Modeling and Analyzing IVR Systems, as a Special Case of Self-services

Nitzan Carmeli

MSc Advisors: Prof. Haya Kaspi, Prof. Avishai Mandelbaum

Empirical Support: Dr. Galit Yom-Tov

Industrial Engineering and Management
Technion

23.06.2014

Introduction

- **Interactive Voice Response (Voice Response Unit)**

"If you're losing patience with our endless automated system and need to run outside and scream, press 44. If you're feeling better now and wish to continue, press 45..."

Introduction

- Agents salaries are typically 60%-80% of the overall operating budget in call centers

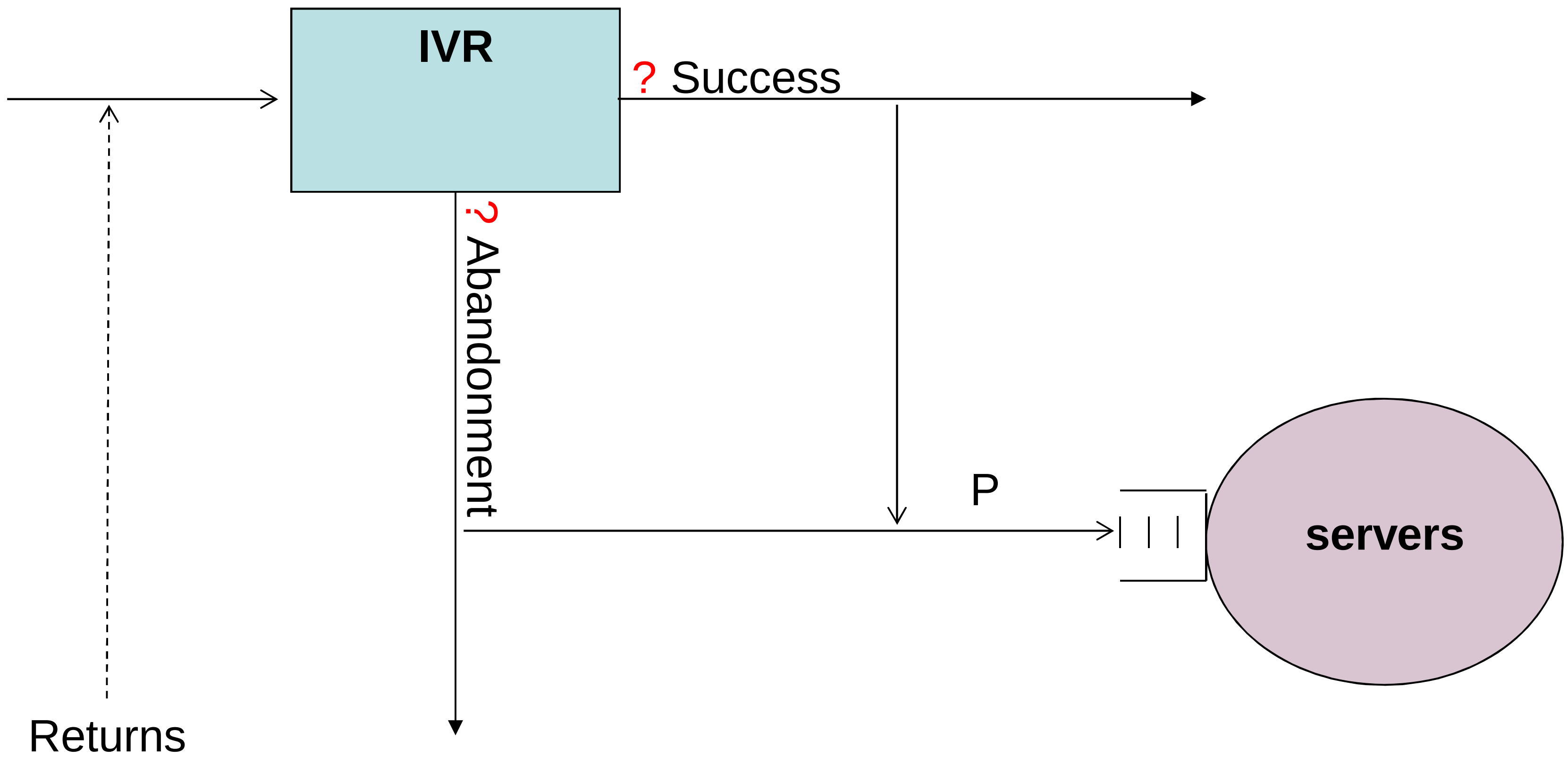
- **Properly designed** IVR system → increase customer satisfaction
  → increase loyalty
  → decrease staffing costs

⇨ **Customers self-serve**

⇨ **Increase profit**

Introduction

- **Call Center with an IVR system**



Literature Review

- Methodologies for evaluating IVRs
  - Quantifying IVR usability and cost-effectiveness
    - Agent time being saved by handling the call, or part of the call, in the IVR
    - Original reasons for calling vs. experience

      (by analyzing end-to-end calls)

    Suhm B. and Peterson P. (2002, 2009)

30% completed service in IVR

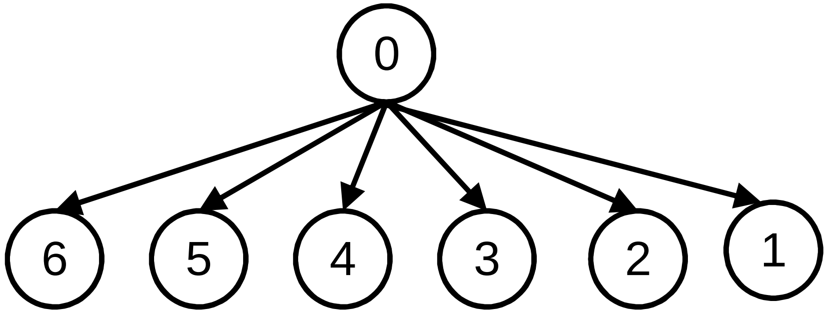
Only 1.6% completed it successfully

- Designing and optimizing IVRs
  - Human-Factor-Engineering
    - Mainly comparing broad (shallow) designs and narrow (deep) designs.

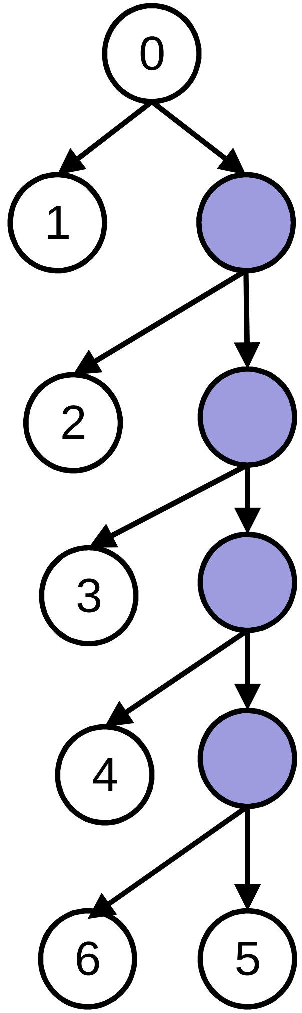
  Examples: Schumacher R. et al. (1995), Commarford P. et al. (2008) and many more.

Literature Review

Broad (shallow) design:



Narrow (deep) design:



Literature Review

- Reducing IVR service times
  - The IVR menu as a service tree – reducing the average time to reach a desired service

$$\Rightarrow \quad Minimize \quad f(T) = \sum_{i=1}^{M} t_i p_i$$

Salcedo-Sanz S. et al. (2010)

- Stochastic search in a Forest
  - Models of stochastic search in the context of R&D projects
  - Optimality of index policy.

