Continuous versus Abrupt Reorganizations

Anton Kolotilin and Hongyi Li (UNSW)

May 2018

Complicated systems

- Many systems are excessively complex ...
  - tax regulations
  - computer programs
  - *organizational bureaucracies and processes*
- ... due to accumulated design inefficiencies – *cruft*, *kludges*, etc.

Entanglements and kludges

- Design elements are interdependent (*entangled* with each other).
- Entanglements inhibit change:

  Fixes create problems elsewhere, necessitate further fixes, etc.
- Change may be delayed → inefficiencies persist and accumulate.

- Examples:
  - MS-DOS → Windows → Windows 95 ...
  - 1960s contracting processes at US Defense Dept
  - Obamacare

Entanglements and kludges

- Design elements are interdependent (*entangled* with each other).
- Entanglements inhibit change:

  Fixes create problems elsewhere, necessitate further fixes, etc.
- Change may be delayed → inefficiencies persist and accumulate.

- Examples:
  - MS-DOS → Windows → Windows 95 ...
  - 1960s contracting processes at US Defense Dept
  - Obamacare

This paper:

When complicated, entangled systems face continuous pressure to change,

- Should they adapt *continuously*?
- Or *abruptly* and episodically?

Abrupt change occurs in various settings:

- radical re-engineering in organizations
- big-bang reforms of public policy
- periodic refactoring in software development

Abrupt change is often associated with technical debt ↓, functionality ↓.

- Occurs with disruptive new products: e.g., iPhone.

This paper:

When complicated, entangled systems face continuous pressure to change,

- Should they adapt *continuously*?
- Or *abruptly* and episodically?

Abrupt change occurs in various settings:

- radical re-engineering in organizations
- big-bang reforms of public policy
- periodic refactoring in software development

Abrupt change is often associated with technical debt $\downarrow$, functionality $\downarrow$.

- Occurs with disruptive new products: e.g., iPhone.

This paper:

When complicated, entangled systems face continuous pressure to change,

- Should they adapt *continuously*?
- Or *abruptly* and episodically?

Abrupt change occurs in various settings:

- radical re-engineering in organizations
- big-bang reforms of public policy
- periodic refactoring in software development

Abrupt change is often associated with technical debt $\downarrow$, functionality $\downarrow$.

- Occurs with disruptive new products: e.g., iPhone.

Product design example

2011: Apple releases Final Cut Pro X (to replace Final Cut Pro 7).

> *Many users have expressed their frustration with a litany of missing features in Final Cut Pro X. To begin with, there's no support for output to tape ... There's no support for EDL or XML export ... There's no ...*
>
> *... because FCPX uses a completely re-architected underlying media handling and editing paradigm, it can't ...*
>
> *— arstechnica.com*

Stylised model of entangled systems

System Design:

- system with continuum of elements
- designer can add and delete elements
- ‘good’ elements randomly turn ‘bad’ over time

Entanglement:

- *exogenous* directed network structure over elements
- element deleted → direct and indirect children also deleted

Preview of results

For highly entangled designs, abrupt reorganization is optimal:

1. Delays in eliminating inefficiencies, then sudden large reorganizations.
    - Less entangled designs: continuous, immediate reorganization is optimal.

2. Driven by *untangling* effect: large one-time reorganization (endogenously) less disruptive than continuous, incremental reorganization.
3. Also driven by intertemporal tradeoff: patient designer will optimally cycle between 'clean' and 'dirty' designs.

4. Abrupt reorganization → discontinuous drop in performance (following gradual decline) ...
    - But performance improves rapidly afterward.

Preview of results

For highly entangled designs, abrupt reorganization is optimal:

1. Delays in eliminating inefficiencies, then sudden large reorganizations.
   - Less entangled designs: continuous, immediate reorganization is optimal.
2. Driven by *untangling* effect: large one-time reorganization (endogenously) less disruptive than continuous, incremental reorganization.
3. Also driven by intertemporal tradeoff: patient designer will optimally cycle between 'clean' and 'dirty' designs.
4. Abrupt reorganization $\rightarrow$ discontinuous drop in performance (following gradual decline) ...
   - But performance improves rapidly afterward.

Preview of results

For highly entangled designs, abrupt reorganization is optimal:

1. Delays in eliminating inefficiencies, then sudden large reorganizations.
   - Less entangled designs: continuous, immediate reorganization is optimal.

2. Driven by *untangling* effect: large one-time reorganization (endogenously) less disruptive than continuous, incremental reorganization.
3. Also driven by intertemporal tradeoff: patient designer will optimally cycle between ‘clean’ and ‘dirty’ designs.

4. Abrupt reorganization → discontinuous drop in performance (following gradual decline) ...
   - But performance improves rapidly afterward.

Preview of results

For highly entangled designs, abrupt reorganization is optimal:

1. Delays in eliminating inefficiencies, then sudden large reorganizations.
   - Less entangled designs: continuous, immediate reorganization is optimal.

2. Driven by *untangling* effect: large one-time reorganization (endogenously) less disruptive than continuous, incremental reorganization.
3. Also driven by intertemporal tradeoff: patient designer will optimally cycle between ‘clean’ and ‘dirty’ designs.

4. Abrupt reorganization → discontinuous drop in performance (following gradual decline) ...
   - But performance improves rapidly afterward.

