Improving AI Decision Modeling Through Utility Theory

Dave Mark
President & Lead Designer
Intrinsic Algorithm LLC

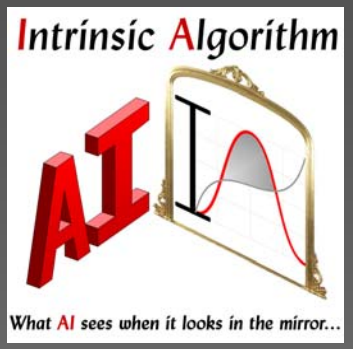
Kevin Dill
AI Engineer
Lockheed Martin

AI Summit GDC 2010

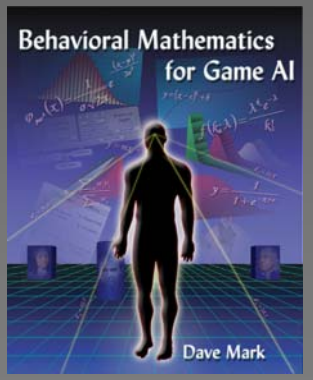
AI GAME PROGRAMMERS GUILD

Dave Mark

- President & Lead Designer of
  - Intrinsic Algorithm LLC, Omaha, NE
    - Independent Game Studio
    - AI Consulting Company



- Author of
  - Behavioral Mathematics for Game AI



AIGameDev.net:
Trends for 2009 in Retrospect

http://aigamedev.com/open/editorial/2009-retrospective/

What's new in 2009 is:

1. There's now an agreed-upon name for this architecture: utility-based, which is much more reflective of how it works. Previous names, such as "Goal-Based Architectures" that Kevin Dill used were particularly overloaded already.

2. A group of developers advocate building entire architectures around utility, and not only sprinkling these old-school scoring-systems around your AI as you need them.

The second point is probably the most controversial.

We do requests…

"Wow… you've got a lot of stuff on utility modeling in here…

You should do a lecture on this stuff at the AI Summit."

Daniel Kline
Outside P. F. Chang's
Stanford Mall
October 2009

What is “Utility Theory”?

http://en.wikipedia.org/wiki/Utility

In economics, **utility** is a measure of the relative satisfaction from, or desirability of, consumption of various goods and services.

Given this measure, one may speak meaningfully of increasing or decreasing utility, and thereby explain economic behavior in terms of attempts to increase one's utility.

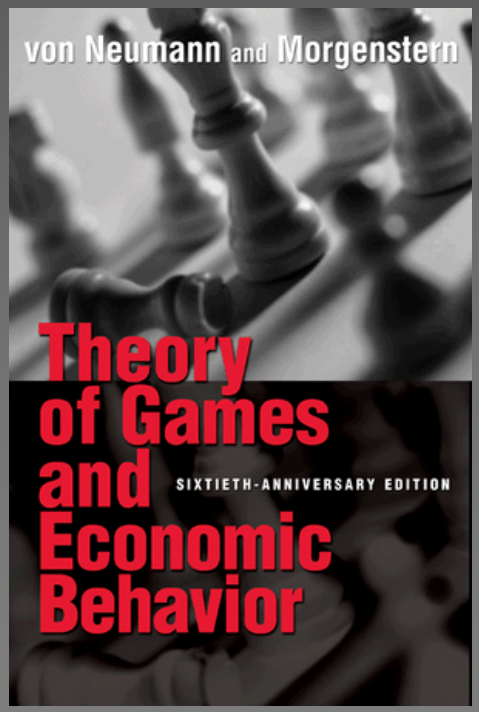
What is “Utility Theory”?

- How much is something worth to me?
- Not necessarily equal to “value”
  - E.g. $20 might mean more or less than $20
- Allows comparisons between concepts
- Allows decision analyses between competing interests
- “Maximization of expected utility”

What is “Utility Theory”?

- Related to…
  - Game theory
  - Decision theory
- Used by…
  - Economics
  - Business
  - Psychology
  - Biology

John von Neumann

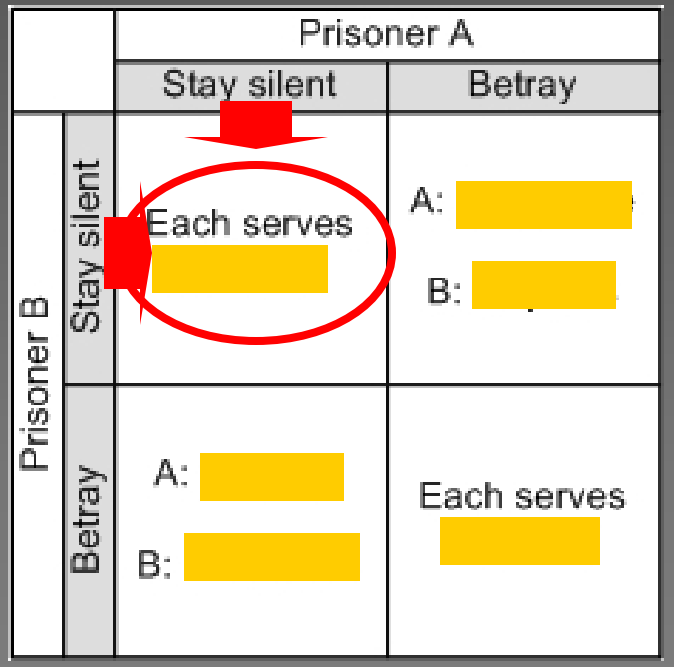


Value Allows Analysis

- Converting raw numbers to usable concepts
  - Distance
  - Ammo
  - Health
- Converting raw numbers to *useful* concepts
  - Distance → Threat
  - Ammo → Reload Necessity
  - Health → Heal Necessity

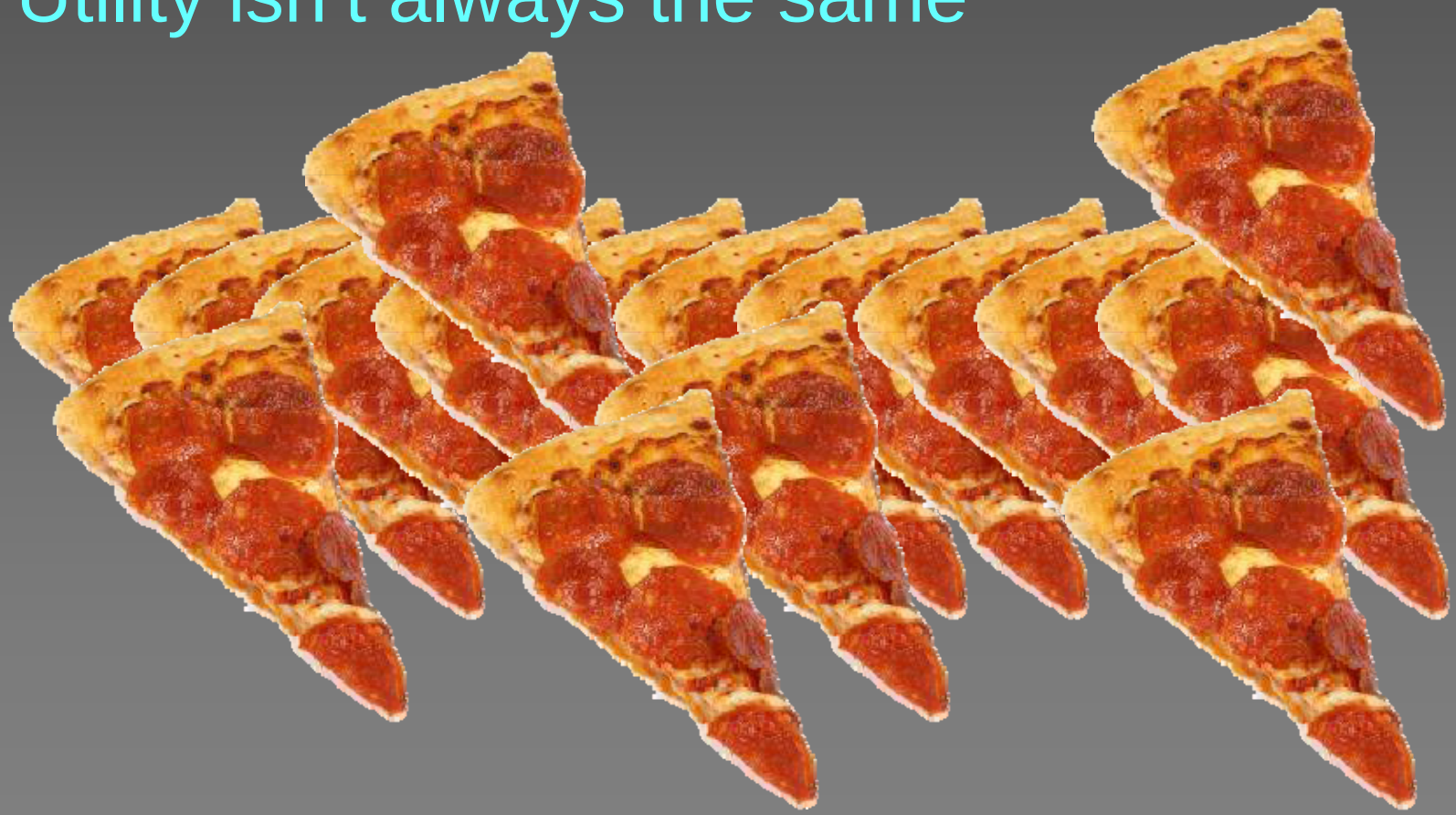
Value Allows Comparisons

- By assigning value to a selection, we can compare it to others
- Von Neumann and Morgenstern's game theory
- Without value, comparisons are difficult… or even impossible!



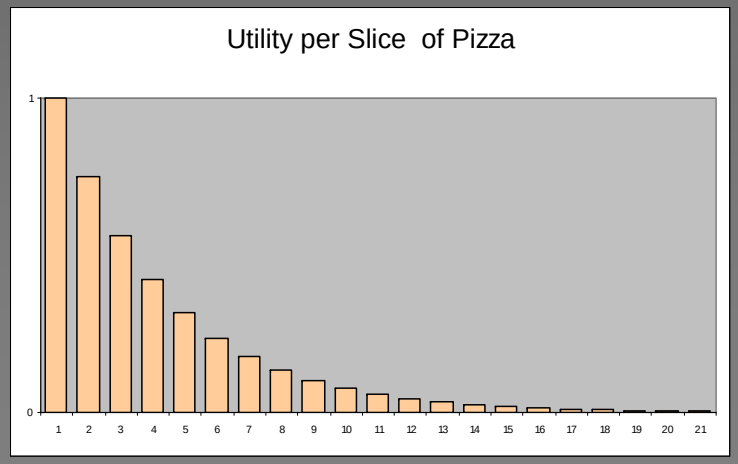
Marginal Utility

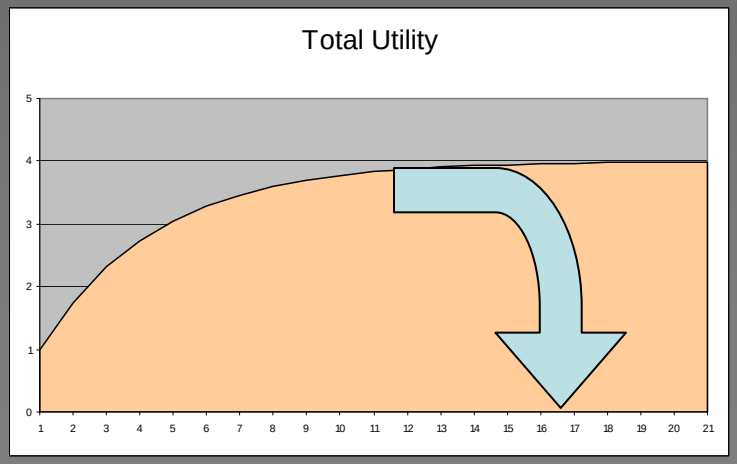
- Utility isn't always the same



Marginal Utility

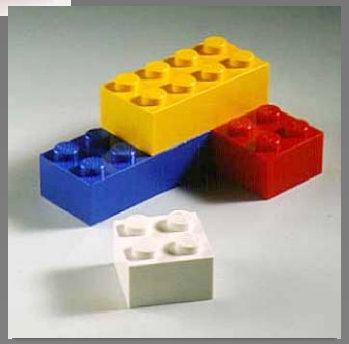
- Decreasing Marginal Utility
  - Each additional unit is worth less than the one before
  - The rate of increase of the *total utility* decreases
  - Utility of 20 slices != 20 * Utility of 1 slice

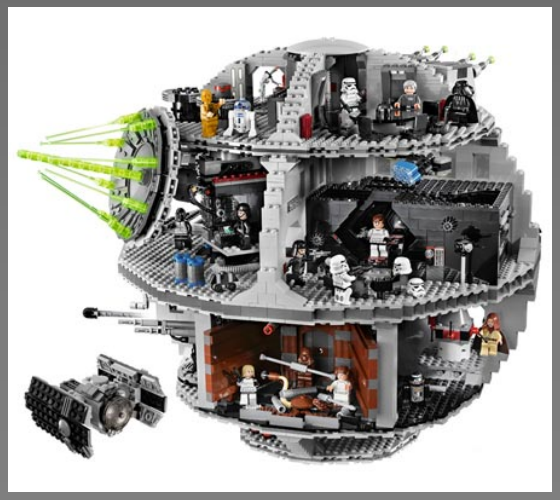




Marginal Utility

- Increasing Marginal Utility
  - Each additional unit is worth more than the one before
  - The rate of increase of the *total utility* increases
  - Utility of 20 Lego != 20 * Utility of 1 Lego





Converting Data to Concepts

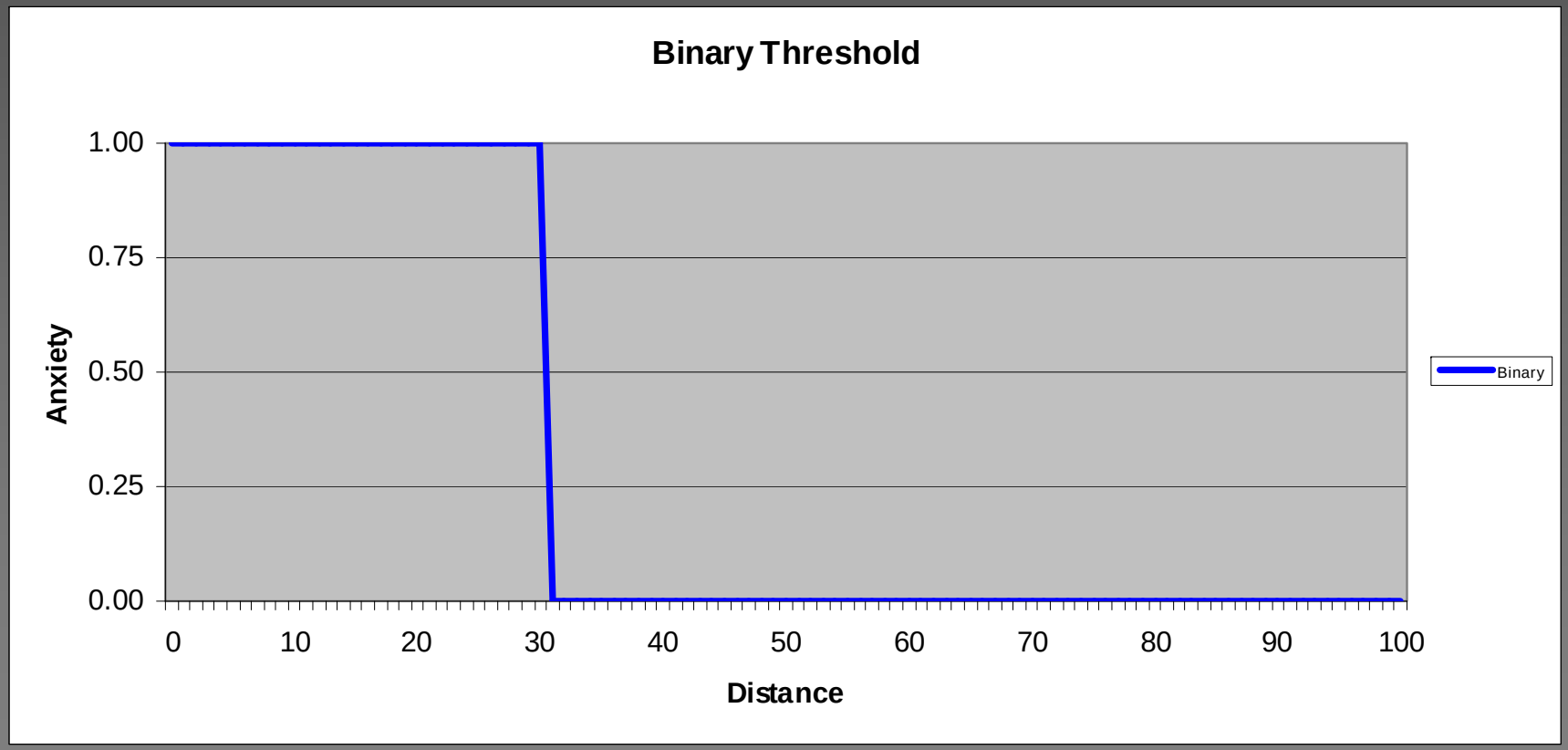
- What does the information say?
- Raw data doesn't mean much without context
- If data is ambiguous, we can't reason on it
- Various techniques to make sense of raw data
  - Conversion formulas
  - Response curves
  - Normalization (e.g. 0..1)

Processing One Piece of Info

As the distance changes, how much anxiety do you have?