Denardo E.V., Rotblum U.G., Van der Heyden L. (2004)

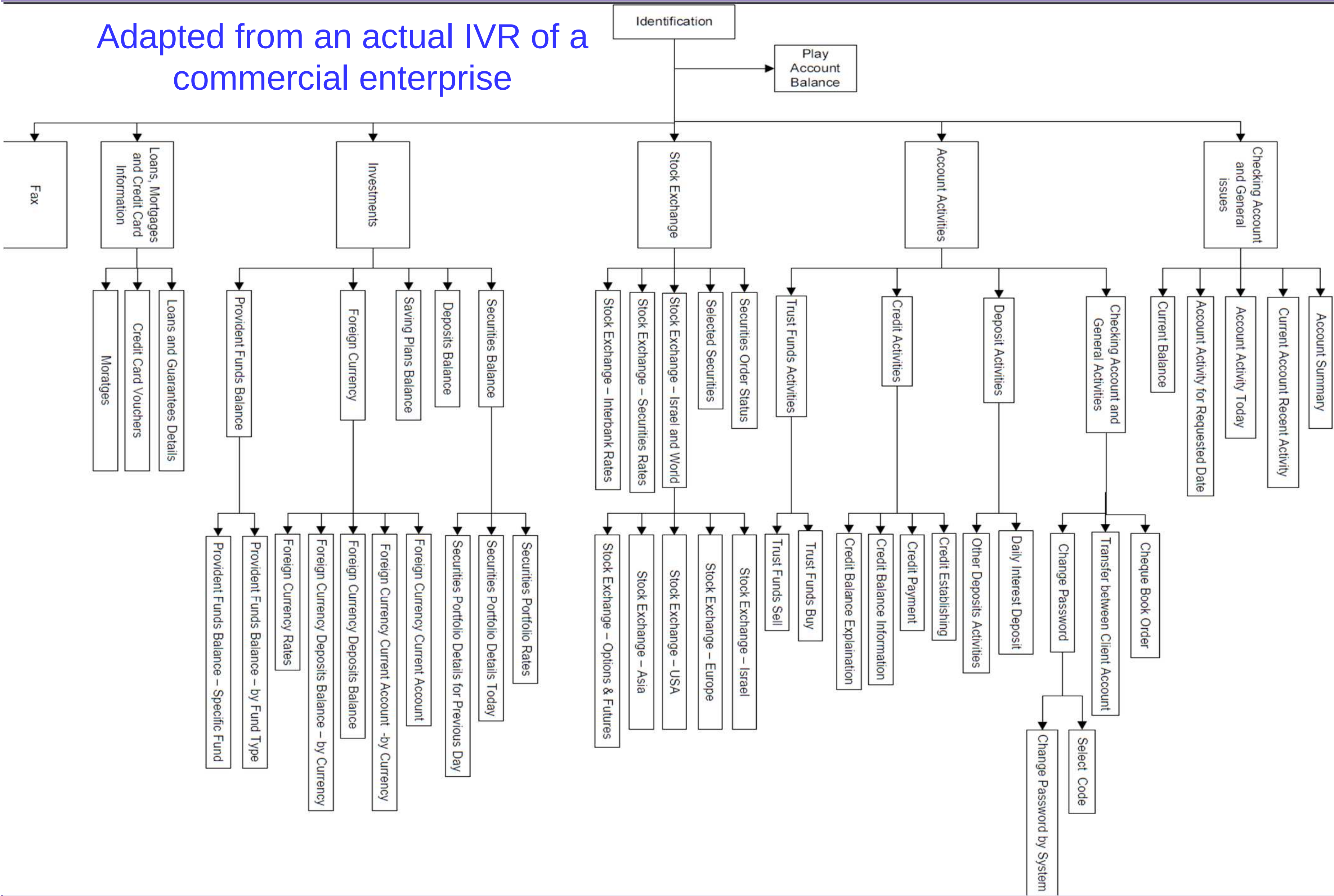
Granot D. and Zuckerman D. (1991)

Research Goals

- **Modeling customer flow within an IVR system**
- **Using EDA to estimate model parameters and identify usability problems: has actually inspired our theoretical model**
- **Using optimal search solutions and empirical analysis to compare alternative designs and infer design implications**

Search Model

Adapted from an actual IVR of a commercial enterprise



Search Model

- **Notations**

$G = (V, E)$ - rooted tree representing the IVR system

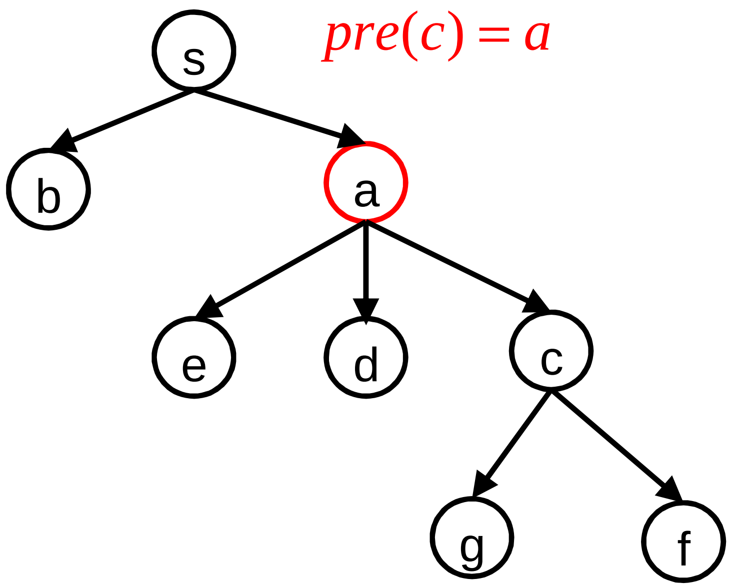
$M \subset V$ - set of tree leaves, representing IVR services

E - Tree edges, representing IVR menu options

s - root vertex, representing the IVR main menu

$A(i)$ - set of immediate successors of vertex i

$pre(i)$ - immediate predecessor of vertex i



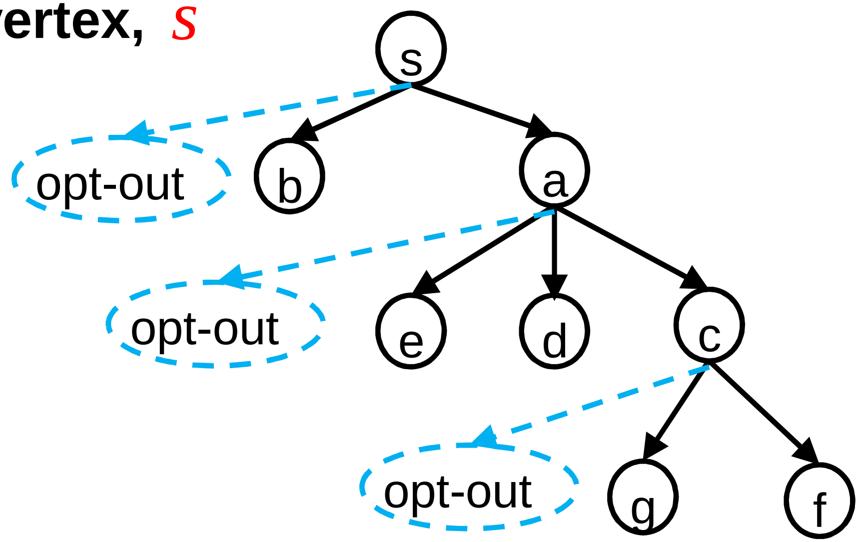
Search Model

- **Customer search**
  - **Customers pay for each unit of time they spend in the IVR**
  - **Customers receive reward when reaching a desired service**
  - **Customers may look for more than one service**
  - **Customers have finite patience**