Lit review

- Learning / experimenting on rugged landscapes
  - Kauffman 1989; Levinthal 1997; Billinger, Stieglitz and Schumacher 2013
- Kludges and long-run complexity
  - Ely 2011; Ellison and Holden 2013; Kawai, Lang and Li forthcoming
- Adaptation, coordination, and communication
  - Dessein and Santos 2006; Alonso, Dessein and Matouschek 2008

Road map

1. Intro
2. Model
3. The Coefficient of Friction
4. Myopic designer
5. Patient designer
6. Conclusion

The Model

- Time is continuous, $t \geq 0$.
- System S_t is a continuum of infinitesimal, equal-weighted elements.
- Good elements independently turn bad with constant decay rate λ.
- Designer's flow payoff depends on masses of good vs. bad elements:

$$\pi_t = m_G(t) - c\, m_B(t).$$

The designer

At each instant t, the designer may:

- Add good elements at bounded rate $a_t \leq \alpha$ (mass per unit time).
- Choose *target set* $D(t) \subseteq S_t$ of elements for deletion.

Elements are entangled:

- *Exogenous* network of directed links between elements.
- Element x targeted $\rightarrow$ all children, grandchildren, ... also deleted.
  * Notation: set D targeted $\rightarrow$ *collateral set* $C(D, S_t) \supset D$ deleted.
- "Mean-field" implementation of network formation/deletion process; similar to Barabási and Albert 1999.

The designer

At each instant t, the designer may:

- Add good elements at bounded rate $a_t \leq \alpha$ (mass per unit time).
- Choose *target set* $D(t) \subseteq S_t$ of elements for deletion.

Elements are entangled:

- *Exogenous* network of directed links between elements.
- Element x targeted $\rightarrow$ all children, grandchildren, ... also deleted.
- ∗ Notation: set D targeted $\rightarrow$ *collateral set* $C(D, S_t) \supset D$ deleted.

- "Mean-field" implementation of network formation/deletion process; similar to Barabási and Albert 1999.

The designer

At each instant t, the designer may:

- Add good elements at bounded rate $a_t \leq \alpha$ (mass per unit time).
- Choose *target set* $D(t) \subseteq S_t$ of elements for deletion.

Elements are entangled:

- *Exogenous* network of directed links between elements.
- Element x targeted $\rightarrow$ all children, grandchildren, ... also deleted.

∗ Notation: set D targeted $\rightarrow$ *collateral set* $C(D, S_t) \supset D$ deleted.

- "Mean-field" implementation of network formation/deletion process; similar to Barabási and Albert 1999.

Network: preview of key features

- Homogenous, ‘detail-free’ network;
  so, ‘big-picture’ view of system is sufficient.

- Entanglement is ‘non-localised’:
  as system grows, each element accumulates more links.

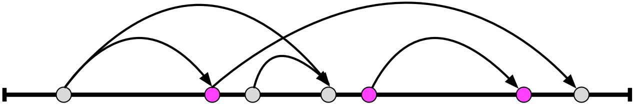
Network – Part 1: link formation

- Each new element x links with finite (i.e., zero mass) random subset of existing elements. Specifically ...

  * Each new element and each existing element become linked with probability $\kappa \cdot dm$, where dm is infinitesimal element mass;

    ($\kappa > 0$ represents density of *entanglement*.)

  * i.e., in expectation, each new element forms $\kappa \cdot m$ links.



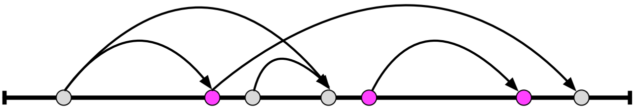
Network – Part 1: link formation

- Each new element x links with finite (i.e., zero mass) random subset of existing elements. Specifically ...

∗ Each new element and each existing element become linked with probability $\kappa \cdot dm$, where dm is infinitesimal element mass;

  ($\kappa > 0$ represents density of *entanglement*.)

∗ i.e., in expectation, each new element forms $\kappa \cdot m$ links.

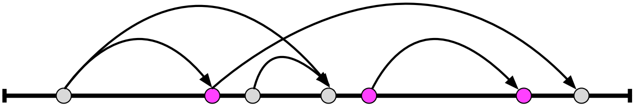


Network – Part 2: link directions

- Elements are ranked. Links are directed towards lower-ranked elements.

* So, network is acyclic – which ensures ‘finite entanglement’.

- Each new element is uniformly randomly assigned a rank.



What does the designer know?

The Designer:

- Observes the type (good or bad) of each element in S_t.
- Understands the network formation process,
  but *doesn't observe time-t network*.

* Upon deleting x, immediately observes deletion of x's descendants.

Continuous ingredients

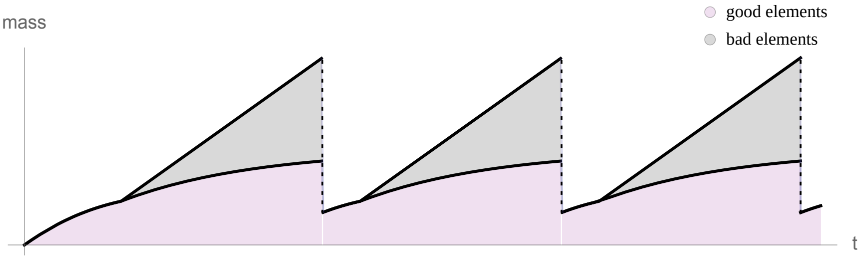
- continuous time
- continuous space (continuum of elements)
- continuous pressure to reorganize (decay process)
- continuous network frictions

$\Rrightarrow$ continuous / discontinuous reorganizations?

