2$, …, continue this process till $\hat{\theta}_k$ converges to $\hat{\theta}^*$, which is our optimization solution.

Optimization

Example: Let $X = \begin{bmatrix} 7 & 1 & 5 & 4 & 7 & 6 \\ 0 & 6 & 3 & 5 & 6 & 7 \\ 1 & 3 & 7 & 7 & 3 & 7 \\ 2 & 6 & 0 & 1 & 1 & 1 \end{bmatrix}$ $C = \begin{bmatrix} 22 \\ 36 \\ 2 \\ 30 \end{bmatrix}$ $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}$

and $\varepsilon \sim N(0,1)$

Optimize f(θ)=(Xθ-C)’(Xθ-C)+ε with Finite Difference procedure,we get the result as follows:

$\hat{\theta}$ =[1.1823 4.1721 -2.5319 -1.0656 3.3252 0.5283]’

It is close to the result of the case where the observations are not influenced by noise:

$[1.1667 \ \ 4.1257 \ -2.6528 \ \ -1.0799 \ 3.3395 \ \ 0.6524]'$

Optimization

(2) Simultaneous Perturbation Stochastic Approximation (SPSA) with Injected Noise (Maryak and Chin, 2001):to obtain global minimum.

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) + q_k \varpi_k$$

$$\hat{g}_k(\theta) = (2c_k\Delta_k)^{-1}[y(\theta + c_k\Delta_k) - y(\theta - c_k\Delta_k)]$$

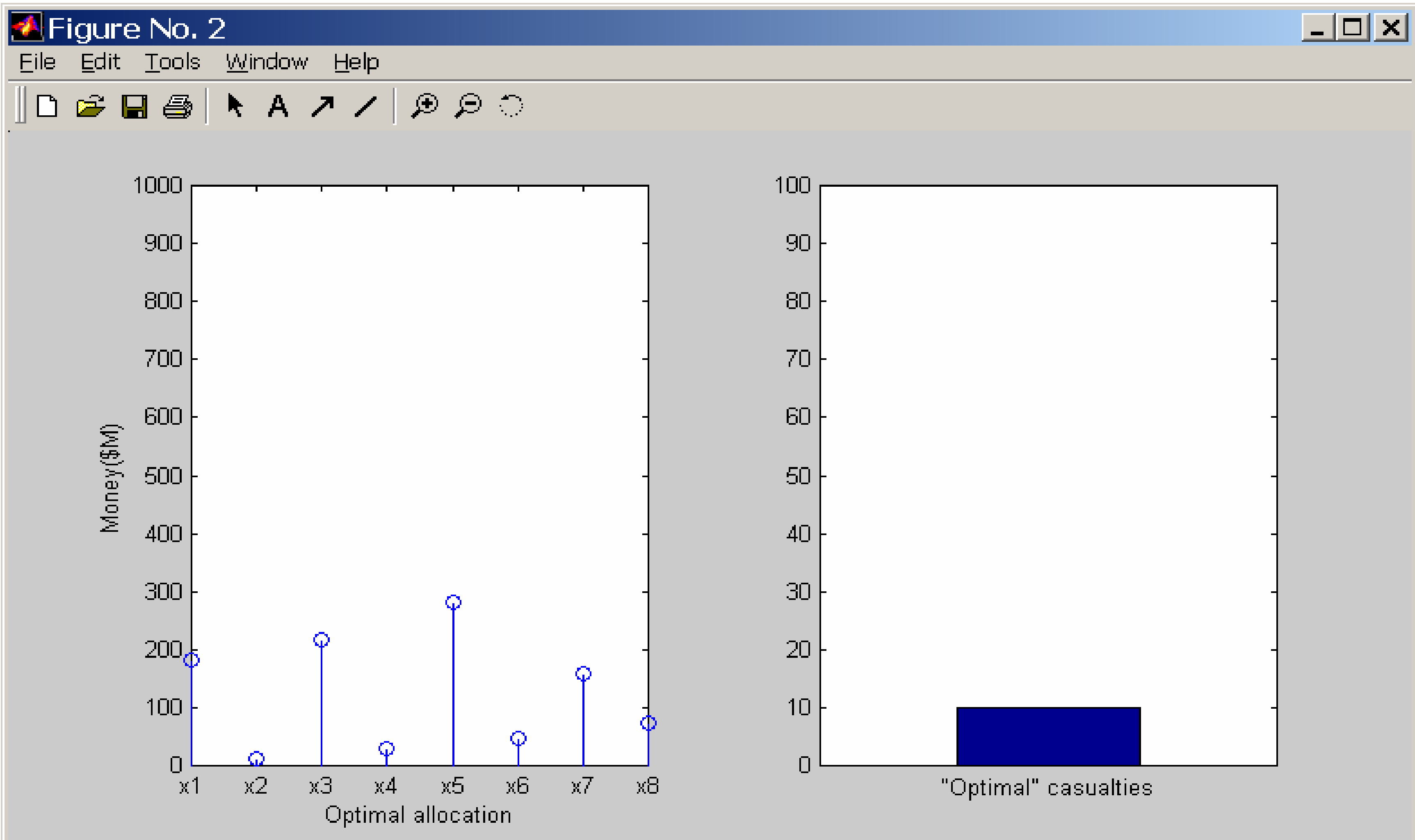
a_k,c_k,q_k:goes to 0 at a rate neither too fast nor too slow;