Search Model

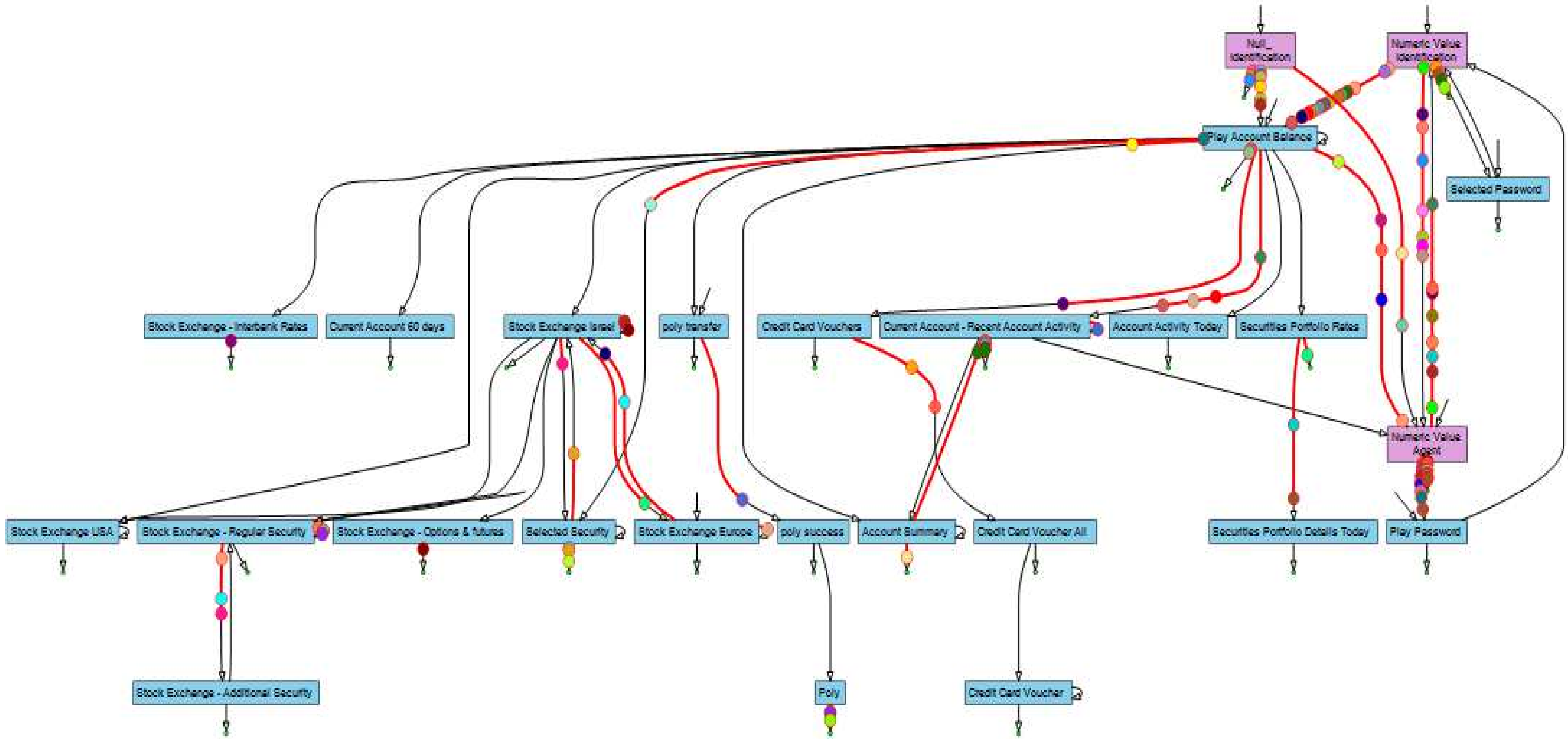
- **Customer search**
  - **At each stage the customer can choose:**

1. **Terminate the search and leave the system**
2. **Terminate the search and opt-out to an agent**
3. **Move forward to one of i's immediate successors,** $j \in A(i)$
4. **Move backward to i's immediate predecessor,** $pre(i)$
5. **Move backward to the root vertex,** s



Search Model

- **IVR flow animation**



Search Model

- **States**

$$S_n = \left(i, T, \overline{N}\right)$$

Where

$i =$ current vertex

$T =$ Total time spent in the serach

$\overline{N}$: $|V| \times 1$ vector representing repeated visits to each vertex

Search Model

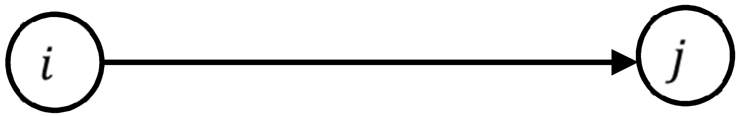
Model Parameters - Edges

$t_{i,j}^{N(i)}$ - exploration time of edge $e=(i,j)$,

given that vertex i was visited $N(i)$ times before

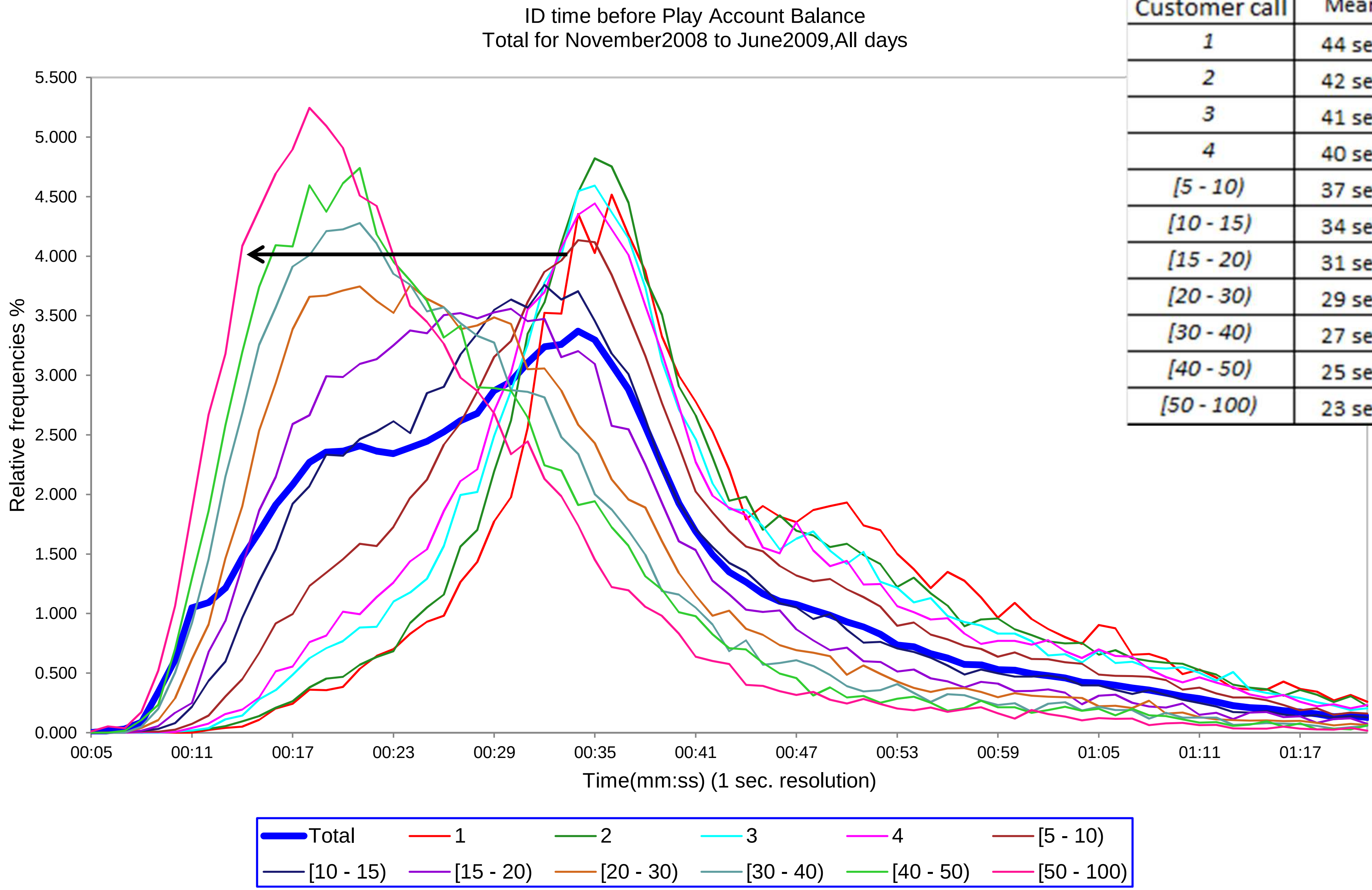
Where

$N(i)$ - number of previous visits to vertex i



$$S_n = \left(i, T, \overline{N}\right) \dashrightarrow S_{n+1} = \left(j, T + t_{i,j}^{N(i)}, \overline{N} + \delta(i)\right)$$