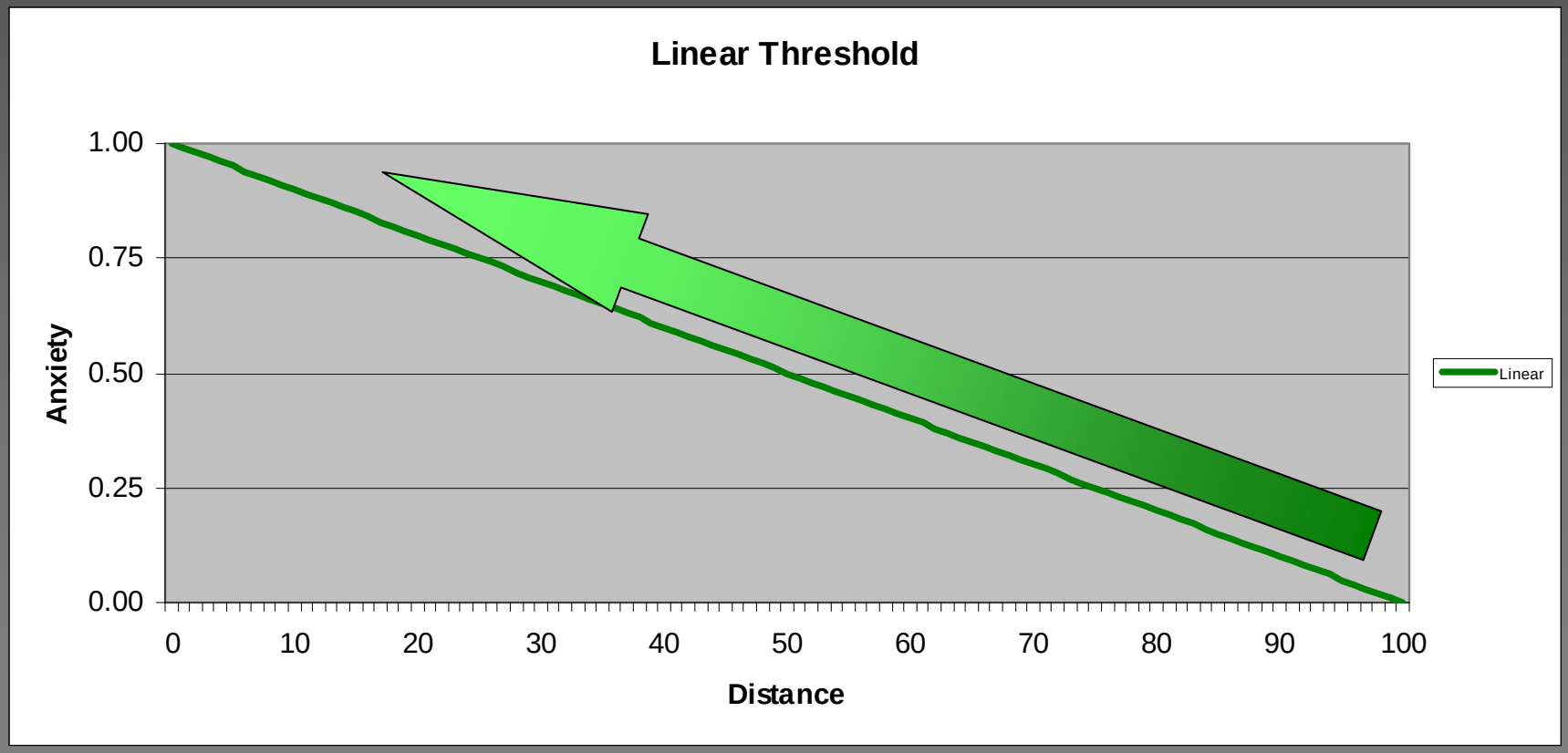
Simple Rule

If distance <= 30 then anxiety = 1



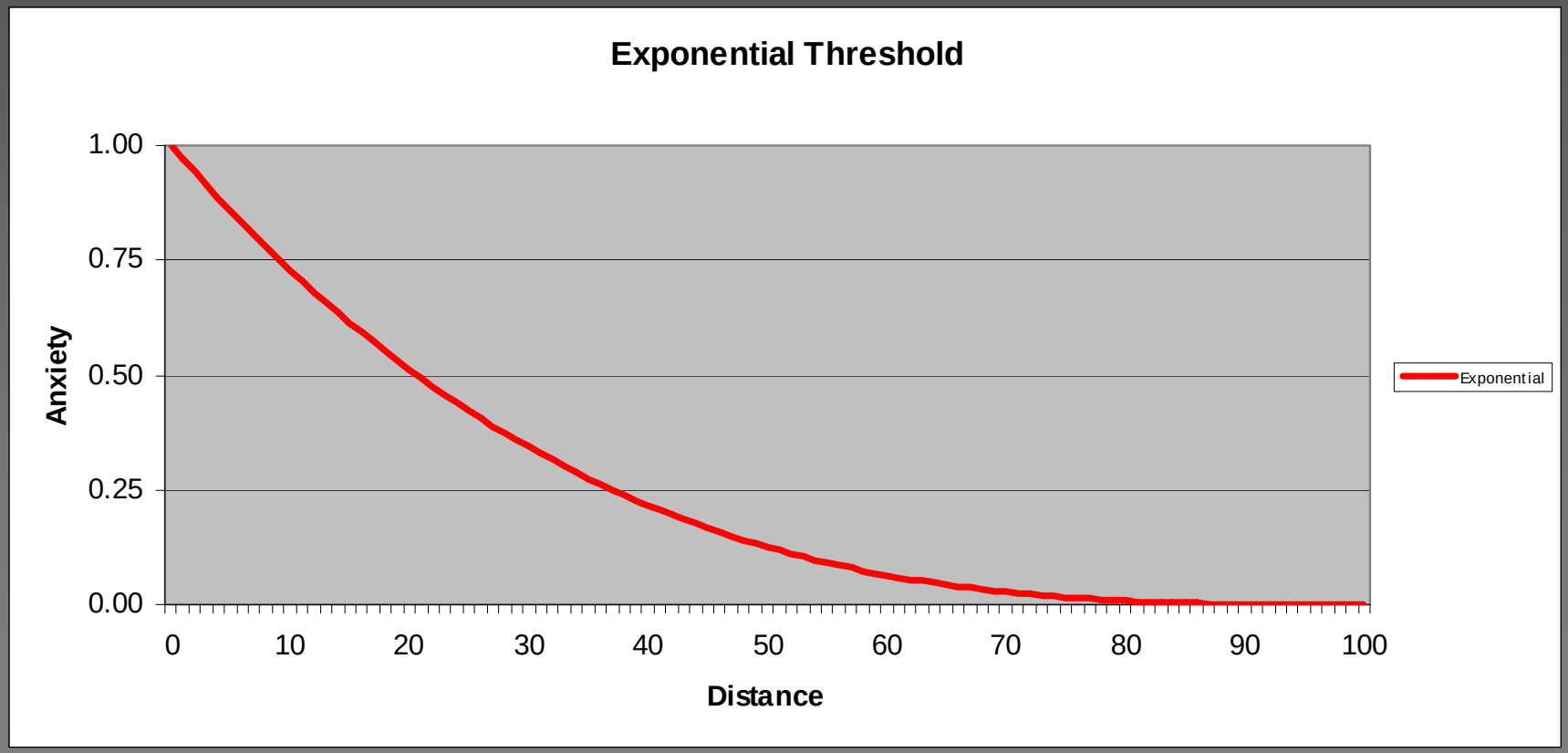
Linear Formula

Anxiety = (100 – distance) / 100



Exponential Formula

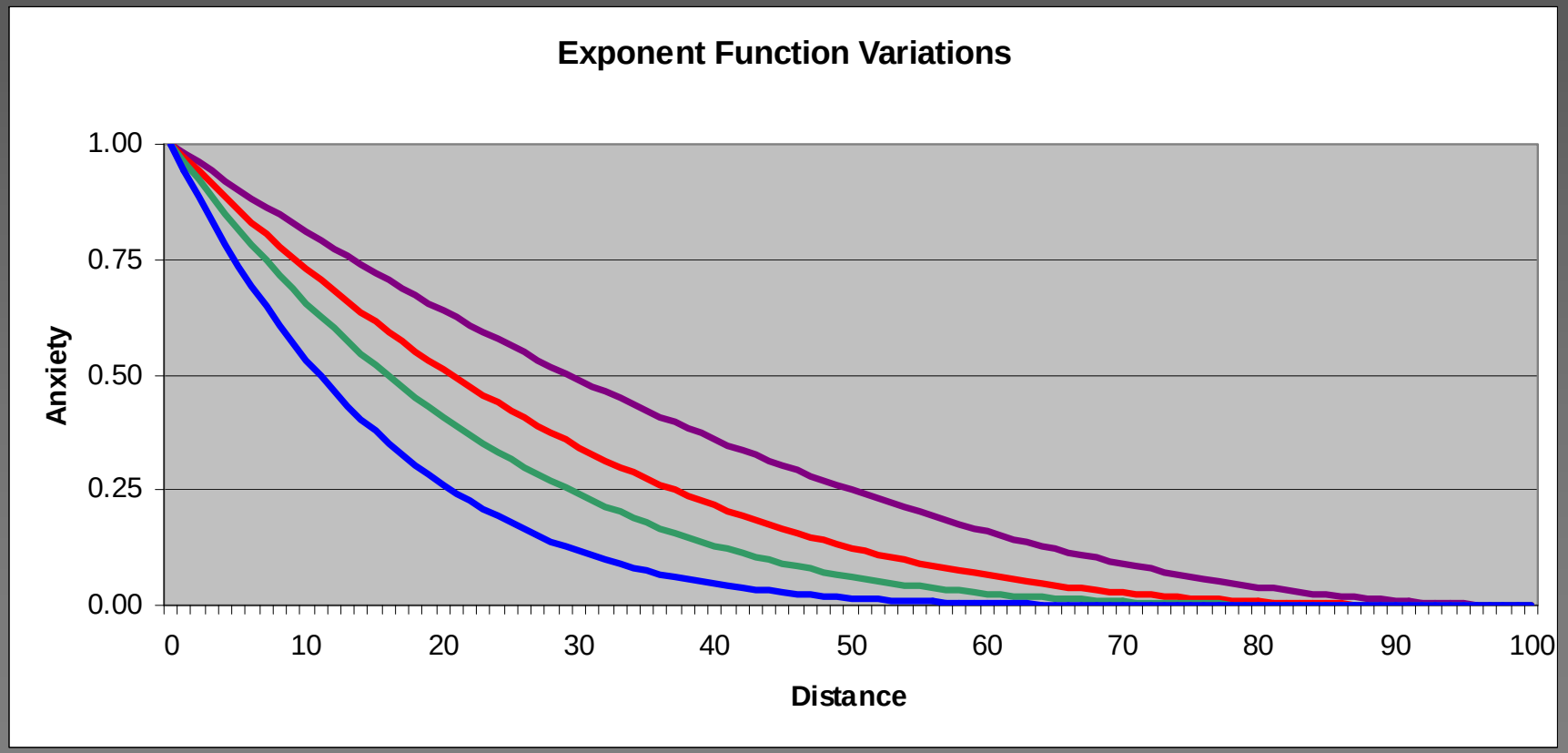
Anxiety = $(100 - \text{distance}^3) / (100^3)$



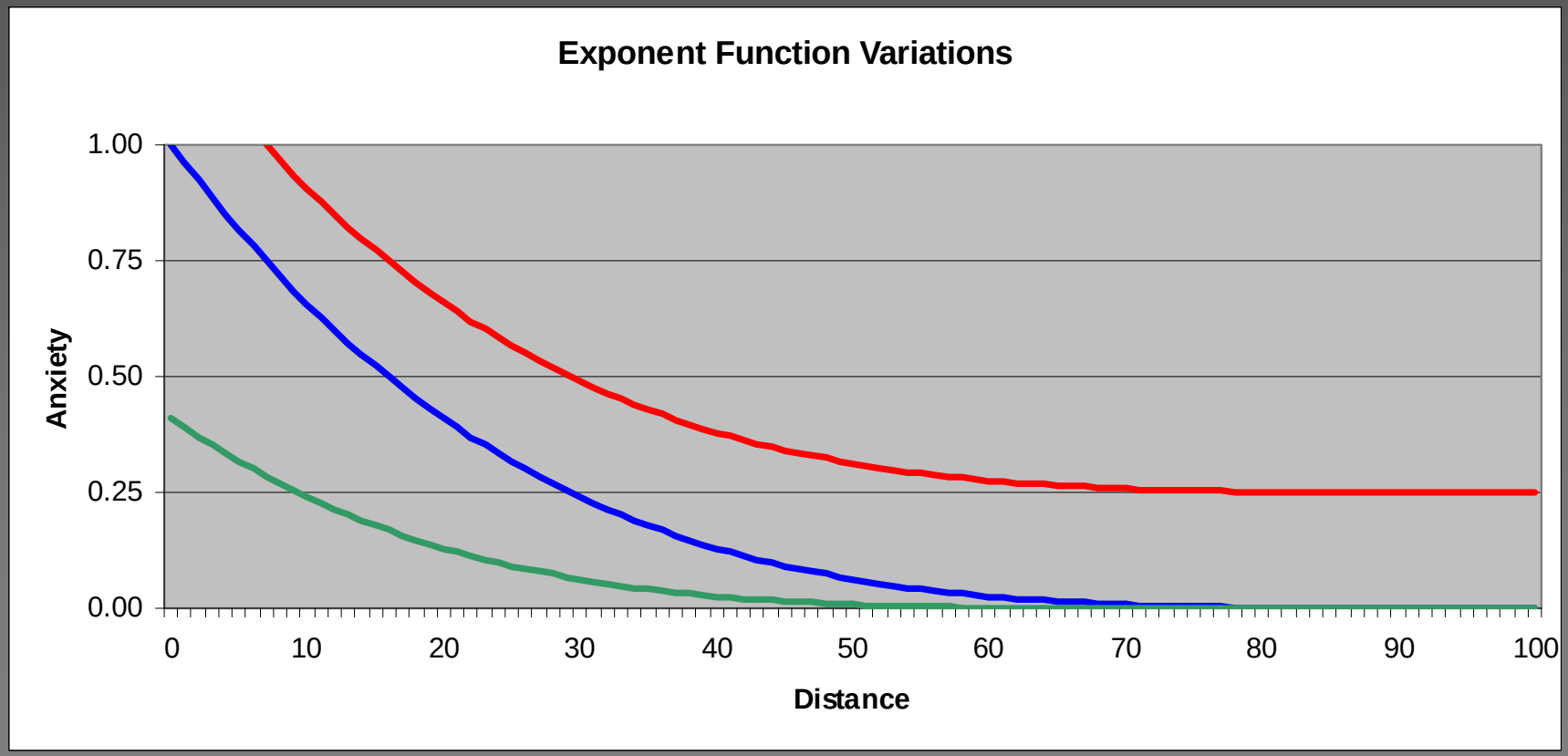
Changing Exponents

Anxiety = $(100 - \text{distance}^k) / (100^k)$

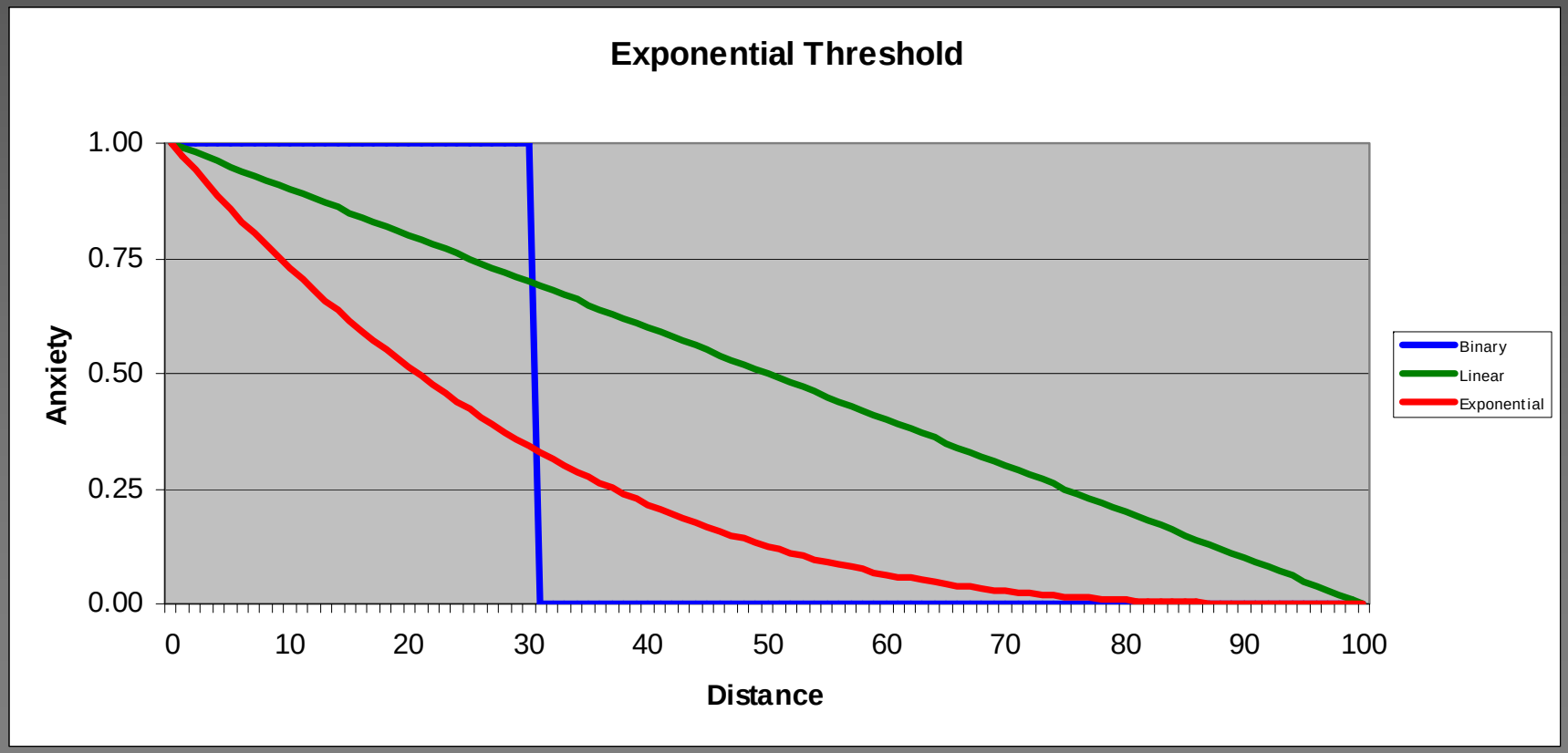
k = 2, 3, 4, 6



Shifting the Curve



Threshold / Linear/ Exponential



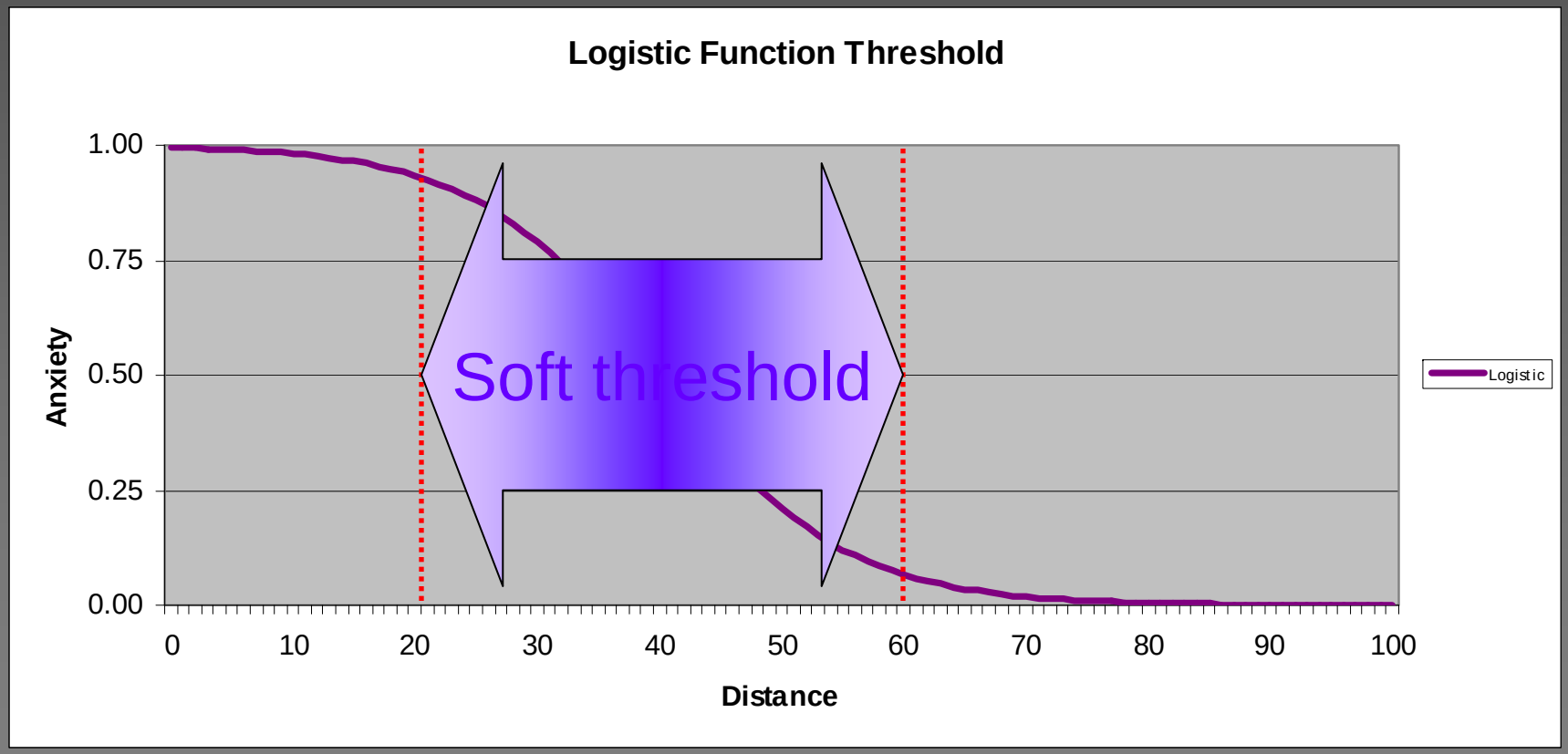
Logistic Function

(One of the sigmoid – or "s-shaped" – functions)

