

Internet Economics and Financial Technology - Reviewed Notes

Dom Hutchinson

December 1, 2020

Contents

1	Economic Theory	3
1.1	General	3
1.2	Micro-Economics	3
1.2.1	Supply-and-Demand Curve	4
1.2.2	Economies of Scale	6
1.2.3	Price Elasticity	7
1.2.4	Price Discrimination	8
1.3	Auction Theory	9
1.3.1	Types of Forward Auction	10
1.3.2	Comparing Forward Auctions	13
2	Market Economics	13
2.1	Market Metrics	16
2.2	Options Markets	18
2.2.1	Options	18
2.2.2	Option Strategies	20
2.3	Trading Algorithms	21
2.4	Prediction Markets	24
3	Economics of the Internet	25
3.1	Digital Goods	27
3.2	Online vs Offline Businesses	29
3.3	Online Price Discrimination Strategies	29
3.4	Economics of Computer Games	30
3.5	Online Auctions	31
3.6	The Crowd Economy	32
4	Financial Technology	32
4.1	FinTech Firms	33
4.2	Blockchain	34

4.2.1	Road to Bitcoin	35
4.2.2	Bitcoin	37
4.2.3	Innovating on Bitcoin	39
4.3	Sentiment Analysis	41
4.3.1	Theory	41
4.3.2	Algorithms	42
4.3.3	Practice	43
0	Reference	II
0.1	Empirical Methods	II
0.2	Option Strategies	IV

1 Economic Theory

1.1 General

Definition 1.1 - *Economic Agent*

An *Economic Agent* is someone who interacts with an economic market. *Economic Agents* are typically split into two categories

- *Producers* - Those making products which are then sold.
- *Consumers* - Those who buy the products.

In practice most people fulfil both roles (e.g. factories sell products but buy raw materials) but not in the same market.

Definition 1.2 - *Externality*

An *Externality* is a cost or benefit caused by the production or consumption of a good which is not financial (e.g. Pollution is a negative externality).

Definition 1.3 - *Monopolised Markets*

A market is *Monopolised* if it is dominated by a single seller. In a *Monopolised* market the *Monopolising* faces little, or no, competition and thus can be *price setters*, rather than price takers.

IRL a firm with over 40% market share is considered to have a monopoly.

Remark 1.1 - *Experimental Economics - Human Trading Behaviours*

Experimental Economics is a field of Economics where lab-style studies are used to study economic questions.

In this unit we are interested in how *Experimental Economics* approaches the study of human trading behaviours. The general approach is the following

- Have a small number of human subjects split into “buyers” and “sellers”.
- Assign each subject a private-value/limit price which they are not allowed to exceed.
- Provide subjects with a market to interact with where they can post prices.
- Observe the subjects behaviours under different market conditions.

Proposition 1.1 - *Experiment Design*

1.2 Micro-Economics

Definition 1.4 - *Micro-Economics*

Micro-Economics is the study of the behaviour of individual economic agents (e.g. individuals or businesses) in the allocation and making of limited resources. Typically these agents are partitioned into consumers and producers.

This can be considered as the study of *Supply-and-Demand*.

Remark 1.2 - *Effect of Scarcity*

Producers have a limited number of resources which they can convert into products and consumers have a limited amount of money with which to purchase goods. This means there

are prices which are too low for a producer to be willing to sell at, and prices which are too high for a consumer to be willing to purchase at.

Definition 1.5 - Production Costs

The cost incurred by a producer to produce a product can be split into three categories. These categories aim to explain how costs are effected by quantity produced.

- *Fixed Costs* - Costs a producer incurs regardless of amount produced (e.g. rent).
- *Variable Costs* - Costs which depend on quantity being produced (e.g. raw materials).
- *Semi-Variable Costs* - Costs which in part stay consistent and in part vary with quantity produced (e.g. commissions in wages).

Remark 1.3 - Production Costs

These categories are very idealised. *Fixed Costs* do actually increase when the quantity produced increases (e.g. needing to rent a large factory) but at a much slower (and likely stepped) fashion that *Variable Costs*.

Definition 1.6 - Net Utility

Net Utility for a product is the difference between the price a consumer pays for it and the perceived value they have for the product. This is similar to *Consumer Excess* and consumers want to maximise their *Net Utility*.

1.2.1 Supply-and-Demand Curve

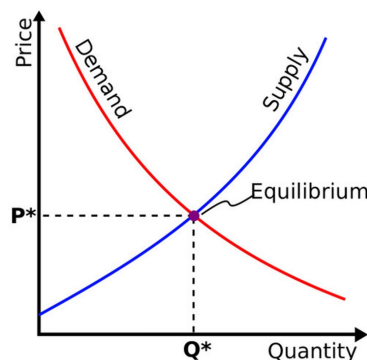


Figure 1: Example of Market Supply-and-Demand Curves

Definition 1.7 - Supply-and-Demand Curve

Supply-and-Demand Curves are plots of price-per-unit against quantity, and summarise the price per unit agents in a market (consumers and producers) are willing to accept for different quantities of a product.

A *Market Demand Curve* shows the total quantity of a product which all consumers in a market would buy at different price points. A *Demand Curve* slopes down as it is assumed that consumers are always willing to accept to a lower price-per-unit but not necessarily a higher price.

A *Market Supply Curve* shows the total quantity of products all producers in a market would produce if they were to receive certain prices. A *Supply Curve* slopes upwards as it is assumed that producers will always accept a higher price, but not necessarily a lower price.

See Figure 1.1 for an example of these curves.

Definition 1.8 - Business Supply Curve

A *Business Supply Curve* is similar to a *Market Supply Curve* except it is only concerned with a single business in the market. The *Business Supply Curve* plots the minimum price-per-unit a business is willing to accept for different quantities of goods sold. Generally this tracks the marginal cost of producing the goods at these quantities.

Business Supply Curves tend to be truncated as there are quantities which are too small for a business to deem them feasible at all.

Remark 1.4 - Shifting Supply-and-Demand Curves

Supply-and-Demand Curves can *Shift*. These *Shifts* occur due to non-price factors such as a pandemic, or improvements in the production process. A *Shift* will cause a movement in the *Market Equilibrium*.

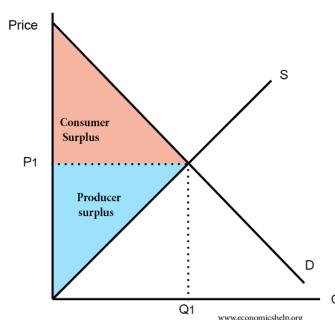


Figure 2: Illustration of excess for given prices

Definition 1.9 - Surplus

Surplus is the difference between the price an agent was willing to accept and the price they actually accepted (ie the market price). All agents want to maximise their surplus (ie Producers want as high a price as possible & Consumers want as low a price as possible).

Definition 1.10 - Equilibrium

A competitive market the *Market Equilibrium* is the price at which the quantity demanded and quantity supplied by a market is equal (ie the intersection of supply and demand curves). This is the price at which *Surplus* is maximised for both consumers and producers.

This can be seen in a *Supply-Demand Graph* as the point where the supply and demand curves intersect.

Remark 1.5 - Dynamic Variation of Equilibrium

In reality *Market Equilibria* are dynamic as consumers and producers often leave the market after completing a transaction. This is more apparent in small markets.

Definition 1.11 - Pareto Efficient Allocation

An allocation of resources is *Pareto Efficient* if any change to the allocation would definitely end up with someone being worse off.

1.2.2 Economies of Scale

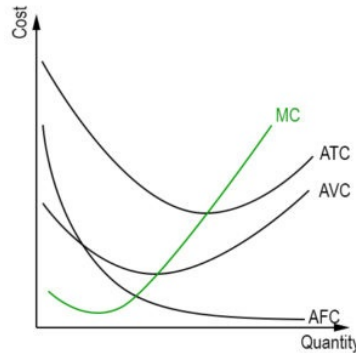


Figure 3: Illustration of typical relationship between quantity produced and: Marginal Cost (MC); Average Total Costs (ATC); Average Variable Costs (AVC); and, Average Fixed Costs (AFC)

Definition 1.12 - Marginal Cost

Marginal Cost of production is the cost to produce one extra unit. This value will vary depending upon the total number of units being produced due to *Economies of Scale*.

Marginal Cost combines *Fixed Costs* and *Variable Costs*. As the quantity of units being produced increases, average-fixed-costs-per-unit will always decrease but average-variable-costs-per-unit may increase.

$$\text{Marginal Cost} = \frac{\text{Change in Production Costs}}{\text{Change in Quantity Produced}}$$

Definition 1.13 - Marginal Cost Curve

Marginal Cost Curves is a plot of price-per-unit against quantity, showing the marginal cost of producing extra units wrt the total number of units being produced. *Marginal Cost Curves* typically have a U-shape.

See Figure 3.

Proposition 1.2 - Minimum Sale Price

In the short-run, for a producer to stay in business, they need the average sale price of each unit to be greater than their *Average Variable Costs* (AVC).

In the long-run, for a producer to stay in business, they need to cover their fixed costs too. Thus, the average sale price of each unit needs to be greater than the *Average Total Cost* (ATC).

These two metrics provide a minimum sales price for both the short- and long-term

Definition 1.14 - Economies of Scale

Economies of Scale are production cost savings which are obtained due to scaling the quantity of products being produced. These occur where the *Marginal Cost Curve* has a negative gradient (ie marginal cost is decreasing).

Economies of Scale occur, in practice, from maximising the potential of all machines/workers and bulk buying raw materials.

Remark 1.6 - Dis-Economies of Scale

The scale of a business can grow too large and *Dis-Economies of Scale* occur. Practically this

can occur due to increased logistical difficulties. *Dis-Economies of Scale* are occurring when the *Marginal Cost Curve* has a positive gradient (ie marginal costs are increasing).

1.2.3 Price Elasticity

Definition 1.15 - Price Elasticity

Price Elasticity is a measure of how changes in price and quantity demanded/supplied affect each other. This is quantified by the inverse gradient of a supply-curve or a demand-curve. The shallower the gradient (ie flatter the curve) the greater the quantity available will change for a given change in price.

$$PE = \left| \frac{\text{Change in Quantity Demanded/Supplied}}{\text{Change in Price}} \right|$$

There are some special cases:

- *Perfect Price Elasticity* ($PE = \infty$) - A change in quantity has no affect on price. This is when the supply/demand-curve is horizontal.
- *Perfect Price Inelasticity* ($PE = 0$) - The same quantity is demanded/supplied regardless of price. This is when the supply/demand-curve is vertical.
- *Unit Price Elasticity* ($PE = 1$) - A $\Delta\%$ change in price leads to a $\Delta\%$ change in quantity demanded/supplied. This is when the supply/demand-curves are 45° .

Remark 1.7 - Elastic v Inelastic

A product is said to *Price Elastic* if $PE > 1$ and *Price Inelastic* if $PE < 1$.

Remark 1.8 - Price Elasticity - Supply

The *Price Elasticity* of supply quantifies how much the price of a product needs to be increased in order for the producer to be able/willing to meet the new demand.

Proposition 1.3 - Factors Affecting Elasticity of Supply

There are several factors which can the *Price Elasticity* of producers:

- Whether there is any spare production capacity.
- Current stock levels of a product. And thus, how feasible it is to store a product.
- How long it takes to produce the product.
- How easy is it to scale. If specialised workers and tools are required then prices will be more inelastic.

Proposition 1.4 - Factors Affecting Elasticity of Demand

There are several factors which can the *Price Elasticity* of producers:

- Necessity of the product. Luxuries will be more *Price Elastic* than necessities.
- Availability of substitutes.
- Ability to postpone consumption.
- Relative price level (as a proportion of total expenditure).

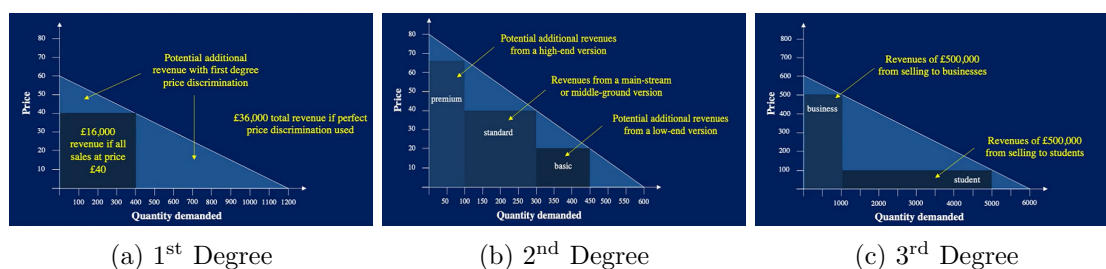


Figure 4: Potential revenue increases for different pricing strategies

1.2.4 Price Discrimination

Definition 1.16 - Price Discrimination

Price Discrimination is the practice of charging different prices to different customers, for the same product. There are three levels of *Price Discrimination*

- 1st Degree (Personalised Pricing) Producers charge each consumer the maximum price they are willing to pay (AKA *Perfect Price Discrimination*).
- 2nd Degree (Versioning) Producers charge different prices for different quantities/qualities of a product. The consumer will pick the version which is most appealing to them.
- 3rd Degree (Group Pricing) Producers charge customers different prices depending on what group the customer falls into. Typically groups are based on age or location. Consumers do not get to pick which group they are a part of.

Remark 1.9 - Utility of Price Discrimination

Price Discrimination helps a producer to increase their revenue (improving their surplus) by setting different price points which capture more of the market. See Figure 1.4 for a plot of how this can be achieved.