Search Model



Customer call	Mean
1	44 sec
2	42 sec
3	41 sec
4	40 sec
[5 - 10)	37 sec
[10 - 15)	34 sec
[15 - 20)	31 sec
[20 - 30)	29 sec
[30 - 40)	27 sec
[40 - 50)	25 sec
[50 - 100)	23 sec

Search Model

- **Model Parameters - Edges**

τ - customer patience

$\tau \sim \exp(\theta)$, assumed

$$\Rightarrow P_{i,j} = P\left(\tau > T + t_{i,j}^{N(i)} \middle| \tau > T\right) = P\left(\tau > t_{i,j}^{N(i)}\right) \text{ (memoryless)}$$

Search Model

- **Model Parameters - Leaves**

P_i - success probability of vertex i

r_i - reward earned from successful completion of vertex i

$t_{ser}(i)$ - time spent in vertex i given that we have been successful in visiting it

$t_F(i)$ - time spent in vertex i given that we have not been successful in visiting it

$$T_i = \begin{cases} t_{ser}(i) & w.p.\ P_i \\ t_F(i) & w.p.\ 1-P_i \end{cases}$$

- For non-leaves vertices, $i \in V \setminus M$:

$$P_i = 1$$

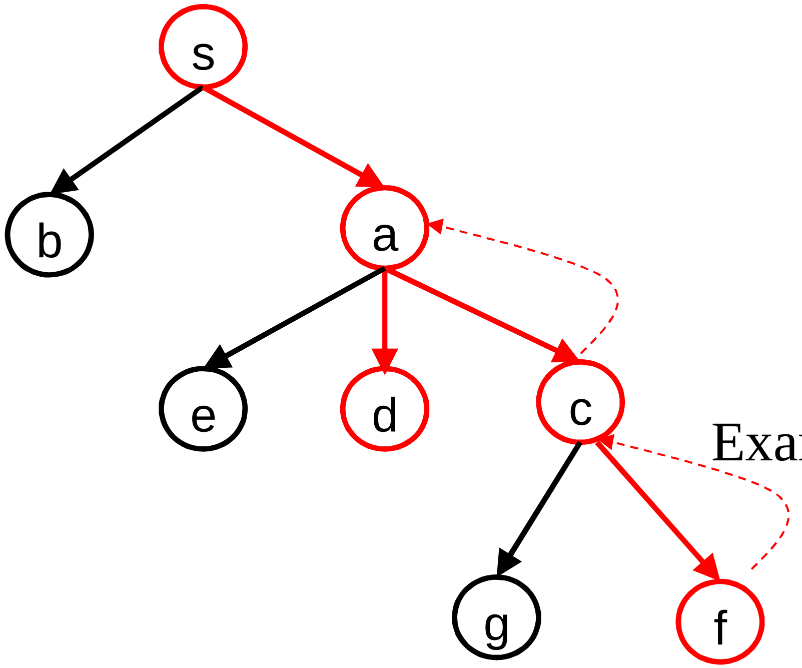
$$r_i = 0,\ t_{ser}(i) = 0,\ t_F(i) = 0$$

Search Model

o Search candidates

A sequence $U = (i_1, \ldots, i_n)$ of vertices in V will be called a *candidate*

Definition 1. A candidate U will be called *proper* if it starts with with s, and if every two sequential vertices is U : i_k, i_{k+1}, satisfies one of the following conditions:



$$i_{k+1} \in A(i_k)$$
$$i_{k+1} = pre(i_k)$$
$$i_k \neq s,\ i_{k+1} = s$$

Example: $U = s \rightarrow a \rightarrow c \rightarrow f \rightarrow c \rightarrow a \rightarrow d$

Search Model

Search candidates

A sequence $U = (i_1, \ldots, i_n)$ of vertices in V will be called a *candidate*

Definition 1. A candidate U will be called *proper* if it starts with with $s,$ and if every two sequential vertices is U : $i_k, i_{k+1},$ satisfies one of the following conditions:

$$i_{k+1} \in A(i_k)$$
$$i_{k+1} = pre(i_k)$$
$$i_k \neq s,\ i_{k+1} = s$$

Λ - The set of all *proper* candidates

Search Model

- **Search candidates**

$X(U)$ - Net revenue earned by fathoming candidate U

$$R(U) = E\left[X(U)\right]$$

(Expectation over: patience, success probabilities, time in vertices and edges)

Our goal is to find a candidate U^* that maximizes $R(U)$; $U \in \Lambda$

Search Model

- **Search candidates**

Proposition 1. A *proper* candidate U could be dominated by another candidate (could not be optimal) if it has at least one of the following properties:

1. There is at least one leaf $i \in M$ which appears more than once in U
2. The candidate U has a subsequence $pre(i) \rightarrow i \rightarrow pre(i),$ where $i \notin M$
3. The candidate U has a subsequence $i \rightarrow s,$ where $i \notin M$
4. The candidate U ends with $i \notin M$
5. The candidate U has a subsequence $i \rightarrow pre(i) \rightarrow i$
6. The candidate U has a subsequence $i \rightarrow s \rightarrow direct\ sequence\ to \rightarrow j,$ where $L(i, j) = pre(i)$

Search Model

- **Search candidates**

D - The set of all *proper* candidates with at least one of the properties of *Proposition 1*

