FoodThought

Seamless money allocation manager

Components

Components

- A build system: Tup

Components

- A build system: Tup
- A logging module: glog

Components

- A build system: Tup
- A logging module: glog
- An event notification library: libevent

Components

- A build system: Tup
- A logging module: glog
- An event notification library: libevent
- A testing framework: gtest

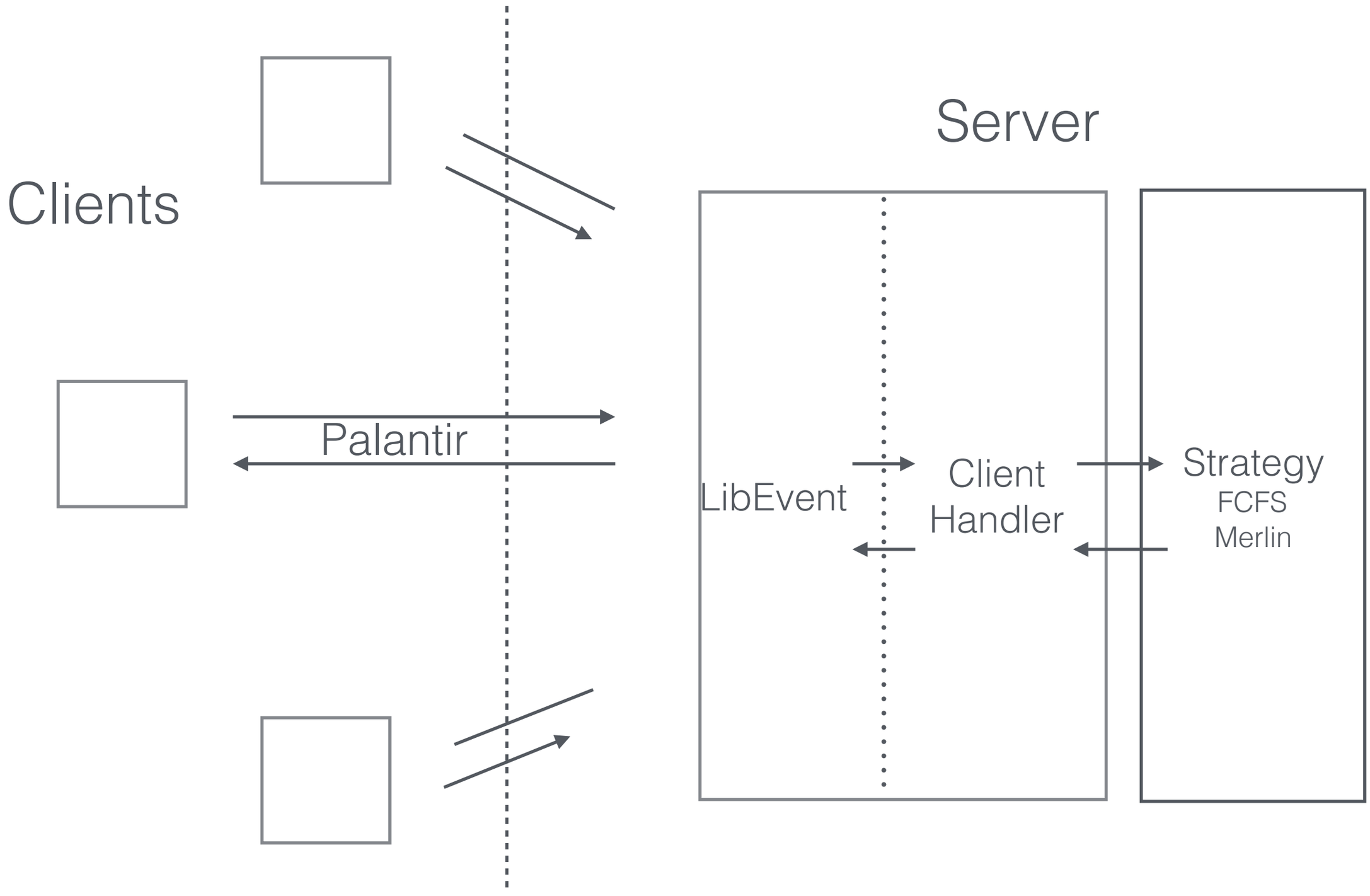
Components

- A build system: Tup
- A logging module: glog
- An event notification library: libevent
- A testing framework: gtest
- C++11/14

