# Exponential Average

Brian K. Schmidt

December 2015



MITRE

# What Is the Exponential Average?

- **A new kind of weighted average for combining capability (or utility) scores in a Decision Analysis tree**

$$eav_a(x_1, \ldots, x_n) = \log_a\left(\sum_{k=1}^{n} w_k \cdot a^{x_k}\right)$$

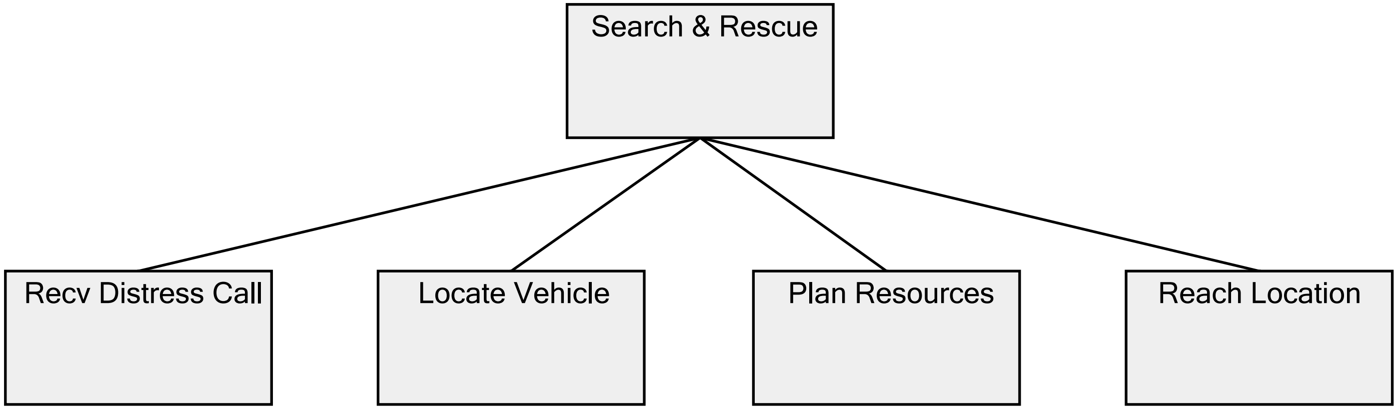
**where "a" is a constant**

**and $w_1, \ldots, w_n$ are weights**

# What’s Wrong with the Old Average?

- **The Imbalance Problem**
  – “Optimal” portfolios derived using a weighted average may neglect some capabilities while over-developing (“gold plating”) others

# Example: Simple Capability Tree

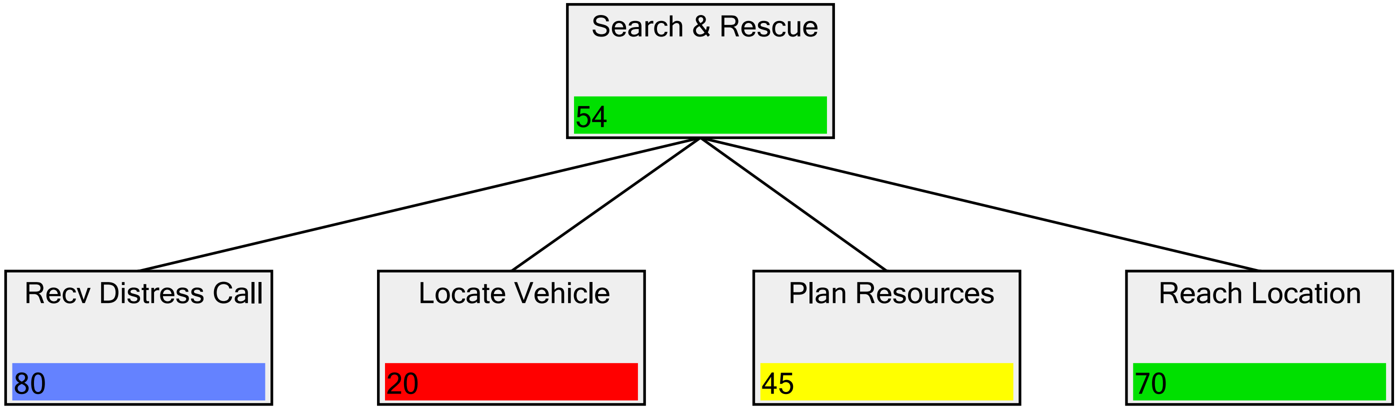



- **Search and rescue mission has been decomposed into four tasks**
- **Overall mission score is the weighted average of the scores of the tasks, with equal weights**