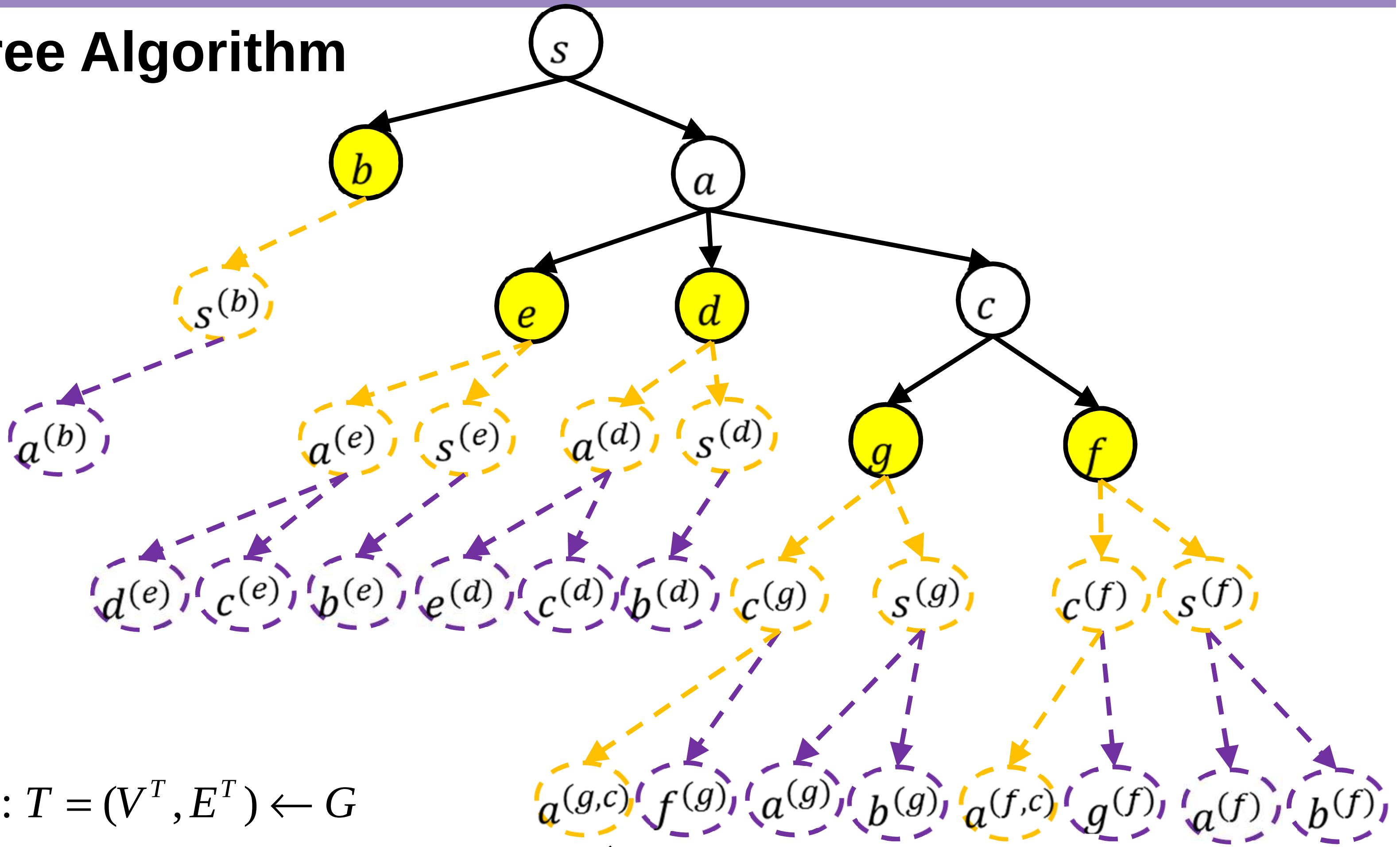
$$\Lambda' = \Lambda \setminus D$$

The set of all *admissible* candidates

Proposition 2. The set Λ' is finite and each of its candidates is a finite path

Search Model

Maximal Tree Algorithm

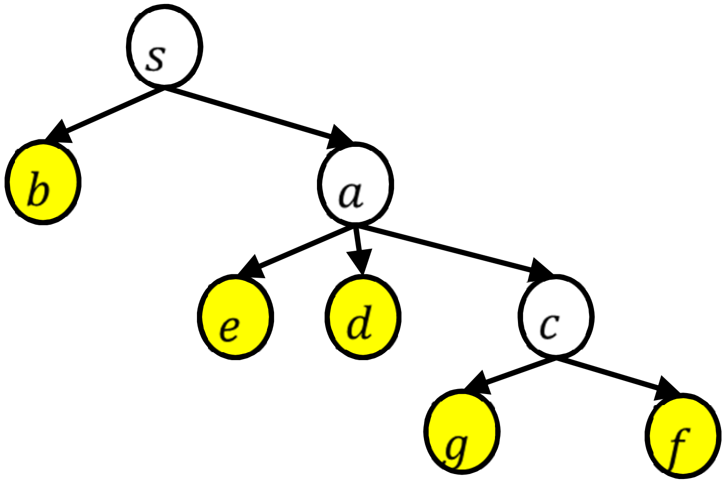


0. Initialization: $T=(V^T,E^T)\leftarrow G$
1. Spanning the tree with backward transitions
2. Spanning the tree with forward transitions
3. Updating T with the set of new vertices, M^T

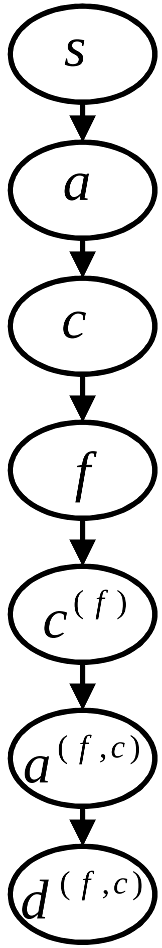
   if $M^T=\varnothing \rightarrow Stop$, else, return to step 1.

Search Model

- **Maximal Tree Algorithm**



$\overline{U} \in T:$



Recall: $S_n = \left(i, T, \overline{N}\right)$

$\forall u \in \overline{U}$

$base(u) =$ current vertex

$index(u) =$ ordered vector representing former backward transitions

Search Model

Theorem. A *proper* candidate U is *admissible* (in Λ') *iff* it is represented by a path starting from the root, $\overline{U} \in T = \left(V^T, E^T\right)$

$\Rightarrow$ The set $\Lambda' = \Lambda \backslash D$ and the output of Maximal Tree Algorithm are equivalent

MDP

- **Backward induction – Index calculation**

$r(v)$ - discounted revenue earned from vertex $v \in V^T$

$$r(v) = \begin{cases} \overbrace{r_v e^{-\alpha t_{ser}(v)}}^{\text{discounted reward from vertex}} - \overbrace{\int_0^{t_{ser}(v)} ce^{-\alpha s} ds}^{\text{payment for time spent in vertex}} & w.p.\ P_v \\ -\underbrace{\int_0^{t_F(v)} ce^{-\alpha s} ds}_{\text{payment for time spent in vertex}} & w.p.\ 1 - P_v \end{cases}$$

MDP

- **Backward induction – Index calculation**

$r^*(v)$ - discounted revenue earned from the tree with root vertex $v \in V^T$

$R^*(v) = E\left[r^*(v)\right]$

In the special case where $v \in M^T$: $r^*(v) = r(v)$

MDP

- **Backward induction – Index calculation**

$r_{u,v}$ - discounted revenue earned earned from edge $(u,v)\in E^T$,

given that we successfully reached vertex u after t_u units of time

$$R_{u,v} = E\left[r_{u,v}\right] =$$

$$E\left[\mathrm{I}_{\{\tau \ge t_u + t_{u,v}\}} \left(e^{-\alpha\left(t_u + t_{u,v}\right)} R^*(v) - \int_{t_u}^{t_u + t_{u,v}} c e^{-\alpha s} ds \right) + \mathrm{I}_{\{t_u + t_{u,v} > \tau \ge t_u\}} \left(-\int_{t_u}^{\tau} c e^{-\alpha s} ds \right) \middle| \tau \ge t_u \right]$$

MDP

- **Backward induction – Index calculation**

We get:

$$R_{u,v} = e^{-\alpha t_u}\left[\mathcal{L}_{t_{u,v}}(\alpha+\theta)\left(R^*(v)+\frac{c}{\alpha}\left(1-\frac{\theta}{\alpha+\theta}\right)\right)-\frac{c}{\alpha}\left(1-\frac{\theta}{\alpha+\theta}\right)\right]$$

While in u, it is enough to calculate:

$$\boxed{R^*_{u,v} = \mathcal{L}_{t_{u,v}}(\alpha+\theta)\left(R^*(v)+\frac{c}{\alpha}\left(1-\frac{\theta}{\alpha+\theta}\right)\right)-\frac{c}{\alpha}\left(1-\frac{\theta}{\alpha+\theta}\right)}$$

MDP

- **Backward induction – Index calculation**

$$if \quad \max_{v\in A(u)} R^*_{u,v} \le 0: \; Set \;\; R^*(u) = E\left[r(u)\right]$$

$$\text{Set} \quad \sigma^*(u) = Stop$$

$$if \quad \max_{v\in A(u)} R^*_{u,v} > 0: \; Set \;\; r^*(u) = \begin{cases} \left(r_u + \max_{v\in A(u)} R^*_{u,v}\right) e^{-\alpha t_{ser}(u)} - \int_0^{t_{ser}(u)} ce^{-\alpha s} ds & w.p. \;\; P_u \\ \left(\max_{v\in A(u)} R^*_{u,v}\right) e^{-\alpha t_F(u)} - \int_0^{t_F(u)} ce^{-\alpha s} ds & w.p. \;\; 1 - P_u \end{cases}$$

$$Set \;\; R^*(u) = \mathrm{E}\left[r^*(u)\right]$$

$$Set \;\; \sigma^*(u) = \arg\max_{v\in A(u)} \left\{R^*_{u,v}\right\}$$

MDP

$$G = (V, E)$$

IVR original tree

↓

$$T = (V^T, E^T)$$

Extended tree representing possible customer actions within the IVR system

↓

Calculating indices using backward induction to find optimal search policy

Case Study

- A medium bank
- About 450 working agents per day (~200 during peak hours)
- Call by call data from May 1st, 2008 to June 30th, 2009
- Overall Summary of calls

IVR Calls	Total	% Out of total	Average per weekday
# Served only by IVR	27,709,543	61.6%	50,937
# Requesting also agent service	17,280,159	38.4%	31,765

10% of IVR calls – Identification only!