Components

- A build system: Tup
- A logging module: glog
- An event notification library: libevent
- A testing framework: gtest
- C++11/14
- Boost

Architecture



StrategyBase

```
 9   class StrategyBase
10   {
11      public:
12         ~StrategyBase() {  }
13
14         // returns false only if we tried to add a client and couldn't.
15         // return true if the name if already present
16         virtual bool addNewClient(const std::string& name) = 0;
17
18         // shutdown/cleanup function to be called at the end. (against RAII)
19         // This can't be put in the destructor because it is part of the static
20         // class which gets destroyed at the end of the overall process. At that stage
21         // things like logger and others are not present
22         virtual bool shutdown() = 0;
23
24         // onTimeout callbacks from server
25         virtual void onTimeout() = 0;
26
27         // queries the amount of money available in the global pool
28         std::string virtual query() = 0;
29
30         using RequestReturnType = std::tuple<bool, std::string>;
31         using DonateReturnType  = std::tuple<bool, std::string>;
32
33         // For request - tuple's first argument tells whether request was successful or not.
34         //      - if successful, a string of who helped and by how much
35         //      - if not, reason (usually no money, or unknown name)
36         // For donate - tuple's first arguments tells whether donation was succssful or not.
37         //      - if successful, string argument to be ignored
38         //      - if not, reason (usually unknown name)
39         virtual RequestReturnType request(const std::string& name, double amount) = 0;
40         virtual DonateReturnType donate(const std::string& name, double amount) = 0;
41   };
```

FCFS

FCFS

- First Come First Serve

FCFS

- First Come First Serve
- Global pool

FCFS

- First Come First Serve
- Global pool
- Unfair

Merlin Strategy

Merlin Strategy

- Tries to be fair while being efficient

Merlin Strategy

- Tries to be fair while being efficient
- Takes into account 3 factors
  - F1 = Time elapsed (γ, T)
  - F2 = Status based on past request/donate pattern (α, d1)
  - F3 = Amount of money asked (β, d2)

Merlin Strategy

- Tries to be fair while being efficient
- Takes into account 3 factors
  - F1 = Time elapsed (γ, T)
  - F2 = Status based on past request/donate pattern (α, d1)
  - F3 = Amount of money asked (β, d2)
- Final decision
  - std::min(1, F1 + F2 * F3)