# Scale for Assessing Node Scores

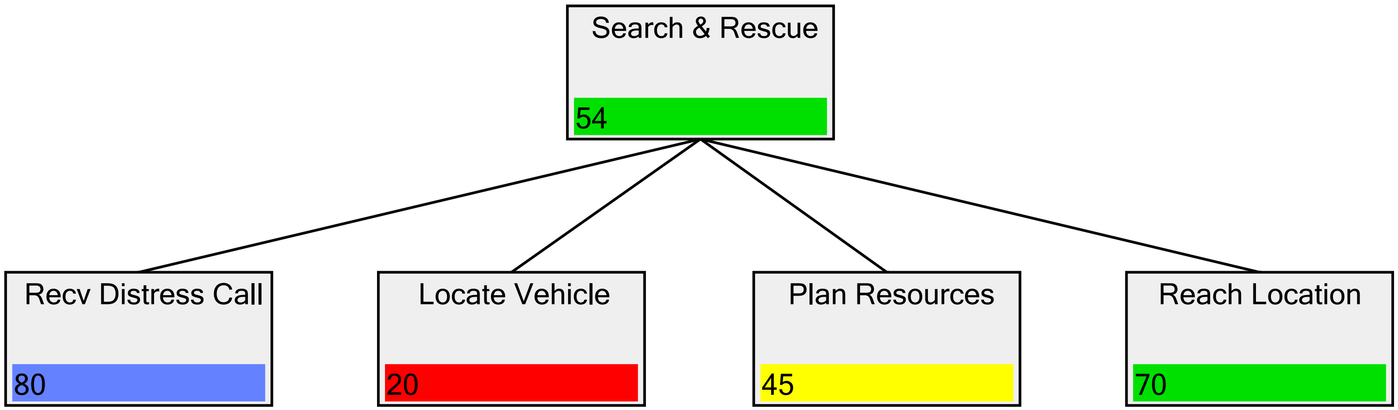
| Color | Interpretation | Numerical Range |
|---|---|---|
| Blue | Exceeds requirements | 75-100 |
| Green | Fully meets requirements | 50 - 75 |
| Yellow | Partially meets requirements | 25 - 50 |
| Red | Does not meet requirements | 0 - 25 |