y(.): the observation of f(.);

Δ_k: distributed as Bernoulli (±1);

w_k: i.i.d. in N_k (0,1).

Optimization of mockup math model using simulated annealing



Ranking

- Rank solutions $X = (X_1, X_2 \ldots X_k)$ obtained via optimization to find the most "efficient" one (Multi-criteria decision making, with the mitigating variables $\theta_1,\theta_2 \ldots \theta_k$ as criteria).
- Goal: define a total order between alternatives, i.e. define a map $\varphi:\Re^k \rightarrow \Re$, so that alternative X is preferred to alternative Y if $\varphi(Y) \leq \varphi(X)$.
- The total order should reflect the degrees of importance of each criterion θ_i in its contribution to the "total score" $\varphi(X)$.

Ranking

- In our ranking problem, the criteria are interactive, e.g. mitigating variables can contribute to damage reduction in combinations.
- Non-linear aggregation operators are proved more efficient in such situation than linear ones but may be computationally prohibitive.
- Approximation can be done with a linear aggregation operator based on simple analysis of interaction between criteria.

Ranking

1. Ranking with AHP

- Linear aggregation operator based on simple analysis of interaction between criteria, namely, pair-wise comparisons.
- Uses fuzzy logic to extract degrees of importance between pairs of criteria, and conducts a synthesis of priorities leading to a weighted average operator.
- Can handle linguistic (qualitative) values of allocations.

Ranking

2. Ranking with Choquet Integral

- Non-linear aggregation operator (more general and axiomatically justified).

Description of Discrete Choquet Integral:

Denote $T = (c_1, c_2, \cdots, c_k)$ to be the set of k criteria,

$x = (x_1, x_2, \cdots, x_k)$ to be the evaluations on the subject x.

A fuzzy measure on power set 2^T satisfies

a. $\mu(0) = 0, \mu(T) = 1$ and,

b. $A \subseteq B$ implies $\mu(A) \le \mu(B)$ for $A, B \subseteq T$.

Discrete Choquet integral with respect to the fuzzy measure is given by

$$C_\mu(x) = \sum_{i=1}^{k} (x_{(i)} - x_{(i-1)}) \mu(A_{(i)})$$

with $x_{(0)} = 0$, and $A_{(i)} = \{c_{(i)}, \cdots, c_{(k)}\}$, where $(x_{(1)}, \cdots, x_{(k)})$ is ranked $(x_1, \cdots, x_k)$ in increasing order, $\{c_{(i)}, \cdots, c_{(k)}\}$ is the subset of criteria corresponding to $\{x_{(i)}, \cdots, x_{(k)}\}$