Remark 1.10 - Consumer Opinion - Personalised Pricing

Consumers tend to perceive personalised pricing as unfair, and typically like transparency in how prices are set.

Proposition 1.5 - When is Price Discrimination Possible

For a producer to be able to effectively price discriminate then must fulfil the following criteria:

- Able to distinguish between customers in order to know what price to charge them.
- Resale is impractical, otherwise arbitrage is possible.
- Have enough market power to be able to set prices above marginal costs.

Criteria i) & ii) are much more common for online businesses, than IRL businesses.

Proposition 1.6 - Price Discrimination in Practice

Here are a few techniques for implementing price discrimination.

- *Loyalty Schemes* - Offer the customer a discount after buying several items, this is 2nd Degree Price Discrimination. Also making profiling customers easier.

- *Price Steering* - By showing customers extremely cheap/expensive products first you can influence the price customers expect to pay, this is can achieve 1st *Degree Price Discrimination*
- *Bundling* - Selling several goods together to encourage customers to buy more products than they otherwise would. This is 2nd *Degree Price Discrimination*. (e.g. Microsoft Office vs the individual softwares).

Remark 1.11 - *Competing with Yourself*

Versioning carries a risk of competing against yourself as customer who are willing to pay a higher price may choose not to if one of your versions has the specific functionality they are seeking.

Example 1.1 - *Economics of Versioning*

Suppose we have two products: Full version X ; and, a cutdown version X_c . And, suppose there are two customers A & B who value our products as follows

	X	X_c
A	£20	£8
B	£8	£7

We assume that a customer will buy a product if they have a positive *Net Utility* for it, and if they have a positive *Net Utility* for both products then they will buy the one with the greater *Net Utility* to them (or the better version). Consider the following pricing structures

(X, X_c)	A's Net Utility	A's Purchase	B's Net Utility	B's Purchase	Total Income
(£10,-)	(£10,-)	X	(-£2,-)	None	£10
(£10,£5)	(£10,£3)	X	(-£2,£2)	X_c	£15
(£19,£7)	(£1,£1)	X	(-£11,£0)	X_c	£26

The last pricing structure is actual the maximum income we can achieve in this scenario.

1.3 Auction Theory

Definition 1.17 - *Auction*

Auctions facilitate the transactions of goods by enforcing a set of rules on participants. Typically there is a single auctioneer (who ensures the rules are followed), several buyers and one/several sellers. There are several categories of *Auction*

- *Posted Auction* - The seller “posts” a price which bidders have to meet (This is how most shops work).
- *Forward Auction* - The bidders offer prices to the seller with the highest price winning. The rules of the auction define how bids are placed. (See 1.3.1 **Types of Forward Auction**).
- *Reverse Auction* - Several sellers offer prices to a buyer with the lowest price winning. (This is typically used for construct contracts).
- *Double Auction* - All buyers and sellers are able to lists prices and whenever a compatible pair of prices are listed a transaction occurs between those parties. (*Continuous Double Auctions* are used by stock markets).

Remark 1.12 - Utility of Auctions

Auctions allow for products to be sold at a price wholly decided by the consumer's willingness to spend (ie how much they value the product). This can be an effective method for *Price Discovery*

Definition 1.18 - Private Value

A bidder's *Private Value* is the value they place on a product in an auction. Each bidder only knows their *Private Value* and this will typically be the limit of what they are willing to spend on the item.

Private Values are typically imagined as being fixed before the auction. If this is not the case then *Interdependence* occurs as the bidder will change their *Private Value* depending on the actions of others.

Definition 1.19 - Incentive Compatible

A style of auction is *Incentive Compatible* if it encourages bidders to bid their true value (ie *Private Value*). These auctions are popular with auctioneers. (English and SPSP auctions are *Incentive Compatible*).

Theorem 1.1 - Revenue Equivalence Theorem

The *Revenue Equivalence Theorem* states

If *Private Values* are IID and all bidders are risk neutral, then any standard auction yields the same expected revenue for the seller.

In practice this theorem does not hold as bidders are risk adverse and interdependence exists.

Definition 1.20 - Signalling

Signalling is the practice of bidders making bids in “weird” ways, with the intent to signal some information to other bidders. This is a form of collusion (although within the rules of the auction) and is typically used when multiple versions of the same product are on offer, allowing all participants to pay less.

Remark 1.13 - Assumptions in Auction Theory

Auction Theory makes the following assumptions (which limit its practical applications)

- All auctions are equally attractive (ie people will turn up).
- There is no collusion between bidders.

1.3.1 Types of Forward Auction**Open Auctions****Definition 1.21 - Open Auction**

An *Open Auction* is a *Forward Auction* where bids are known to all participants. There are two main types of *Open Auction*: An *English Auction* (As used in *Homes under the Hammer*); and, a *Dutch Auction* (As used in Dutch tulip markets).

Definition 1.22 - English Auction

An *English Auction* (AKA *Open Ascending-Price Auction*) is an *Open Auction* where bidders post bids which are ascending in value until only one bidder is left. That bidder is the highest bidder and thus the winner, paying the value of their final bid.

The typical process of an *English Auction* is

- i). The auctioneer announces an initial (low) price.
- ii). A bidder who is willing to pay this price raises their hand (placing a bid). If no bids are placed then the price is lowered until someone does.
- iii). The auctioneer announces the current lead bidder (i.e. whoever raised their hand first in the last round) and announces a new (higher) price.
- iv). Steps ii)-iii) are repeated until no bids are made at the new price.
- v). The product is sold to the person who made the last bid, at the price of the last bid.

Proposition 1.7 - *English Auction - Optimal Strategy*

Consider a bidder with *Private Value* v . In an *English Auction* the optimal strategy for this bidder is to bid on any price $p \leq v$ (ie Bid upto their *Private Value*).

The optimality of this strategy can be seen by considering the other scenarios

- A bidder should not bid at $p > v$ as they will make a loss.
- A bidder should not stop bidding at a price $p < v$ as they still have surplus demand.

Definition 1.23 - *Dutch Auction*

A *Dutch Auction* (AKA *Open Descending-Price Auction*) is an *Open Auction* where an auctioneer announces prices in a descending fashion and the first bidder to bid wins, paying the price they bid at.

The typical process of a *Dutch Auction* is

- i). The auctioneer announces an initial (high) price.
- ii). The auction announces progressively lower prices.
- iii). Step ii) repeats until a bidder raises their hand (placing a bid).
- iv). The product is sold to the first (thus highest) bidder, at the price the auction was stopped at.

Proposition 1.8 - *Dutch Auction - Optimal Strategy*

Consider a bidder with *Private Value* v . In a *Dutch Auction* the optimal strategy for this bidder is to shade their bid (ie enter at a price $p < v$). To optimally shade their bid the bidder will estimate the private value of all other bidders and enter at a price just above the second greatest private value.

Sealed Auction

Definition 1.24 - *Sealed Auction*

A *Sealed Auction* is a *Forward Auction* where bidding is done in private and each bidder makes a single bid (ie Bidders only know the price they have bidden). There are two main types of *Sealed Auction*: A *First-Price Sealed Bid Auction* (As used in the Scottish housing

market); and, A *Second-Price Sealed Bid Auction*.

Definition 1.25 - First-Price Sealed Bid Auction (FPSB)

A *First-Price Sealed Bid Auction* (FPSB) is a *Sealed Auction* where the winning bidder is whoever bids the greatest price and they pay the price they bid.

The typical process of a *FPSB Auction* is

- i). The auctioneer announces the auction is open and how long it is open for.
- ii). Each bidder makes a single secret bid.
- iii). The auctioneer reads all the bids and announces the highest bidder the winner, paying the price they bid (The *First-Price*)

Proposition 1.9 - FPSB - Optimal Strategy

Consider a bidder with *Private Value* v . In an *FPSB Auction* the optimal strategy for this bidder to shade their bid (ie enter at a price $p < v$). To optimally shade their bid the bidder will estimate the private value of all other bidders and enter at a price just above the second greatest private value.

Definition 1.26 - Second-Price Sealed Bid Auction (SPSB)

A *Second-Price Sealed Bid Auction* (SPSB) is a *Sealed Auction* where the winning bidder is whoever bids the greatest price, but they pay the price of the second highest bid.

The typical process of a *SPSB Auction* is

- i). The auctioneer announces the auction is open and how long it is open for.
- ii). Each bidder makes a single secret bid.
- iii). The auctioneer reads all the bids and announces the highest bidder the winner, paying the price of the second greatest bid (The *Second-Price*).

Proposition 1.10 - SPSB - Optimal Strategy

Consider a bidder with *Private Value* v . In an *SPSB Auction* the optimal strategy for this bidder is to bid a price $p = v$ (ie Bid their *Private Value*).

For the optimality of this strategy see **Proof 1.1**.

Proof 1.1 - SPSB - Optimal Strategy

Consider a bidder with *Private Value* v . Let p be the price our bidder bids at and c be the greatest value of a competitor. We assume our bidder would never bid above their private value. There are two possible bidding strategies

- Bidding our private value $p = v$:
 - Our bidder *wins* if $c < p = v$ for a surplus of $(v - c)$.
 - Our bidder *loses* if $v = p < c$. (Our bidder can never win if $v < c$).
- Shading our bid $p < v$
 - Our bidder *wins* if $c < p < v$ for a surplus of $(v - c)$. (Same as when $p = v$)
 - Our bidder *loses* if $p < v < c$. Our bidder can never win if $v < c$. (Same as $p = v$).

- Out bidder *loses* if $p < v < p$. Out bidder would have won in this scenario if they bid $p = v$ as $c < v$.

These scenarios show that bidding $p < v$ is a suboptimal strategy when compared to $p = v$.

Remark 1.14 - Revenue Generation

For an identical set of bids, an *FPSB Auction* will generate greater revenues than an *SPSB Auction* as the highest price is paid (not the second). Thus, in scenarios where bidders are risk adverse (ie will bid their private value in order to not miss out) an *FPSB Auction* is preferable for the seller.

1.3.2 Comparing Forward Auctions

Remark 1.15 - Robustness

Collusion is much easier in an *Open Auction* as *Signalling* is possible, this means *Open Auctions* are less robust than *Closed Auctions*.

Remark 1.16 - Weak Bidders

It is harder for a weak bidder (ie any bidder without the greatest private value) to win an *English Auction* as stronger bidders can always accept the next stated price.

Definition 1.27 - Strategic Equivalence

Two types of auction are said to be *Strongly Strategically Equivalent* if the same bidding strategy is optimal for both auctions.

Two types of auction are said to be *Weakly Strategically Equivalent* if the same bidding strategy is optimal for both auctions if and only if all *Private Values* are private and *Interdependence* does not occur.

Proposition 1.11 - English & SPSB Auctions are Equivalent

English and *SPSB Auctions* are *Weakly Strategically Equivalent*.

Proposition 1.12 - Dutch & FPSB Auctions are Equivalent

Dutch and *FPSB Auctions* are *Strongly Strategically Equivalent*.

2 Market Economics

Remark 2.1 - Market Economics & Theoretical Computer Science

The field of *Market Based Control Algorithms* utilises *Market Economics* as a useful metaphor for problems in computer science which concern scarce resource allocation.

Definition 2.1 - Bull & Bear Assets

A *Bull Asset* is an asset whose value is expected to increase.

A *Bear Asset* is an asset whose value is expected to decrease.

Proposition 2.1 - Apparent Supply-and-Demand Curves for a CDA Auction

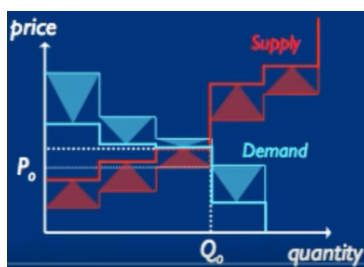


Figure 5: The arrows show the displacement from an agents hidden price and their quoted price

We can plot *Quote Prices* from a CDA Auction to produce “*Apparent Supply-and-Demand Curves*” for the market, but ordering the buy prices in decreasing order and the sell prices in increasing order. In practice these curves are stepped as discrete quantities are offered by each agent.

Any prices to the left of the equilibrium (intersect) are able to exchange in the market or increase their margin, whereas those to the right of the equilibrium will need to decrease their margin in order to make a trade.

Remark 2.2 - No-One Trades their Marginal Price

In practice producers do not want to sell at their minimum price (i.e. marginal cost), nor do consumers want to buy at their maximum price (i.e. private value). To avoid paying these prices agents will quote a price some distance away. This produces “*Apparent Supply-and-Demand Curves*” made up of these *quote prices* (See Figure 1.5).

Some agents will be able to trade at these *quote prices* and those that don't will use them to help inform their next price. The expectation is for the distance between quote and marginal price to decrease over time.

Definition 2.2 - Long Selling

Long Selling is when an agent buys a security with the intention to sell it at some point in the future. You *Long Sell* something you expect to *increase* in value.

Definition 2.3 - Short Selling

Short Selling is when an agent *A* borrows a security from agent *B*, agent *A* then sells the borrow security to another agent *C*. *A* will at some point in the future need to buy an equivalent security in order to reimburse *B*. *A*'s hope is that the price of the security will have decrease in this time.

You *Short Sell* something you expect to *decrease* in price. *Short Selling* can be used as insurance/mitigation against market events.

Definition 2.4 - Arbitrage

Arbitrage is the act of simultaneously buying & selling across markets to make a risk-free profit. This occurs when the price you can buy an asset in one market, is less than the price it can be sold for in another.

Arbitrage opportunities don't last for long as the increased demand for the products which make it up will cause the both their prices to contract together.