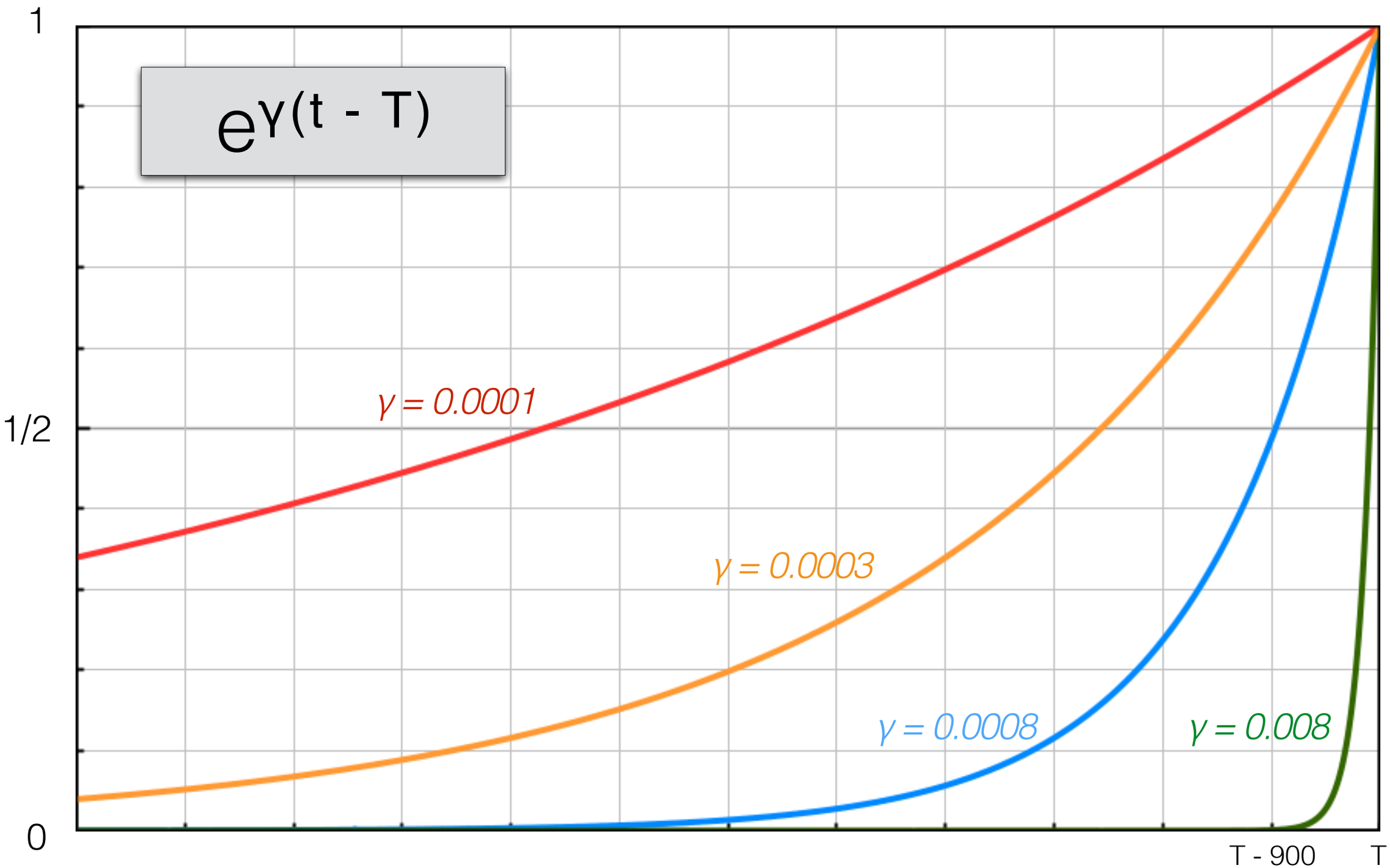
Merlin Strategy

- Tries to be fair while being efficient
- Takes into account 3 factors
  - F1 = Time elapsed (γ, T)
  - F2 = Status based on past request/donate pattern (α, d1)
  - F3 = Amount of money asked (β, d2)
- Final decision
  - std::min(1, F1 + F2 * F3)
- Random numbers by Mersenne Twister