Main result: a preview

Proposition (Informal)

The designer's optimal strategy involves 'abrupt reorganizations' iff entanglement κ is sufficiently high.



Simplifying the problem

Designer's time-t problem:

Given system $S(t)$ and (beliefs about) network $E(t)$,
Choose growth rate g_t and deletion set $D(t)$

to maximize

$$\int_0^\infty e^{-rt} \underbrace{(m_G(t) - c\, m_B(t))}_{\text{flow payoff}} dt.$$

Simplifying the problem

Myopic Designer's time-t problem:

Given system $S(t)$ and (beliefs about) network $E(t)$,
Choose growth rate g_t and deletion set $D(t)$

to maximize

$$\frac{d}{dt} \underbrace{(m_G(t) - c\, m_B(t))}_{\text{flow payoff}}.$$

The details don't matter

Following any history h_t, the designer believes that:

- Links are uniformly randomly distributed across element-pairs.
- Each element's rank is uniformly randomly distributed.

$\Rightarrow$ So, all good elements look alike; all bad elements look alike.

In the optimal strategy,

- Only bad elements are targeted.
- Good elements are added at maximal rate: $a(t) \equiv \alpha$.

$\Rightarrow$ So, designer simply chooses *how many* bad elements to target.

The details don't matter

Following any history h_t, the designer believes that:

- Links are uniformly randomly distributed across element-pairs.
- Each element's rank is uniformly randomly distributed.

$\Rightarrow$ So, all good elements look alike; all bad elements look alike.

In the optimal strategy,

- Only bad elements are targeted.
- Good elements are added at maximal rate: $a(t) \equiv \alpha$.

$\Rightarrow$ So, designer simply chooses *how many* bad elements to target.

The coefficient of Friction

Given:

- system S and network E
- target set $D \subset S$ of bad elements
- collateral set $C(D,S)$ with mass Δ_B of bad elements

The (coefficient of) *Friction*

$$F(\underbrace{m}_{\text{mass}}, \underbrace{m_G/m_B}_{\text{ratio}}, \underbrace{\Delta_B}_{\text{scale}}) = \frac{\Delta_G}{\Delta_B}$$

is the ratio of good to bad elements in $C(D,S)$.

Friction = endogenous reorganization cost

At time t, the Designer chooses

flow rates of deletion $\beta_G(t), \beta_B(t)$

discrete masses of deletion $\Delta_G(t), \Delta_B(t)$

to control the system $(m_G(t), m_B(t))$

$$dm_G(t) = \underbrace{\alpha dt}_{\text{growth}} - \underbrace{\lambda m_G(t)dt}_{\text{decay}} - \underbrace{\big(\beta_G(t)dt + \Delta_G(t)\big)}_{\text{removal}},$$

$$dm_B(t) = \underbrace{\lambda m_G(t)dt}_{\text{decay}} - \underbrace{\big(\beta_B(t)dt + \Delta_B(t)\big)}_{\text{removal}}$$

subject to Frictional constraints

$$\frac{\Delta_G(t)}{\Delta_B(t)} = F\left(m(t), m_G(t)/m_B(t), \Delta_B(t)\right),$$

$$\frac{\beta_G(t)}{\beta_B(t)} = F\left(m(t), m_G(t)/m_B(t), 0\right).$$

Friction = endogenous reorganization cost

At time t, the Designer chooses

flow rates of deletion $\beta_G(t), \beta_B(t)$

discrete masses of deletion $\Delta_G(t), \Delta_B(t)$

to control the system $(m_G(t), m_B(t))$

$$dm_G(t) = \underbrace{\alpha dt}_{\text{growth}} - \underbrace{\lambda m_G(t)dt}_{\text{decay}} - \underbrace{\left(\beta_G(t)dt + \Delta_G(t)\right)}_{\text{removal}},$$

$$dm_B(t) = \underbrace{\lambda m_G(t)dt}_{\text{decay}} - \underbrace{\left(\beta_B(t)dt + \Delta_B(t)\right)}_{\text{removal}}$$

subject to Frictional constraints

$$\frac{\Delta_G(t)}{\Delta_B(t)} = F\left(m(t), m_G(t)/m_B(t), \Delta_B(t)\right),$$

$$\frac{\beta_G(t)}{\beta_B(t)} = F\left(m(t), m_G(t)/m_B(t), 0\right).$$

Friction = endogenous reorganization cost

At time t, the Designer chooses

flow rates of deletion $\beta_G(t), \beta_B(t)$

discrete masses of deletion $\Delta_G(t), \Delta_B(t)$

to control the system $(m_G(t), m_B(t))$

$$dm_G(t) = \underbrace{\alpha dt}_{\text{growth}} - \underbrace{\lambda m_G(t) dt}_{\text{decay}} - \underbrace{\left(\beta_G(t) dt + \Delta_G(t)\right)}_{\text{removal}},$$

$$dm_B(t) = \underbrace{\lambda m_G(t) dt}_{\text{decay}} - \underbrace{\left(\beta_B(t) dt + \Delta_B(t)\right)}_{\text{removal}}$$

subject to Frictional constraints

$$\frac{\Delta_G(t)}{\Delta_B(t)} = F\left(m(t), m_G(t)/m_B(t), \Delta_B(t)\right),$$

$$\frac{\beta_G(t)}{\beta_B(t)} = F\left(m(t), m_G(t)/m_B(t), 0\right).$$

Friction: properties

Friction $F(m, m_G/m_B, \Delta_B)$ is deterministic;

$\Rightarrow$ Designer can deterministically control evolution of (m_G, m_B).