# Current Situation




- **We are considering buying one of two upgrades**

# Which Should We Buy?




**becomes 41**

**becomes 95**

**Radar upgrade (cost 30)**

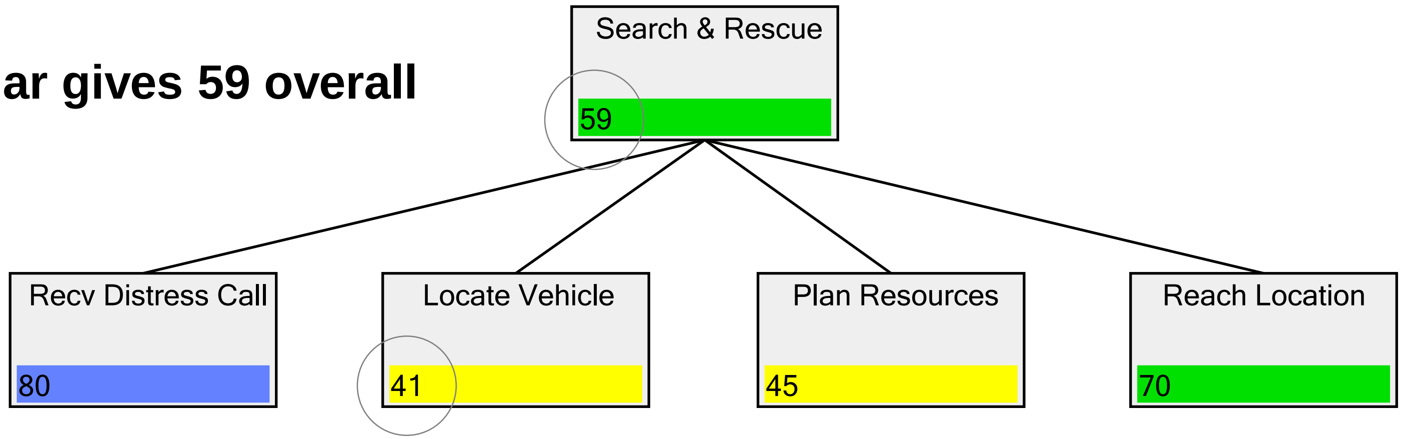
**This fixes a weakness.**

**Rescue vehicle upgrade (cost 30)**

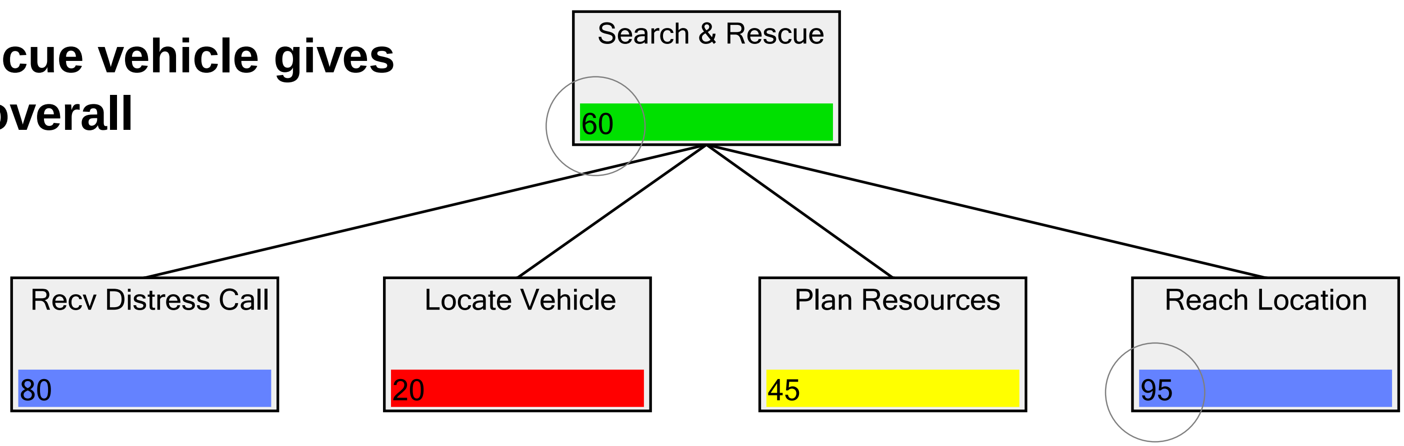
**This is gold-plating.**

# Comparison of Overall Scores

**Radar gives 59 overall**




**Rescue vehicle gives 60 overall**



# Outcome

- **So the “optimal” solution is gold plating**
- **This is disturbing**
- **This has happened in studies of real systems**

# Further Thoughts

- **We could avoid this problem if the overall score was the “min” (smallest value) of the task scores**
  - This eliminates gold plating, because if any task has score 0, the overall score is 0
  - But min is too harsh
    - We don’t get any credit for improving scores unless we improve the worst score
- **What we want is something “between” an average and a min**

# Approach: Generalized Average

- **Definition**

$$f(x_1, \ldots, x_n) = g^{-1}\left(\sum_{k=1}^{n} w_k \cdot g(x_k)\right)$$

- **g is called the “scaling function”**

- **Motivation:**
  - We want low values of $x_k$ to dominate the average. We can do this by “blowing them up” with g(x).

# Examples of Generalized Averages

| Scaling Function | Name of Average | Domain |
|---|---|---|
| $g(x) = x$ | regular average | $(-\infty, \infty)$ |
| $g(x) = x^2$ | root mean square | $[0, \infty)$ |
| $g(x) = x^p$ | power mean | $[0, \infty)$ |
| $g(x) = \log(x)$ | geometric mean | $(0, \infty)$ |
| $g(x) = 1/x$ | harmonic mean | $(0, \infty)$ |

# Generalized Average: Basic Properties

- **Stays in range**
  – The answer is between min and max of the x's
- **Associative**
  – $f( f(x_1,x_2), y ) = f( x_1, x_2, y )$ (when weights are adjusted)
- **Preferentially independent**
  – This is an important Decision Analysis property
  – Definition: Pref. Ind. means that the preferences between $x_j$ and $x_k$ do not depend on the values of the other x's

# Motivation for Choice of g(x)

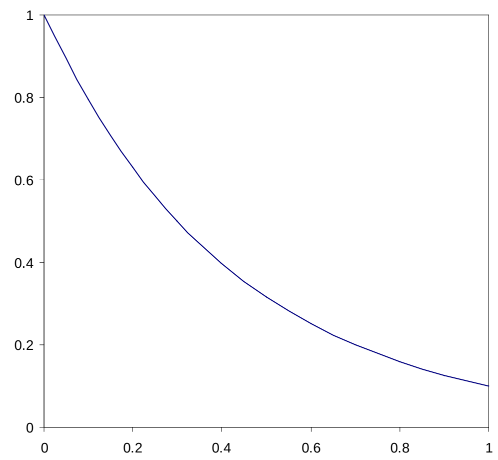
- **We want "far smaller" values to dominate the average**
  - Suppose "far smaller" means "less by U"
  - Suppose "dominate" means g(x) is twice as big
  - Then we want g(x-U) = 2g(x), g(x-2U) = 4g(x), g(x-3U) = 8g(x), …
  - This suggests $g(x) = a^x$