$$y = \frac{1}{1 + e^{-x}}$$

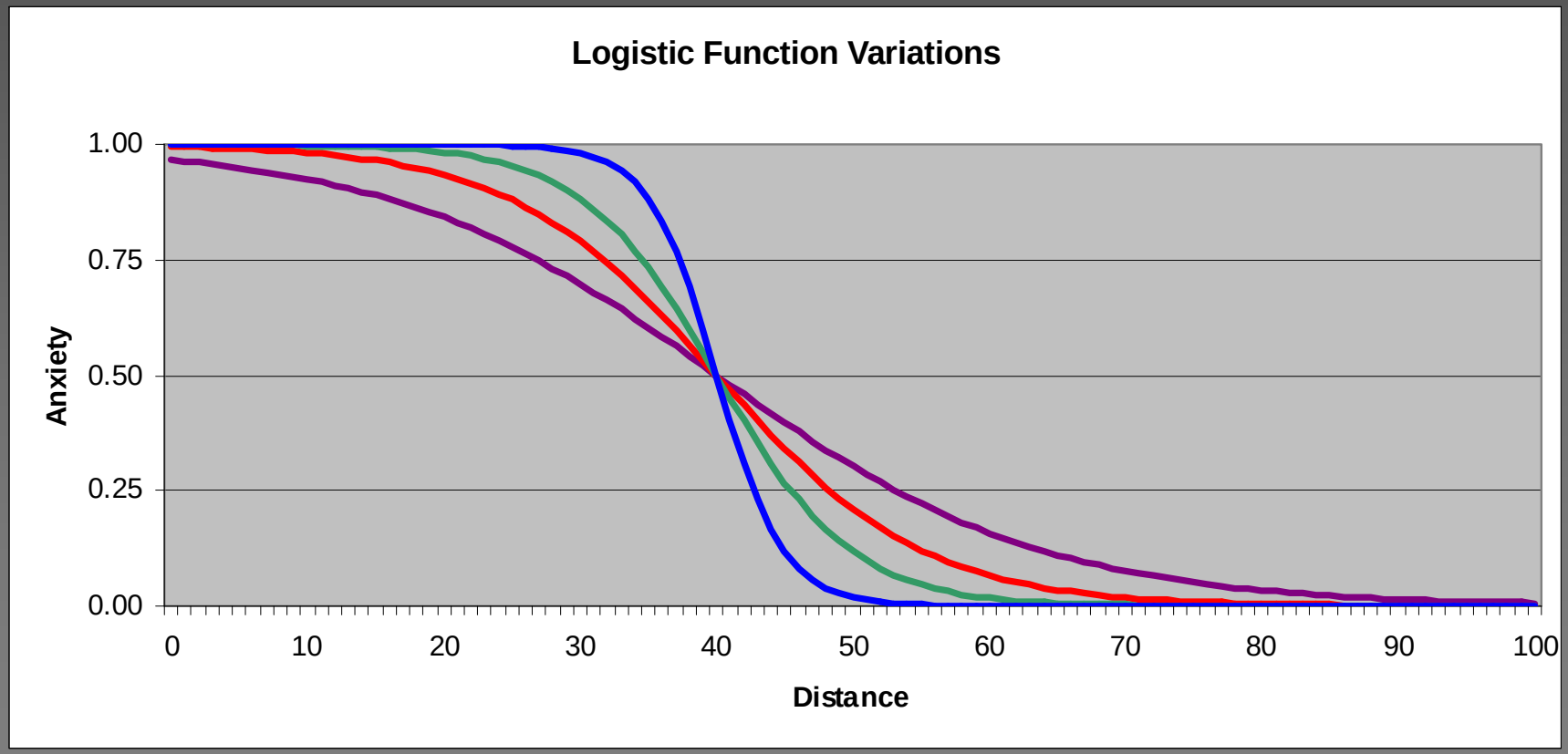
Logistic Function

Anxiety = $1/(1+(2.718 \times 0.45)^{distance+40})$



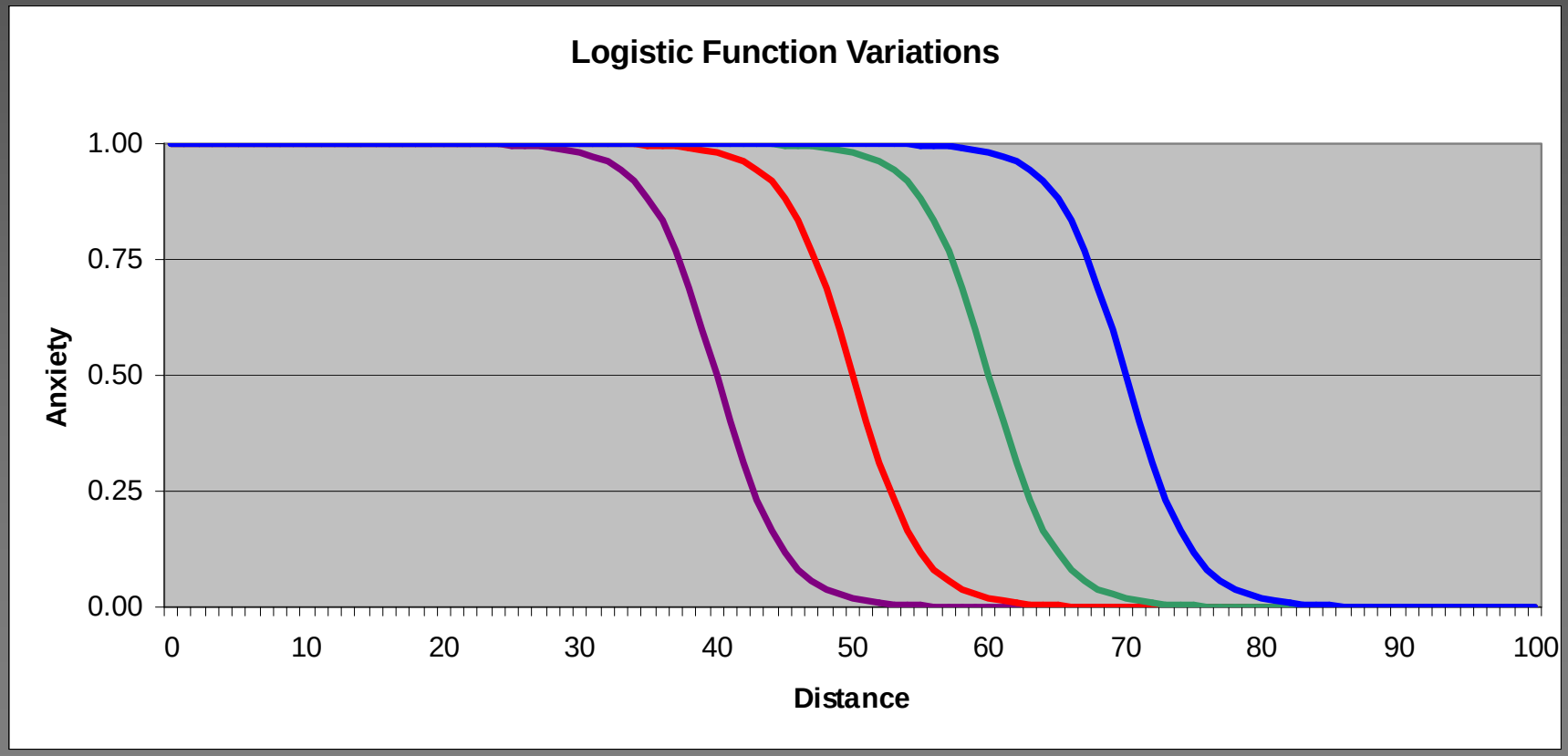
Variations on the Logistic Curve

$$\text{Anxiety} = 1/(1+(2.718 \times 0.45)^{\text{distance}+40})$$

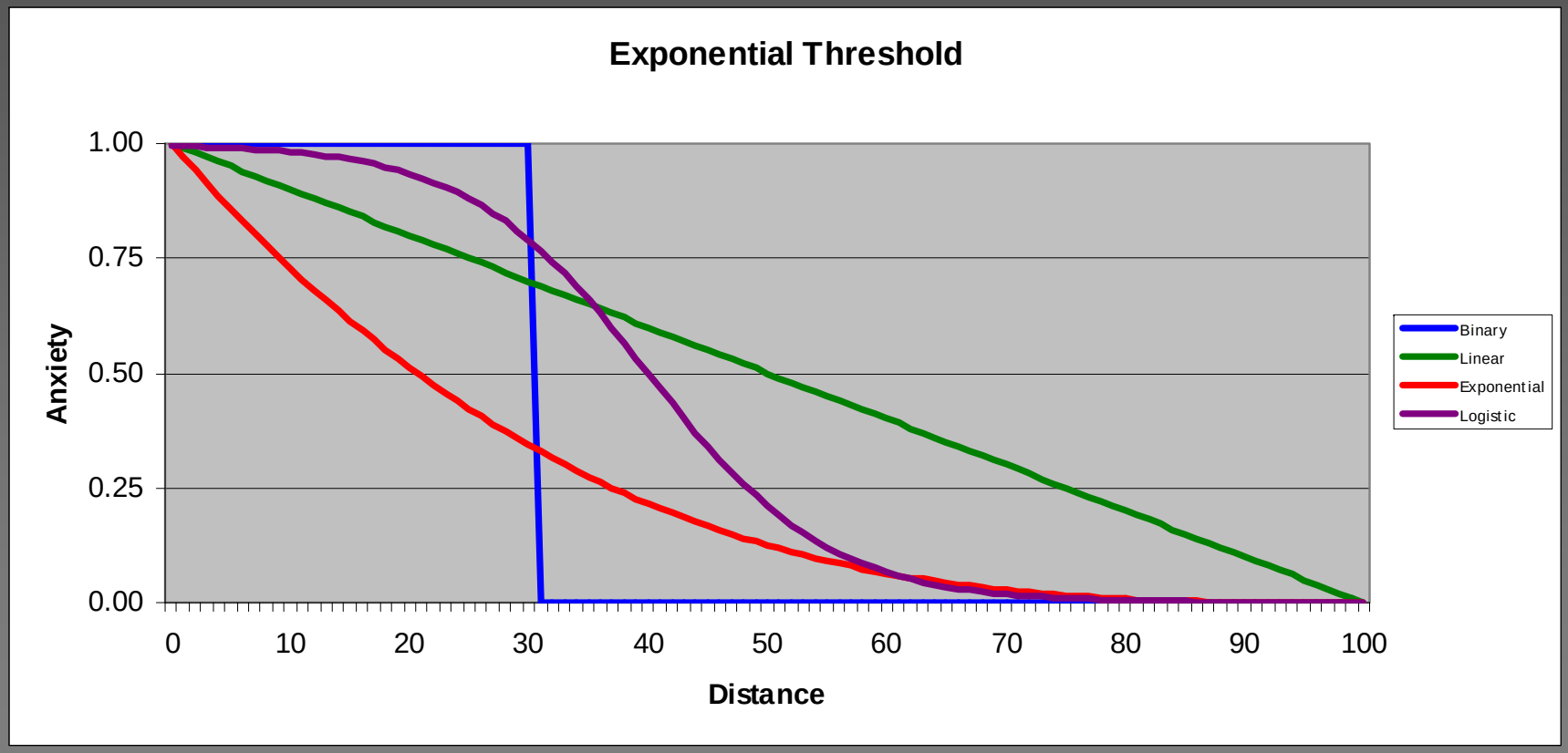


Shifting the Logistic Function

$$\text{Anxiety} = 1/(1+(2.718 \times 0.45)^{\text{distance}+40})$$



Curve Comparison

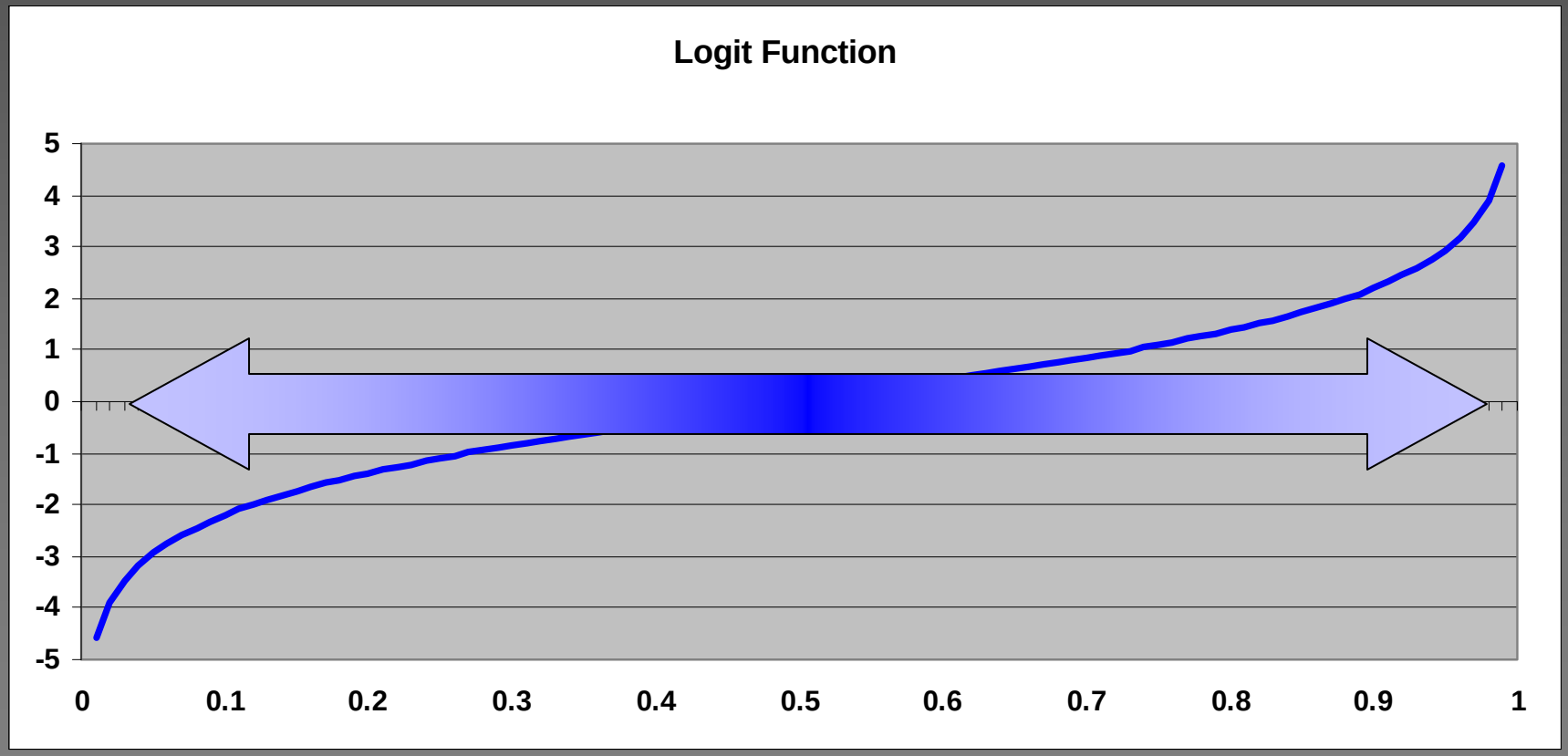


Logit Function

$$y = \log_e \frac{1}{1-x}$$

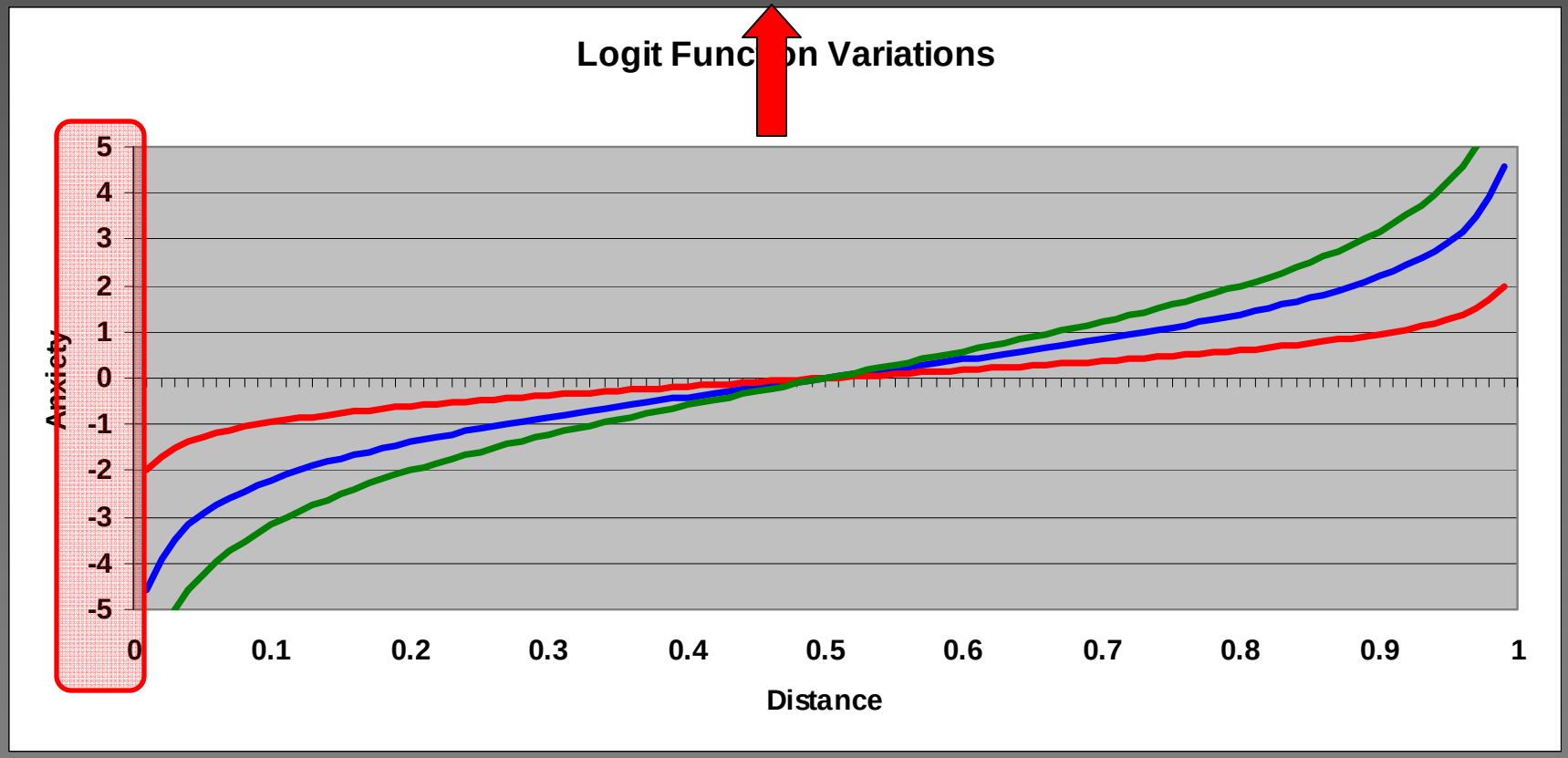
Logit Function

$$y = \log_e(x/(1- x))$$



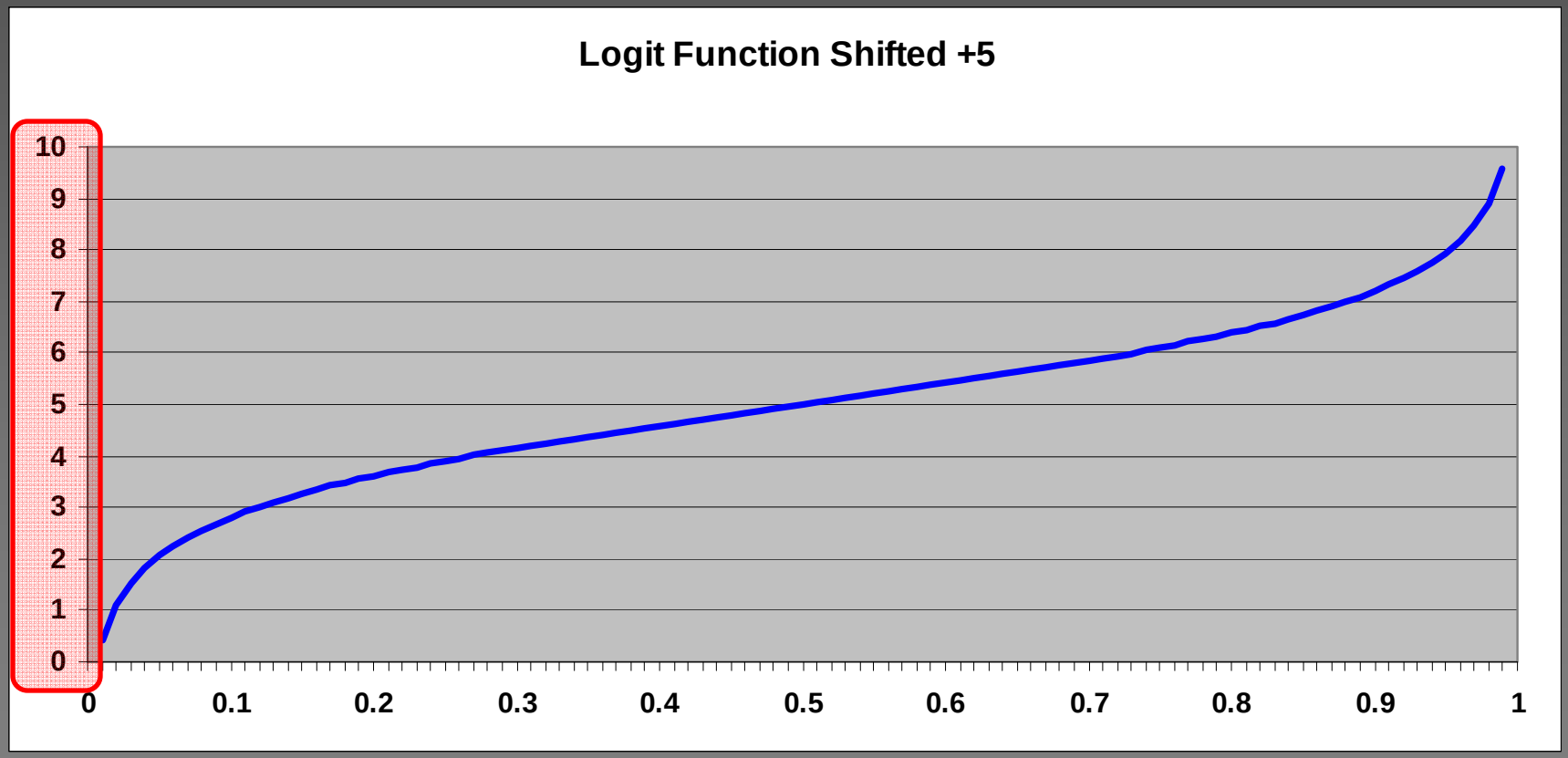
Logit Function

$y = \log_{?}(x/(1- x))$



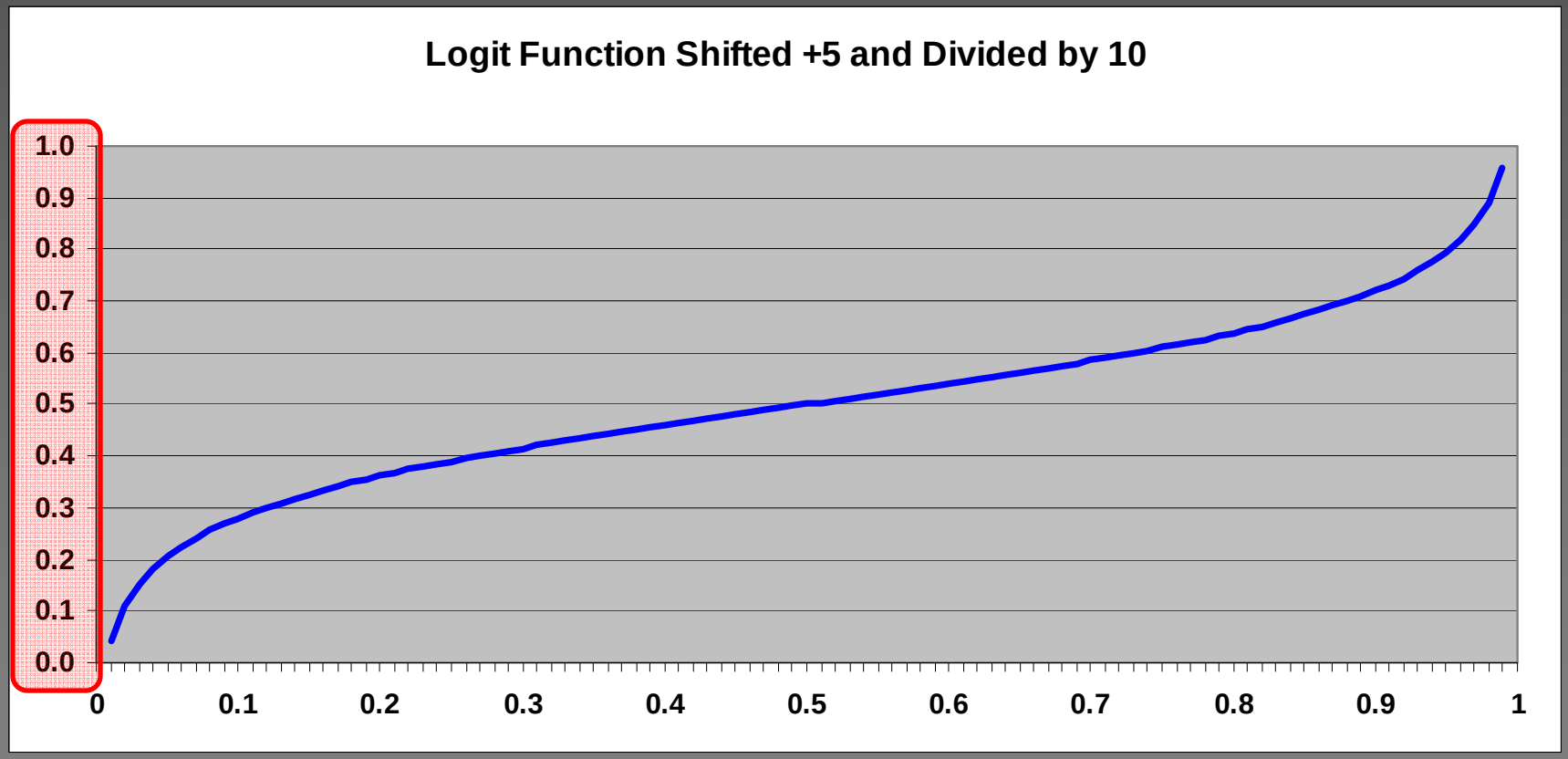
Logit Function