Time Factor

$$e^{\gamma(t - T)}$$

Time Factor



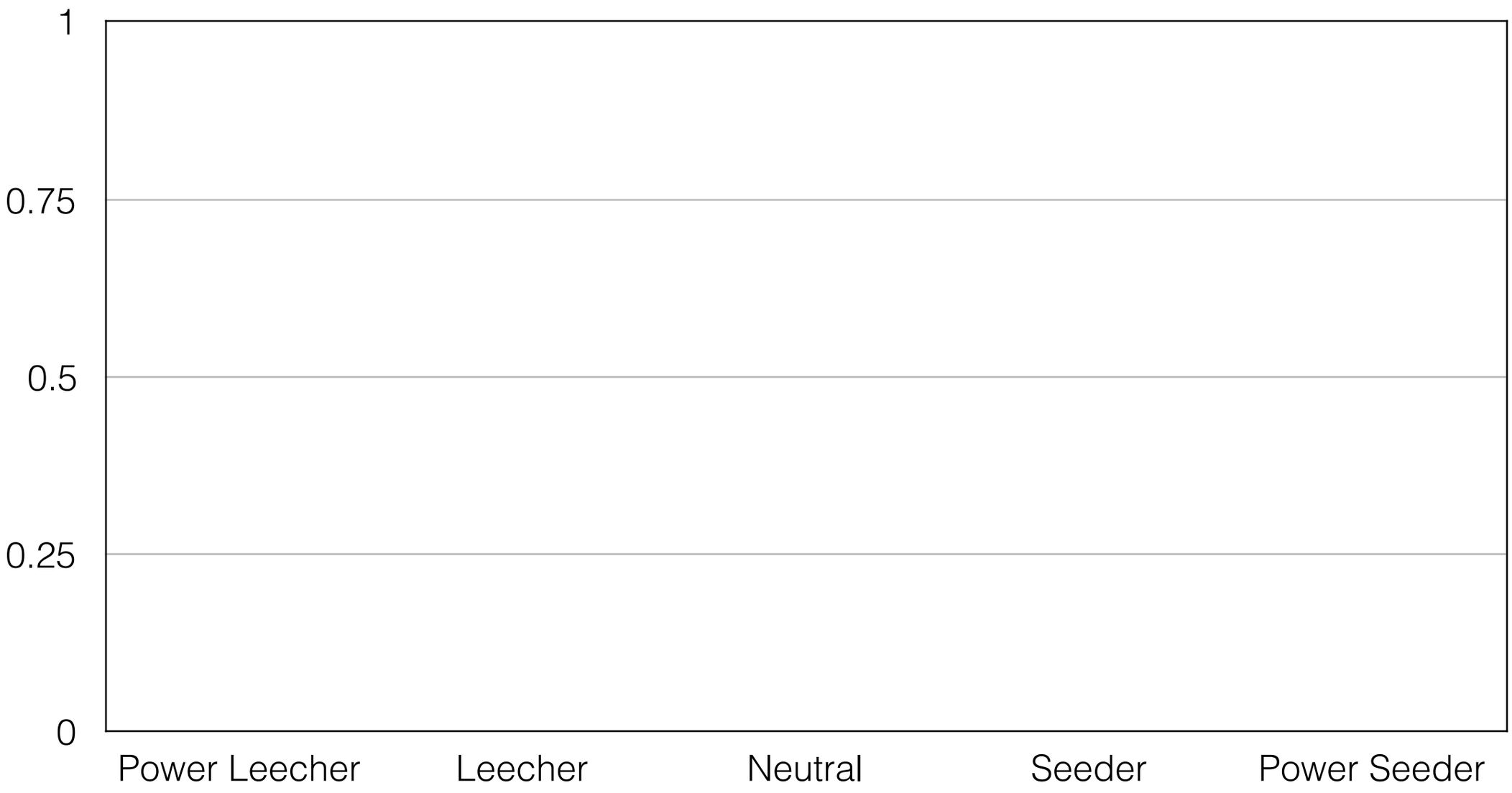
https://graphsketch.com

```
std::min(1.0, a + n(n+1)/2 * d)
```