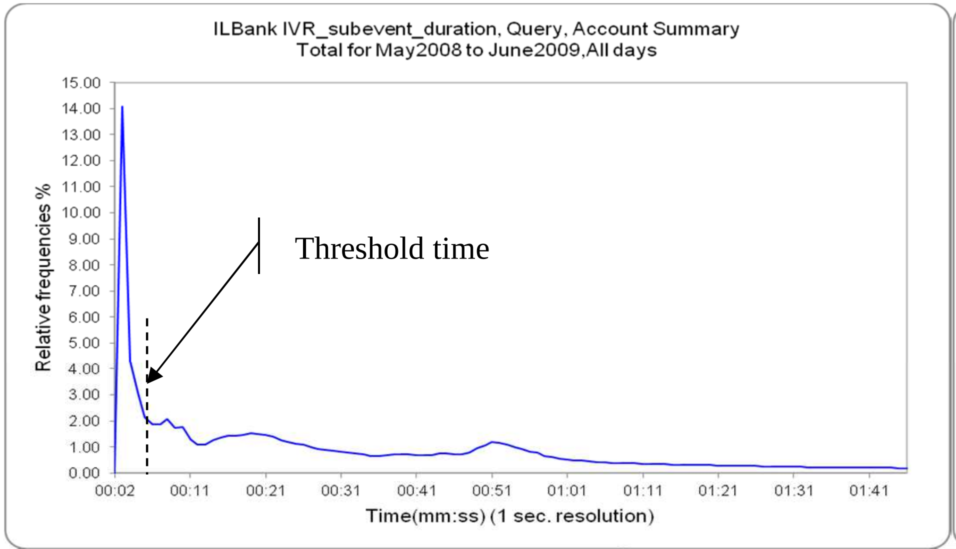
Case Study

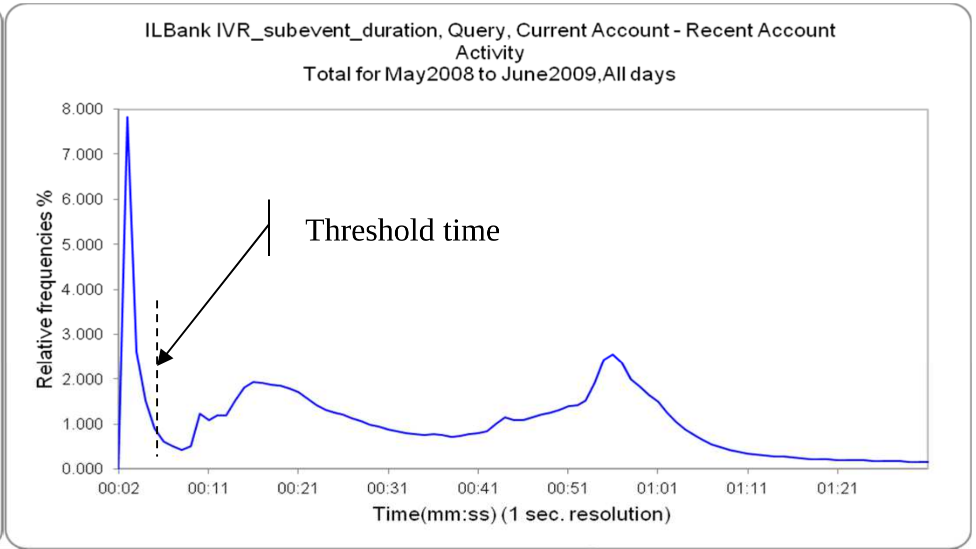
IVR service	Frequency out of average number of calls per day
Play Account Balance	72.25%
Current Account - Recent Account Activity	12.03%
Account Summary	5.70%
Credit Card Vouchers	3.95%
Stock Exchange Israel	2.98%
Selected Security	2.58%
poly transfer	2.36%
Account Activity Today	2.27%
Selected Password	1.88%
Current Account 60 days	1.60%
Stock Exchange - Interbank Rates	1.38%
Stock Exchange Europe	1.33%
Securities Portfolio Rates	1.22%
Stock Exchange USA	1.14%
Stock Exchange - Regular Security	1.12%
Current Balance	0.96%
Change Password	0.66%
Securities Portfolio Details Today	0.58%
Deposits	0.54%
Stock Exchange - Options & futures	0.43%
Provident Funds Balance	0.36%
Stock Exchange Asia	0.29%
Stock Exchange - Securities	0.19%
Savings Plans	0.12%
Cheque Book Order	0.08%
Daily Interest Deposit	0.06%
Credit Balance	0.04%
Foreign Currency Current Account Activity	0.03%
Credit Establishing	0.02%
Foreign Currency Deposit Activity	0.01%
Trust Funds Sell	0.01%
Credit Information	0.01%
Trust Funds Buy	0.00%
Foreign Currency Rates Activity	0.00%
Foreign Currency Current Account	0.00%
Foreign Currency Deposits	0.00%
Provident Fund Balance	0.00%
Credit Payment	0.00%
Mortgage	0.00%