$$y = \log_e(x/(1- x))+5$$



Logit Function

$$y = (\log_e(x/(1- x))+5)/10$$

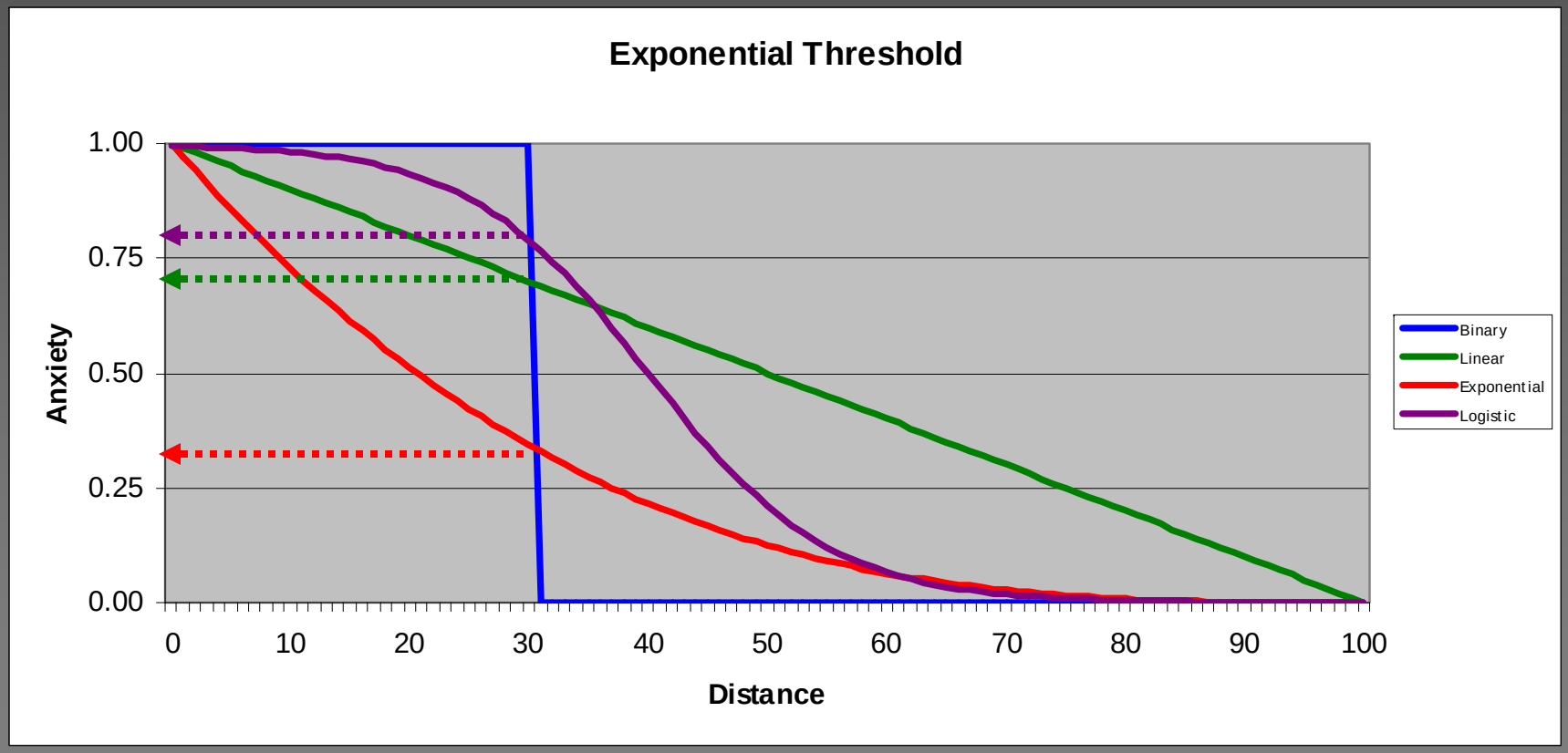


How Do We Model Our Information?

- Increasing or Decreasing?
- Rates of change
  - Steady or Variable?
  - Inflection Point?
- Amount of change
  - Constrained or Infinite?
  - Asymptotic?

But What Good Is It?

When Anxiety > *n* then…



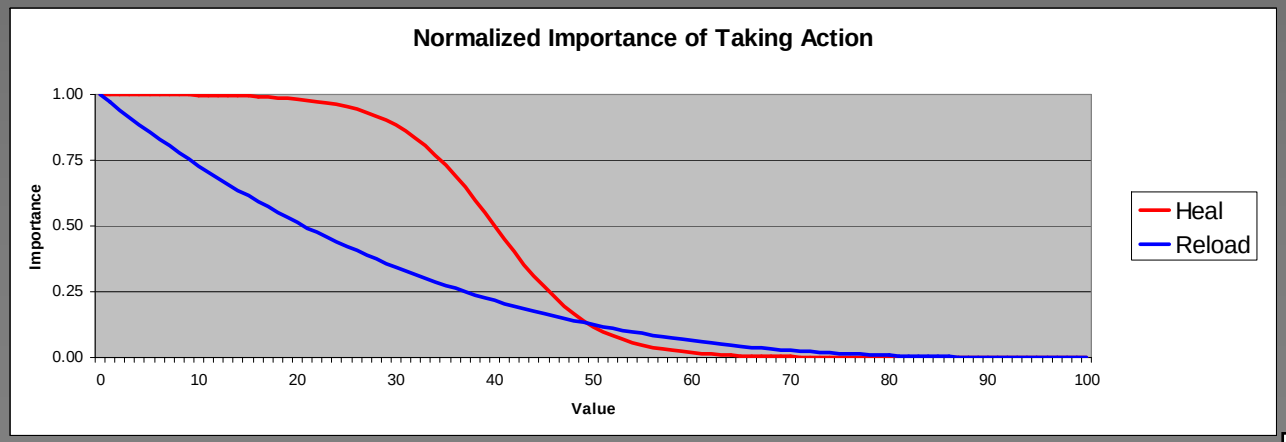
Comparing Apples and Ammo

- By using normalized utility values, we can define relationships and comparisons that otherwise would have been obscure
  - Risk vs. Reward (game theory)
  - Fear vs. Hate
  - Ammo vs. Health

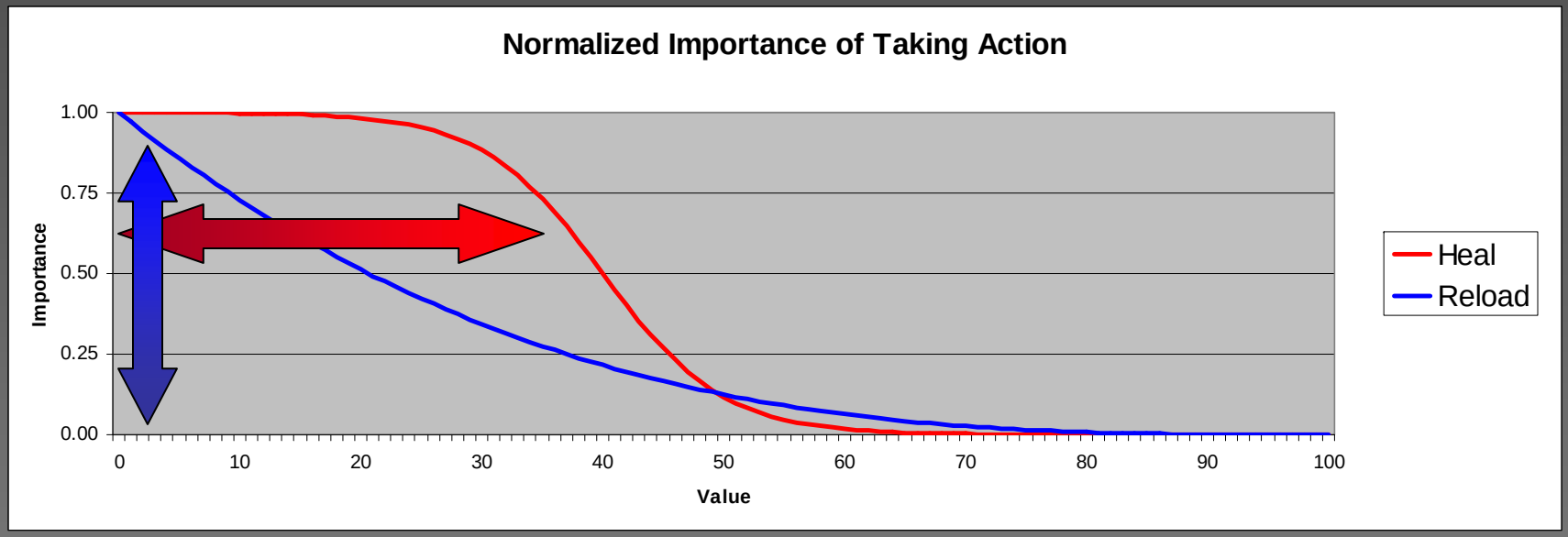
Comparing Apples and Ammo

- 100 Health (Max)
- 75 Health
- 50 Health
- 25 Health (??)
- 5 Health (!!!)