Ranking

Example: *Rank ten 5-component consequence vectors, with the fuzzy measure μ=[0.07 0.07 0.07 0.27 0.27 0.35 0.37 0.37 0.37 0.37 0.47 0.47 0.55 0.47 0.59 0.27 0.27 0.27 0.29 0.29 0.39 0.39 0.39 0.57 0.57 0.63 0.63 0.65 0.75 0.75]:*

Consequence vectors	Choquet integrals	Rank	Consequence vectors	Choquet integrals	Rank
(14,1,1,1,1)	6.33	2	(13,1,1,1,2)	6.29	3
(11,2,1,1,2)	5.59	7	(15,1,1,1,1)	6.74	1
(13,2,1,1,1)	6.14	4	(10,3,1,1,2)	5.40	8
(9,1,1,1,4)	5.39	9	(12,3,1,1,1)	5.95	5
(11,1,1,1,3)	5.84	6	(6,2,1,1,5)	4.64	10

Ranking

3. Identification of fuzzy measures

Fuzzy measures have to satisfy the monotone constraints.

Two methods to identify fuzzy measures μ:

i) *Supervised learning*

a) Quadratic programming

b) Neural network

ii) *Unsupervised learning (Method of Entropy)*

Viewing criteria as a random vector, and allocations as random sample. Estimating all joint partial density functions. Using entropies of subsets of criteria as fuzzy measure value.

Ranking

a. Identification of fuzzy measures with quadratic programming

For the following data

x_1 =(1 1 .9 .7 .5) x_2 =(.3 .1 1 .9 .6) x_3 =(.5 .7 .3 .5 .9)

x_4 =(1 .5 .4 .1 .5) x_5 =(.8 .6 .8 .8 .7) x_6 =(.4 .0 .2 .7 .9)

x_7 =(.9 .8 .9 1 .3) x_8 =(.5 1 1 .5 .1) x_9 =(.7 .9 .8 .2 .7)

$Y = (y_1, y_2,\ldots,y_9)$ =(.7 .6 .5 .3 .8 .5 .7 .5 .4).

By our algorithm, we get

μ = [0.0700 0.0700 0.0700 0.2700 0.2700

0.3500 0.3700 0.3700 0.3700 0.3700

0.4700 0.4700 0.5500 0.4700 0.5900

0.2700 0.2700 0.2700 0.2900 0.2900

0.3900 0.3900 0.3900 0.5700 0.5700

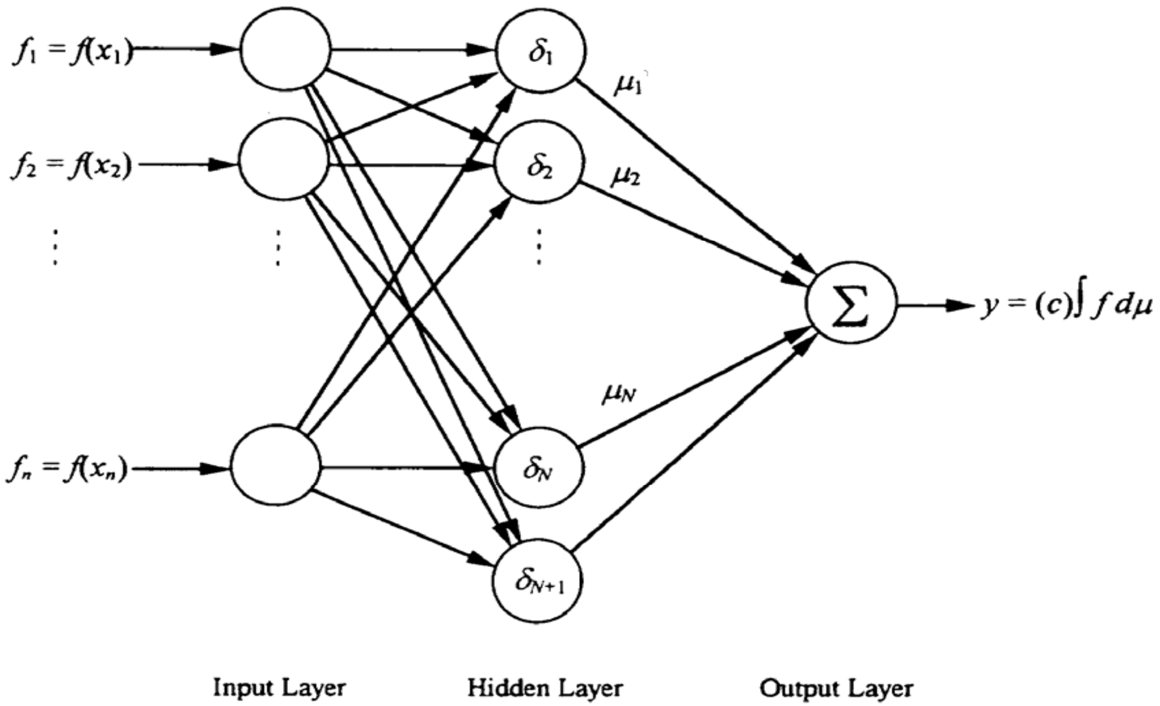
0.6300 0.6300 0.6500 0.7500 0.7500]