Proposition 2.2 - Order Types

There are several different types of orders which can be made in a stock market. Each type

Acronym	Order Type	Description
MO	<i>Market Order</i>	Buy/Sell a defined number of shares at the current market rate.
LIM	<i>Limit Order</i>	Buy/Sell a defined number of shares at a specified limit price, or better.
GFD	<i>Good For Day</i>	A <i>Limit Order</i> which expires at the end of the trading day.
FOK	<i>Fill Or Kill</i>	The order only executes if it can be fully filled immediately.
AON	<i>All Or Nothing</i>	The order only executes if it can be fully filled, but not necessarily immediately.
IOC	<i>Immediate Or Cancel</i>	The order must be executed immediately, but not necessarily in full.
ICE	<i>Iceberg</i>	Execute a large order as a sequence of smaller orders.
LOO	<i>Limit On Open</i>	A <i>Limit Order</i> which is added to the market when it opens.
LOC	<i>Limit On Close</i>	A <i>Limit Order</i> which is added to the market when it closes.
MOO	<i>Market On Open</i>	A <i>Market Order</i> which is added to the market when it opens
MOC	<i>Market On Close</i>	A <i>Market Order</i> which is added to the market when it closes.
OCO	<i>One-Cancels-Other</i>	A pair of orders, if one is fulfilled then the other is cancelled.
OSO	<i>One-Sends-Other</i>	A pair of orders, once the first is executed the second is sent to the market.

Table 1: Selection of common order types available in stock markets.

requires the agent to specify a price and quantity, but then offer further options to specify: how an order is filled (fully or partially); when the order is cancel; or, further orders to send (See Table 1).

Definition 2.5 - Dark Pool

A *Dark Pools* is a marketplace where participants do not want to give away their personal identity. Typically this is because if their identity was known the value of the product would change (e.g. a CEO selling shares in their company).

The identities of participants is only known by the dark pool operator, and there are examples of operators blackmailing/extorting participants.

Remark 2.3 - Dark Pools in Practice - Turquoise LSE

Turquoise is a sub-exchange of the LSE and has two LOBs: *Turquoise Lit*, a standard LOB; and *Turquoise Plato*, a *Dark Pool* where the LOB is not published.

Turquoise Plato has two interesting features

- *Uncross* - Orders are allowed to accumulate on the LOB and at random intervals (5-10 seconds) bids & offers are matched, in a so-called *Uncross Event*.
- *Block Discovery Process* - Agents are allowed to place large, confidential, non-binding orders called *Blocks*, with a specified minimum-size they are willing to accept as a partial fulfilment of their order.

Between *Uncross Events* the owners of matched blocks are asked to confirm their order (An *Order Submission Request*, OSR). The owners are then expected to respond with a firm order (A *Qualifying Block Order*, QBO), which if done means the orders can be completed by the exchange.

Agents can pull out of *Block* orders after receiving an *OSR*, but this will negatively affect their *reputation score* which may lead to them being denied access to the *Block Discovery Process*.

2.1 Market Metrics

Price Description

Definition 2.6 - Bid-Offer Spread

The *Bid-Offer Spread* of a market is the absolute difference in value between the highest bid price and the lowest ask price.

$$\text{B-O Spread} := |\min(P_{\text{ask}}) - \max(P_{\text{bid}})|$$

Definition 2.7 - Mid-Price

The *Mid-Price* of a market is the midpoint between the highest bid price and the lowest ask price (independent of the quantities available at each price).

$$\text{Mid-Price} := \frac{1}{2} (\min(P_{\text{ask}}) + \max(P_{\text{bid}}))$$

The *Mid-Price* is an estimate of the *Market Equilibrium Price*, but single stray prices can make it stray significantly.

Definition 2.8 - Micro-Price

The *Micro-Price* of a market is the weighted mean of the highest bid price and the lowest ask price, each weighted by the quantity available at the other order-type (ie bid by ask, and ask by bid).

$$\text{Micro-Price} := \frac{Q(P_{\max \text{ bid}}) \cdot P_{\min \text{ ask}} + Q(P_{\min \text{ ask}}) \cdot P_{\max \text{ bid}}}{Q(P_{\max \text{ bid}}) + Q(P_{\min \text{ ask}})}$$

where $Q(\cdot)$ is the quantity available at a given price and order-type (bid xor ask).

Definition 2.9 - Price Depth

Price Depth is the number of price levels currently being quoted for a particular product, for a specific order-type (bid xor ask).

Definition 2.10 - Volume Depth

Volume Depth is the total quantity available of a given product, across a specific order-type (bid xor ask).

Market Efficiency**Definition 2.11 - Smith's α**

Smith's α is a metric of the variation of transaction prices $\{P_t\}_{t \in T}$ around the theoretical equilibrium price P_0 . This is normalised to be a percentage.

$$\alpha := \frac{100}{P_0} \sqrt{\frac{1}{|T|} \sum_{t \in T} (P_t - P_0)^2}$$

Lower values of *Smith's α* are better as they indicate a greater concentration of prices

Definition 2.12 - Allocative Efficiency

Allocative Efficiency is a metric of how effective a market is at extracting *Surplus* for its agents A . This is normalised to be a percentage.

$$100 \times \frac{\sum_{a \in A} S_a}{\text{Theoretical Max. Total Surplus}}$$

where S_a is the surplus earned by agent a .

Definition 2.13 - Single Agent Efficiency

Single Agent Efficiency is a metric for how well an individual agent a performs in a market, compared to other agents. (ie how much surplus S_a they receive compared to other agents). This is normalised to be a percentage.

$$100 \times \frac{S_a}{\mathbb{E}[S_A|P_0]}$$

where $\mathbb{E}[S_A|P_0]$ is the expected surplus if all trades occur at price at the theoretical equilibrium P_0 .

Remark 2.4 - Is Intelligence Required for Allocative Efficiency?

Experimental Economics has demonstrated that very simple *Zero-Intelligence Agents* (ZIC), which quote random prices which do not exceed their assigned limit price, trade in a very

human-like manner. Suggesting that most of the intelligence required for a market to be efficient is in the market mechanisms, not the traders.

2.2 Options Markets

Definition 2.14 - *Derivative Contracts*

A *Derivative Contract* is based on the *Marginal Change* in the value of an asset (the difference in price at two points in time), rather than the absolute value of the asset. *Derivative Contracts* do not require you to own the asset.

Here are two types of *Derivative Contracts*

- *Futures Contract* - Have a *Forward Price* and a *Delivery Date*. The transaction must occur on the delivery date, for the forward price (Common in crop yields).
Buyers of a *Future* take a long position (expect the asset to appreciate) while sellers take a *Short* position (expect the asset depreciate).
- *Options Contract* - Have a *Strike Price* and a *Delivery Date*. The transaction may occur on the delivery date, for the strike price (ie there is no obligation).
- There is also *Forwards* and *Swap* contracts.

Remark 2.5 - *Futures & Options are Standardised and Exchange Tradable*

Both *Futures & Options Contracts* are said to be *Standardised* as the price and delivery date of the contracts are prespecified. This means they can be traded on exchanges.

It is important for these contracts to be *Exchange Tradable* so that investors who do not have the cash to exercise the full value of their contract^[1] can still realise the value of it (by selling to someone who can afford to exercise the option).

Remark 2.6 - *Betting Exchanges*

Betting Exchanges are similar to *Options Markets* as they allow people with differing views about some future outcome to wager on it. Similarly, the winner makes a profit but the loser loses their whole investment (Whereas, in a stock-market, the loser can recoup some of their investment by selling the asset.).

2.2.1 Options

Proposition 2.3 - *Pricing an Option*

There are a few factors of an *Options Contract*:

- *Intrinsic Value* - The amount of money exchange if the option is exercised.
- *Volatility Premium* - The volatility in the value of the underlying asset.
- *Time Value* - How long until the exercise date

^[1]i.e. Buy the amount of the underlying asset they have the right to. In this case your only cost is the commission for buying the contract, and not in actually exercising the contract. You receive (part of) the profit from realising the contract.

Remark 2.7 - Commissions

Commissions are paid to set up an *Options Contract*. Commission grows linearly with the number of contracts. You need to make a certain amount of profit on the contract in order to cover all costs.

Remark 2.8 - American & European Style Options Contracts

There are two styles of *Options Contracts* which affect how the contract can be exercised.

- *American Style* - The contract can be exercised on any day upto the *Delivery Date*.
- *European Style* - The contract can be exercised only on the *Delivery Date*.

Definition 2.15 - Gearing

Buy an options contract is much cheaper than buying the underlying asset, this means you can buy more options contracts and thus increase the amount of profit/loss you can make for the same investment. This is known as *Gearing*, and is inherently more risky than buying the underlying asset as if the price does not go in the direction you desire you loss your entire investment.

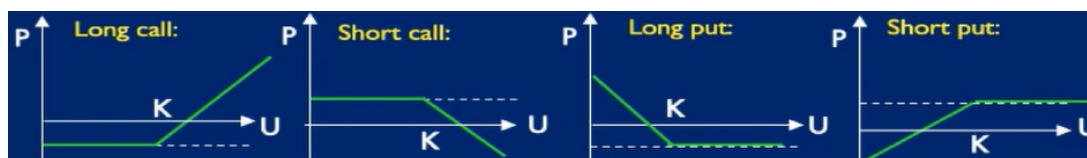


Figure 6: Payoff diagrams for the four basic options contracts. K is the *Strike Price*, the green line plots the payoff to the agent P depending on the underlying value of the asset U on the delivery date.

Proposition 2.4 - Types of Options Contract

There are two types of *Options Contract*, depending on whether you want the right to buy or to sell.

- *Call Option* - Gives the holder the right to buy a set amount of the underlying asset, on some future date, at a specified *Strike Price*. The holder is in the money if the value of the underlying asset is greater than the *Strike Price* on the delivery date.
- *Put Option* - Gives the holder the right to sell a set amount of the underlying asset, on some future date, at a specified *Strike Price*. The holder is in the money if the value of the underlying asset is less than the *Strike Price* on the delivery date.

Each of these contracts has two positions an agent can take

- *Long Position* - The holder of the option (ie the agent with the right to buy/sell at the agreed price).
- *Short Position* - The writer of the option (ie the agent who must buy from/sell to the holder, if the holder wishes to exercise the option).

This gives four different options contract positions an investor can enter into: Long Call; Short Call; Long Put; or, Short Put. Each of these have different pay offs depending on the change in value of an asset. See Figure 6.

Remark 2.9 - Cash Flow Short vs Long Position

Agents taking *Long Positions* pay a commission to the agent taking the *Short Position*. This commission is the only way for the *Short Position* agent to make money, whereas the *Long Position* agent makes money depending on the value of the asset on the delivery date.

The commission is the maximum amount the *Long Position* agent can lose, but the maximum amount the *Short Position* agent can make.

Due to the commission the initial cash-flow is negative for *Long Positions* and positive for *Short Positions* (as seen in Figure 6).

2.2.2 Option Strategies

Definition 2.16 - Option Strategies

Option Strategies are linear combinations of Long Calls; Short Calls; Long Puts and Short Puts. *Option Strategies* are used to limit potential losses/profits, or to allow an agent greater flexibility in the position they are taking.

Below is a table of a selection of *Option Strategies* (For figures see 0.2 Options Strategies)

Name	Description	Constituent Contracts	Fig
Bull Spread	Bounded profit and loss for a forward position.	i) Long call at K_1 ii) Short call at K_2	12
Bear Spread	Bounded profit and loss for a short position.	i) Short put at K_1 ii) Long call at K_2	13
Long Butterfly	Profit if underlying value does <u>not</u> move, bounded loss otherwise.	i) Long call at K_1 ii) 2x Short calls at K_2 iii) Long call at K_3	14
Short Butterfly	Bounded profit if underlying value moves, in either direction.	i) Short call at K_1 ii) 2x Long calls at K_2 iii) Short call at K_3	15
Long Straddle	Unbounded profit if underlying value changes, in either direction	i) Long call at K_1 ii) Long put at K_1	16
Short Straddle	Profit if underlying value does <u>not</u> move, unbounded loss otherwise.	i) Short call at K_1 ii) Short put at K_1	17
Long Strangle	Unbounded profit if the underlying value moves more than some specified amount, in either direction.	i) Long put at K_1 ii) Long call at K_2	18
Short Strangle	Profit if the underlying value does <u>not</u> move more than some specified amount, otherwise unbounded loss.	i) Short put at K_1 ii) Short call at K_2	19
Ratio Call Spread	Profit if the underlying value does <u>not</u> move. Loss bounded if underlying value decreases but, unbounded if value increases.	i) Short call at K_1 ii) 2x Long calls at K_2	20
Ratio Put Spread	Profit if the underlying value does <u>not</u> move. Loss unbounded if underlying value decreases but, bounded if value increases/	i) 2x Short Puts at K_1 ii) Long Put at K_2	21
Ratio Call Backspread	Profit if the underlying value moves: Bounded profit if underlying value decreases; Unbounded profit if underlying value increases.	i) Short call at K_1 ii) 2x Long Calls at K_2	22
Ratio Put Backspread	Profit if the underlying value moves: Unbounded profit if underlying value decreases; Bounded profit if underlying value increases.	i) Short Put at K_1 ii) 2x Long Put at K_2	23

Remark 2.10 - Combining Option Contracts**2.3 Trading Algorithms****Definition 2.17 - Automated Trading Algorithms**

Automated Trading Algorithms are algorithms which trade on behalf of an agent in a market. These algorithms are given a limit price and are tasked to maximise their operators surplus. The algorithms have access to the prices offered by other agents in the market, but not their private-value/limit-price.