- 100 Ammo (Max)
- 75 Ammo
- 50 Ammo
- 25 Ammo
- 5 Ammo



Comparing Apples and Ammo



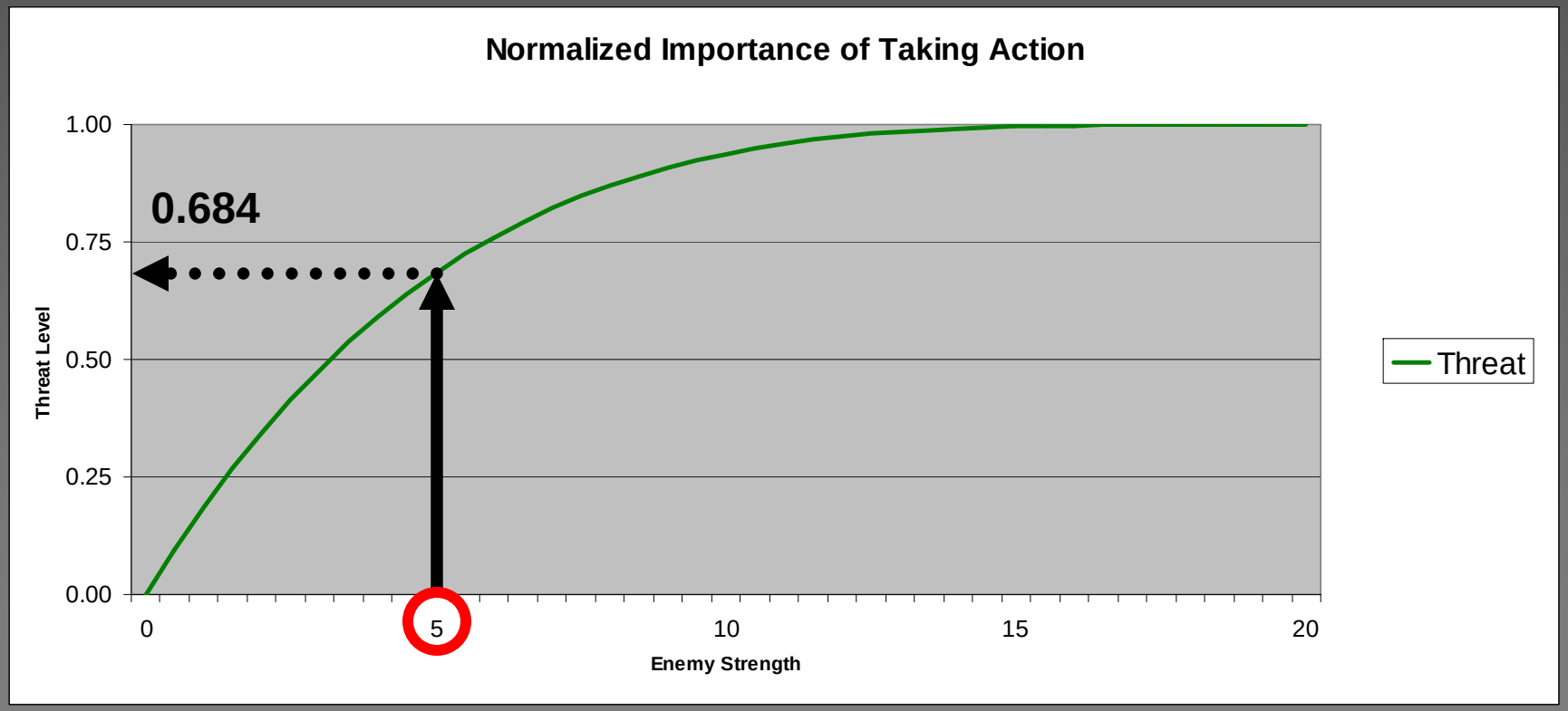
- As health decreases, urgency to heal increases
- Make sure we don't get too low on health!

- As ammo decreases, urgency to reload increases
- Urgency hits maximum when we are out of ammo

Comparing Apples and Ammo

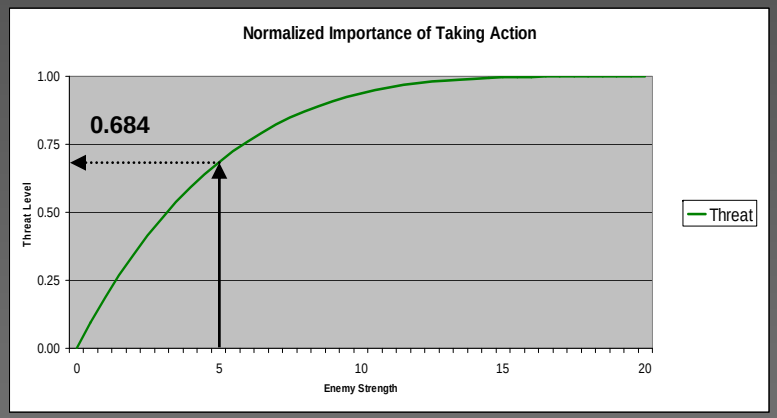
- Collect current states of independent variables
- Normalize using response curves
- (Combine as necessary)
- Compare normalized values and select:
  - Highest scoring selection
  - Weighted random from all choices
  - Weighted random from top *n* choices

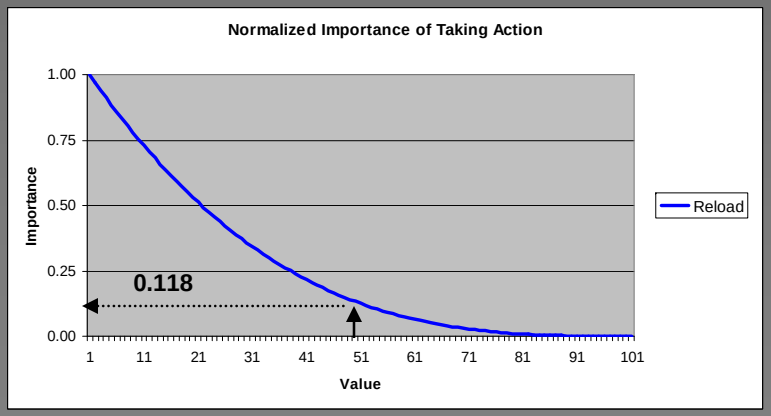
Comparing Apples and Ammo

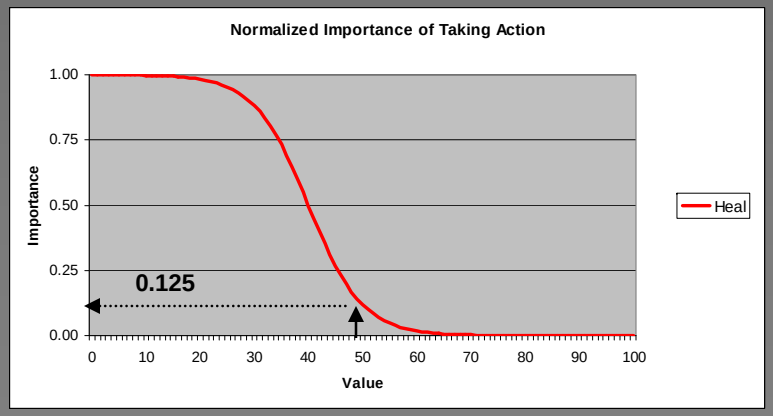


Comparing Apples and Ammo

	Enemies	Ammo	Health
Value	5	50	50
Utility	0.684	0.118	0.125

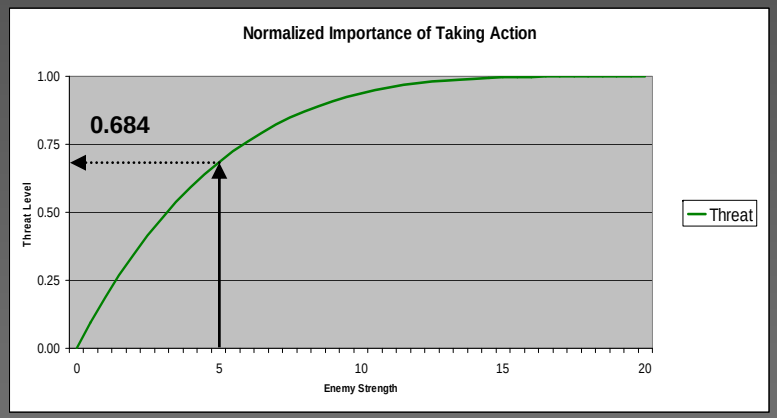


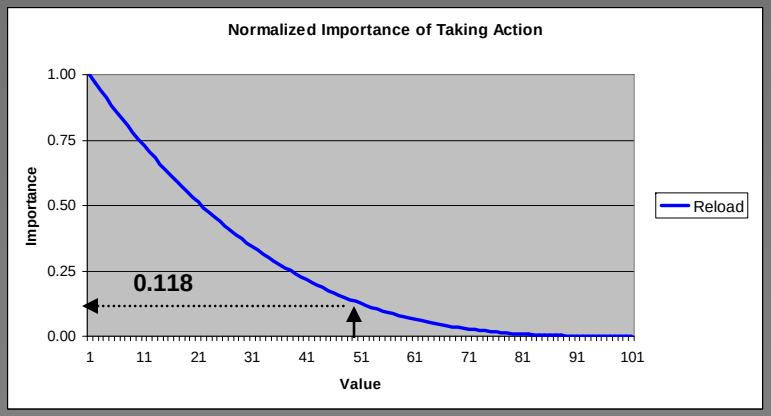


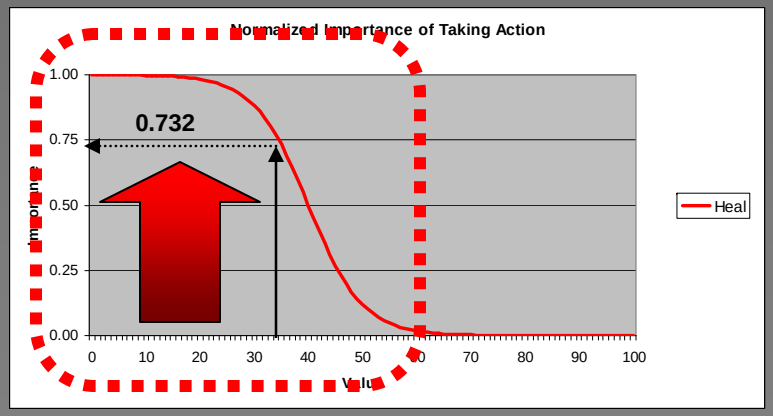


Comparing Apples and Ammo

	Enemies	Ammo	Health
Value	5	50	35
Utility	0.684	0.118	0.732

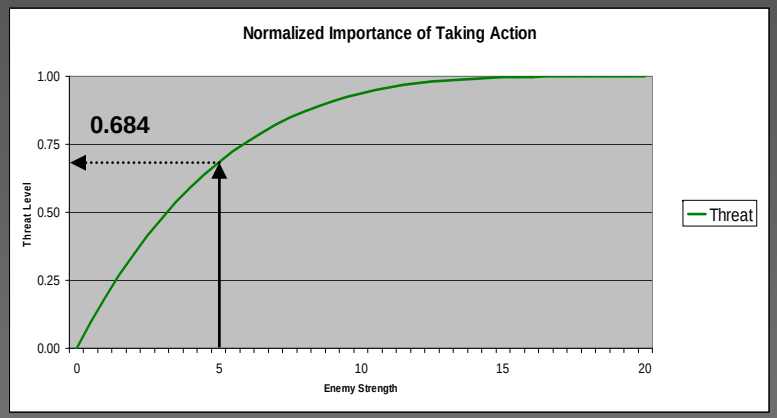


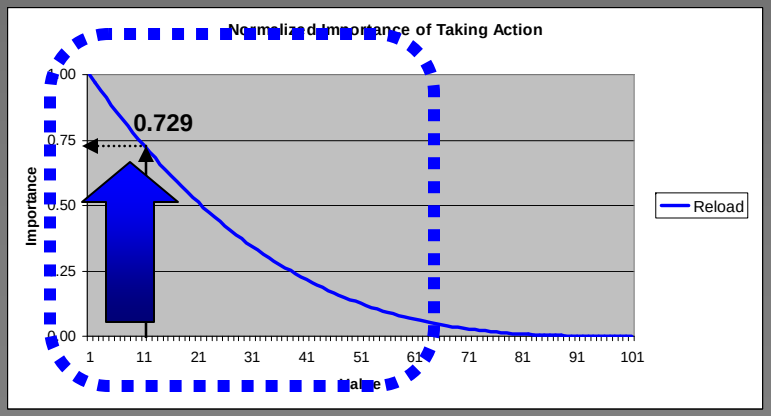


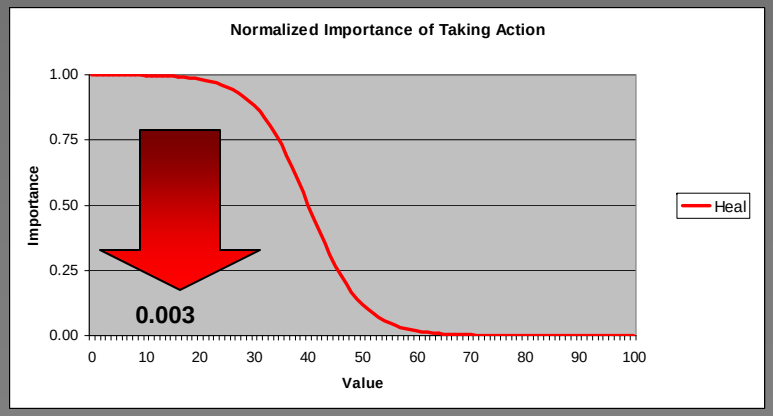


Comparing Apples and Ammo

	Enemies	Ammo	Health
Value	5	10	85
Utility	0.684	0.729	0.003

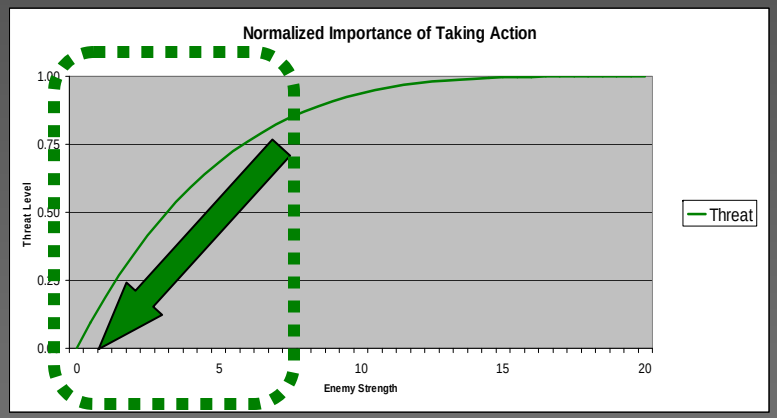


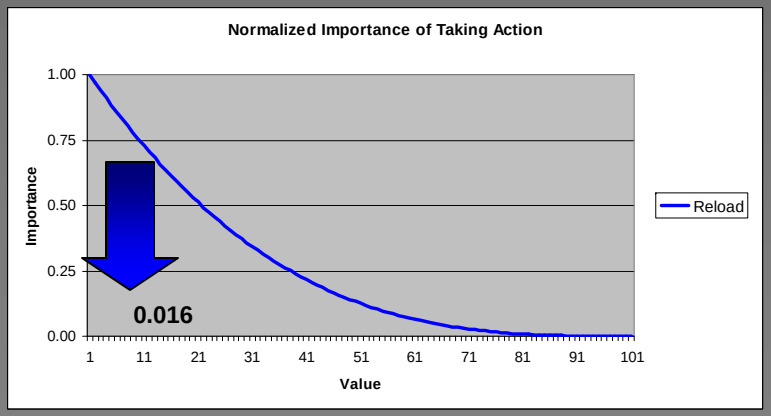


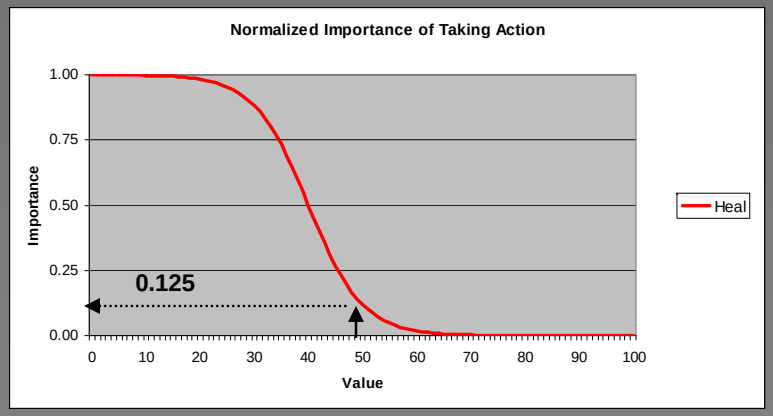


Comparing Apples and Ammo

	Enemies	Ammo	Health
Value	0	75	50
Utility	0.000	0.016	0.125







Comparing Apples and Ammo

- Don’t simply process 1 potential action at a time
  - Should I attack?
  - Should I reload?
  - Should I heal?
  - Should I have a beer?
- Compare all potential actions to each other
  - Of all of the things I could do, which is the most important *at this moment?*

Beyond Apples and Ammo

- Utility measurements can model more than simply tangible data
- They can model abstract concepts:
  - Threat
  - Safety
  - Morale
  - Emotions

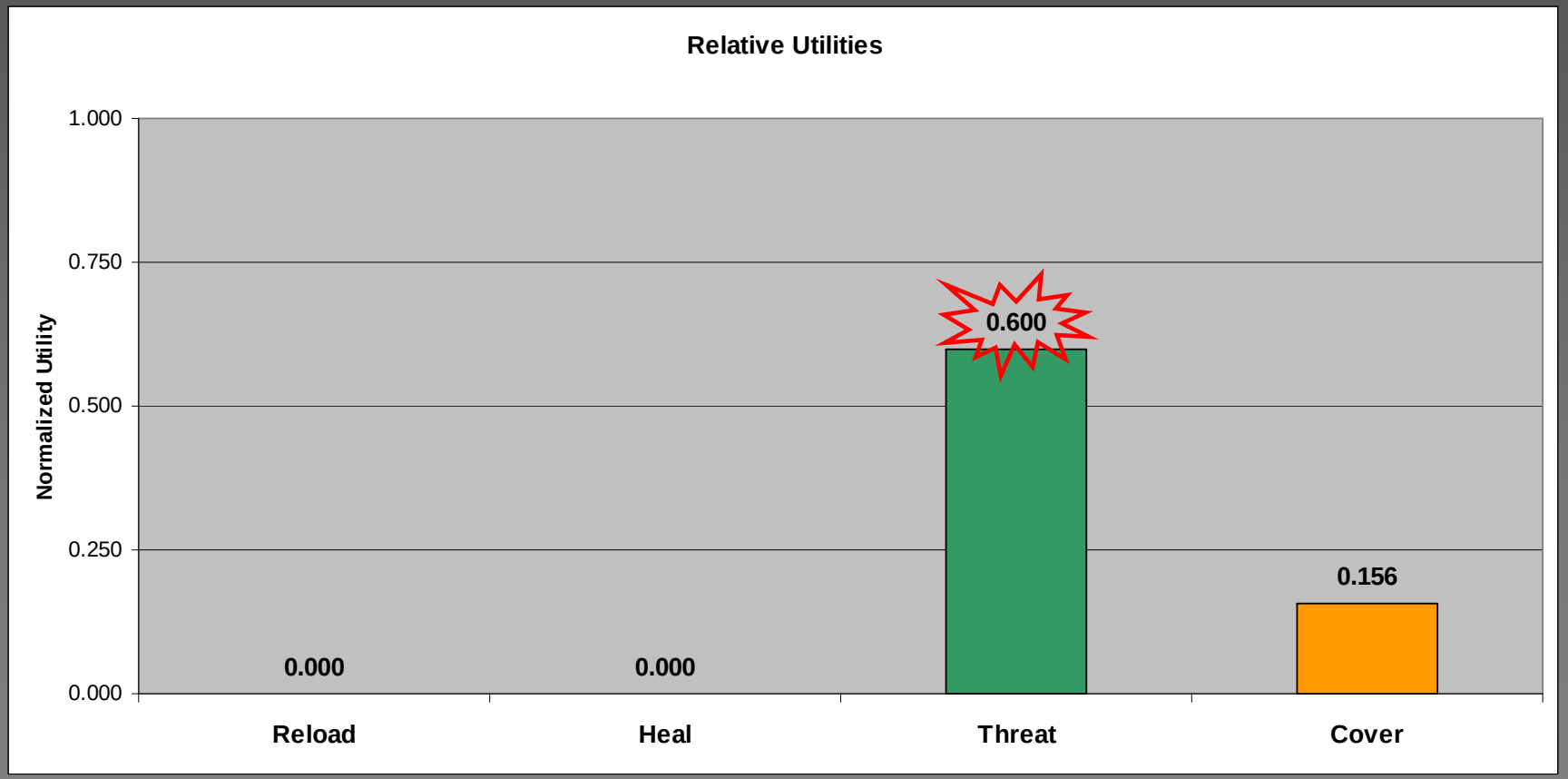
Stacking Apples and Ammo

- Individual utility value can be combined to form new conceptual utilities
- “Need to take cover”
  - Amount of fire being taken (Threat)?
  - Is it almost time to reload?
  - Is it almost time to heal?