Lemma

Friction is:

1. *increasing in mass m*
2. *increasing in good/bad ratio m_G/m_B*
3. *increasing in entanglement κ*
4. ***decreasing in scale of reorganization Δ_B***

Friction: properties

Friction $F(m, m_G/m_B, \Delta_B)$ is deterministic;

$\Rightarrow$ Designer can deterministically control evolution of (m_G, m_B).

Lemma

Friction is:

1. *increasing in mass m*
2. *increasing in good/bad ratio m_G/m_B*
3. *increasing in entanglement κ*
4. *decreasing in scale of reorganization Δ_B*

Friction: properties

Friction $F(m, m_G/m_B, \Delta_B)$ is deterministic;

$\Rightarrow$ Designer can deterministically control evolution of (m_G, m_B).

Lemma

Friction is:

1. *increasing in mass m*
2. *increasing in good/bad ratio m_G/m_B*
3. *increasing in entanglement κ*
4. ***decreasing in scale of reorganization Δ_B***

Friction: intuition

C comprises (i) target set D and (ii) descendants D' of targets:

$$\underbrace{C}_{\text{collateral set}} = \underbrace{D}_{\text{bad elements only}} \bigcup \underbrace{D'}_{\text{random draws from } S}$$

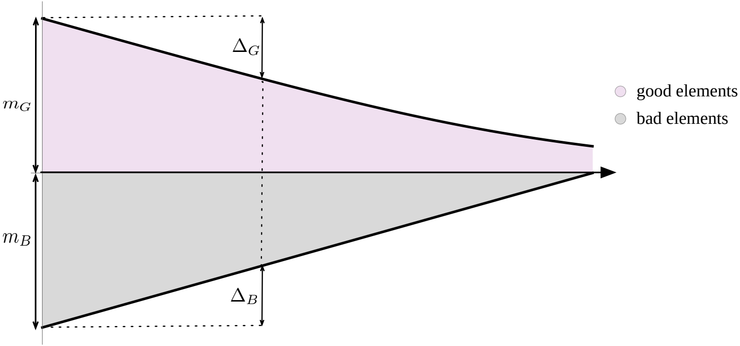
Friction F is good/bad ratio of $C = D \cup D'$:

1. As mass m increases, D' increases in size $\Rightarrow F$ increases.
2. As ratio $\frac{m_G}{m_B}$ increases, more good elements in $D' \Rightarrow F$ increases.
3. As entanglement κ increases, D' increases in size $\Rightarrow F$ increases.
4. **As scale Δ_B increases ... ?**

How does friction F change with scale Δ_B?

Compare:

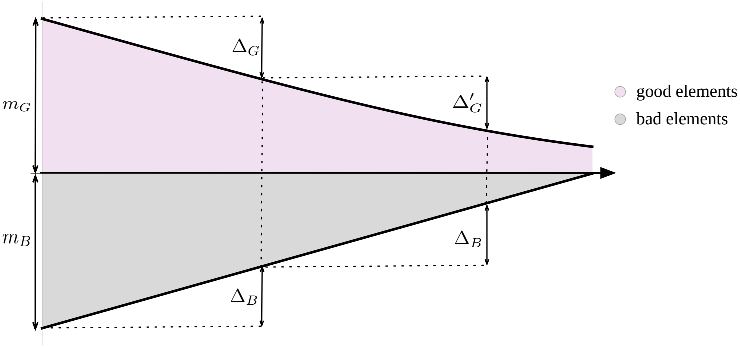
$$\underbrace{\frac{\Delta_G}{\Delta_B}}_{\text{friction for initial } \Delta_B}$$



How does friction F change with scale Δ_B?

Compare:

$$\underbrace{\frac{\Delta_G}{\Delta_B}}_{\text{friction for initial } \Delta_B} \quad \text{vs.} \quad \underbrace{\frac{\Delta'_G}{\Delta_B}}_{\text{friction for incremental } \Delta_B}$$

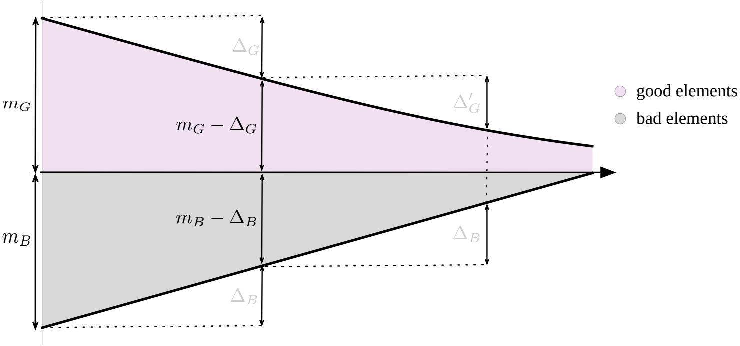


‘Decontamination’ effect ...