Here are a selection of *Trading Algorithms*, which represent a timeline progression of these algorithms

- *Zero Intelligence (ZI)* - Offer prices at random, ignoring private-value/limit-price. (Useless).
- *ZI-Constrained (ZIC)* - Offer prices at random but never offer prices which would result in a negative surplus for the operator.
- *ZI-Plus (ZIP)* - Use machine learning to adapt how far away the prices it offers are from its assigned limit price, given actions in the market.
- *Kaplan's Sniper* - Waits until sufficient information is known about the market before offering a price. The aim is for this price to “snipe” other prices late into the trading window.
- *Gjerstad-Dickhaut (GD)*
- *Modified GD (MGD)*
- *Extended GD (GDX)*
- *Adaptive-Aggressive (AA)*

Definition 2.18 - ZIP Traders

The *ZI-Plus* algorithm uses machine learning to adapt how far away the prices it offers are from its assigned limit price, given actions in the market.

Let L be the limit price of the agent, $p(t)$ be the price offered by the agent at time t and $q(t)$ be the last quoted price in the market (ie from any agent). $\mu(t)$ is a parameter we seek to learn which models the profit margin the agent tries to make. We define the following learning rules

$$\begin{array}{ll}
 \Delta(t) & := \beta(T(q(t)) - p(t)) & \text{Step towards target} \\
 \Gamma(0) & := 0 & \text{Initial Momentum} \\
 \Gamma(t+1) & := \gamma\Gamma(t) + (1 - \gamma)\Delta(t) & \text{Build momentum towards target} \\
 \mu(t+1) & := \frac{1}{L}(p(t) + \Gamma(t)) - 1 & \text{New margin} \\
 p(t) & := L(1 + \mu(t)) & \text{New price}
 \end{array}$$

Here β is a learning rate parameter, γ is a damping parameter and $T(\cdot)$ is a stochastic function of the last quoted market price $q(t)$. ($T(\cdot)$ is the “target value” which aims to be slightly better than the last quoted price).

Definition 2.19 - Kaplan's Sniper

Kaplan's Sniper trading algorithm waits until one of the following conditions is met before offering a price (and only offers a price if the achieved margin is above a defined threshold).

- i). The *Bid-Offer Spread* is sufficiently small.
- ii). Offers are better than the best transaction prices from the previous trading period.
- iii). There is only a short time left in the trading period.

This algorithm is based around not giving information away and hopes that the price offered will “snipe” all existing offers (and other traders will not have time to react).

Remark 2.11 - Kaplan’s Sniper & the Market

Kaplan’s Sniper does not adapt to market activity (ie its threshold do not change) and does not help the market achieve an equilibrium.

Kaplan’s Sniper relies on other traders being active in order to be effective. A market full of snipers would be very inefficient.

Definition 2.20 - GD Traders

The *Gjerstad-Dickhaut* (GD) trading algorithm produces an estimate of the probability $f(p)$ of a given price p being accepted and then quotes the price p_{quote} which maximises the agent’s expected gain.

$$p_{\text{quote}} := \operatorname{argmax}_p p \cdot f(p; H)$$

The belief function for price p , given a history of quote prices H , is defined as

$$f(p; H) = \frac{\operatorname{acc}(p; H) + \operatorname{off}(p; H)}{\operatorname{acc}(p; H) + \operatorname{off}(p; H) + \operatorname{rej}(p; H)}$$

where $\operatorname{acc}(p; H)$ is the number of bids made below price p which were accepted during H ; $\operatorname{off}(p; H)$ is the number of offers made below price p during H ; and, $\operatorname{rej}(p; H)$ is the number of bids made above price p which were rejected during H .

Remark 2.12 - GD Belief Function in Practice

The belief function $f(p; H)$ used by *GD Traders* is a stepped function, but, in practice, a cubic-spline interpolation is used to smooth the function.

Definition 2.21 - MGD Traders

The *Modified GD* (MGD) trading algorithm modifies the belief function $f(p; H)$ st it shows zero probability for bids lower than the previous lowest trade price and for offers greater than the previous greatest trade price. This reduces volatility

$$\begin{aligned} f_{\text{bids}}(p; H) &= \begin{cases} 0 & \text{if } p < \min(H_{\text{acc}}) \\ \frac{\operatorname{acc}(p; H) + \operatorname{off}(p; H)}{\operatorname{acc}(p; H) + \operatorname{off}(p; H) + \operatorname{rej}(p; H)} & \text{otherwise} \end{cases} \\ f_{\text{offer}}(p; H) &= \begin{cases} 0 & \text{if } p > \max(H_{\text{acc}}) \\ \frac{\operatorname{acc}(p; H) + \operatorname{off}(p; H)}{\operatorname{acc}(p; H) + \operatorname{off}(p; H) + \operatorname{rej}(p; H)} & \text{otherwise} \end{cases} \end{aligned}$$

where H_{acc} is the set of prices where trades were successful made in H .

Definition 2.22 - GDX Traders

The *GDX* trading algorithm is an extension of the *GD* algorithm which introduces *real-time dynamic programming* to formulate bidding strategies in auctions which are characterised by sequential bidding and continuous clearing (e.g. CDA).

The *Dynamic Programming* considers the “state” of the agent (their current holdings) and calculates transition probabilities to other states using the market event history H and an algorithm similar to the belief algorithm $f(p; H)$ used in *GD*.

Definition 2.23 - Adaptive-Aggressive Traders

The *Adaptive-Aggressive* (AA) trading algorithm uses an *Aggressiveness Parameter* which represents an agent’s willingness to trade-off profit for actually transacting (more aggressive prices are more likely to be accepted). The underlying *Aggressiveness Function* is updated using *Smith’s* α (A estimate of market volatility). In a more volatile market, a small change in aggressiveness leader to a greater change in bidding behaviour.

The weighted moving average of previously accepted trades is used to estimate the current market equilibrium price \hat{P}_0 and *Smith’s* α . The estimate \hat{P}_0 is used to determine whether the current quoted price $p(t)$ is intra-marginal (ie aggressive) or extra-marginal (ie not aggressive).

Remark 2.13 - Algorithms vs Humans

Remark 2.14 - Comparing Trading Algorithms

Performing a full *Analytical Analysis* of the capabilities of each trading algorithm (e.g. using game-theory) is either so difficult that it may as well be impossible or require too many simplifications for the findings to be relevant.

Thus, an *Empirical Analysis* (e.g. simulation experiments) are much more appropriate. There are several simulations which can be inciteful for assessing an algorithms ability to make a market efficient

- *Market Homogeneous* - All agents use the same algorithm. Varying the supply and demand dynamics can be interesting.
- *One-in-many* - Where one traders uses algorithm A while all other traders use algorithm B . This assesses vulnerabilities in a homogeneous market of B and the ability of A to exploit such a market.
- *Balanced-Groups* - An equal proportion of traders use each algorithm (with the same distribution of limit prices for each group for fairness).

Remark 2.15 - Market Homogeneity

For most applications, including financial markets, *Market Homogeneity* is implausible. There are some scenarios, such as *Market-Based Resource Allocation* (e.g. resources in a network) where it is reasonable to assume that all traders use the same algorithm due to the centralised nature of these scenarios.

	<i>Market Homogeneous</i>
ZIC	Slow but did converge
Sniper	No convergence
ZIP	Converged quickly
MGD	Converged quickly

Table 2: Description of how each algorithm performed in a homogeneous market.

Proposition 2.5 - Trading Algorithm Performance

Tables 2, 3, 4 provide summaries of how the trading algorithms performed in the tests described in Remark 2.9.

	Many ZIC	Many Sniper	Many ZIP	Many GD	Many MGD
One ZIC	-	-.238	-.124	-.005	-.105
One Sniper	.011	-	.011	.068	.018
One ZIP	.091	.117	-	.028	-.022
One GD	.130	-.039	-.290	-	-
One MGD	.158	.235	-.162	-	-

Table 3: The average improvement of introducing a single agent with a different algorithm to a homogenous market

(Wins)	ZIC	ZIP	Sniper	GD	MGD	(Efficiency)	ZIC	ZIP	Sniper	GD	MGD
ZIC	-	0-100	100-0	1-99	1-99	ZIC	-	.955	.964	.978	.960
ZIP	100-0	-	99-1	63-36	29-71	ZIP	.955	-	.982	.996	.997
Sniper	0-100	1-99	-	7-93	2-98	Sniper	.964	.982	-	.978	.993
GD	99-1	36-63	93-7	-	-	GD	.978	.996	.978	-	-
MGD	99-1	71-29	98-2	-	-	MGD	.960	.997	.993	-	-

Table 4: Results of a balanced-group simulation

Table 3 shows that introducing a *ZIC* trader always reduces market efficiency; Introducing a *Sniper* always improves market efficiency; Introducing a *ZIP* trader improves market efficiency unless the rest of the market use *MGD*; And, similarly, introducing a *MGD* trader improves market efficiency unless the rest of the market use *ZIP*.

Table 4 shows that *MGD* is the superior algorithm in a *Balanced Group* simulation. It has been shown that *AA* traders outperforms *GDX* & *ZIP* in *Balanced Group* simulations, but is not dominant (ie there are some ratio of groups where *AA* doesn't win. See Figure 7).

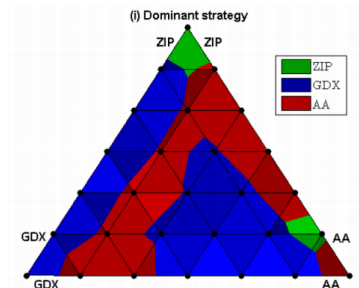


Figure 7: Which trading algorithm wins for different mixtures of the population. None is dominant!

2.4 Prediction Markets

Definition 2.24 - Prediction Markets

Prediction Markets are an exchange-traded market which attempt to draw on the *Wisdom-of-crowds* to determine the outcome of a future events. Agents in the market are able to trade “coupons” which will payout if a particular outcome occurs (e.g. a particular party wins an election). As this is a free market and agents are only motivated by finances (not political preference), the outcomes the crowd deems more likely should eventually be priced higher.

The payout of these “coupons”

- *Winner-Takes-All* - A fixed value is payed if a particular event occurs.
- *Vote-Share* - The amount payed out depends on a quantitative outcome (e.g. Number of seats won in an election).

The total price of all the “coupons” should equal the total payout.

3 Economics of the Internet

Definition 3.1 - Network Externality

A *Network Externality* are non-monetary effects which occur to a consumer due to other consumers using compatible products. Products with positive network externalities are known as *Network Goods*.

Phones have *Network Externalities*. The more people that own a phone the more people you can call with yours. Conversely, if no-one else owns a phone then yours is effectively useless.

Remark 3.1 - Network Externalities - Positive Feedback Loop

Positive Network Externalities can produce a *Positive Feedback Loop* where consumers choose to buy products which are compatible with their friends, rather than necessarily the best product (AKA *Brand Value*). This increases the size of the network and thus the magnitude of the *Positive Network Externalities*.

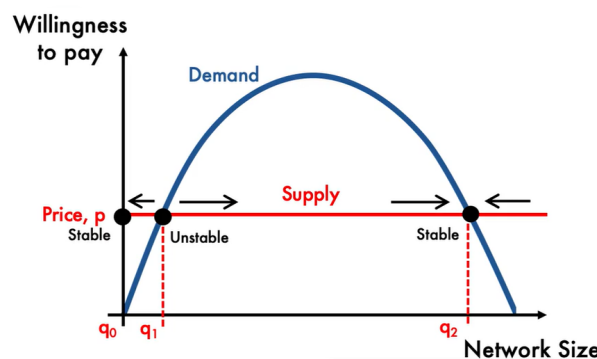


Figure 8: Network Effect Demand Curve.

Definition 3.2 - Network Effect Demand Curve

A *Network Effect Demand Curve* shows the price consumers are willing to pay for a product given the size of the network using that product (See Figure 8). A *Network Effect Demand Curve* has an upside-down U shape.

The gradient of the *Network Effect Demand Curve* is initially positive as the *Marginal Value* of each extra user is higher, but this diminishes as the network grows.

Proposition 3.1 - Equilibria Points of a Network Effect Demand Curve

For any price P , there are three network sizes which produce equilibria q_0, q_1, q_2 on the *Network Effect Demand Curve* (See Figure 8). q_0 is a network of size 0 and thus is naturally stable.

q_1 is an unstable-equilibria, known as the *Critical Mass*. If the network is larger than q_1 then it will grow to size q_2 , but if it is smaller than q_1 it will shrink to q_0 (ie non-existent).

Definition 3.3 - Disruptive vs Sustaining Innovation

There are two types of innovation:

- *Sustaining Innovations* - Incremental improvements of traditional performance metrics for existing products.

- *Disruptive Innovations* - Vast improvement of *non*-traditional performance metrics which generate new markets & products (Typically underperform on traditional performance metrics).

Definition 3.4 - Combinatorial Innovation

Combinatorial Innovation occurs when multiple innovations can be combined & recombined to create new innovations. (The Internet is a combinatorial innovation due to its standardised & open-source nature).

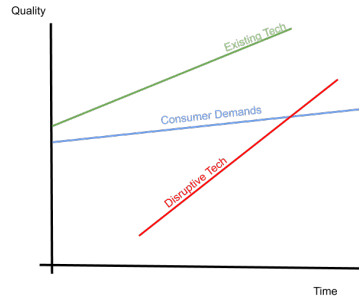


Figure 9: Timeline of how a disruptive technology can establish itself in a market.