Cover = (0.2 + Reload + (Heal x 1.5)) x (Threat x 1.3)

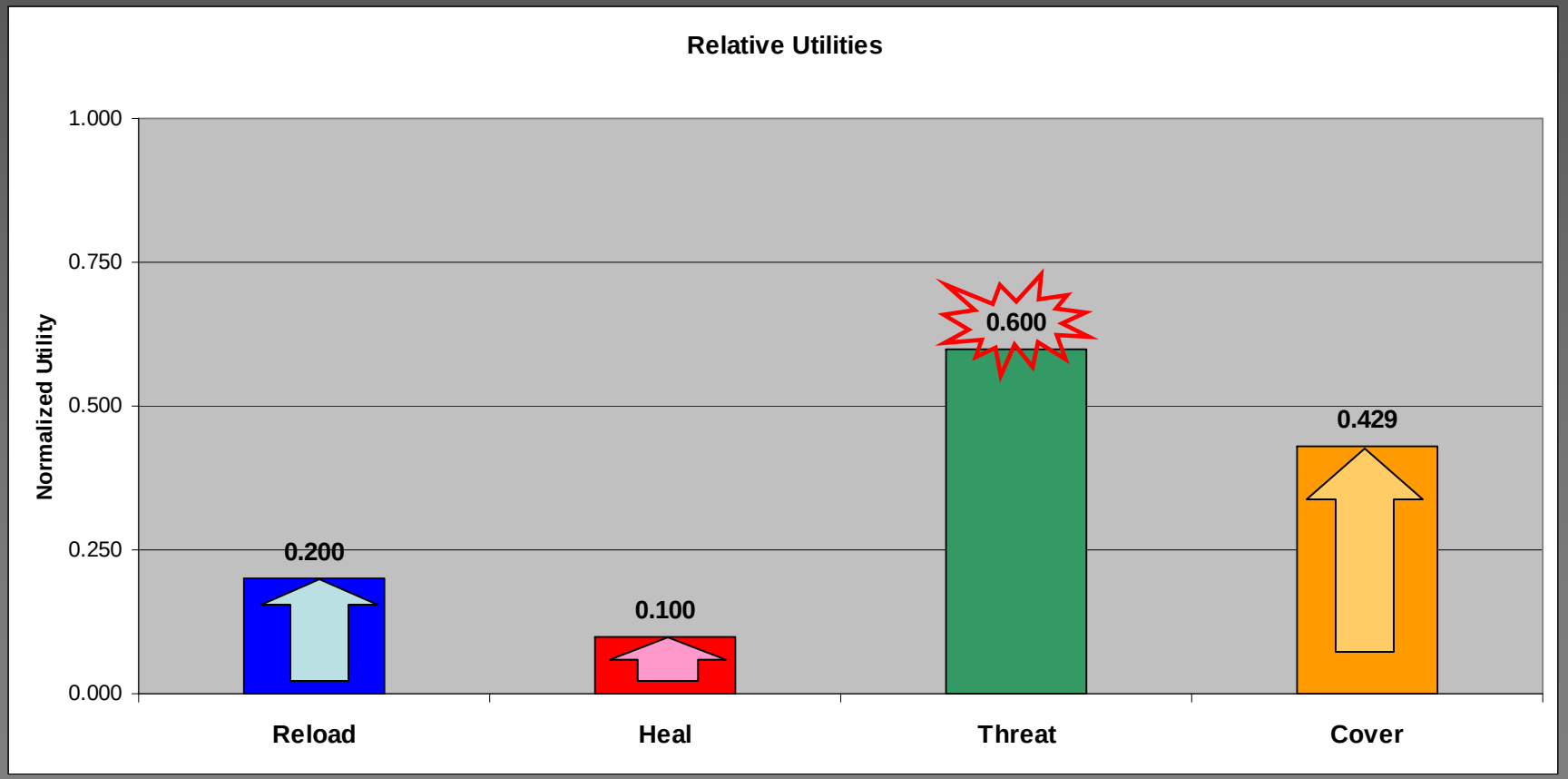
Stacking Apples and Ammo

Cover = (0.2 + Reload + (Heal x 1.5)) x (Threat x 1.3)



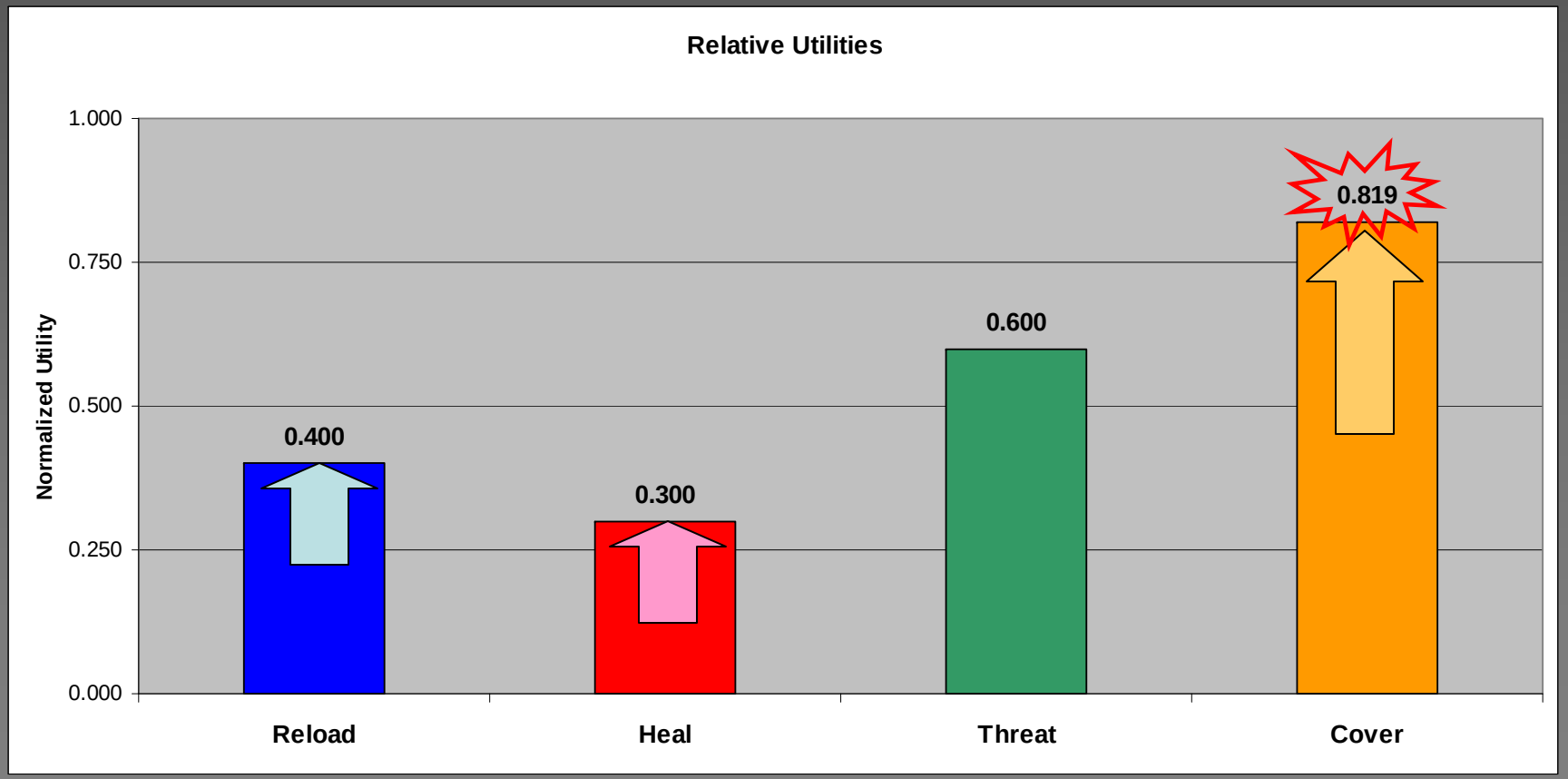
Stacking Apples and Ammo

Cover = (0.2 + Reload + (Heal x 1.5)) x (Threat x 1.3)



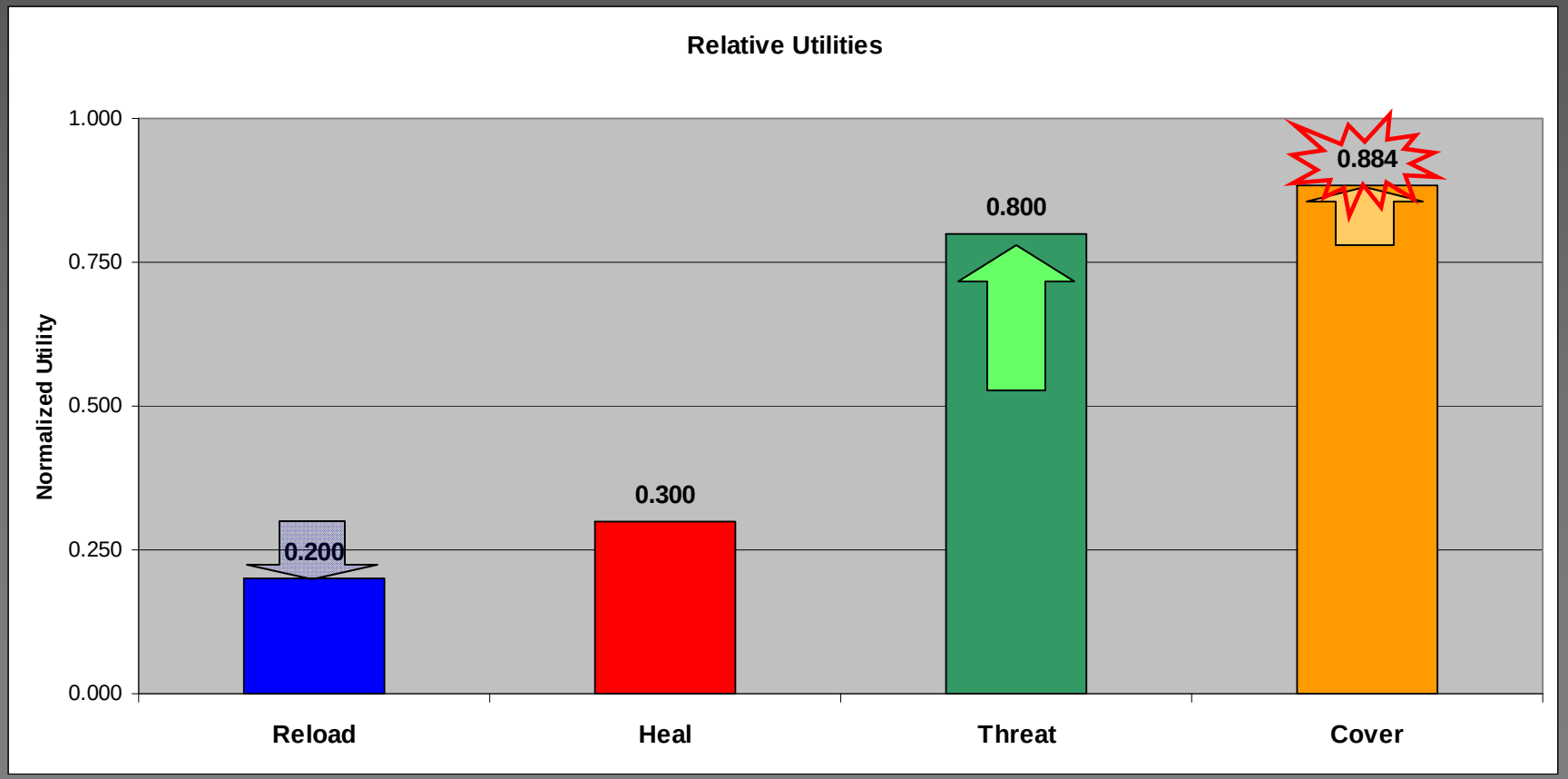
Stacking Apples and Ammo

Cover = (0.2 + Reload + (Heal x 1.5)) x (Threat x 1.3)



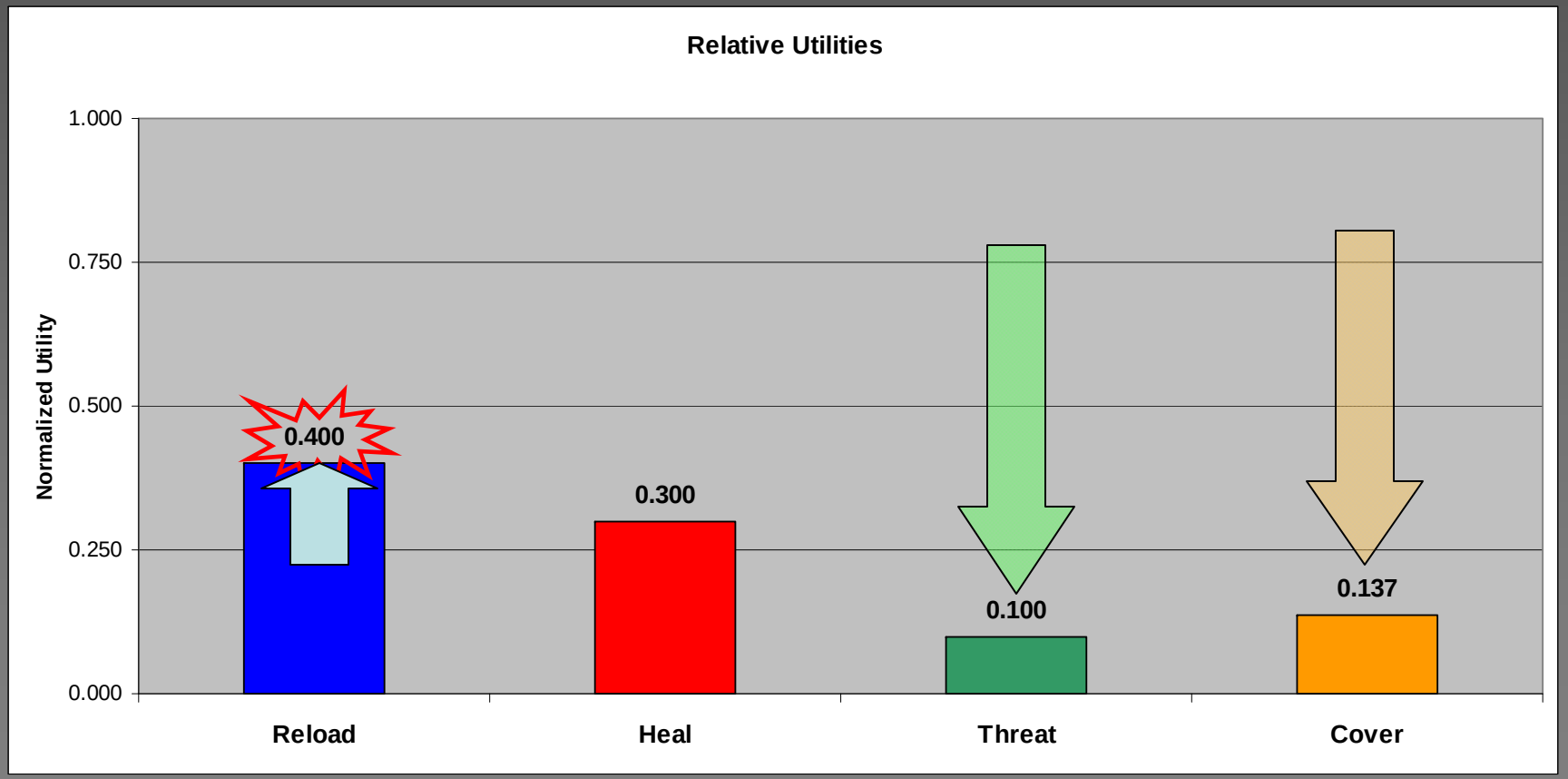
Stacking Apples and Ammo

Cover = (0.2 + Reload + (Heal x 1.5)) x (Threat x 1.3)