Status: a=α, d=d1
Money:a=β, d=d2

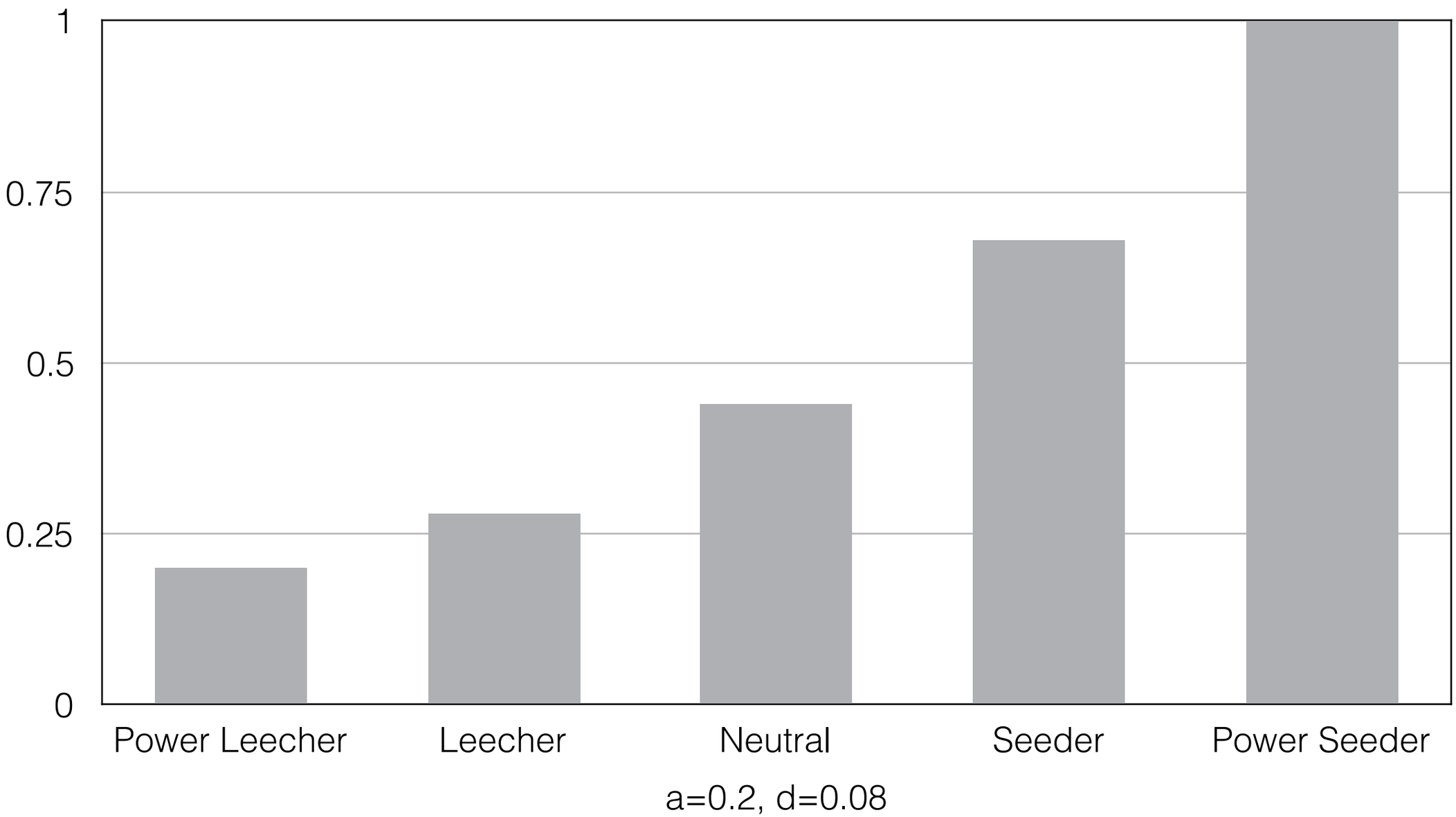
std::min(1.0, a + n(n+1)/2 * d)