Remark 3.2 - Disruptive Technology - Established Producers

Establish companies are typically slow to invest in *Disruptive Technologies* as these new technologies do not meet the needs of the company's clients.

Often, by the time the disruptive technology does reach the requirements of these clients, there is already an establish company in the new market and the original company will struggle to compete. The product of the original company may still be of greater quality than the new technology, but it severely underperforms in the non-traditional performance metrics customers now care about (See Figure 9).

Remark 3.3 - Timeline of Disruptive Technology

Here is the typical timeline of a successful *Disruptive Techonology* (As illustrated in Figure 9).

- i). *Disruptive Technology* is invented (Often by an established company).
- ii). The *Disruptive Technology* does not meet the needs of the establish company's clients, and thus the company does not focus on the new technology.
- iii). New companies form to pursue the *Disruptive Technology* (Often by ex-employees of the established company).
- iv). The *Disruptive Technology* sufficiently improves along traditional performance metrics and becomes main-stream/commercially successful.
- v). The establish companies tries to enter the new market but is too late.

Definition 3.5 - Switching Costs

Switching Costs are costs a customer incurs when switching services, both monetary and non-monetary costs. When *Switching Costs* are too high a customer becomes *Locked-In* as they

will choose to stay with an inferior service than incur the switching costs of moving to a better service.

Potential *Switching Costs* include:

- *Setup Costs*.
- *Training Costs* - Having to learn a new service.
- *Network Effects* - Losing positive network externalities of the existing service.
- *Initial Loss in Quality* - The new service will not know you as well as the existing service so may offer inferior services initially (e.g. Show recommendations on Netflix).

3.1 Digital Goods

Proposition 3.2 - *Digital vs Physical Goods*

There are several common economic differences between digital and physical goods

- *Costs*: Digital goods are expensive to produce, but cheap to reproduce. (i.e. High fixed costs, low variable costs).
- *Capacity Constraints*: There is no limit to the number of times a digital good be reproduced.
- *Sunk Costs*: Investment into digital goods are sunk costs. (i.e. You can recoup costs by selling unneeded shop-buildings, but you cannot recoup the cost of software developer time).
- *Ascertaining Value*: Digital goods are more likely to be *Experience Goods*, meaning customers cannot know whether they will like the good before purchasing it. (ie customers cannot assign a value).
- *Search Costs*: It is much easier for a customer to search for/compare competing digital products online, than competing physical goods offline.
- *Network Externalities*: Digital goods are more likely to have strong positive network externalities.
- *Switching Costs*: Digital goods are more likely to have greater switching costs, and try to lock-in customers.

Proposition 3.3 - *Cost Curve for Digital Goods*

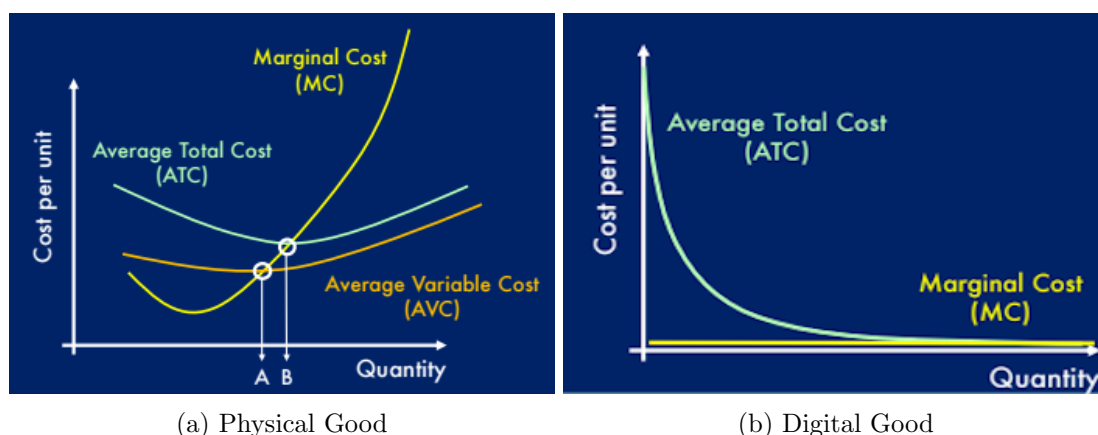
Digital goods typically have high fixed costs and low variable costs, meaning the *Marginal Cost* of digital goods is effectively zero and average costs tend to zero. The *Cost Curve* for digital and physical goods are very different (See **Figure 10**).

Remark 3.4 - *Minimum Price of Digital Goods*

As *Digital Goods* have effectively zero *Marginal Cost* producers can justify selling their product for nothing.

They will need to recoup the initial fixed costs at some point in the future. Many businesses give away their product for free until they reach critical mass, at which point they attempt to monetise it.

Remark 3.5 - *Competition between Producers of Digital Goods*



(a) Physical Good

(b) Digital Good

Figure 10: Typical cost curve for physical and digital goods.

Due to very low marginal costs, when there is competition in a market producers of similar digital goods quickly drop their price to zero. This makes it hard for new companies to enter, due to high fixed costs.

Companies will then compete on the quality of their product and try to build positive network externalities, with lock-in.

Definition 3.6 - Freemium Business Model

The *Freemium Business Model* offers a free version of a digital good and then encourages customers to upgrade to the premium version.

Definition 3.7 - File Formats

File Formats are a standardised ways for encoding digital information. There are often multiple formats for the same type of object (e.g. JPEG v SVG v PNG).

Remark 3.6 - Proprietary Formats

Some companies choose to encoding their digital information with *Proprietary Formats* (ie formats only their software knows how to read). This increases the switching costs for customers, but reduces the shareability of the information. The reduced shareability is good for some information (e.g. copyrighted-music) but not for others (e.g. word documents).

There are examples (e.g. MS Word's .DOC) of *Proprietary Formats* being reverse engineered by other companies/groups in order to piggy back the network externalities of these products.

Remark 3.7 - Industry-Wide Standards

Sometimes *Industry-Wide Standards* are agreed, allowing customers to share files between providers. This increases the network effect, likely attracting new customers.

The trade-off for producers is between having a large part of a small pie; or a small part of a large pie.

Proposition 3.4 - How Industry-Wide Standards Develop

There are three common ways for *Industry-Wide Standards* to develop

- i). A single, major, producer sets a standard by opening up their proprietary format(e.g Adobe with PDF).
- ii). A “war” occurs between multiple producer wishing to set the standard (Generally to the

deteriment of all).

- iii). Several producers negotiate a standard. There is always a risk that during this process one party will pull out and use their own proprietary format.

3.2 Online vs Offline Businesses

Remark 3.8 - *Laws of Economics are Invariant*

The *Laws of Economics* (e.g supply-and-demand) are fundamentally the same for online & offline businesses. It is the characteristics of online business activities which result in different markets.

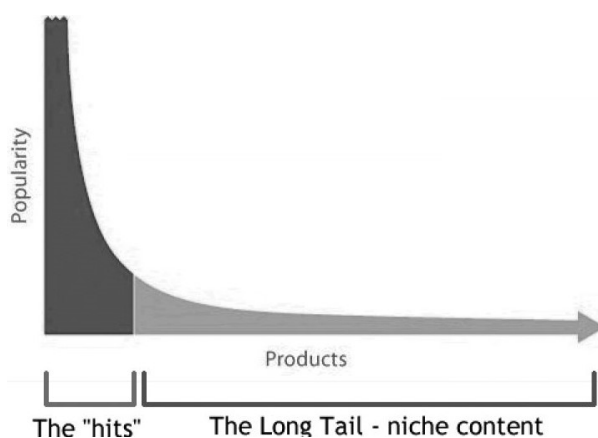


Figure 11: Illustration of how popularity dictates which products are sold.

Proposition 3.5 - *The Long Tail*

The variety of products an offline business can sell is limited by the shelf space they have. This means that only the best selling products are stocked and sold by the business.

However, online business either have much more space for physical goods (due to using warehouses) or unlimited space if selling digital goods. This means online business can stock a greater variety of products. Moreover, online businesses selling digital goods should offer any product which makes at least one sale as it costs them (almost) nothing to do so (See Figure 11).

Proposition 3.6 - *Utilising the Long Tail*

There are a few steps an online business can take to utilise the idea of the long tail

- i). Make it easy for customers to find new products (ie good recommender systems).
- ii). Offer everything, someone will buy it.
- iii). Reduce prices, as economies-of-scale reduce costs.

3.3 Online Price Discrimination Strategies

Remark 3.9 - *Personalised Pricing*

Personalised Pricing is, in-theory, much easier online as producers have more information about their customers. However, it is not believed that any major companies use personalised pricing due to potential public backlash. Rather, they choose to use this information to promote items/sales they believe a customer is more likely to be interested in.

European data protection laws require producers to specify, to their customers, how any data collected is used. This means a company would have to state they use the data for personalised pricing (likely leading to backlash and people trying to game the system).

Remark 3.10 - *Digital Product Versioning*

As digital goods have near-zero reproduction costs and software can easily be customised (by blocking access to certain features), *Product Versioning* is common with digital goods.

Bundling digital goods together is easy to setup and at very low cost to the producer.

Proposition 3.7 - *Unbundling*

Due to it being easy for a customer to purchase digital goods, customers are willing to make lots of small purchases which they may not for physical goods (i.e. you wouldn't and buy penny sweets every day). This can lead to the practice of *Unbundling* where producers sell the components, of what is typically larger product, individually (e.g. articles of a magazine).

3.4 Economics of Computer Games

Remark 3.11 - *In Game vs IRL Currencies*

In-Game Currencies are in-essence very similar to IRL currencies, as they are stimulated by players collecting raw materials, producing goods and then trading (Especially in multi-player games).

This means they share similar problems around stability (ie balancing money entering and leaving the system). If this balance is not good then *MUDflation* occurs (ie inflation). It is important for in-game currencies to be stable otherwise the game's longevity will be shortened.

Remark 3.12 - *Meta-Currencies*

In cases where in-game currencies have proven unstable, there are examples of player bases developing *Meta-Currencies* where they decide an alternative unit of trade (e.g. cigarettes/fish in prisons).

Remark 3.13 - *IRL Value*

For many popular games there are IRL markets where players sell their in-game assets for Fiat-currencies (This is essentially an FX market).

Gold Farming is a practice of players grinding tasks in a game in order to acquire assets they intend to sell in IRL markets. This can prove a profitable endeavour, especially in countries with lower wages. *Gold Farming* typically breaks the EULA of a game but it is hard for game makers to counter.

Remark 3.14 - *Game Development - Budgets*

The budget and size of teams involved in creating AAA titles has grown exponentially (Tracking the powerfulness of computers). However, the *average* game budget has decreased as it is easier for individuals to create passion-project games (due to better tools being available).

Interestingly, the mid-size studio (with budes in low millions) have almost disappeared due to them either being absorbed by larger firms or splintering into smaller firms.

Remark 3.15 - *Game Development - Employee Specialisation*

As the desired quality of games has increased, employees have become increasing specialised.

This means individual tasks now take longer and more people are required to create a high quality game. This increases the production costs.

3.5 Online Auctions

Remark 3.16 - *Online Auctions*

Due to well-defined rules of auctions and the fact there is no need to have a physical auctioneer, it is easy to write algorithms to perform auctions digitally. Moreover, the internet enables auctions to be run quickly and cheaply with the only limit being server capacity and bandwidth.

It is possible for bidders to automate their bidding process too. This has led to more complex auctions being introduced.

Remark 3.17 - *Intermediaries*

Although the internet makes it viable for small businesses to hold their own auctions, most choose to use an intermediary marketplace (e.g. Amazon & eBay) due to the greater exposure these intermediaries offer.

These marketplaces also make payments easier and improve trustworthiness of both buyers and sellers.

Definition 3.8 - *Snipping*

Snipping is the practice of placing bids close to the expiry time of the auction, in the hope that other buyers do not have enough time to outbid you. This means that information about a bidder's *Private Value* is kept secret for as long as possible.

Remark 3.18 - *Estimating Demand Curves for Online Auctions*

In online auctions it is easier to track attempted bids^[2] you can plot an estimate for the demand curve. This estimate can be used to derive optimal sale strategies and for estimating revenue.

Remark 3.19 - *eBay*

eBay is an online auctioneer which specialises in consumer-to-consumer and producer-to-consumer auctions (and some business-to-business auctions). *eBay* uses an *Open-Ascending Auction with a Deadline*, this similar to an *English Auction* but has a predefined expiry date.

Reviews of sellers act to *Lock-In* sellers and thus attract buyers, this is in part how *eBay* has an effective monopoly.

Remark 3.20 - *Snipping on eBay*

Snipping is/was a common practice on *eBay* with specialist software being written to help enable it. To combat this *eBay* introduced a system of “*proxy bidding*” where bidders state the maximum they are willing to bid and the system places the minimum bid required for them to be winning the auction at any point in time (provided this is below the specified maximum). If all bidders use this system then an *eBay* auction becomes an *SPSB Auction* with a time limit.

Remark 3.21 - *Google Ad Auctions*

Google uses auctions for its *Ad-Space* (multiple auctions per user-search results page). In order

^[2]Some bids may not be accepted due to the price being too high when that bidder enters the market.

to maintain a good user experience, *Google* considers both *bid price* and *advert quality* (assessed by historic click-through rate, relevancy and landing page quality) when choosing the winner of an auction.

Originally *Google* used *FPSB Auctions* but this lead to server overload as bidders checked the results of auctions (to see if they could lower their pricing). Switching to *SPSB Auctions* solved this issue due to *SPSB Auctions* being incentive compatible.