- Collateral set “over-samples” bad elements: $\frac{\Delta_G}{\Delta_B} < \frac{m_G}{m_B}$.
- So, after initial deletion, remaining good/bad ratio increases,

$$\underbrace{\frac{m_G}{m_B}}_{\text{before}} < \underbrace{\frac{m_G - \Delta_G}{m_B - \Delta_B}}_{\text{after}}.$$

$\Rightarrow$ Tendency for initial friction < incremental friction.

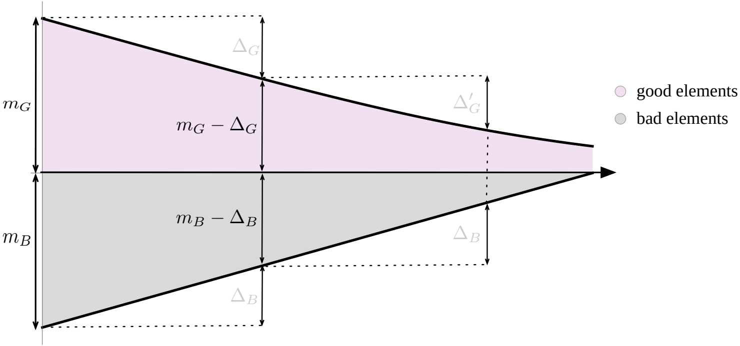


... versus 'Untangling' effect

- As elements get deleted, links w/ remaining elements vanish;
- So, after deletion, number of descendants/element decreases,

$$\underbrace{\kappa(m_G + m_B)}_{\text{before}} > \underbrace{\kappa(m_G - \Delta_G + m_B - \Delta_B)}_{\text{after}}.$$

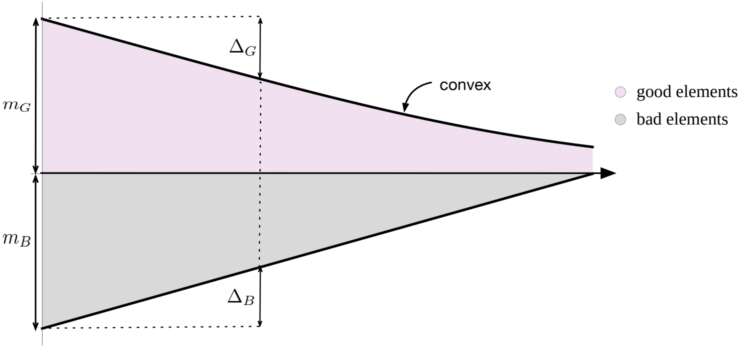
$\Rightarrow$ Tendency for initial friction > incremental friction.



Comparing effects

Untangling effect dominates decontamination effect, so

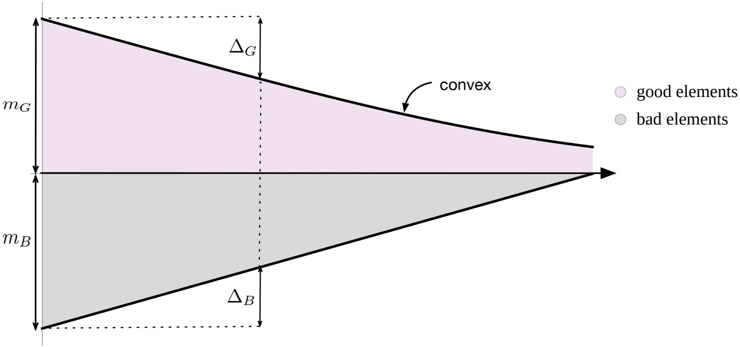
$$\underbrace{\frac{\Delta_G}{\Delta_B}}_{\text{friction for initial } \Delta_B} > \underbrace{\frac{\Delta'_G}{\Delta_B}}_{\text{friction for incremental } \Delta_B}$$



How does friction F change with scale Δ_B?

Conclusion: friction decreases as scale increases, i.e.,

$$\frac{\Delta_G}{\Delta_B} \text{ is decreasing in } \Delta_B.$$



Myopic designer performs full cleansing

Consider a myopic designer: i.e., maximizes $\frac{d}{dt}\left(m_G(t) - cm_B(t)\right)$.