Status: a=α, d=d1
Money:a=β, d=d2



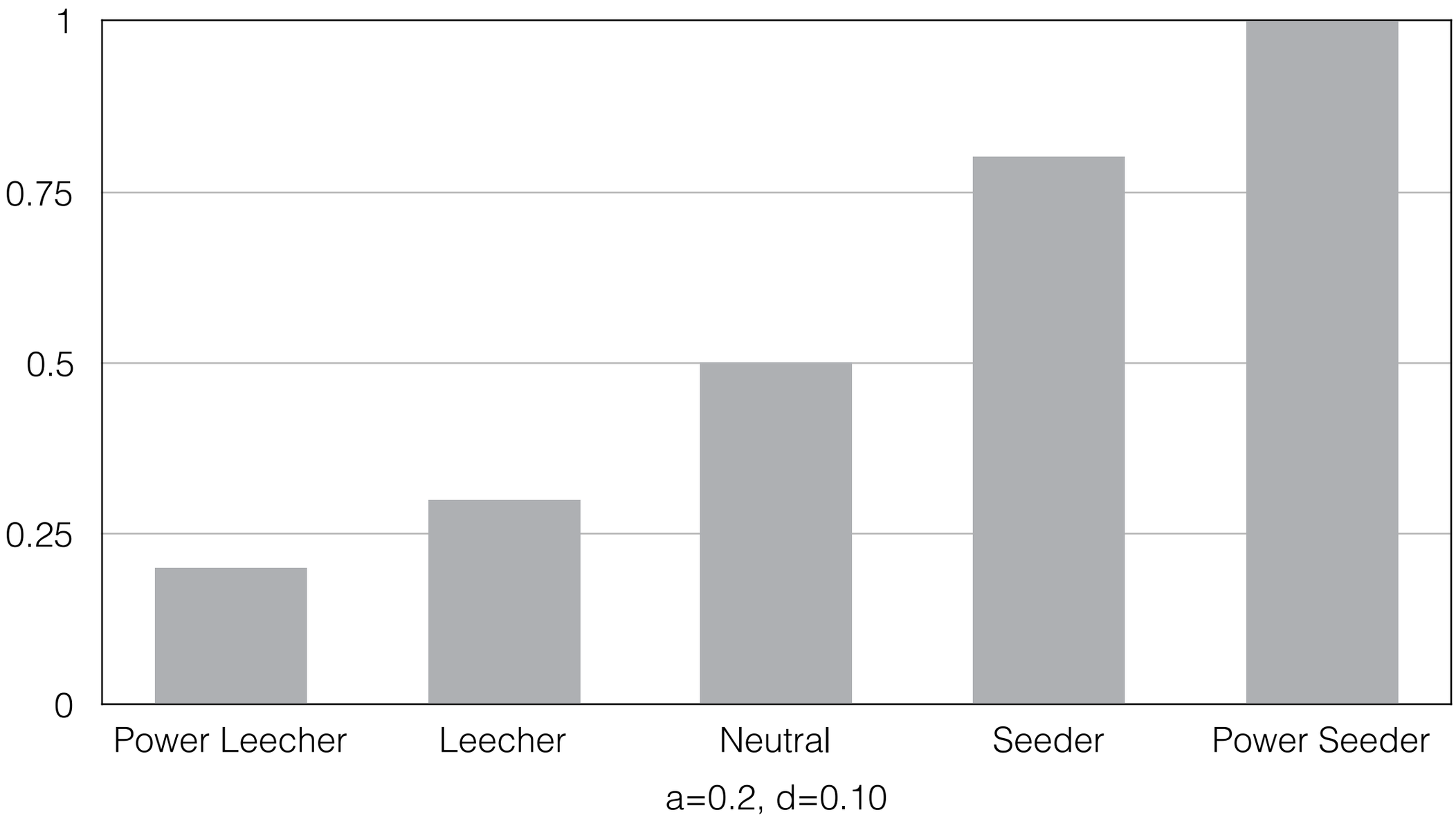
std::min(1.0, a + n(n+1)/2 * d)