# Exponential Average

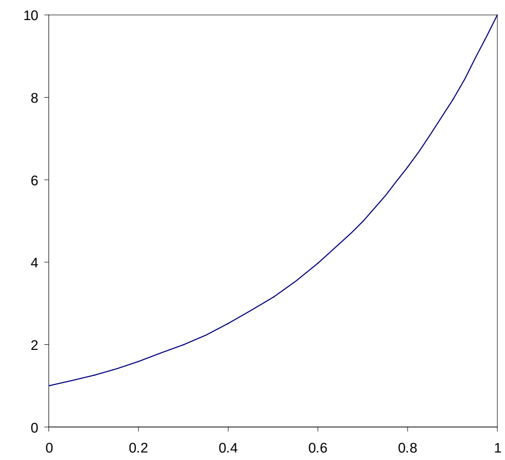
- **Select g(x) = $a^x$**

- **Properties**
  - As a → 0, eav → min
  - As a → 1, eav → regular weighted average
  - As a → infinity, eav → max

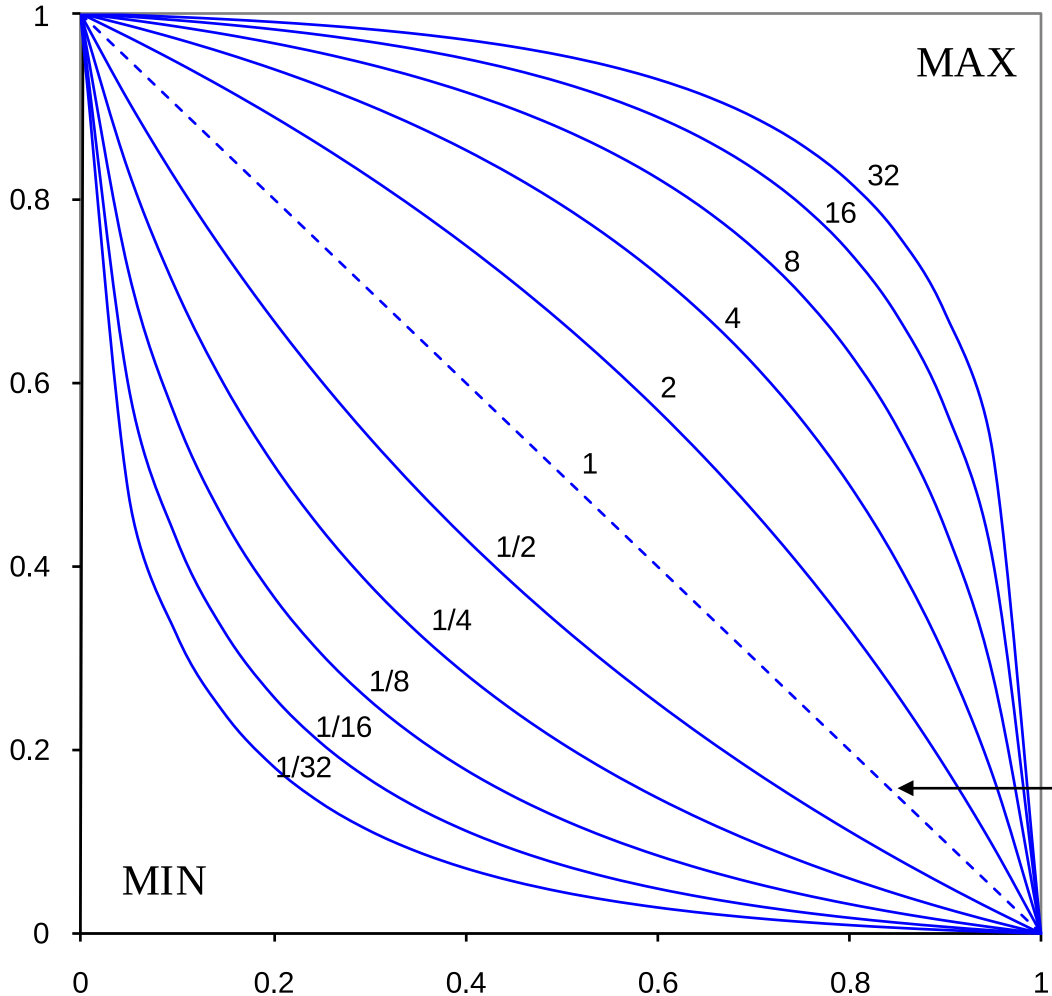
**(a = 0.1)**




**(a = 10)**



# Tradeoff Curves for Exponential Average




**Graph shows all points with same goodness as (0,1)**

**Value of "a" is written beside each curve**

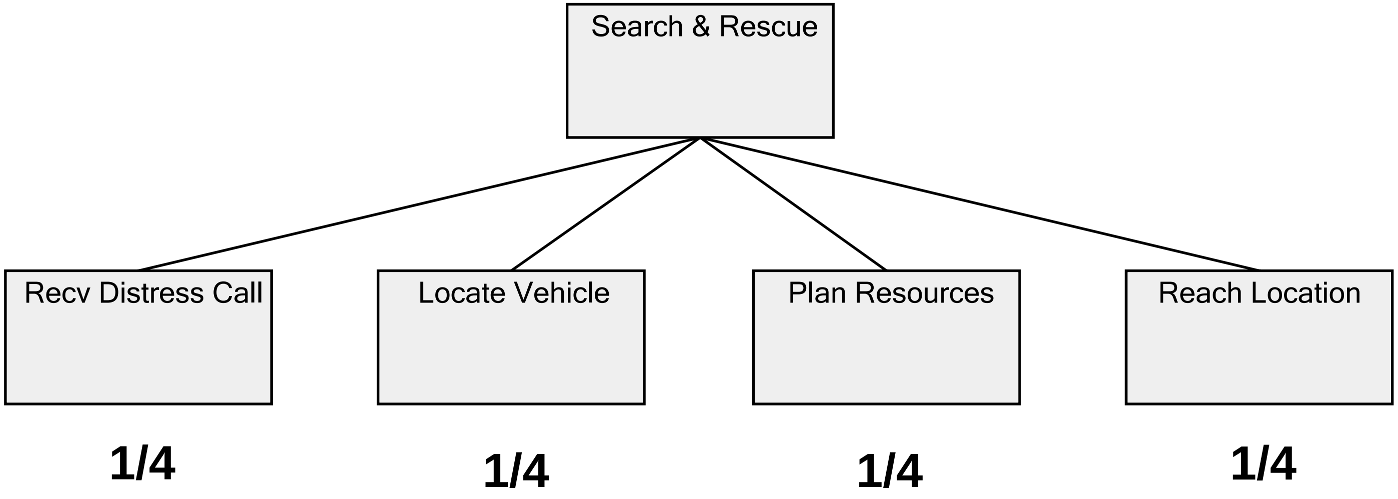
**Regular average**

# Is This What We Are Looking For?

- **Maybe... because:**