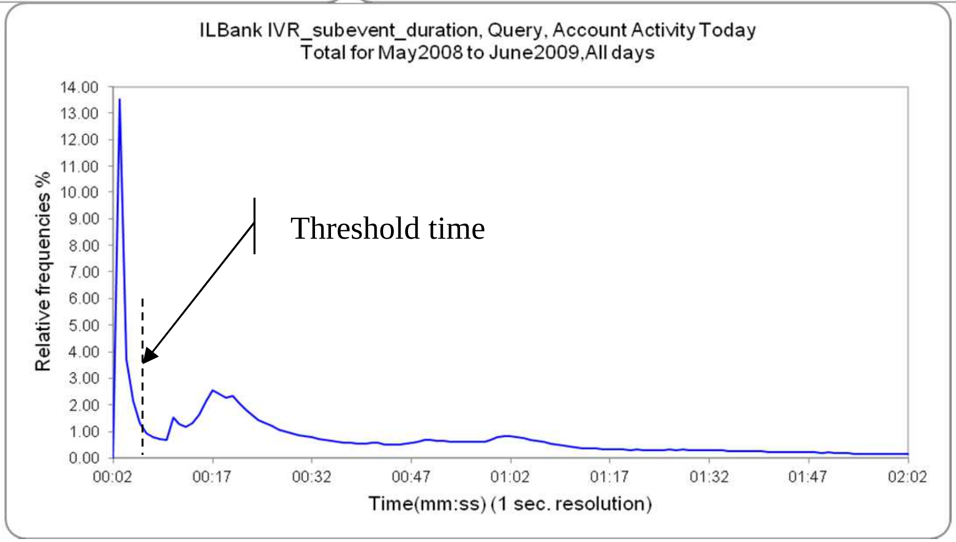
Requested by less than 2% of the calls

Case Study

- **Estimating abandonments**

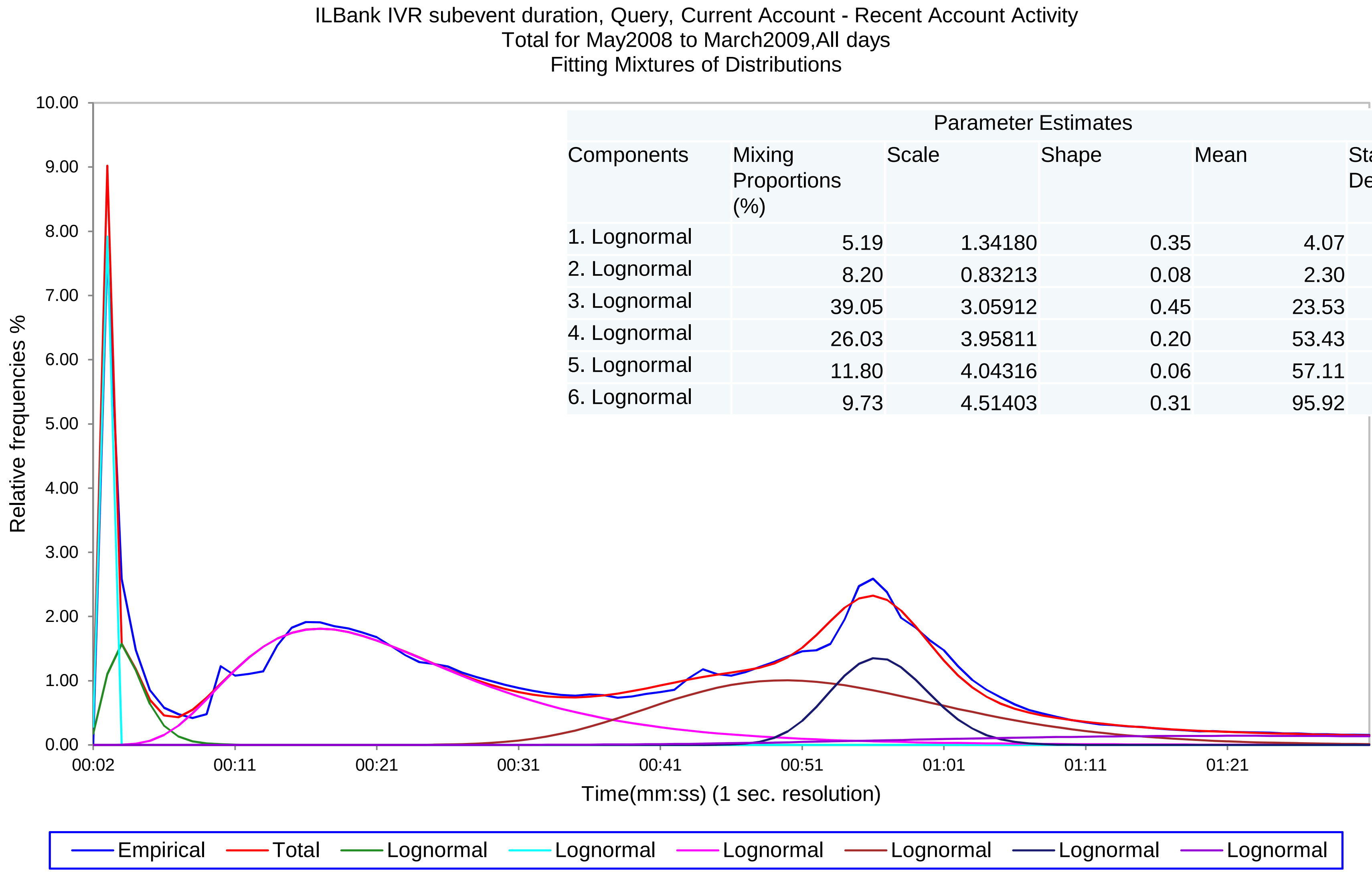






Case Study

- **Estimating abandonments**

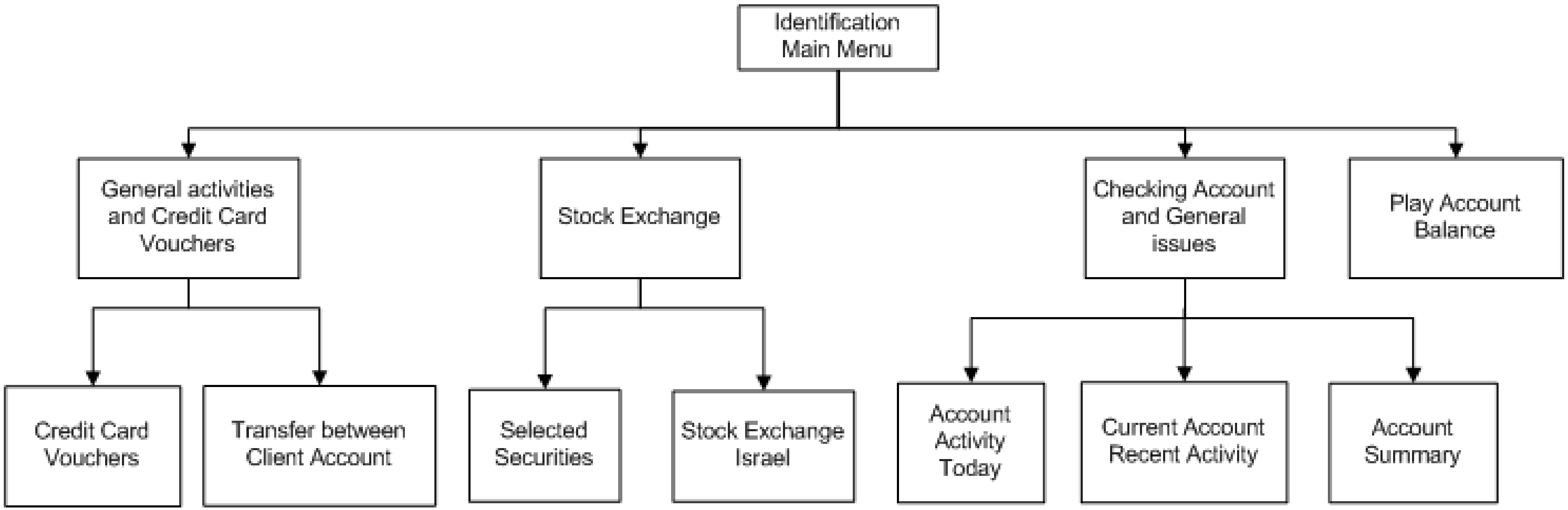


Parameter Estimates					
Components	Mixing Proportions (%)	Scale	Shape	Mean	Standard Deviation
1. Lognormal	5.19	1.34180	0.35	4.07	1.49
2. Lognormal	8.20	0.83213	0.08	2.30	0.18
3. Lognormal	39.05	3.05912	0.45	23.53	11.03
4. Lognormal	26.03	3.95811	0.20	53.43	10.85
5. Lognormal	11.80	4.04316	0.06	57.11	3.47
6. Lognormal	9.73	4.51403	0.31	95.92	30.95

Model Implications

- **Example - Comparing different designs**

  (based on frequent services in our database)



Model Implications

- **Example - Comparing different designs**
  - $c = 0.05$ NIS/sec
  - $\alpha = 0.05$
  - $\theta = 0.004 \Rightarrow$ average customer patience $= 250$ sec
  - t_{ser}, t_F exponentially distributed
  - $t_{i,j}$ exponentially distributed for every i, j

Service	r Type 1	t_{ser} (mean, sec)	t_F (mean, sec)	P
Play account balance	200c	21	6	0.75
Account Summary	0	51	5	0.765
Recent Account Activity	400c	45	3	0.87
Account Activity Today	0	50	3	0.78
Selected Securities	0	43	3	0.7
Stock Exchange Israel	0	16	7	0.775
Transfer between Account	0	17	3	0.6
Credit Card Vouchers	600c	49	3	0.915