Status: a=α, d=d1
Money:a=β, d=d2



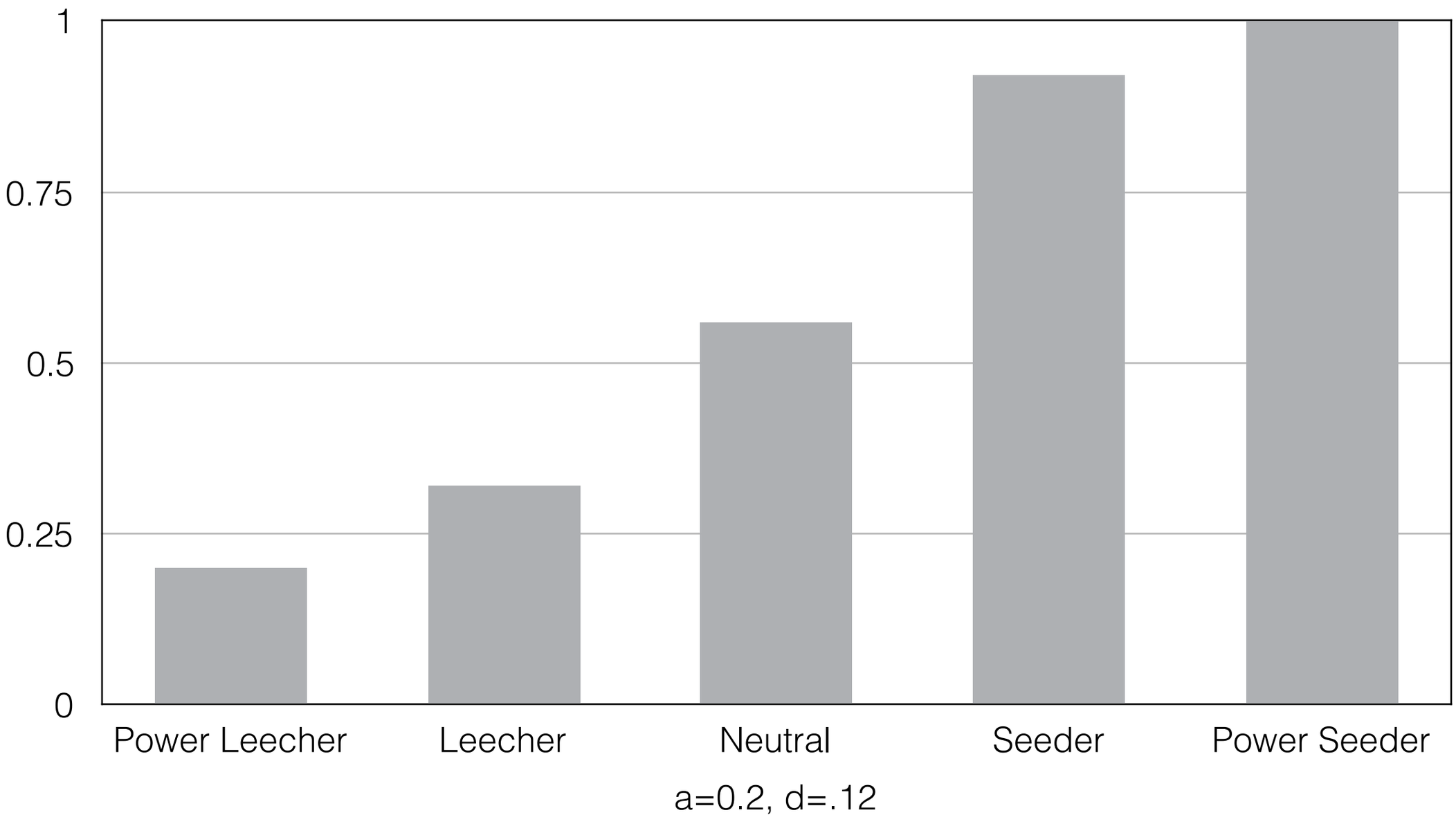
std::min(1.0, a + n(n+1)/2 * d)