3.6 The Crowd Economy

Definition 3.9 - Crowd Economy

The *Crowd Economy* is the group of platforms which rely on active online participation from large crowds of people, in order to provide their services (e.g. *Wikipedia* & *AirBnB*).

Definition 3.10 - Crowdsourcing

Crowdsourcing is the practice of using crowds to complete some task (e.g. image labelling, writing articles). *Crowdsourcing* is often used for tasks which are difficult for a computer to do.

Proposition 3.8 - Finding Participants for Crowdsourcing

Here are some practices which business can use to encourage participation

- *Pay Them* - Typically only a few pence, as on *Mechanical Turk*.
- *Trick Them* - Require them to complete the task to complete something they actually want to (e.g. *Amazon's Mechanical Turk*).
- *Motivate Them* - Make the task fun or meaningful (e.g. for medical research).

Definition 3.11 - Crowdfunding

Crowdfunding is the practice of raising funds for a project (e.g. producing a project) from crowds of people. This can take several forms

- *Donation Based* - Funders give money for no reward, except for good feels.
- *Reward Based* - Give funders rewards based on how much they pledge.
- *Equity Based* - Funders become share holders.
- *Debt Based* - Accept funding from many funders with the agreement to repay with interest (effectively a cheap loan), this is AKA *Peer-to-Peer Funding*.

Remark 3.22 - Crowdfunding Regulation

In the UK *Equity Based* and *Debt Based Crowdfunding* are regulated by the FCA, in the same way traditional services are.

4 Financial Technology

Remark 4.1 - The Underbanked

The *Underbanked* are the $\sim 70\%$ of the world, typically outside the western world, do not have access to basic banking facilities. These people are prime customers for *FinTechs*, as these customers desire easy-access and simplicity, over quality products.

More people have access to mobile phones than banking services, so mobile based banking services are particularly attractive. This is why *FinTechs* often focus on mobile-centric services.

4.1 FinTech Firms

Proposition 4.1 - Key Players

There are three key types of company which are involved in *Financial Technology*

- *Traditional Financial Services* - Focus on *Incremental Innovation* to improve their traditional business functions and productivity (through *Digitisation*).
- *Start Ups* (FinTechs) - Focus on *Disruptive Innovation* in order to create new financial services markets for them to enter.
- *Big Tech* (TechFins) - Seek to leverage their existing technological dominance to offer financial services to existing customers. They are able to do this due to their positive network externalities and likely high *Lock-In*.

Remark 4.2 - Regulators vs FinTechs

The goal of regulators is to: Protect participants; Improve market competitiveness; And, build trust.

New financial technologies may require new regulation. Some national regulators are more welcoming to new technologies, taking a “*Watch and See*” approach where they allow new technologies to enter the market and then work with the business to implement regulations (Known as *Sandboxing*).

Remark 4.3 - China

China uses a *development-led* approach to new technologies, allowing new technologies to develop first and adding regulation later (once it is clear it is needed).

Remark 4.4 - India

India uses a *development-lagged* approach to new technologies, heavily restricting the ability of new technologies to enter the market before they are well understood.

Definition 4.1 - LASIC Principles

The *LASIC Principles* are a set of principles, suggested by *Less & Teo*, a *FinTech* should follow in order to succeed:

- *Low Margin* - Selling services with a *Low Profit Margin* allows the business to service, whilst attracting as many customers as possible.
- *Asset Light* - Fewer assets mean lower *Fixed Costs*. Piggy-backing on existing infrastructure (e.g. Mobile Phones) will help.
- *Scalable* - Having a business model and technologies which are able to scale, without affecting efficiency or cost, helps build *Positive Network Effects*.
- *Innovative* - Innovative technologies encourage customers to adopt your services.
- *Compliance Easy* - Not being restricted by regulators helps a business be scalable.

	High Banking Prevalence	Low Banking Prevalence
High Venture Capital Investment	<u>Partnering</u> (e.g. Western Europe) Competition is likely to come from outside traditional banking. It is Possible for a <i>FinTech</i> to thrive but, it is easier to do so by partnering with an establish bank.	<u>Tech Dominance</u> (e.g. China) Ideal for <i>FinTechs</i> as there is little established competition.
Low Venture Capital Investment	<u>Bank Dominant</u> (e.g. Eastern Europe) Most competition occurs between established business, and not form new <i>FinTechs</i> .	<u>Race-to-Finish</u> (e.g. Sub-Saharan Africa) Telecom companies often the most significant tech-firms in these countries. Meaning mobile phones are the key technology to gaining a foothold in these markets.

Table 5: Four classes of national banking maturity.

Proposition 4.2 - National Banking Maturity

The viability of a new financial technology varies by country, depending on the maturity of its banking infrastructure. There are four main categories countries can fall into (See Table 5).

4.2 Blockchain**Definition 4.2 - Blockchain**

A *Blockchain* is a list of records (“*blocks*”) where each record contains a hash of the previous record, linking records together in a chain.

Remark 4.5 - Storing a Blockchain

A *Blockchain* can be stored in a single location, but it is more common for multiple parties to each store different parts of the blockchain (with multiple copies of each part). These parts can be compared to verify the integrity of the *Blockchain*, and as they are distributed it is harder for the *Blockchain* to be changed retrospectively

Remark 4.6 - Changing a Blockchain

Blockchains are designed to be resistant to modification. Modifying a block will change it’s hash and thus all latter blocks would need to be changed as well. This is effectively impossible if there are multiple copies of the blockchain (or parts of it) stored by different parties, as all copies need to be updated and it is unlikely all parties will agree to this.

Proposition 4.3 - Forks in a Blockchain

A *Fork* occurs in a *Blockchain* if there are two versions currently being circulated. There are defined procedures for what to do in this scenario. In general a longer chain is deemed to be correct.

Suppose block Z is the globally agreed end of the chain and suppose two blocks A, B are successfully added onto the blockchain at the same time, by different miners. This means two chains (1) & (2) are being propagated around the network.

$$\begin{aligned} (1) \quad & \cdots \rightarrow Z \rightarrow A \\ (2) \quad & \cdots \rightarrow Z \rightarrow B \end{aligned}$$

Suppose a node which currently believes chain (1) to be correct, successfully mins block C onto

the chain (3) and propagates it out.

$$(3) \quad \cdots \rightarrow Z \rightarrow A \rightarrow C$$

This will cause a fork for any nodes which currently believe chain (2) to be correct. For these nodes, then will drop their current chain (2) and accept (3) to be the correct chain as it is longer. Eventually, chain (3) will be the globally accepted truth.

Definition 4.3 - Bitcoin

Bitcoin is a decentralised-cryptocurrency which uses a *Distributed Blockchain* to record transactions.

Remark 4.7 - R3

R3 is a consortium of banks which are developing a *Blockchain* technology for recording inter-bank transactions. This has been criticised as *Blockchain* technology is generally much slower and more expensive than existing technology.

4.2.1 Road to Bitcoin

Remark 4.8 - Early Internet Payments

In the early life of the Internet, secure HTTP protocols did not exist. This meant good encryption could not be used, making sending payment details very risky.

Remark 4.9 - Road to Bitcoin

Here are a few technologies which were part of the learning process which lead to Bitcoin.

- i). *Credit-Based Payment Architectures* - An intermediary facilitates payment between the seller and the buyer.
- ii). *Cash-Based Payment Systems* - Customers buy “tokens” with fiat currencies. These “tokens” are then exchanged with sellers, who can then use the “tokens” for fiat currency.
- iii). *HashCash* - Users had to complete a cryptographic puzzle in order to send an email. This was to discourage spam emails.
- iv). *Ledgers for Timestamping* - A trusted server stores copies of documents along with a timestamp for when the document was received, in order to keep a trusted record.

Definition 4.4 - Credit-Based Payment Architectures

Credit-Based Payment Architectures are based around an intermediary who facilitates a payment between the seller and the buyer. Typically the intermediary securely stores the payment details of both parties, thus not requiring the parties to share their payment details with each other. When a payment request from one/both parties is received, the intermediary debits the buyer and subsequently credits the seller (Typically after some dispute window).

Here are two popular early implementations *Credit-Based Payment Architectures*

- *FirstVirtual* - A predecessor to *PayPal*. Both buyers and sellers had to be registered with *FirstVirtual*.

To reduce fraud *FirstVirtual* allowed buyers a 90 day dispute window and would not credit sellers until after this window, for this reason *FirstVirtual* was not popular with sellers.

- *SET Architecture Standard* - A system developed by *Visa & MasterCard*. Buyers did not have to enrol with the system, rather they install an application on their device which would store their details and when a purchase was made the app would encrypt these details and send to *Visa* or *MasterCard*.

Buyers were required to certify all transactions which was a complicated and time-consuming process, making *SET* unpopular with buyers.

Definition 4.5 - Cash-Based Payment System

Cash-Based Payment Systems allow users to exchange fiat-currencies for digital “tokens” and visa-versa. Secure mechanisms are then provided for exchanging the “tokens”, and even if these systems are compromised user payment details would not be at risk.

The value of “tokens” are typically pegged to a fiat-currency (ie they are not a free floating currency. See **Remark 4.9**).

- *DigiCash* was a *Cash-Based Payments System* which used *eCash* coins. Owners chose a unique private ID for each of their tokens, this could be as complicated as they liked, and stating this ID was proof of ownership. A centralised databased ensured no double spending.

DigiCash failed as sellers were not anonymous (while buyers were), and it did not allow buyer-to-buyer transactions.

Remark 4.10 - Free-Floating Currency

A currency is *Free-Floating* if its value is based on its supply-and-demand relative to other currencies, rather than being explicitly being defined in terms of another currency (like most in-game currencies).

For a currency to be *Free-Floating* it must have some scarcity. For a digital currency, scarcity will have to form part of the design of its creation process. Typically this is done by making the minting new money computational expensive (e.g. having to complete cryptographic puzzles).

Definition 4.6 - HashCash

HashCash was a *proof-of-work protocol* designed to reduce spam emails. Before a user could send an email *HashCash* made them complete a puzzle and then the puzzle + solution would be sent with the email, the recipient’s inbox would then check the solution against the puzzle and if the solution was incorrect the email would be deleted.

Each puzzle was unique and independent as they depended upon the sender, receiver, email content and time. The puzzles were designed to be hard to solve (~ 1 min), but quick to verify. Puzzles were made harder as hardware improved.

HashCash disappeared due to improved spam filters.

Definition 4.7 - Ledgers for Timestamping

Ledgers for Timestamping is the practice of having a central, trusted service which documents could be sent to: A record would be kept of when that version of the document was received; and, a certificate issued to the sender. This prevents timestamps being changed retrospectively.

Haber & Stornetta introduced this idea in 1991. For efficiency they suggest grouping documents into blocks and then linking blocks into a tree or chain.

4.2.2 Bitcoin

Proposition 4.4 - *Innovations of Bitcoin*

The first generation of *Bitcoin* innovated on the technologies discussed in Section 4.2.1 to produce the following features

- Simple to store and transact coins (An adaption of *Credit-Based Payment Architectures*).
- User-to-user transactions and anonymity for all (An adaption of *Cash-Based Payment Systems*).
- A decentralised record of transactions (An adaptation of *Ledgers for Timestamping*).
- A puzzle-based mining system to regulate currency creation (An adaptation of *HashCash*).

Features

Definition 4.8 - *Blocks*

A *Block* of the *Bitcoin Blockchain* contains a header and a list of the details (parties, price, time & miner's fee) of $\sim 4,000$ transactions. The header of each *Block* contains the following

- *Timestamp* - Time *Block* was mined.
- *Version Number* - Version of *Bitcoin* being used.
- *Merkle Root* - Summary of the new transactions in the block.
- *Nonce Value* - Specified by the miner such that when the header is hashed used SHA256² the resulting 256-bit has at least $x^{[3]}$ leading 0s (This is the miner's *proof-of-work*).

Definition 4.9 - *Bitcoin Mining*

Mining is a *proof-of-work protocol* used to add *Blocks* to the *Bitcoin Blockchain* in a consistent & complete manner. This requires the miner to find a *Nonce Value* which when hashed with the header of the block produces a 256-bit string with at least $x^{[3][0]}$ leading 0s (This is a laborious process).

The found *Nonce Value* is added to the header of the *Block* so that other miners can verify its correctness and the work has been completed.

It is common for multiple miners to be working on the same block

Remark 4.11 - *Bitcoin Rewards*

Each transaction in the a *Block* specifies a reward (in Bitcoins). The miner who successful adds the *Block* to the *Bitcoin Blockchain* receives all the rewards specified in the block. Users can specify greater rewards in order to encourage their block to be mined faster.

Additionally, a fixed reward is paid for the completion of each block. This is how new *Bitcoins* are minted (ie enter circulation) and thus this reward is periodically halved in order to limit the introduction of the coins.

Remark 4.12 - *Bitcoin Mining Practices*

Miners typically use GPUs to perform mining operations as their many small-cores are typically faster than traditional CPUs.

^[3] x is the target difficulty, specified by the network. Smaller x values are easier to solve for.

Field Programmable Gate Arrays (FPGAs) have been popular as it is easy to customise their configuration to achieve quicker speeds. However, they are significantly more expensive than GPUs so the cost-per-performance gains are negligible.

Definition 4.10 - Bitcoin Scripts

Bitcoin Scripts is the scripting language used for performing *Bitcoin* transactions. *Bitcoin Scripts* have native support for basic arithmetic, logic & cryptographic operations (for hashing and signature verification), but not for loops.^[4]

There is a limit to the number of instructions in a *Bitcoin Script* to prevent infinite loops, which would trap miners.