- Friction is decreasing in scale Δ_B, so ...
- Whenever *any* bad elements are removed, *all* bad elements are removed:

Lemma

At any instant t, if either $\beta_B(t) > 0$ or $\Delta_B(t) > 0$, then

$$\Delta_B(t) = m_B(t).$$

Optimal modes for a myopic designer

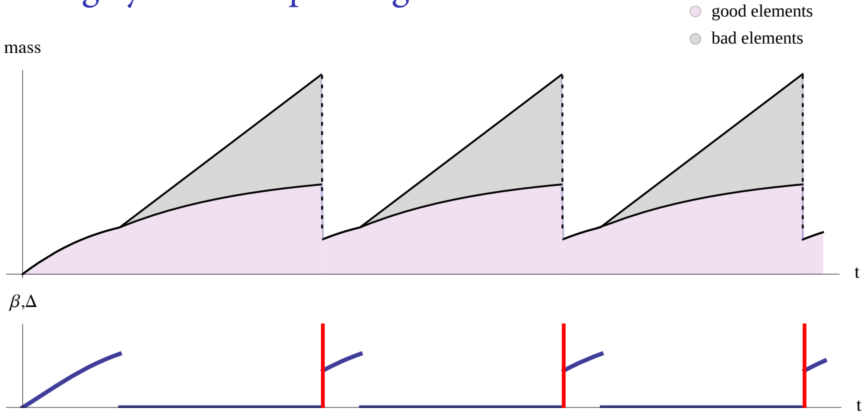
Proposition

The myopic designer's optimal strategy is unique. Depending on parameters, it takes one of two forms:

1. *a 'cleansing-cycles' mode.*
2. *a 'steady-state' mode.*

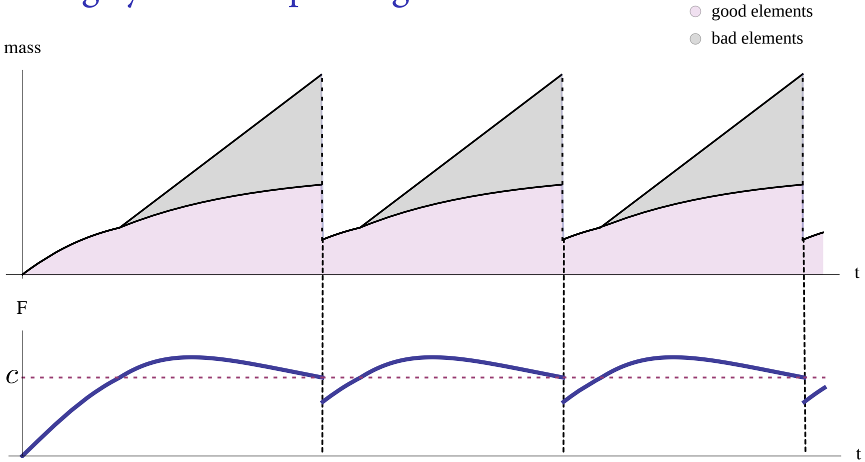
If entanglement κ is sufficiently high, then cleansing cycles are optimal.

Cleansing cycles: abrupt reorganizations



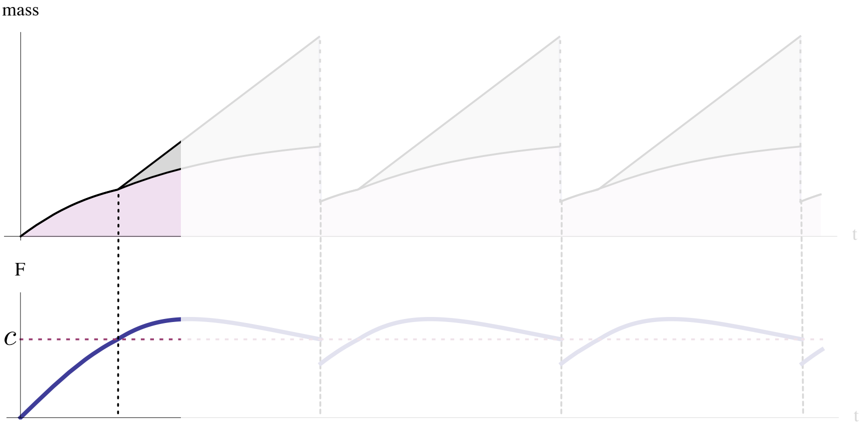
Reorganization occurs episodically:
abrupt cleansing → continuous cleansing → no cleansing → abrupt ...

Cleansing cycles: abrupt reorganizations



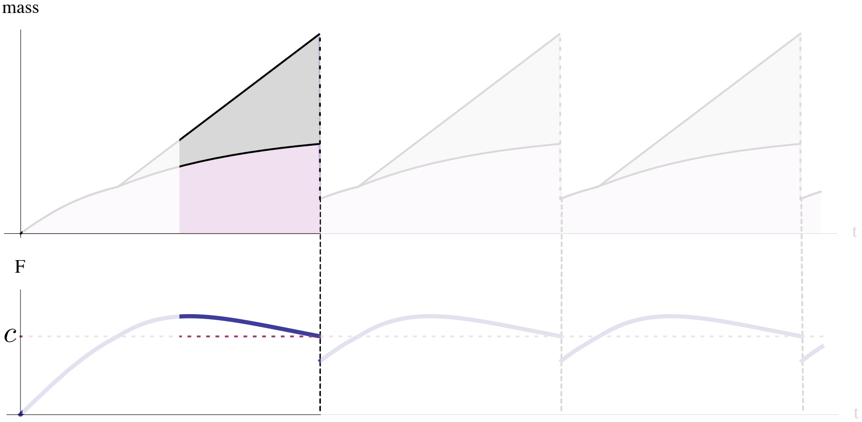
Reorganization occurs whenever friction is low: $F < c$.