  - We can tune it anywhere between min and average
  - It has all the nice properties of generalized averages
- **Side benefit**
  - We can also tune it between max and average
    - Good for analyzing risk
- **Before we try it, let's get a better understanding of how it works**

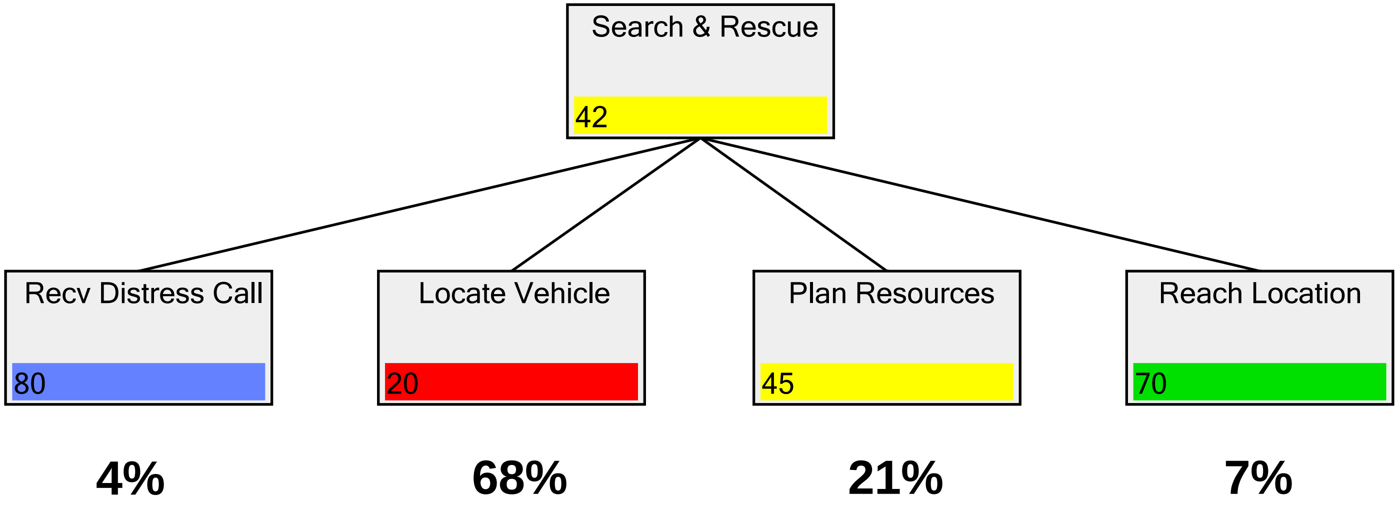
# The Concept of Return

- **Define the "return" of a task to be the partial derivative of overall score with respect to task score**
- **Ultimate problem with regular weighted average:**
  - Task returns stay the same, regardless of excess capability in some areas