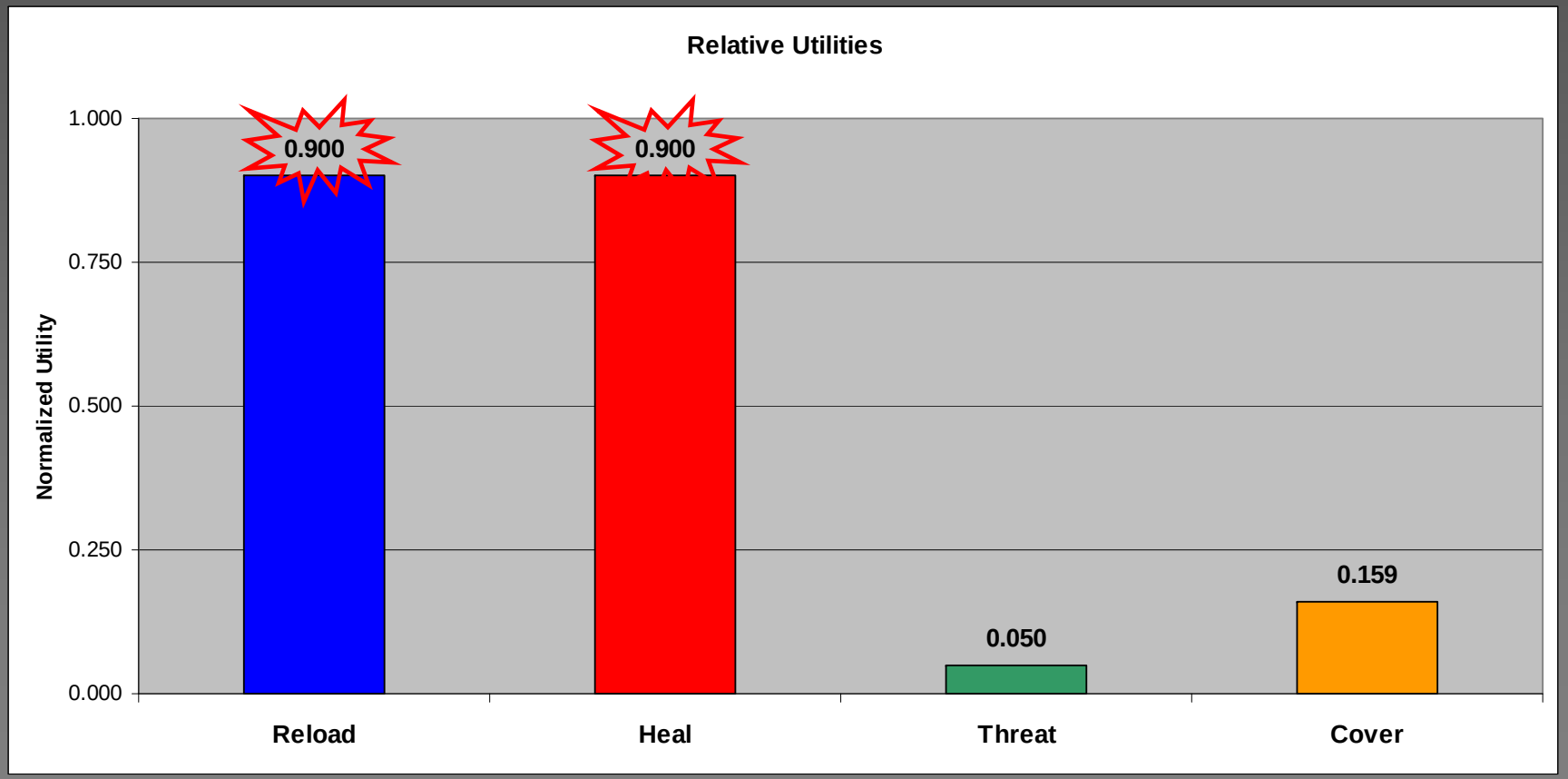
Stacking Apples and Ammo

Cover = (0.2 + Reload + (Heal x 1.5)) x (Threat x 1.3)



Stacking Apples and Ammo

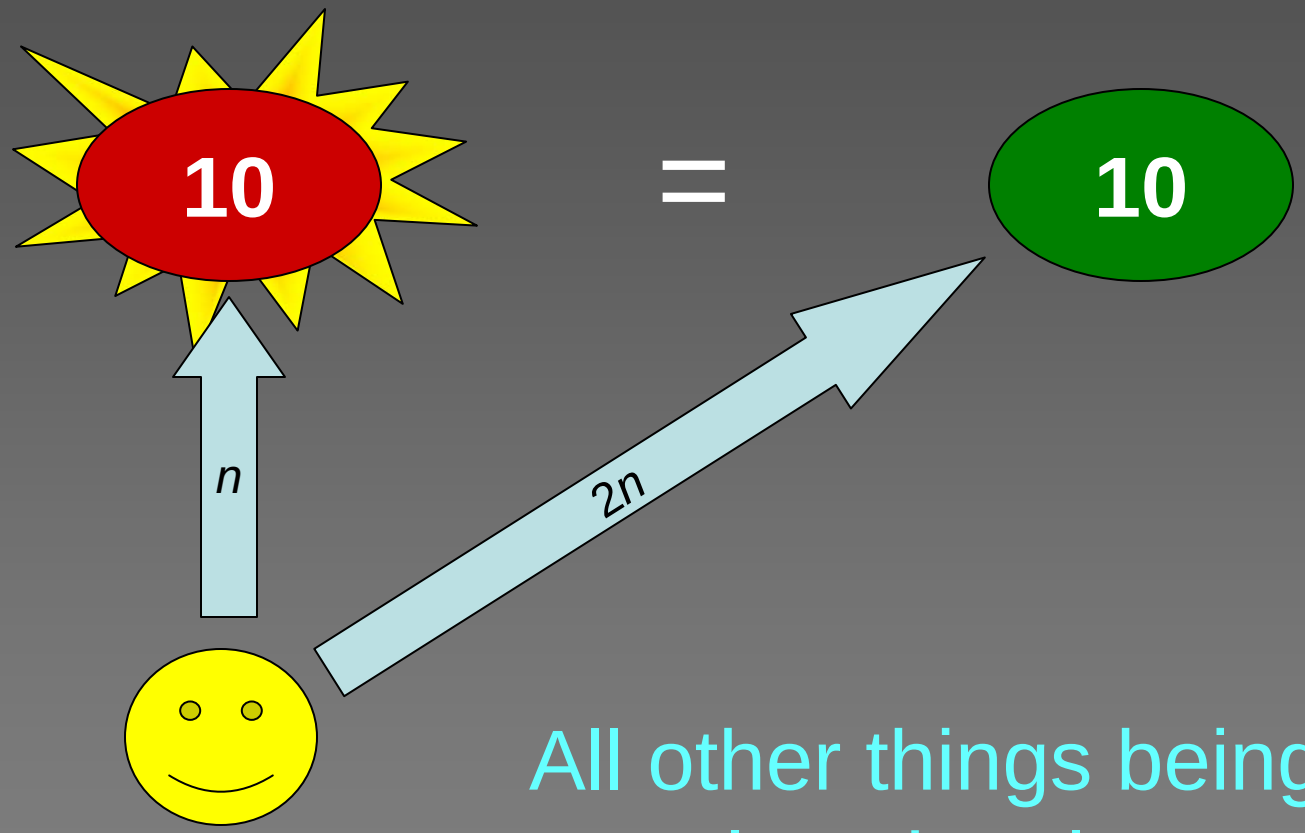
Cover = (0.2 + Reload + (Heal x 1.5)) x (Threat x 1.3)



Utility of Time

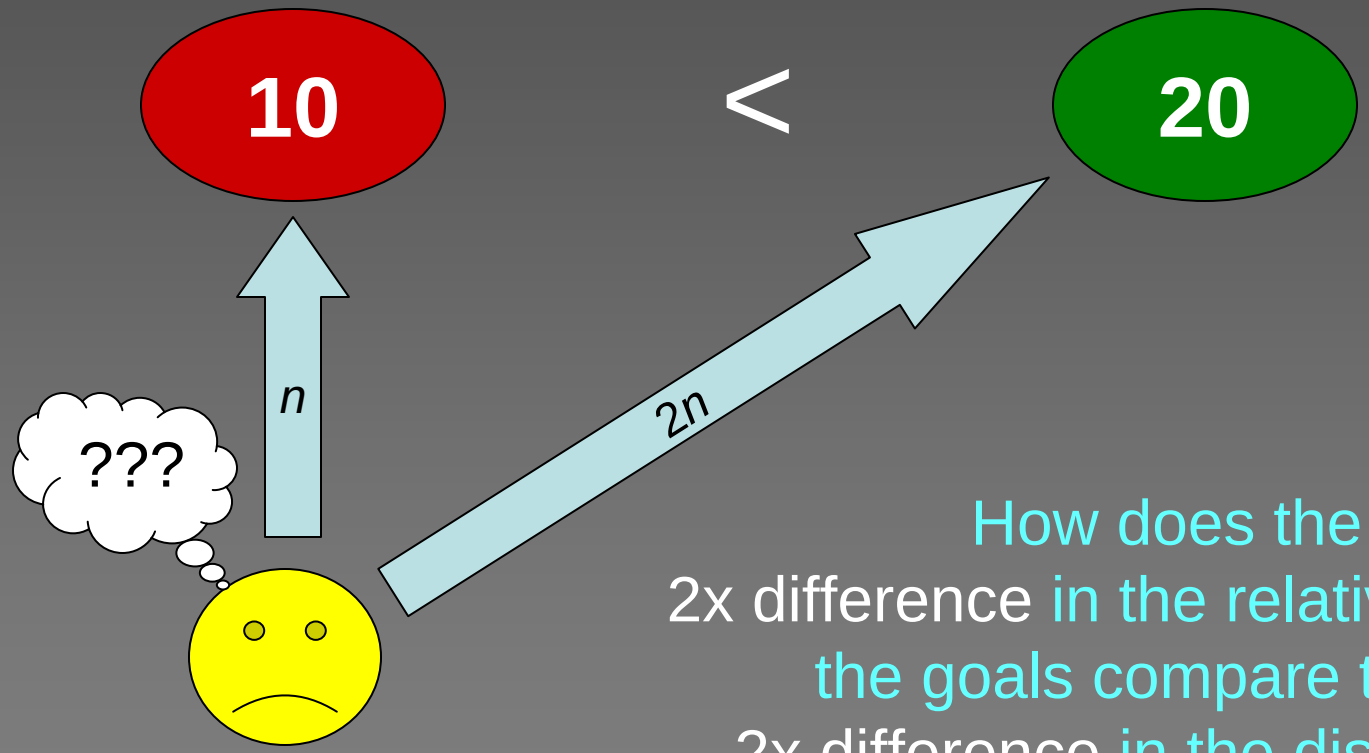
- Time can be converted into a utility value
  - Time to travel over distance
  - Time to complete something
- Utility of time can be used for comparisons
- Utility of time can modify other utilities

Utility of Time

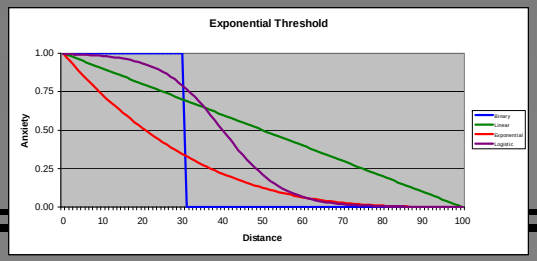


All other things being equal, select the closest goal

Utility of Time

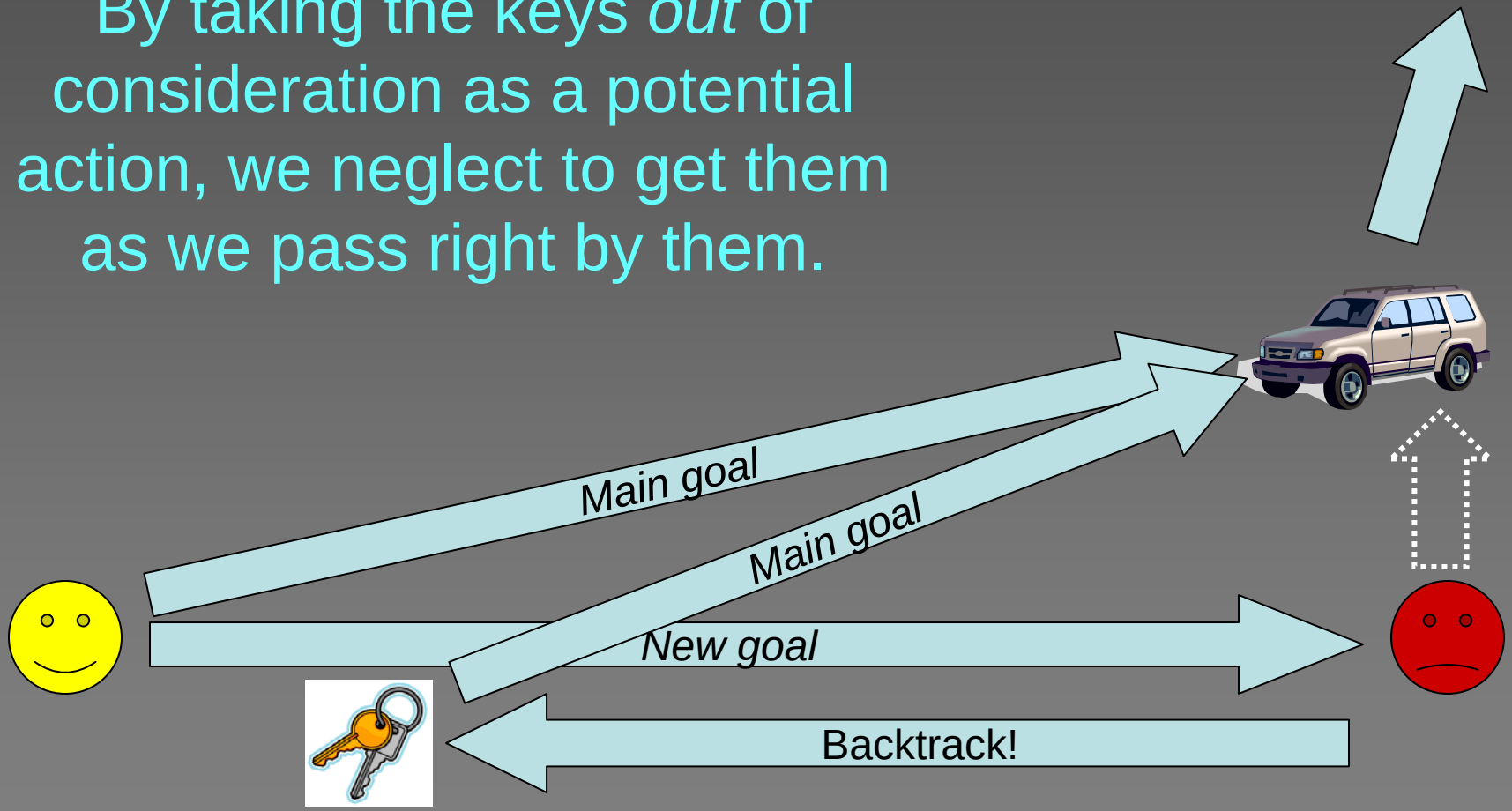


How does the
2x difference in the relative utility of
the goals compare to the
2x difference in the distances?



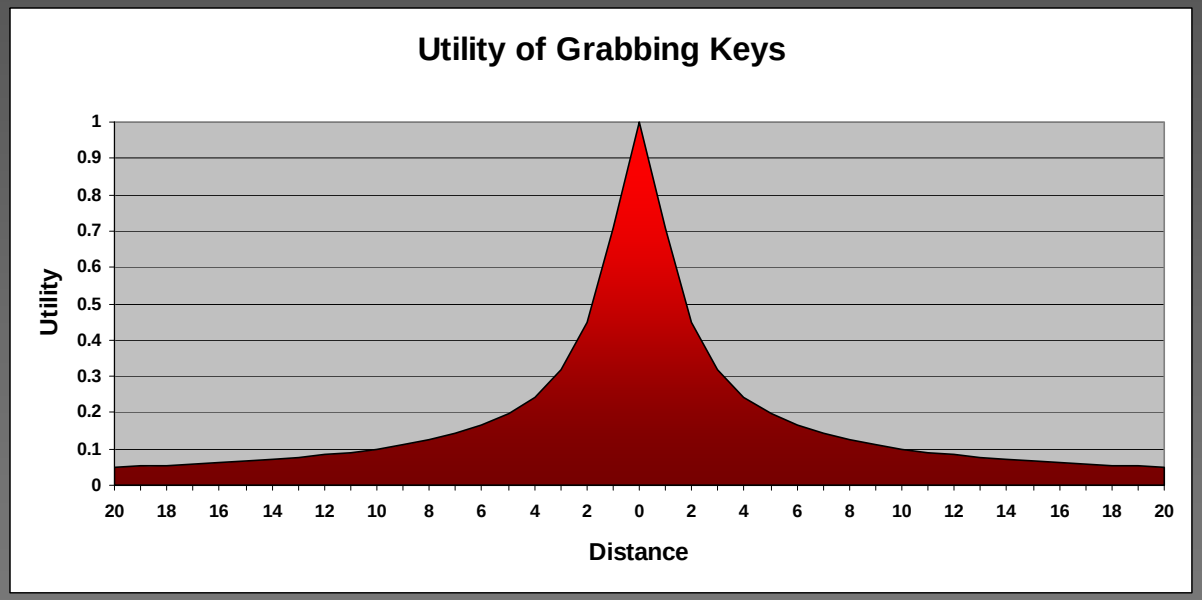
Utility of Time

By taking the keys *out* of consideration as a potential action, we neglect to get them as we pass right by them.