Cleansing cycles: a walkthrough



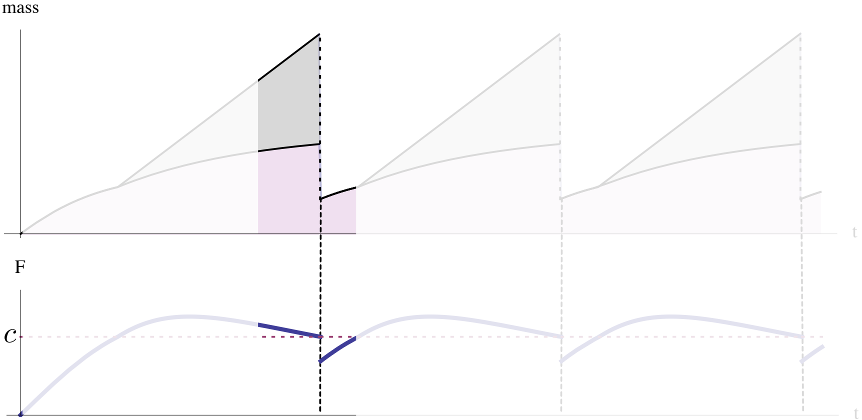
Initially, friction F increases as system grows.
When $F > c$, designer stops cleansing;
$\Rightarrow$ contamination begins (m_G/m_B decreases).

Cleansing cycles: a walkthrough



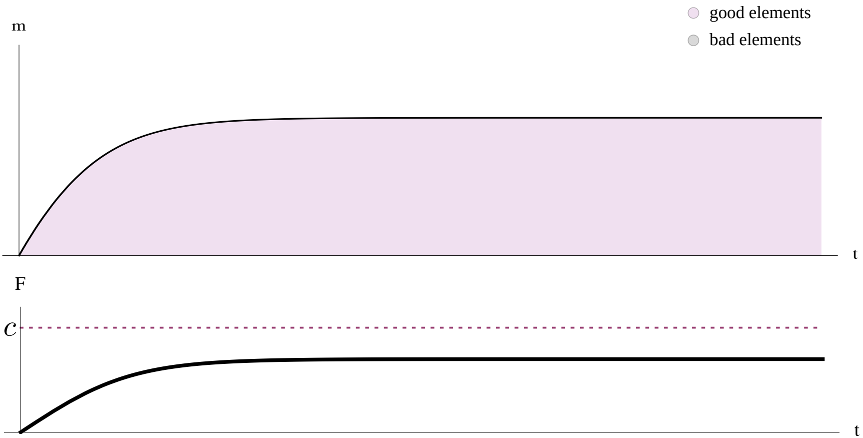
Eventually, ‘contamination effect’ starts to dominate;
⇒ friction F starts decreasing.

Cleansing cycles: a walkthrough



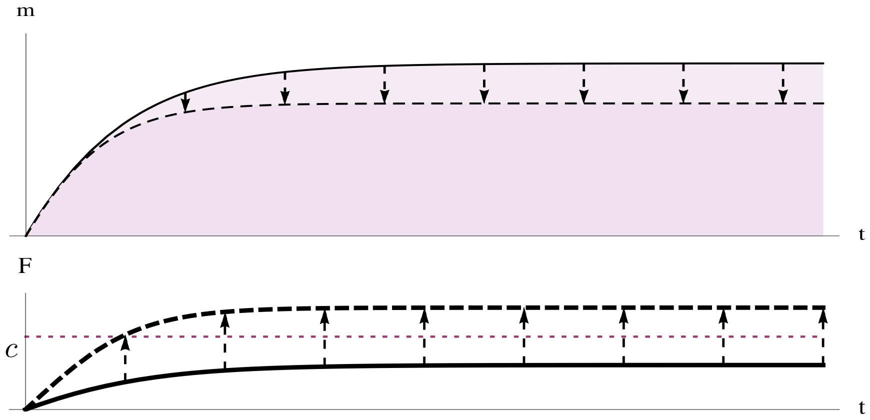
When friction $F = c$, cleansing event is triggered.
Abrupt 'full cleansing' due to 'untangling' effect.

Steady-state mode



In steady-state mode: cleansing occurs constantly.

Entangled systems and abrupt reorganizations



Increase in entanglement κ → steady-state mode is inefficient
→ cleansing cycles are optimal

Comparative statics

Proposition

Cleansing Cycles are optimal when:

1. ***entanglement** (κ) **is high**.*
2. *burden imposed by bad elements (c) is low.*
3. *productivity/innovativeness of designer (α) is high.*
4. *rate of decay (λ) is low.*

Patient designer

Suppose designer is non-myopic: maximizes

$$\int_0^\infty e^{-rt} \underbrace{(m_G(t) - c\, m_B(t))}_{\text{flow payoff}} dt \text{ with } r < \infty.$$

Simplifying assumption: network is (countably) *dense*, i.e., $\kappa \to \infty$.

Dense network: $\kappa \to \infty$

Consider a small target set D.

Recall: C comprises (i) target set D and (ii) descendants D' of targets,

$$\underbrace{C}_{\text{collateral set}} = \underbrace{D}_{\text{bad elements only}} \bigcup \underbrace{D'}_{\text{random draws from } S}$$

Given $\kappa \to \infty$,

- D' (vastly) outnumbers D, so ...
- C is approx. random sample of $S = (m_G, m_B)$.