# But with Exponential Average...

- Returns change as the portfolio changes
- Neglected tasks have high returns



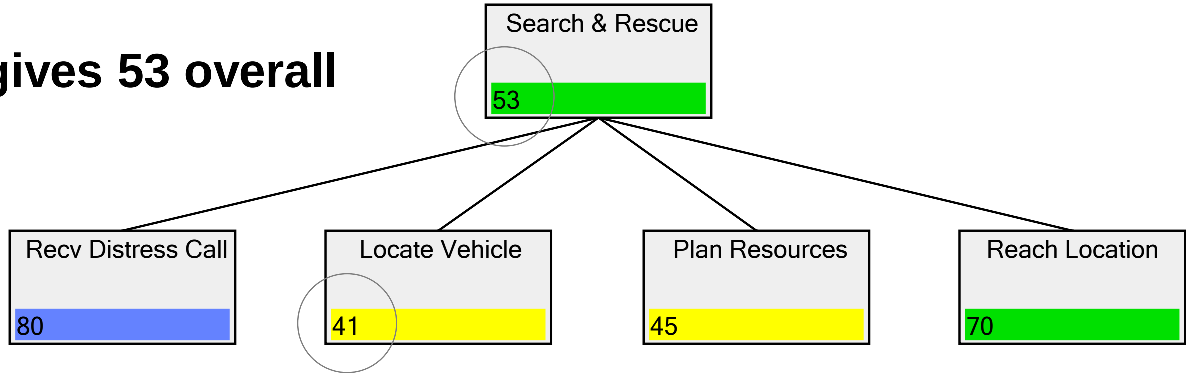

**(a = 0.955)**

**See backup slides for the formula for computing returns**

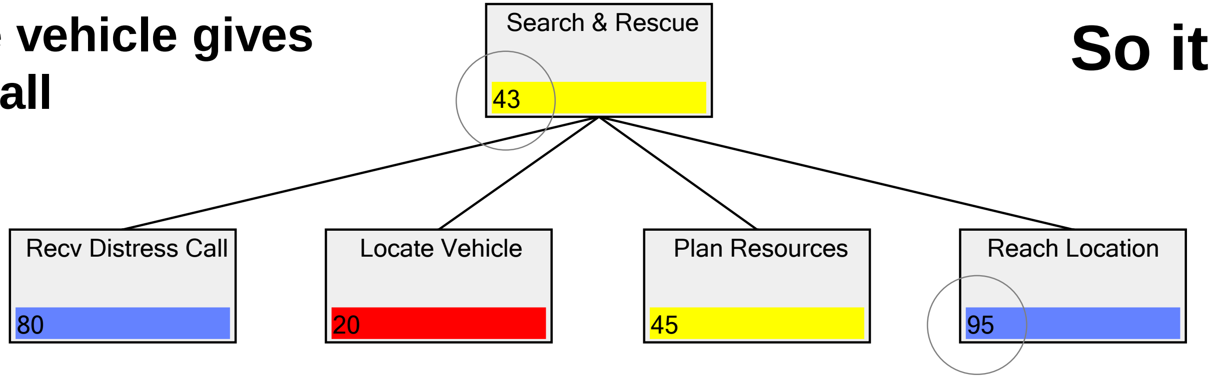
# Result

- **With exponential average, the optimal solution is to fix what is broken**

**Radar gives 53 overall**




**Rescue vehicle gives 43 overall**




**So it works!**

# And Did You Notice?

- **Returns for tasks add up to 1**
  - – Echoes a property of regular average
    - For regular average, the returns are the weights
  - – Is surprising (a type of linearity in a non-linear function)
  - – Makes the returns more meaningful
    - We can think of the return as the relative importance of improving the given task

# But How Do I Set the Constant “a”?

- **Swing Weighting**
  - User specifies a table of child (task) scores, with the parent (overall) score for each
  - From this data, we can solve for “a” and the weights

# Example: Swing Weighting

| $x_1$ | $x_2$ | $x_3$ | overall score |
|---|---|---|---|
| 50 | 80 | 80 | 60 |
| 80 | 50 | 80 | 60 |
| 80 | 80 | 50 | 70 |

- **After some algebra*, we get the equation:**

$$2a^{30} - a^{20} - 2a^{10} + 1 = 0$$

- **Solution:**
  – Weights (3/7, 3/7, 1/7), a = 0.933

* **See backup slides for details**

# Advantages of Swing Weighting

- **We get the weights as well as “a”**
  - Weights have always been hard to justify
  - Swing weighting is a widely accepted Decision Analysis technique

# But What Is So Special About THIS Average?

- **Why not use some other concept?**
  – We could create hundreds of other generalized averages by choosing arbitrary scaling functions g(x)