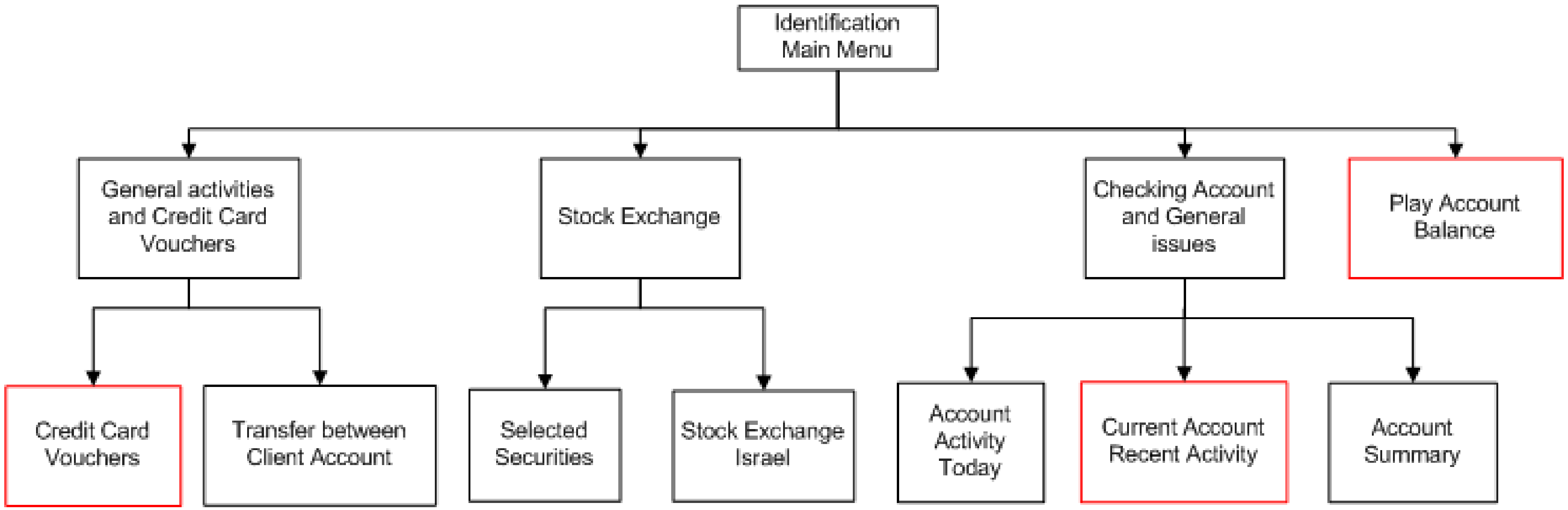
Model Implications

- **Example - Comparing different designs**
- **Design A - original:**

Optimal candidate :

U: Main menu $\rightarrow$ Play Account Balance $\rightarrow$ Main Menu $\rightarrow$ Checking account $\rightarrow$ Recent Account Activity $\rightarrow$ Main menu $\rightarrow$ General Activities $\rightarrow$ Credit Card Vouchers

Customer revenue: $R(U) = 3.38$



Model Implications

- **Example - Comparing different designs**
- **Design B - deep:**

Optimal candidate:

U: Main Menu $\rightarrow$ Play Account Balance

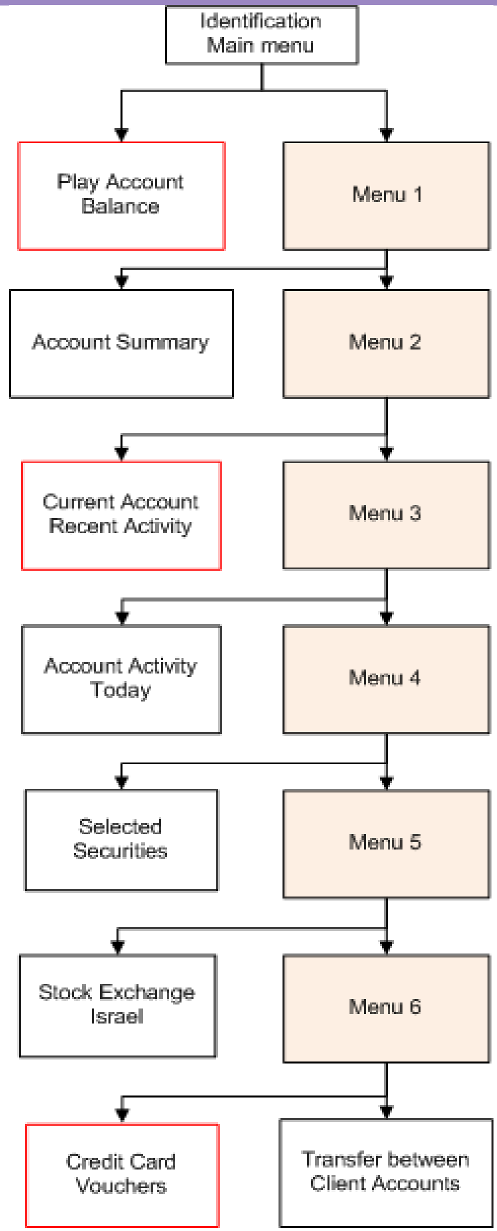
$\rightarrow$ Main Menu $\rightarrow$ M1 $\rightarrow$ M2

$\rightarrow$ Recent Account Activity

$\rightarrow$ M2 $\rightarrow$ M3 $\rightarrow$ M4 $\rightarrow$ M5

$\rightarrow$ M6 $\rightarrow$ Credit Card Vouchers

Customer revenue: $R(U) = 2.74$



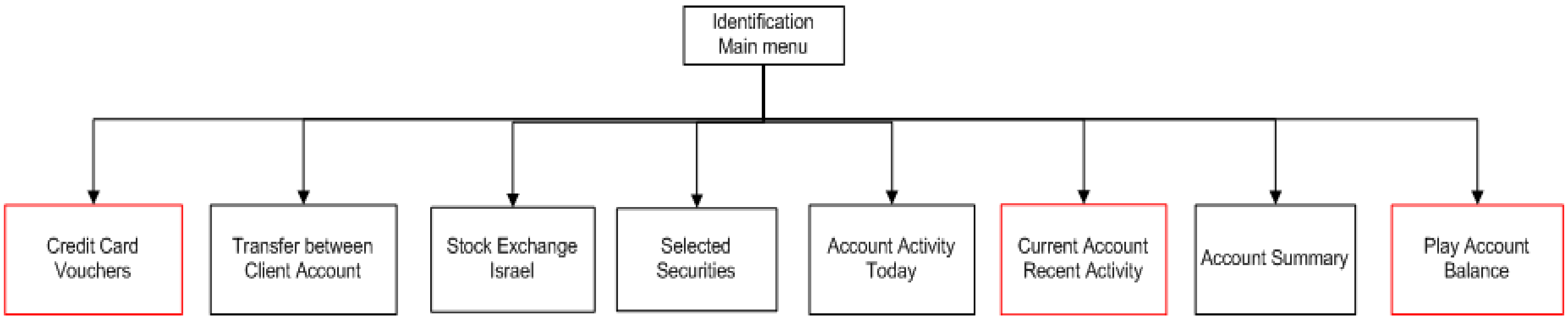
Model Implications

- **Example - Comparing different designs**
- **Design C - shallow:**

Optimal candidate:

$$U: \text{Main Menu} \rightarrow \text{Play Account Balance} \rightarrow \text{Main Menu} \rightarrow \text{Recent Account Activity} \rightarrow \text{Main Menu} \rightarrow \text{Credit Card Vouchers}$$

Customer revenue: $R(U) = 3.61$



Model Implications

- **Comparing different designs – Organization point of view**

Example, assume:

- **10,000 calls per day**
- **Cost per unit of time of communication = 0.003 NIS/sec**
- **Cost per second of agent call = 0.03 NIS/sec (based on UG project)**

	Design A - Original	Design B – Deep	Design C - Shallow
Customer revenue	3.38	2.74	3.61
Cost of communication	4,736	5,606	4,826
Revenue from saved agent time	30,564	30,564	30,564
Total organization profit	25,828	24,958	25,738

Model Implications

- **How to design an IVR tree to encourage the use of certain services?**
- **Customer learning?**
- **Do customers navigate optimally?**
- **Services with low demand – lack of interest or lack of patience?**
- **Sensitivity to customer patience?**
- **Anticipating long waiting in agent queue – does it affect customer behavior?**
- **Using IVR and state information to control customers behavior examples: return later, use IVR for service**

Summary and Future research

- Stochastic models of customer flow within the IVR
  - Comparing IVR designs
  - Insights for better designs
  - Supplementing previous knowledge in HFE, CS fields
  - Data analysis as a basis for theoretical models
  - Can be easily modified to other self-service systems

  Example: Web search engines

  Hassan A. et al. (2010)

- Interesting open questions:
  - How does one recognize an abandonment from self-service when “seeing” one?
  - Interrelation between customer-IVR interaction and customer-agent interaction?

Thank you,
Questions?

"Thank you for calling Customer Service.
If you're calm and rational, press 1.
If you're a whiner, press 2.
If you're a hot head, press 3...."