Proposition 4.5 - T-of-N MULTISIG

T-of-N MULTISIG is an example of a “*smart-contract*” operation support by *Bitcoin Scripts*. It requires *T-of-N* signatures to agree for something to be verified.

It is common to use a 2-of-3 MULTISIG as only the buyer and seller need to sign for a transaction to be verified, but if there is a dispute a single third-party can decide the outcome.

Proposition 4.6 - Bitcoin Transaction Process

The following are the steps required for two parties to complete a *Bitcoin* transaction

- i). *Agree Transaction* - Two parties agree on a transaction and a fee to be paid to the miner.
- ii). *Prepare Transaction Message* - The buyer creates a message specifying the following details about the transaction
 - The addresses of the buyer and seller.
 - The reference(s) of the transaction from which the buyer received the *Bitcoins* they are about to spend.
 - Number of *Bitcoins* to be paid to each party.^[5]
- iii). *Sign Transaction Message* - The buyer encrypts the transaction message using their private key.
- iv). *Broadcast Transaction Message* - The buyer broadcasts the encrypted transaction message & their public key to their immediate peers in the network. These peers verify the message (ie $\text{inputs} \geq \text{outputs}$, etc.) and then broadcasts to their peers.^[6]
- v). *Verify Transaction Message* - Miners mine the transaction into a *Block* and compete to add the new *Block* onto the *Blockchain*.

Comments

Remark 4.13 - A Libertarian Dream

Some have deemed *Libertarian Dream* due to it being decentralised and anonymous (No Tax? Hype!). This is unlikely to be fully realised as the majority of mining is done by a few professional companies.

Remark 4.14 - Tracking Bitcoin Payments

^[4] *Bitcoin Scripts* is not Turing Complete.

^[5] change is returned to the buyer

^[6] The transaction is not on the blockchain yet.

In practice *Bitcoin* payments are only pseudo-anonymous as the *Wallet ID* of each party of a transaction is stored in the *Blockchain*. This means that if you know someone's *Wallet ID* you can see and track all their transactions (although you may not be able to tell who they are to/from).

Remark 4.15 - Who is Mining?

Most mining is done professionally, leading to a centralisation of mining power. And, around 70% of all mining occurs in China.

Remark 4.16 - Shortcomings of Bitcoin

There are several shortcomings of *Bitcoin*, inc.:^[7]

- *Lossing Access* - If a user forgets/loses their *Private Keys* they will completely lose access to their *Bitcoin* wallet. This is equivalent to losing a physical wallet or forgetting a credit-card's PIN.
- *Bitcoin Scripts* - *Bitcoin Scripts* is not Turing complete, which limits the possible contracts people can write.
- *Not Truly Anonymous* - *Bitcoin* transactions are only pseud-anonymous (See Remark 4.13).
- *Technology Attacks* - The network is susceptible to *DDoS* & *Sybil Attacks* (Or an *EMP Bomb*).
- *Energy Consumption* - A lot of electricity is used for *Bitcoin* mining globally (Approx. same electricity consumption as Switzerland).
- *Future* - As the system is decentralised it relies on agreement within the community in order to innovate/adapt, this is not guaranteed.

Remark 4.17 - Updating Bitcoin Protocols

Updating *Bitcoin* protocol requires a hard-fork of the *Blockchain*, where the vast majority of nodes in the network agree to a change. This is unlikely to happen!

4.2.3 Innovating on Bitcoin

Remark 4.18 - New Cryptocurrencies

Here are three newer cryptocurrencies: *Ethereum*, *Cardano*; and, *HyperLedger Fabric*. All of which seek to innovate on *Bitcoin* in some way.

Ethereum

Proposition 4.7 - Ethereum

Ethereum introduces the *Ethereum Virtual Machine* (*Ethereum VM*) which extends *Bitcoin Scripts* to be Turing complete and allows for code to be written/executed directly on the *Blockchain*. This allows for

- *Smart Contracts* - Code which can be executed in the *Blockchain*.

^[7]Many of these are shared with traditional physical currency.

- *Decentralised Applications (DApps)* - A collection of *Smart Contracts* with a front-end web interface which users can interact with. A user's information is stored locally, giving them greater control over their data.^[8]

Remark 4.19 - Gas

To stop people writing computationally intensive programs onto the *Ethereum Blockchain*, users are required to pay “Gas”, in *Ethereum*. “Gas” costs increases with the number of instructions executed and some instructions cost more than others. “Gas” is payed to whomever executes the script (similar to mining rewards in *Bitcoin*).

Users specify a *Gas Limit* which causes the program to halt when reached, preventing infinite loops.

Cardano

Proposition 4.8 - Cardano

Cardano seeks to reduce the work required to keep it's *Blockchain* consistent and secure. This is done by introducing the following

- *Multi-Layered Approach* - Transactions & coins are separated from code & contracts. This helps scalability.
- *Proof-of-Stake Protocol* - To decide which miner gets to add the next block onto the *Blockchain*, *Cardano* uses a *proof-of-stake protocol* which is much more computationally-efficient than *Bitcoin's proof-of-work protocol* (See Proposition 4.9).
- *Private Networks* - Permissions are introduced, creating private networks.

Proposition 4.9 - Proof-of-Stake Protocol

Cardano's proof-of-stake protocol decides which miner m will add the next block onto the *Cardano Blockchain* by doing the following.

- Calculate the total number of tokens in circulation, and the number of tokens each miner currently owns, by reading the existing *Blockchain*.
- Choose the next miner at random, where the probability of a given miner being picked is equal to their share of the total stake

$$\mathbb{P}(\text{Next} = M) = \frac{\# \text{ M's Tokens}}{\# \text{ Total Tokens}}$$

Remark 4.20 - Ouroboros Protocol

The *Ouroboros Protocol* is the specific *proof-of-stake protocol* used by *Cardano*. It ensures miners are chosen randomly by perform a secure multi-party coin-flipping protocol.

HyperLedger Fabric

Definition 4.11 - HyperLedger Fabric

HyperLedger Fabric seeks to make *Blockchain* technology more suitable for businesses by introducing a *plug-and-play* architecture which offers versatility and modularity. Some features include: consensus, privacy, membership services.

^[8] *DApps* typically raise money by selling tokens for their services.

4.3 Sentiment Analysis

4.3.1 Theory

Definition 4.12 - *Sentiment*

Sentiment is a view/opinion that has been expressed about something (generally how much they like/dislike a product). It is easy for someone to write a review online and some review sites allow for users to label their sentiment numerically using a star-rating system.

Remark 4.21 - *Yelp Reviews*

For restaurants on *Yelp*, a one-star increase in average rating leads to an $\sim 5 - 9\%$ increase in revenue. Although this is typically only seen in independent restaurants, not chains.

Over the last few years the market share of independent restaurants has increased, likely due to *Yelp* helping independent restaurants penetrate the market.

Definition 4.13 - *Sentiment Analysis*

Sentiment Analysis is the process of computationally identifying the sentiment of a piece of text. Typically *Sentiment* is measured using a *Polarity Value* of $[-1, 1]$, but often is generalised to either: Like (1); Dislike (-1); or, Neutral (0).

There are three levels of *Sentiment Analysis* which can be done to a piece of text

- i). *Document-Level Analysis* - Identify the overall sentiment of the whole document.
- ii). *Sentence-Level Analysis* - Identify the sentiment of a given sentence.
- iii). *Aspect-Level Analysis* - Identify the opinions for each topic of a document.

Remark 4.22 - *Utility of Sentiment Analysis*

Sentiment Analysis allows for the opinion piece of text to be nicely summarised without the writer having to specify their opinion (e.g. with a star rating). This data can then be analysed to judge public/expert opinion on a product and thus make more informed business decisions about the future of said product.

Proposition 4.10 - *Collecting Public Opinion*

There are two main approaches to collect data about public opinion

- *Traditional* - Focus groups, customer surveys & opinion polls. These are expensive and slow processes.
- *Sentiment Analysis* - Computationally derive opinions from online postings. This is hard and relies on a representative spectrum of users making public posts.

Definition 4.14 - *Topic Modelling*

Topic Modelling is the practice of computationally identifying the topics mention in a piece of text.

Remark 4.23 - *Sentiment Analysis vs Topic Modelling*

Topic Modelling is typically much easier than *Sentiment Analysis* as identifying key-words will also identify most key-topics. *Sentiment Analysis* can be paired with *Topic Modelling* to classify the sentiment of a writer towards each topic in the text, individually.

4.3.2 Algorithms

Remark 4.24 - *Supervised Machine Learning for Sentiment Analysis*

Many review sites allow users to numerically label their reviews with a “star rating”, this provides labelled data which can be used for supervised machine learning. Typically we are more interested in the *classification-version* of the problem so the numerical star ratings are translated into three classes: Negative (0-2); Neutral (3-7); And, Positive (8-10).

Proposition 4.11 - *Document-Level Sentiment Analysis - Supervised ML*

“Bag-of-Words” is a popular SML approach to *Document-Level Sentiment Analysis*. In the “Bag-of-Words” approach the frequency of words in each sentence is considered (ignoring context) and then traditional ML techniques (e.g. Naïve-Bayes & SVM) are applied to these frequencies to learn a model which fits the data.

Proposition 4.12 - *Document-Level Sentiment Analysis - Unsupervised ML*

One UML approach to *Document-Level Sentiment Analysis* is to consider the relationship between words (ie how often pairs of words co-occur). Classification is then done by considering the

Here is an example of the flow of this process

- i). Extract all phrases P which are made up of a pair of words: An adjective or an adverb, followed by a noun. (Adjective=Sentiment, Noun=Context).
- ii). For all possible pairs of words in the document calculate the *Pointwise Mutual Information* (PMI). This is a measure of how strongly the pair of words is semantically-associated.

$$PMI(w_1, w_2) := \log_2 \left(\frac{\mathbb{P}(w_1 \text{ then } w_2)}{\mathbb{P}(w_1) \cdot \mathbb{P}(w_2)} \right)$$

- iii). For each extract phrase in P calculate the *Semantic Orientation* (SO) of the phrase wrt a pair of reference words: one positive, one negative (e.g. “excellent”, “poor”).^[9]

$$SO(p) := PMI(p, \text{“excellent”}) - PMI(p, \text{“poor”})$$

- iv). Calculate the average *Semantic Orientation* of the document. If this average is positive then the document should be classified as positive (and visa-versa).

Remark 4.25 - *Aspect-Level Sentiment Analysis*

It is often hard to determine which object a phrase is referring to, especially when a single phrase refers to multiple topics. The problem of *Aspect-Level Sentiment Analysis* can be decomposed into identifying the following

- i). *Aspect Extraction* - What object is being discussed? (e.g. “iPhone”) Moreover, what aspect of the object is being discussed? (e.g. price)
- ii). *Groups of Aspects* - Identify aspects of an object which can be considered synonyms of each other (e.g. “battery life” & “battery power”).
- iii). *Aspect Sentiment Classification* - What is the sentiment being expressed about each group of aspects? (See **Proposition 4.8**).

^[9]*Semantic Orientation* can be estimated by querying a search engine with each phrase and noting the number of hits where the phrase is within a certain number of words of the reference word.

- iv). *Extract Opinion Holder & Time* - Who's sentiment is this and when was it expressed? (Metadata can make this a lot easier).

Proposition 4.13 - *Aspect Sentiment Classification Algorithm*

Here is a simple algorithm for identifying the sentiment being expressed about a particular aspect of an object

- i). Use a dictionary to assign sentiment values (+1 for positive, -1 for negative) to words and phrases which express sentiment.
- ii). Identify any words which shift sentiment (e.g. "not", "never") and flip the sentiment values of the words/phrases which are shifted.
- iii). Identify any "but" and if the sentiment of one side of the "but" cannot be identified then assume it to be the opposite of the other side.
- iv). Sum the sentiment values, weighting words wrt their distance from the aspect word.
- v). If the sum value is positive then assume a positive sentiment is being expressed (and visa-versa).

4.3.3 Practice

Remark 4.26 - *Twitter*

There have been successful demonstrations of using *Sentiment Analysis* on public tweets to predict the outcomes of events. Including:

- *Box-Office Revenue* - From tweets in the week prior to the movie's release.
- *Election Results* (Although there have been failures to replicate).
- *Stock Market Movements* - By just evaluating the "mood" of tweets, not their context. (Again, there have been failures to replicate).

Basically, correlation between sentiment and an outcome does not imply causation.

Remark 4.27 - *Brushing*

Brushing is a scam used by *Amazon* sellers to make fake reviews look more legit and to artificially inflate a products position in *Amazon's* ranking algorithm. The general process is

- i). The seller "gifts" one of their products to a previous customer. This will show in Amazon's system as the customer purchasing the product.
- ii). The seller will leave a review "on behalf of" the customer (likely highly positive).
- iii). This review will be listed as being from a "Verified Purchaser", giving greater credence to it.

Remark 4.28 - *Gaming The System*

As it is easy for anyone to post online, and it is easy to masquerade as someone you are not online, there are many examples of people trying to game the system by posting fake reviews about products they have a vested interest in, in order to encourage/discourage sales.

Definition 4.15 - *Sockpuppetry*

A *Sockpuppet* is a fake online identity used with the intent to deceive. Typically, the *Sockpuppet* will pose as someone independent of the actual operator and will then act positively impact public opinion about the operator (e.g. positive reviews & defending in forums).