# Uniqueness Theorem

- **The exponential average is the only generalized average (other than the regular average) satisfying either of the conditions below:**
  - $f( x_1+u, \ldots, x_n+u ) = f( x_1, \ldots, x_n ) + u$
  - The returns add up to 1

$$\sum_{k=1}^{n} \partial f / \partial x_k = 1$$

**These conditions pertain to rescaling and interpretation of returns, two important concepts**

MITRE

# Exponential Average: Summary

- **Still a kind of weighted average**
- **Behavior is continuously adjustable between weighted average and min**
- **Also adjustable to resemble max**
- **Only one additional parameter needed**
- **Associative**
  - Several children (tasks) can be treated as one
- **Preferentially independent**
  - This is an important Decision Analysis property
- **Behaves well under changes of scale (see backup slides)**
- **Parameter “a” and weights can be found by swing weighting**
- **Unique**
  - No other generalized average has the same properties

# Backup Slides

# Returns for Exponential Average

- **Differentiating the formula for eav gives:**

$$\partial f / \partial x_k = \frac{\mathrm{w_k}\, a^{x_k}}{\sum_{i=1}^{n} \mathrm{w_i}\, a^{x_i}}$$

- **Hence it is easy to confirm that**

$$\sum_{k=1}^{n} \partial f / \partial x_k = 1$$

# Some Properties of Exponential Average

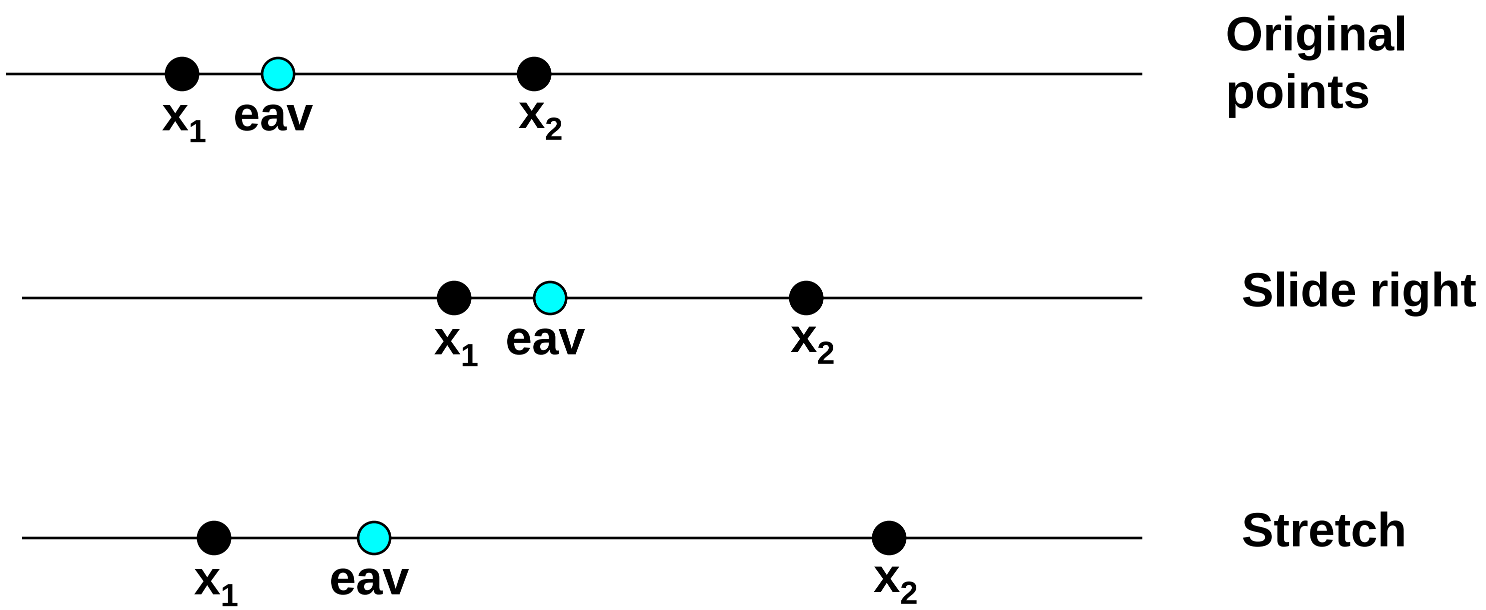
- **Behaves well under changes of scale:**

$$eav(\ x_1+u,\ \ldots,\ x_n+u\ ) = eav(\ x_1,\ \ldots,\ x_n\ ) + \mathrm{u}$$

$$eav_a(\ r \cdot x_1,\ \ldots,\ r \cdot x_n\ ) = r \cdot eav_{a^r}\ (\ x_1,\ \ldots,\ x_n\ )$$

- **This is important because:**
  – Changing scale is common in practical situations
  – In Decision Analysis, “value” is defined on an arbitrary scale

# Geometry of Rescaling




- **Relative position of average is maintained**
  - Requires adjusting “a” when we stretch