Status: a=α, d=d1
Money:a=β, d=d2



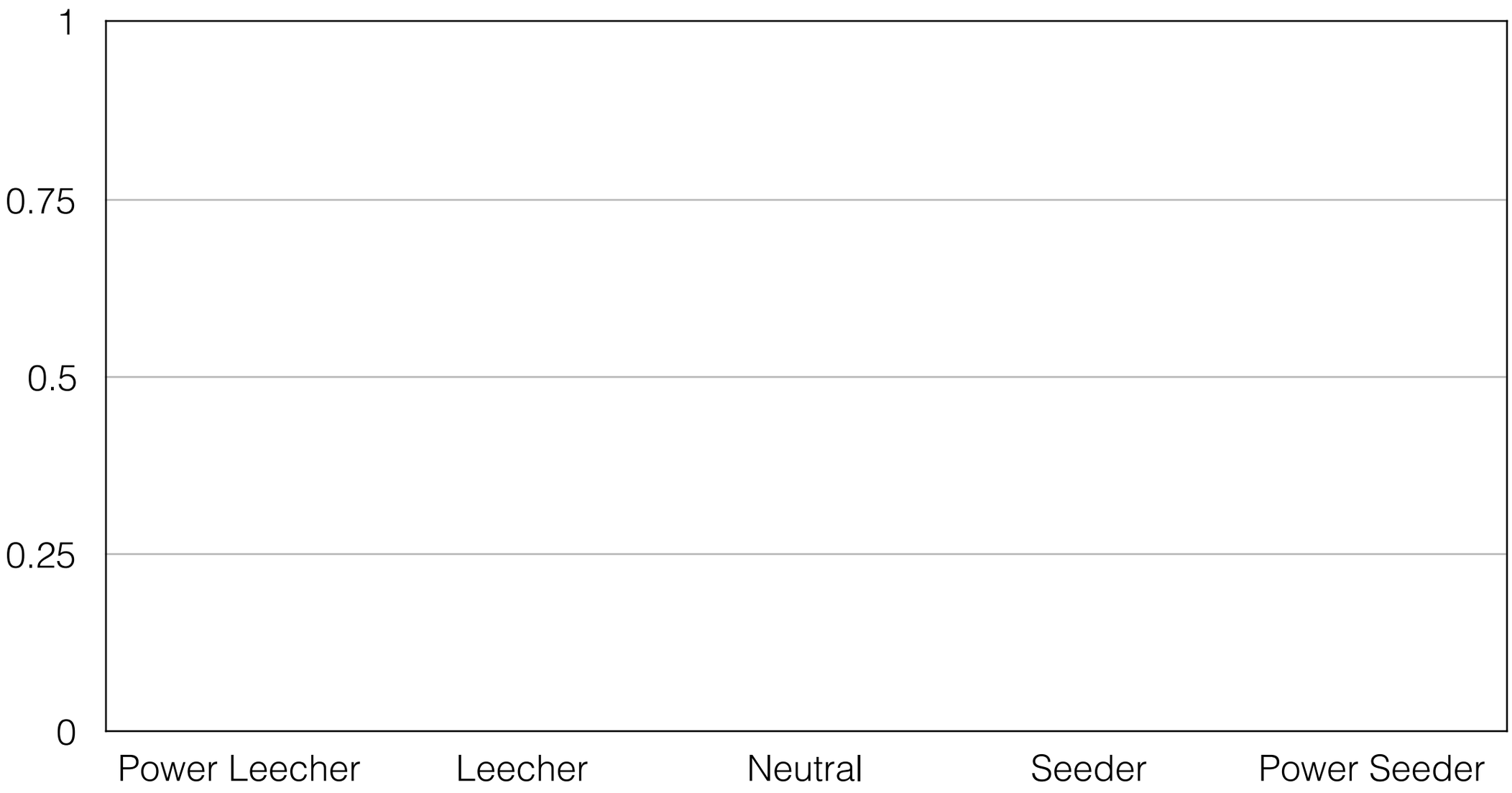
std::min(1.0, a + n(n+1)/2 * d)

Status: a=α, d=d1
Money:a=β, d=d2



std::min(1.0, a + n(n+1)/2 * d)

Status: a=α, d=d1
Money:a=β, d=d2



FoodThought

Seamless money allocation manager

FoodThought

~~Seamless money allocation manager~~

Seamless allocated money assistant

https://github.com/skgbanga/FoodThought