Remark 4.29 - *Sockpuppetry - Crowd Opinion*

Generally an operator will have multiple *Sockpuppet* in order to utilise *Crowd Psychology*, and in the hope that real people will join their fake-crowd's momentum.

Proposition 4.14 - *Sockpuppetry Practices*

Here are some common *Sockpuppetry* practices

- *Ballot Stuffing* - Casting multiple votes in of the operator.
- *A Sybil Attack* - Multiple sockpuppets are used to create influence within a peer-to-peer online network (e.g. Facebook Groups), subverting any reputation systems.
- *Stealth Marketing* - Multiple sockpuppets mention positive aspects of the operators product in general discussions, rather than just reviews of the product.
- *Starman Sockpuppet* - A sockpuppet makes a strawman argument against the operator, making it easy for the operator to rebuff the argument and make it appear foolish.
- *Astroturfing* - Multiple sockpuppets are used to make it appear a message is coming from a grassroots movement, rather than an established entity.

Definition 4.16 - *Persona Management Software*

Persona Management Software is software which has been created with the intent to help operators manage their sockpuppet network. Some features include

- Create diverse & plausible online personas, each with a unique static IP address.
- Randomly selecting different sockpuppets for different tasks.
- Blending sockpuppet traffic in with real traffic to create cover/plausible-deniability.

Proposition 4.15 - *Sockpuppet Detection*

Detecting sockpuppets generally requires creating a network and somehow identifying that these accounts are cluster in some unnatural way. Common features to look out for are

- Similar IP addresses.
- Similar naming patterns.
- Similar/short registration times.
- Similar/unnatural login patterns.

0 Reference

Definition 0.1 - Widrow-Hoff Learning Rule

The *Widrow-Hoff Learning Rule* adjusts the actual output at time t $A(t)$ wrt some desired output at time t $D(t)$, with rate parameter β and damping factor $\gamma \in [0, 1]$.

$$\begin{aligned}\Delta(t) &:= \beta(D(t) - A(t)) \\ A(t+1) &:= \gamma A(t) + (1 - \gamma)\Delta(t) \\ &= \gamma A(t) + \beta(1 - \gamma)(D(t) - A(t))\end{aligned}$$

NOTE the desired value $D(\cdot)$ varies over time, but how this happens is defined separately (on a problem-by-problem basis).

0.1 Empirical Methods

Remark 0.1 - When to use a t -Test

A T -Test can only be used when it is reasonable to assume the underlying distribution is a normal distribution.^[10]

Proposition 0.1 - A - B Test

An A - B Test determines the best of two methods: A, B . The general approach is

- i). Calculate a confidence interval of the mean for samples from A .
- ii). Calculate a confidence interval of the mean for samples from B .
- iii). If the intervals overlap then there is insufficient evidence to deem on method better than the other. *Otherwise*, the method with the greater predicted mean is deemed better.

Definition 0.2 - Wilcoxon-Mann-Whitney U Test

The *Wilcoxon-Mann-Whitney U Test* is a test of whether two samples are from the same population, or not. It is a *non-parametric* version of the t -Test. This test makes some assumptions (See **Proposition 0.4**).

Let A be a sample of size n_A and B be a sample of n_B , with $n_A > n_B$.

$$\begin{aligned}H_0: & A \text{ and } B \text{ come from the same population.} \\ H_1: & A \text{ and } B \text{ come from different populations}\end{aligned}$$

- i). Combine A and B into one list and sort it (keeping track of which data point is from which sample).
- ii). Rank the list from lowest to highest. Awarding average rank when ties occur.
- iii). Sum the ranks for A and B to get R_A, R_B .
- iv). Compute $U_B := n_A n_B + \frac{1}{2} n_A (n_A + 1) - R_A$ and $U_A := n_A n_B + \frac{1}{2} n_B (n_B + 1) - R_B$.
- v). Define $U := \min\{U_A, U_B\}$.
- vi). Compare U to the critical value from $U_{n_A, n_B, \alpha}$ from a lookup table.
- vii). If $U < U_{n_A, n_B, \alpha}$ then reject H_0 .

^[10]There are normality tests to verify this.

Proposition 0.2 - *Assumptions of U Test*

- The Independent variable data is in two groups (Control and Treatment).
- The dependent variable is at least in an ordinal scale.
- Data is randomly selected samples from the two groups
- The population distributions of the dependent variable for the two groups have a similar shape, but have differences in measures of central tendency.

Remark 0.2 - *Chaining U Tests*

The *U Test* is a pairwise test, but it is bad practice to use a series of *U Tests* to test more than 2 samples as the error of the test grows quickly.^[11]

Definition 0.3 - *Kruskal-Wallis H Test*

The *Kruskal-Wallis H Test* is a generalisation of the *U Test*, extending it to allow multiple group comparisons.

Consider samples A, B, C, D

- Calculate the mean ranking for each group's entries in the sorted union list (As in *U Test*).
- Compute a statistic H from the mean rank and the sample size n .

$$H := \frac{12}{n(n+1)} \sum_{g \in \{A, B, C, D\}} n_g \left(\bar{r}_g - \frac{N+1}{2} \right)^2$$

- Use a χ^2 -test to determine whether H is significant.

^[11]The probability of four tests at a 95% significance level all being correct is $.95^4 = .814 \dots \ll .95$

0.2 Option Strategies

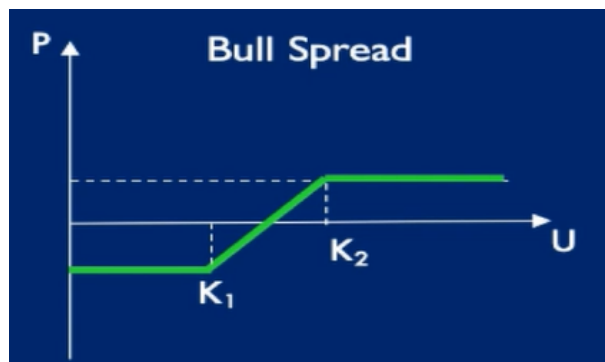


Figure 12: Pay-Off Diagram for *Bull Spread*.

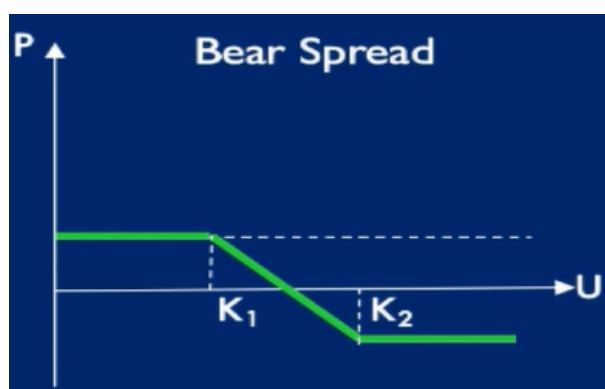


Figure 13: Pay-Off Diagram for *Bear Spread*.

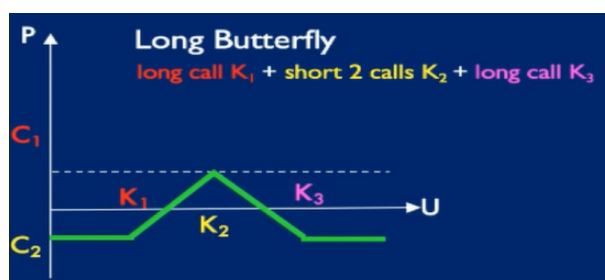


Figure 14: Pay-Off Diagram for *Long Butterfly*.

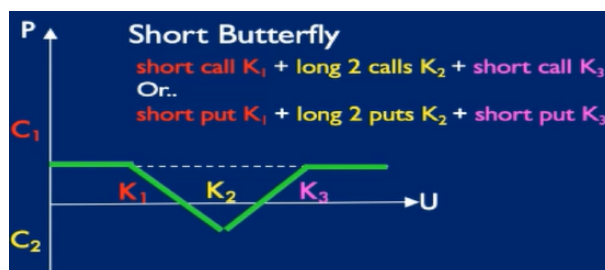
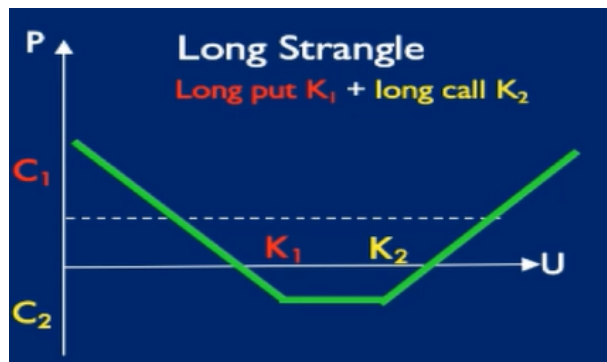
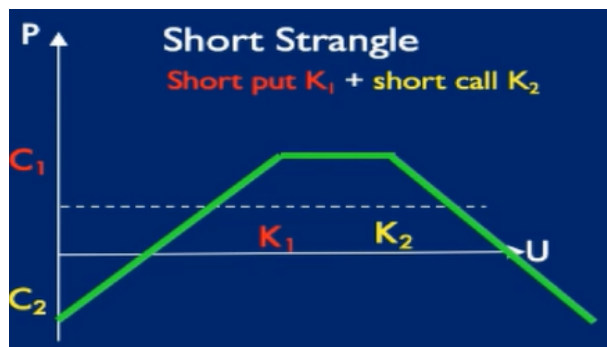
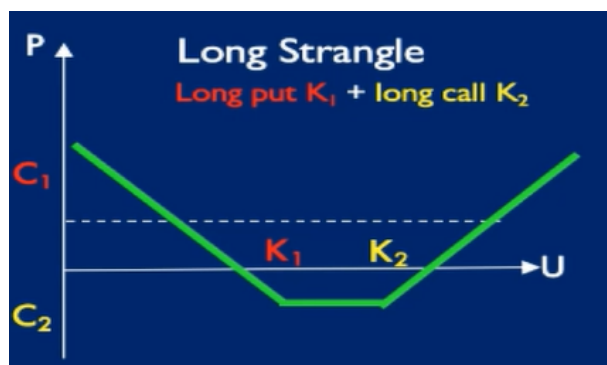
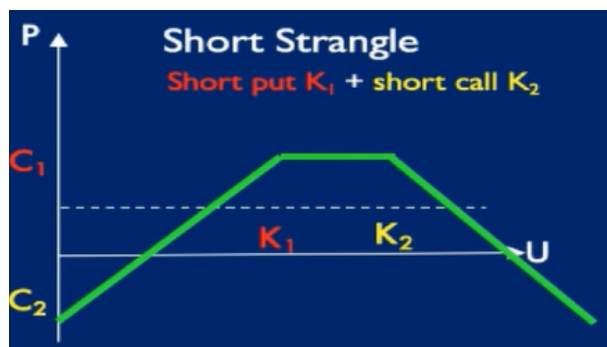
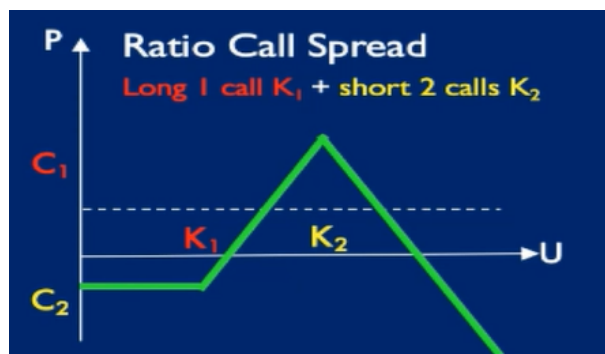
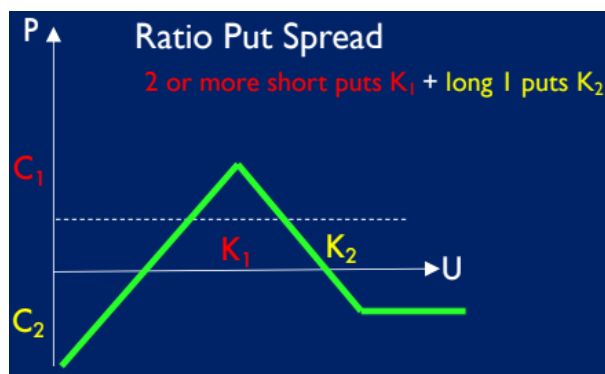
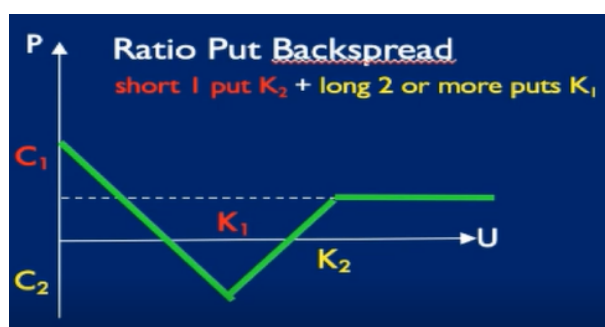


Figure 15: Pay-Off Diagram for *Short Butterfly*.

Figure 16: Pay-Off Diagram for *Long Straddle*.Figure 17: Pay-Off Diagram for *Short Straddle*.Figure 18: Pay-Off Diagram for *Long Strangle*.Figure 19: Pay-Off Diagram for *Short Strangle*.

Figure 20: Pay-Off Diagram for *Ratio Call Spread*.Figure 21: Pay-Off Diagram for *Ratio Put Spread*.Figure 22: Pay-Off Diagram for *Ratio Call Backspread*.Figure 23: Pay-Off Diagram for *Ratio Put Backspread*.