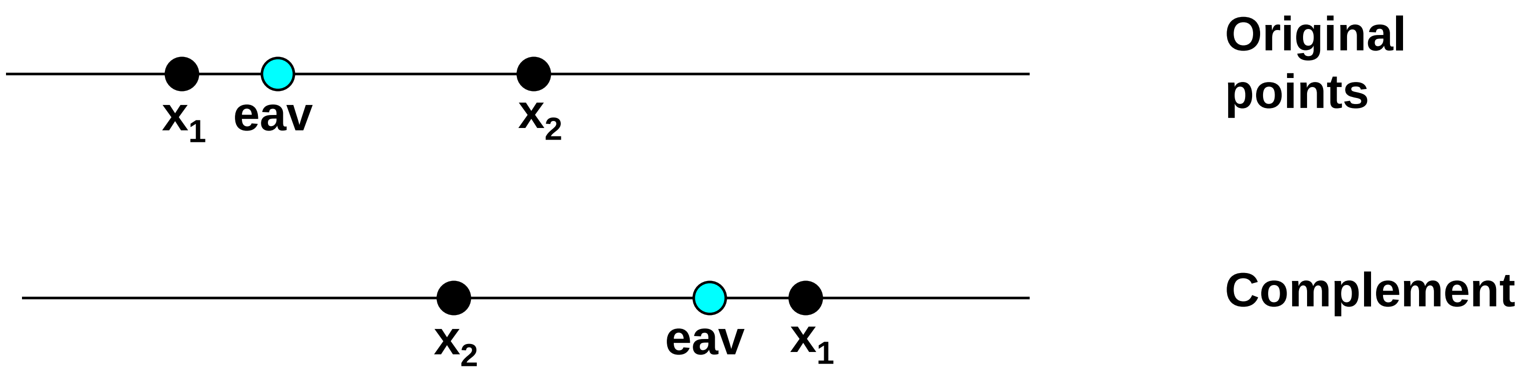
# More on Rescaling

- **Behaves well under complement:**

$$eav_a(\, 100\text{-}x_1, \ldots, 100\text{-}x_n \,) = 100 - eav_{1/a}(\, x_1, \ldots, x_n \,)$$

- **This is useful for relating two models:**
  - capability (rule is like min)
  - risk (rule is like max)

# Geometry of Complement




- **Relative position of average is maintained**
  – Requires taking reciprocal of “a”

# Algebra of Swing Weights Example - 1

- **These slides explain how to solve for the weights and the parameter “a” from a swing table**
- **To simplify the algebra, we add a row of all 80’s at the beginning**
  - This is valid, because the average of all 80’s is 80

| $x_1$ | $x_2$ | $x_3$ | overall score |
|---|---|---|---|
| 80 | 80 | 80 | 80 |
| 50 | 80 | 80 | 60 |
| 80 | 50 | 80 | 60 |
| 80 | 80 | 50 | 70 |

# Algebra Example - 2

**The table gives us the equations:**

$$a^{80} w_1 + a^{80} w_2 + a^{80} w_3 = a^{80}$$
$$a^{50} w_1 + a^{80} w_2 + a^{80} w_3 = a^{60}$$
$$a^{80} w_1 + a^{50} w_2 + a^{80} w_3 = a^{60}$$
$$a^{80} w_1 + a^{80} w_2 + a^{50} w_3 = a^{70}$$

**Subtracting row 2 from row 1 gives:**

$$(a^{80} - a^{50}) w_1 = a^{80} - a^{60}$$
$$w_1 = (a^{80} - a^{60}) / (a^{80} - a^{50})$$

**Similarly:**

$$w_2 = (a^{80} - a^{60}) / (a^{80} - a^{50})$$
$$w_3 = (a^{80} - a^{70}) / (a^{80} - a^{50})$$

# Algebra Example - 3

**Now:**

$w_1 + w_2 + w_3 = 1$

**So:**

$(a^{80} - a^{60}) / (a^{80} - a^{50}) + (a^{80} - a^{60}) / (a^{80} - a^{50}) + (a^{80} - a^{70}) / (a^{80} - a^{50}) = 1$

**This gives:**

$(a^{80} - a^{60}) + (a^{80} - a^{60}) + (a^{80} - a^{70}) = a^{80} - a^{50}$

$3*a^{80} - 2*a^{60} - a^{70} = a^{80} - a^{50}$

$2*a^{80} - a^{70} - 2*a^{60} + a^{50} = 0$

# Algebra of Swing Weights - Theory

- More generally, the equation is:

$$(n-1)*a^{xregular} - \sum ( a^{yreduced} ) + a^{xreduced} = 0$$

- The solution always exists and is unique (if we ignore the trivial solution a=1)
- Numerical methods must be used to solve

# Swing Weights - Extreme Cases

- **In extreme cases, the solution for "a" may be too small or too large to be practical**
- **In these cases, changing the scale may correct the problem.  Or we may use a different rollup function: e.g., min, max, or a different generalized average**