At the $\kappa \to \infty$ limit, $F(m, m_G/m_B, \Delta_B) \equiv m_G/m_B$.

$\Rightarrow$ Friction is constant in scale Δ_B; i.e., ‘untangling’ effect is absent.

Dense network: $\kappa \to \infty$

Consider a small target set D.

Recall: C comprises (i) target set D and (ii) descendants D' of targets,

$$\underbrace{C}_{\text{collateral set}} = \underbrace{D}_{\text{bad elements only}} \bigcup \underbrace{D'}_{\text{random draws from } S}$$

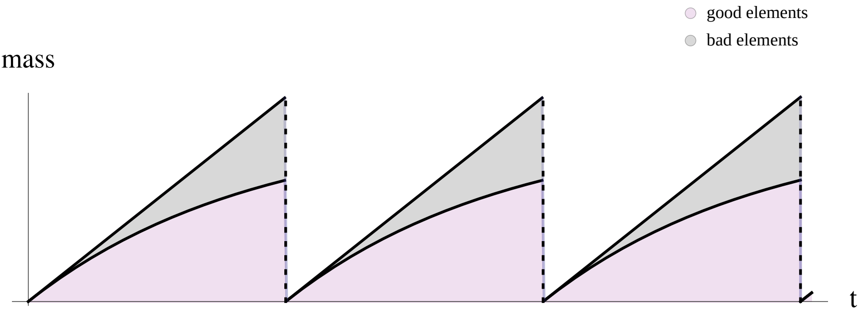
Given $\kappa \to \infty$,

- D' (vastly) outnumbers D, so ...
- C is approx. random sample of $S = (m_G, m_B)$.

At the $\kappa \to \infty$ limit, $F(m, m_G/m_B, \Delta_B) \equiv m_G/m_B$.

$\Rightarrow$ Friction is constant in scale Δ_B; i.e., ‘untangling’ effect is absent.

Dense networks and non-myopic designer



Proposition

With non-myopic designer and dense network:
Cleansing cycles are strictly optimal.

Why cleansing cycles (even without untangling)?

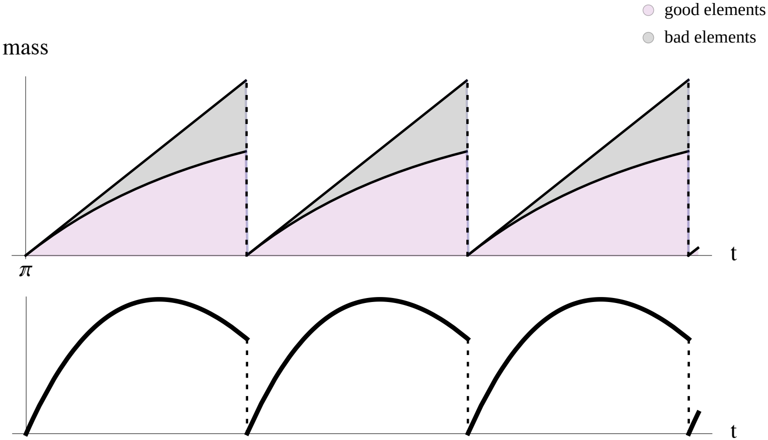
Intuition – designer has two conflicting objectives:

- Maintain high m_G, low m_B over time (on average).
- Reorganize cheaply, i.e., when friction $F = \frac{m_G}{m_B}$ is low.

How to reconcile these objectives?

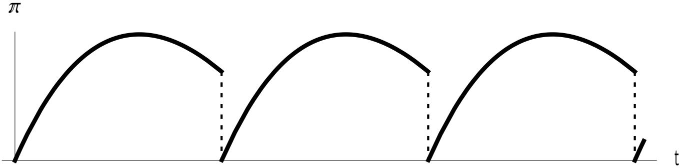
- Allow system to cycle between 'clean' (high F) and 'dirty' (low F).
- Concentrate reorganizations on times with low F.

How does reorganization affect productivity?



Reorganization: discontinuous drop in payoffs
(foreshadowed by gradual decline).

How does reorganization affect productivity?



For a patient designer,

- Drop in payoffs from reorganization is worthwhile:
  * Following reorganization, payoff decline is reversed.

- Designer delays past peak payoff to reorganize:
  * Maximizes time spent at peak / near-peak payoffs.

1. Intro
2. Model
3. The Coefficient of Friction
4. Myopic designer
5. Patient designer
6. Conclusion

Recap

1. Abrupt reorganizations iff system is highly entangled.
2. Abrupt reorganization $\rightarrow$ functionality $\downarrow$, technical debt $\downarrow$.
   - Introduction of disruptive new products: e.g., iPhone.
3. Abrupt reorganization $\rightarrow$ discrete drop in performance.
   - But performance improves rapidly afterward.
4. Reorganizations are not triggered by discrete technological shock.
   - e.g., incremental improvements in battery, touchscreen, CPU, storage tech $\rightarrow$ iPhone.

What's next

Immediate:

- Extend results to general case: $\kappa < \infty$ and $r < \infty$.

On the agenda:

- How to model different network structures?
- How to endogenize entanglement? (i.e., modularization.)
- How to introduce decentralization / strategic interactions?