Utility of Time

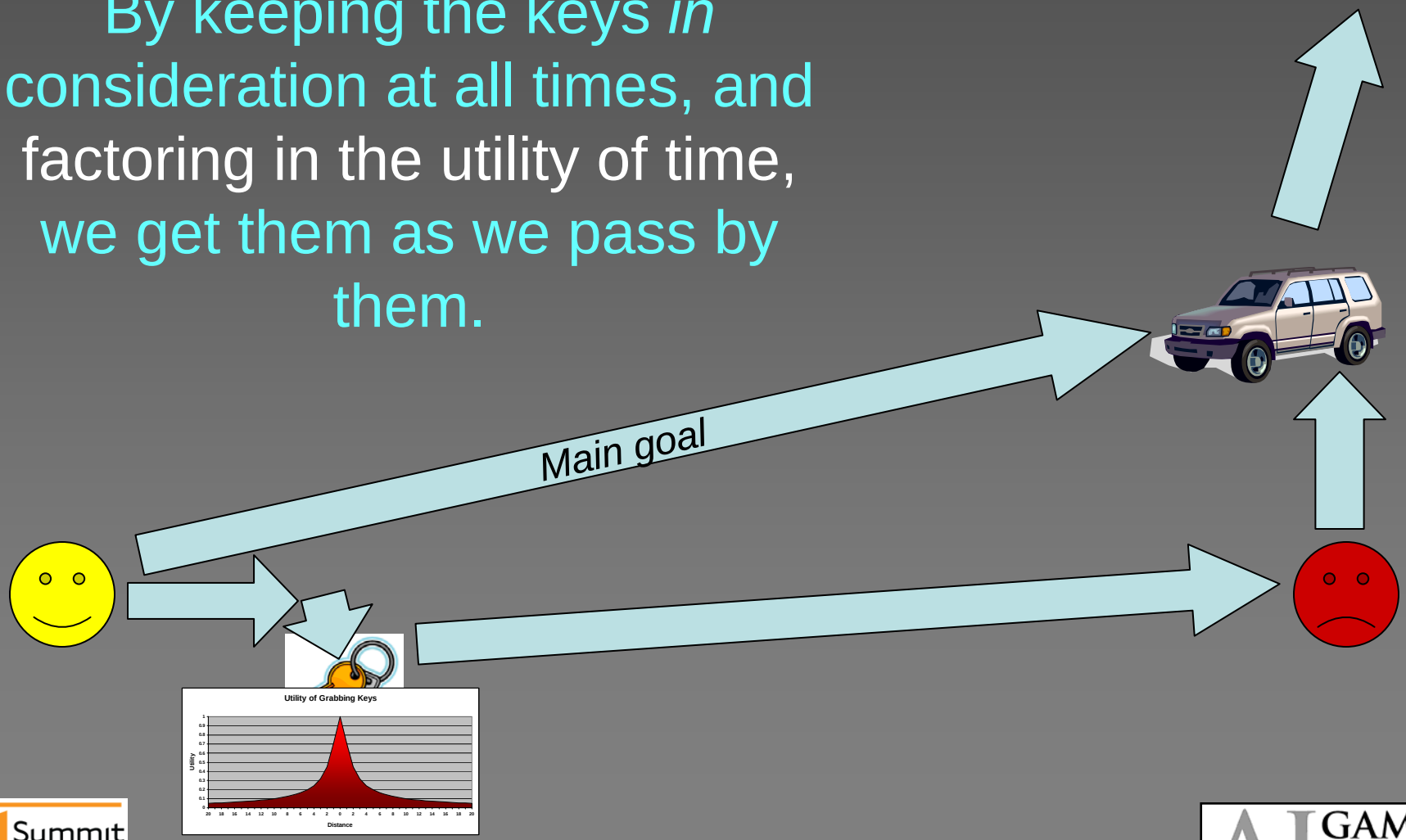
$$\frac{1}{Dist_{keys}}$$



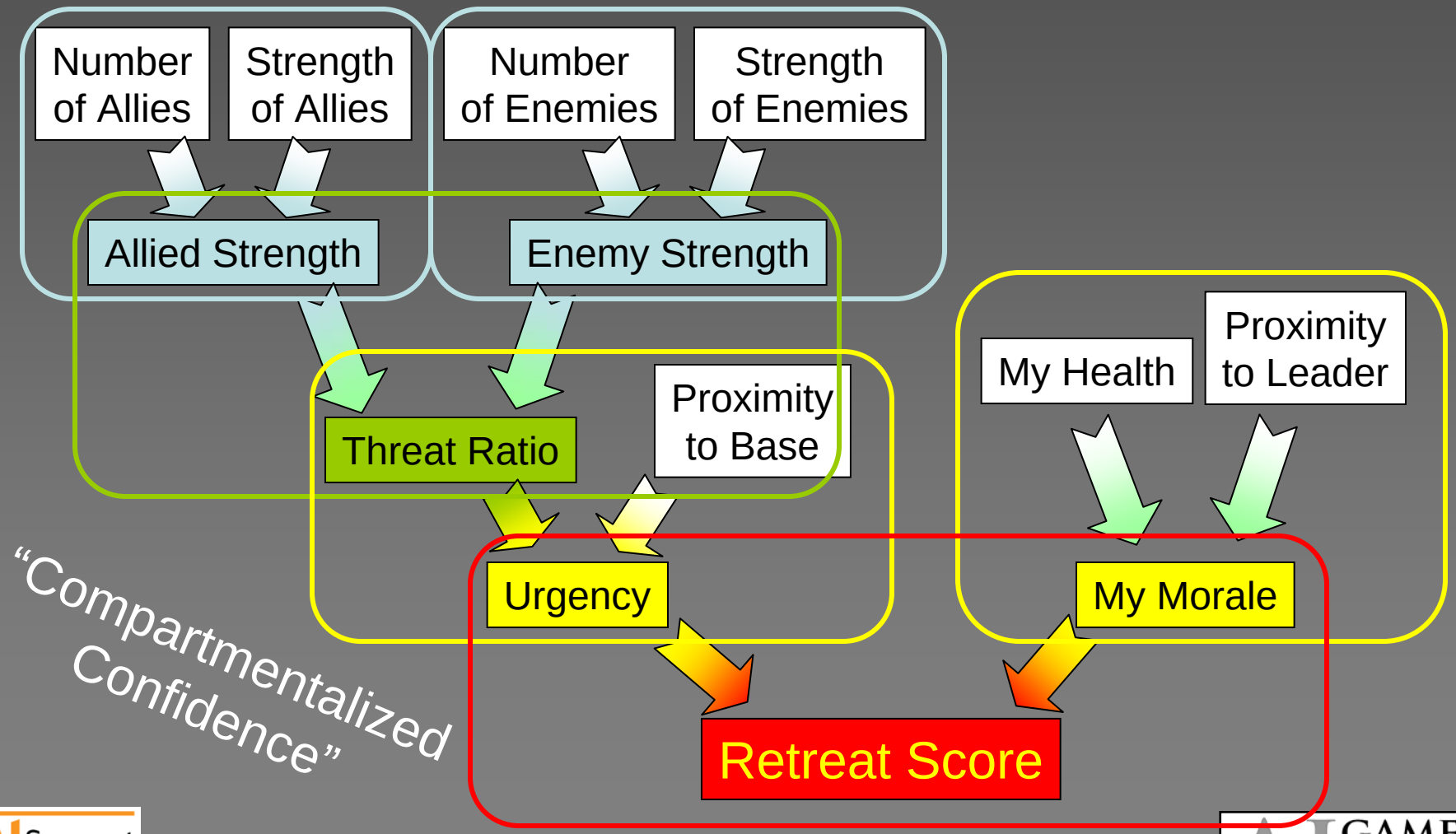
- Normalized distance utility as inverse of distance
- Use as coefficient to modify base utility of getting keys

Utility of Time

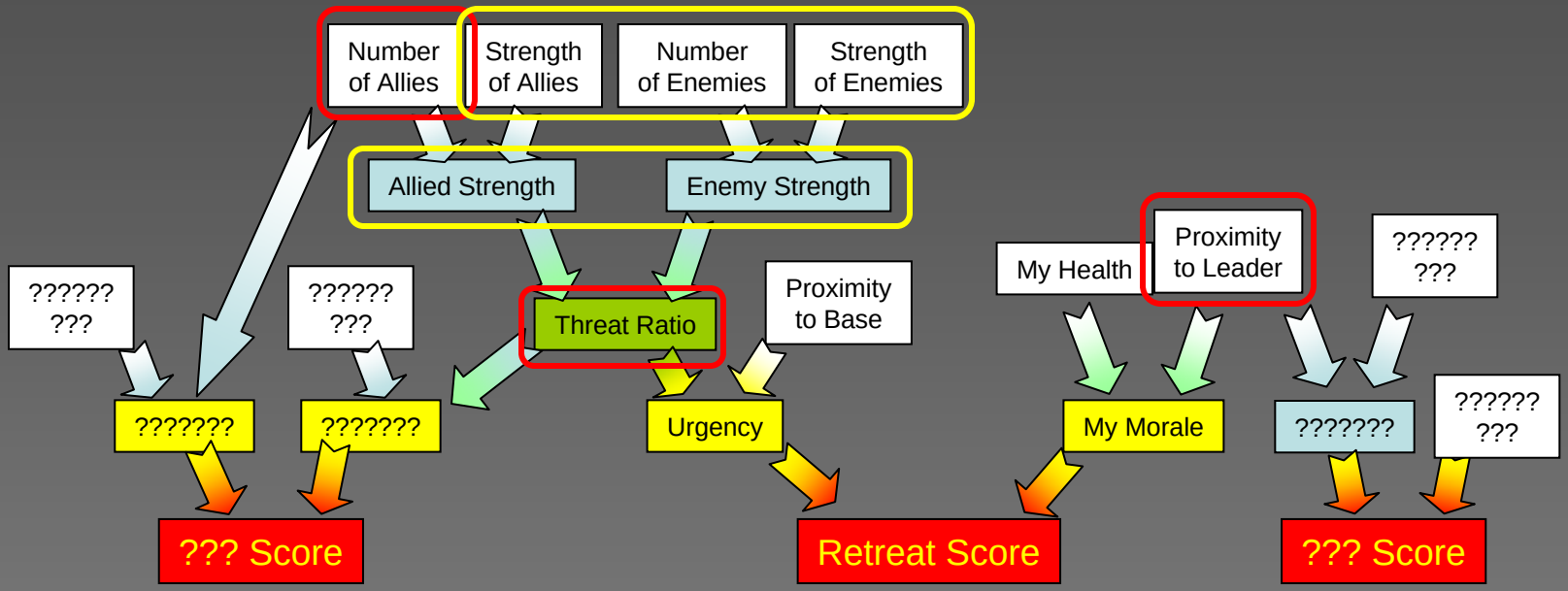
By keeping the keys *in* consideration at all times, and factoring in the utility of time, we get them as we pass by them.



Stacking It All Up



Spreading It All Out



Data processing != Decision processing

Managing Scalability

- Don’t perform all calculations every frame
  - Every *n* frames
  - Use triggered updates
- Split data calculation off into separate processes
  - Used by multiple utility calculations for same agent
  - Used by decision calculations for multiple agents
  - Blackboard architecture to manage and store
- Lends itself well to multi-threading

Everything is Relative

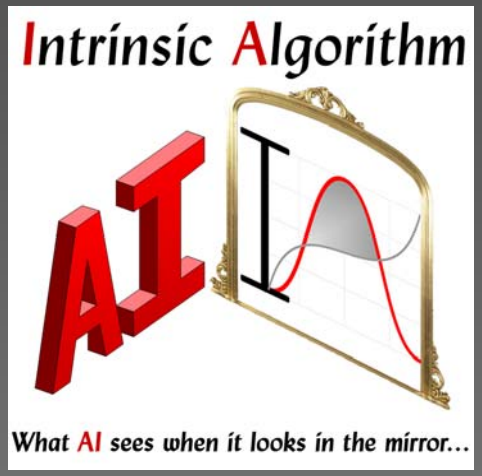
- Many AI decision processes (BTs, FSMs):
  - Examine one choice at a time and ask “should I do this one thing?”
  - Are certain parameters met to justify that choice?
  - If not, move on to the next one in a pre-specified order
- What happens if *no* options meet their criteria?
  - Fall back (idle) behavior may not be appropriate
  - Very susceptible to edge cases

Everything is Relative

- Utility-based architectures:
  - Continuously analyze all options (rather than just one)
  - Rate all options based on their respective factors
  - Select the option that is most appropriate at the time
- Not based on arbitrary, independent thresholds
- Handles situational edge cases better
- Easier to manage potentially conflicting logic

Dave Mark

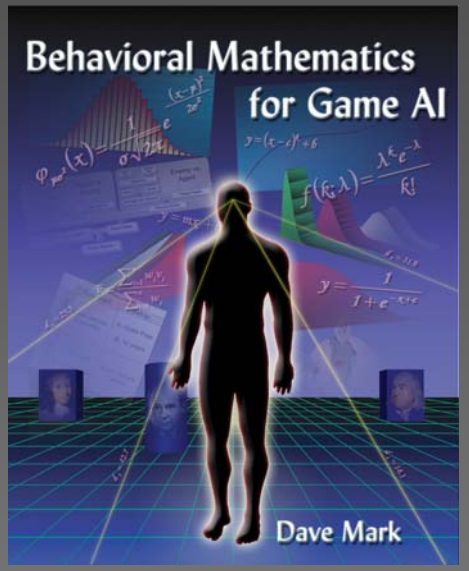
Intrinsic Algorithm LLC



Dave@IntrinsicAlgorithm.com

www.IntrinsicAlgorithm.com

(402) 208–7054



IADaveMark
on:
Yahoo – AIM – Skype – Twitter

AI Summit GDC 2010

AI GAME PROGRAMMERS GUILD

Kevin Dill

- 9 year industry veteran
- Staff Software Engineer, Lockheed Martin
- Lecturer, Boston University
- Technical Editor, Charles River Media

Example: Apartment Shopping

606 Automobile Way, Apt 316

Pros	Cons
- Beautiful loft apartment	- Great view… of a used car lot
- Convenient shopping district	- No off-street parking
- Close to work	

10-B Placid Avenue

Pros	Cons
- Low rent	- 45 minute commute
- Electricity & water included	- No shopping nearby
- Beautiful wooded lot	- Landlady lives upstairs
- Nearby parks, bike trails	

Key Insight

- We have 12 distinct pros and cons, but only 4 ***types*** of considerations:
  - Cost
  - Distance to _____
    - Could be the distance to work, shopping, parking, etc.
  - Aesthetic (i.e. how nice looking is the place)
    - Could be interior or exterior
    - Obviously, there is a lot of variability in what people consider to be “nice”
  - Noise restrictions
- Many of these are reusable in other contexts!!

Example: Apartment Shopping

606 Automobile Way, Apt 316

Pros	Cons
- Beautiful loft apartment	- Great view… of a used car lot
- Convenient shopping district	- No off-street parking
- Close to work	

10-B Placid Avenue

Pros	Cons
- Low rent	- 45 minute commute
- Electricity & water included	- No shopping nearby
- Beautiful wooded lot	- Landlady lives upstairs
- Nearby parks, bike trails	

Executive Joe

- Joe is a high-powered executive at a big bank
- Joe makes lots of money
  - Cost doesn't matter
- Joe works late most every night
  - Exterior aesthetics don't matter when the sun is down
  - Distance to recreation doesn't matter – who has time?
  - Distance to work, shopping, and parking matter a lot
- Joe likes to throw big parties
  - Interior aesthetics are very important
  - Joe is not fond of noise restrictions

Example: Apartment Shopping

606 Automobile Way, Apt 316

Pros	Cons
- Beautiful loft apartment	- Great view… of a used car lot
- Convenient shopping district	- No off-street parking
- Close to work	

10-B Placid Avenue

Pros	Cons
- Low rent	- 45 minute commute
- Electricity & water included	- No shopping nearby
- Beautiful wooded lot	- Landlady lives upstairs
- Nearby parks, bike trails	

Stan the Family Man

- Stan goes to work to put in his time and get home to his family
- Stan's wife wants a nice place with lots of recreation for the kids… Stan wants something he can afford
- The apartment needs to be kid-friendly
- Stan likes to drive – it gives him some quiet time
- Stan's family is in bed by 10:00

Example: Apartment Shopping

606 Automobile Way, Apt 316

Pros	Cons
- Convenient shopping district - Close to work	- Beautiful loft apartment - Great view… of a used car lot - No off-street parking

10-B Placid Avenue

Pros	Cons
- Low rent - Electricity & water included - Beautiful wooded lot - Nearby parks, bike trails	- 45 minute commute - No shopping nearby - Landlady lives upstairs

High Principles

- Modular
  - A decision is made up of atomic pieces of logic, called “considerations”
  - We can easily add and remove considerations
- Extensible
  - We can easily create new types of considerations
- Reusable
  - From decision to decision
  - From project to project

Terminology & Architecture

- ***Reasoner*** has a list of possible ***choices***
  - E.G. play with a ball, build a swordsman unit, select a particular weapon, play a particular animation, etc.
- Each choice has a list of ***considerations***
  - Considerations evaluate one aspect of the situation
- Considerations generate ***appraisals***
- Appraisals inform our final selection

Consideration

- Encapsulates one aspect of a larger decision
  - Distance
  - Cost
  - Etc.
  - Selection History
  - Benefit
- Parameterized for easy customization
  - For *this decision* and for *this character*

Consideration Interface

```
class IConsideration
{
public:
    // Load the data that controls our decisions
    void        Load(const DataNode& node) = 0;

    // Evaluate this consideration
    Appraisal   Evaluate(const Context& c) = 0;
}
```

Relies on:

- DataNode
- Context
- Appraisal

DataNode

- XML (or equivalent) that contains the parameterization
- May be tool-generated (not hand-generated)
- May be part of a larger AI specification

Context

- Contains all of the information the AI needs to make a decision
- Provides an abstraction layer between the AI and the game
  - If well implemented, can facilitate porting your considerations from game to game

Appraisal

- Generated by the Evaluate() function
- Drives our final decision
- Common techniques include:
  - Boolean logic (e.g. all appraisals must return TRUE)
  - Highest score
  - Weight-based random
  - Optimize resource allocation to maximize utility
- Experience has taught me to start as simple as possible, extend only when necessary

Simple Utility-Based Appraisals

- Each appraisal contains two components:
  - **Base Score**: a floating point indicating how good we think this choice is (based on our one consideration)
  - **Veto**: a Boolean allowing each consideration to prevent us from selecting the associated choice
- Calculating total utility for a choice:
  - If any consideration sets Veto to false, utility is 0
  - Otherwise, add all of the base scores together

Example: Weapon Selection

- "Tuning consideration" provides a base score
  - A tuning consideration always returns the values specified in data, regardless of the situation
- "Range consideration" can add utility or veto as needed
  - Pistols are better at short ranges, sniper rifles at long
- "Inertia consideration" adds utility to current choice
  - So we don't change without a good reason
- "Random noise consideration" has a random base score
  - So we don't always pick the same thing
- "Ammo consideration" checks if we have ammo
- "Indoors consideration" prevents grenade use indoors
- Select the weapon with the best total score

Appraisal With A Multiplier*

- Replace Veto parameter with a “Final Multiplier”
  - Add all base scores together, then multiply by each of the final multipliers
  - A multiplier of 0 is still a veto
- Allows you to scale utility more smoothly/cleanly
  - For example, scale sniper rifle utility at short range
- Other things you could add:
  - Exponents
  - Polynomials
  - Etc.

* (This is Kevin’s preferred approach.)

Multi-Utility Appraisals

- Add a Priority attribute to the appraisal
- When combining appraisals, take the max Priority
  - In other words, if one consideration sets the priority to be high, keep that priority
- Only consider choices with max priority
  - Allows you to say “If X is true, only consider this small set of options.”
  - For example, force the use of a melee weapon at short range, a ranged weapon at long range

Summary

- Modular
- Extensible
- Reusable
- Applicable to a wide variety of game genres and reasoner architectures
  - *Kohan 2: Kings of War* and *Axis & Allies*
  - Prototype dog AI
  - *Iron Man* boss AI
  - *Red Dead Redemption*
    - Weapon Selection
    - Dialog Selection
  - Event selection in *All Heroes Die*

References

Dill, Kevin, “Prioritizing Actions in a Goal-Based RTS AI,” *AI Game Programming Wisdom 3*

Dill, Kevin, “A Pattern-Based Approach to Modular AI for Games,” *AI Game Programming Gems 8*

Mark, Dave, *Behavioral Mathematics for Game AI*

Alexander, Bob, “The Beauty of Response Curves,” *AI Game Programming Wisdom*

Harmon, Vernon, “An Economic Approach to Goal-Directed Reasoning in an RTS,” *AI Game Programming Wisdom*

Orkin, Jeff, “Constraining Autonomous Character Behavior with Human Concepts,” *AI Game Programming Wisdom 2*

Garces, Sergio, “Extending Simple Weighted Sums,” *AI Game Programming Wisdom 3*

Questions?

Dave Mark

Intrinsic Algorithm LLC

Dave@IntrinsicAlgorithm.com

www.IntrinsicAlgorithm.com

(402) 208–7054

IADaveMark
on:
Yahoo – AIM
Skype – Twitter

Kevin Dill

Lockheed Martin

kdill4@gmail.com