The corresponding quadratic error is 0.0084.

Ranking

b. Identification of fuzzy measures with neural networks

Fuzzy measures are set up as weights of a feed forward neural network, which are found by training the network with back-propagation algorithm. (Wang and Wang, 1997).



Ranking

Example: identification of fuzzy measures with neural networks

Sample	Feature 1	Feature 2	Feature 3	Evaluation
1	0.56	0.78	0.92	0.742984
2	0.05	0.36	0.18	0.143036
3	0.97	0.95	0.84	0.881246
4	0.00	0.62	0.06	0.090632
5	0.22	0.15	0.00	0.064790
6	1.00	0.75	0.33	0.522212
7	0.49	0.55	0.76	0.608632
8	0.89	0.37	0.97	0.794288
9	0.64	0.59	1.00	0.771720
10	0.11	0.00	0.03	0.038632

Our network produces the following fuzzy measure: $\mu\{\}= 0$, $\mu\{1\}= 0.2$, $\mu\{2\}= 0.1$, $\mu\{1,2\}=0.3386$, $\mu\{3\}= 0.399$, $\mu\{1,3\}= 0.7544$, $\mu\{2,3\}= 0.5772$ $\mu\{1,2,3\}=1$, which satisfies the monotone constraints.

Conclusions and Future Work

- Data structure
- Experts and decision-makers’ assistance
- Optimization methods
- Ranking procedures
- Some other issues

References

[1] Deb, K. (2002). Multi-Objective Optimization Using Evolutionary Algorithms. J. Wiley.

[2] Grabisch, M., Nguyen, H.T. and Walker, E.A. (1994). Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference. Kluwer Academic.

[3] Keeney, R.L., and Raiffa (1993), H. Decisions with Multiple Objectives. Cambridge University Press.

[4] Maryak, J. L. and Chin, D. C. (2001). “Global Random Optimization by Simultaneous Perturbation Stochastic Approximation”. Proceedings of the American Control Conference. 756-762.

[5] Robbins, H. and Monro,S. (1951). A stochastic approximation method. Ann. Math. Statist (22), no 3, 400-407.

[6] Saaty Thomas L., (2000), Fundamentals of Decision making and Priority Theory, RWS publications.

[7] Spall, J.C. (1992), "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," IEEE Trans. Automat. Control, 37, 332-341.

[8] Verikas, V. , Lipnickas, A. and Malmqvist, K. (2000). Fuzzy measures in neural network fusion. Proceedings of the 7th International Conference on Neural Information Processing, ICONIP-2000, 1152 -1157.

[9] Yin, G. (1999), "Rates of Convergence for a Class of Global Stochastic Optimization algorithms," SIAM J. Optim.,10, 99-120.

[10] Wang, J. and Wang, Z. (1997) Using neural network to determine Sugeno measure by statistics. Neural Network (10), No 1, 183-195.