RESOURCE-DRIVEN ACTIVITY NETWORKS: AN ANALYSIS ON AN ANONYMOUS BANK CALL CENTER

Vacation scholars: Zhichong Lu, Neel Date and Samantha Young Supervisors: Peter Taylor and Mark Fackrell

Introduction

Resource-Driven Activity Networks(RANs), which are are established by Avishai Mandelbaum, Mor Armony, Nitzan Carmeli, Petar Momcilovic and Galit Yom-Tov[1], are data-based models of networked activities, carried out by resources. It mainly consists of activity types (e.g. nurse admission and calling service) and resource pools (e.g. doctors, nurses, customers and servers with various types). It captures a dynamic mechanism: A unit of resource follows a path of its different sub-resource (e.g. arriving customer and available server) and activities consume sub-recourses and then produce (possibly different) sub-resources once completed.

As a starting point, We focus on the call centre data from an Anonymous Bank in Israel in 1999. The data includes all the calls handled by the call center, over the period of 12 months from January 1999 until December 1999. In this project, we apply the RAN paradigm to the data and use simulation to help call center's manager plan and control hourly and daily performance such as how call centers manage their calls so that 80% of them are answered within 30 seconds. Specifically, we focus on the data from November weekdays. The reason is that there are large number of calls in November and there is no holidays which can avoid biased service time.

Model

There are five different types of service: PS - regular activity; PE - regular activity in English; IN - internet consulting; NE -stock exchange activity; NW - potential customer getting information.

Set up

- Five activities (five service types): PS, PE, IN, NE, NW
- Six resources: five types of customers with five different service demand and one type of server (Temporarily, we assume each server can deal with five different services, we will relax this assumption later on)
- Six sub-resources: five types of arriving customers and one type of available servers
- Exogenous net-flow: cumulative net-flow of sub-resources

$$\Lambda^T = \left[\epsilon_{PS}(t)\ \epsilon_{PE}(t)\ \epsilon_{IN}(t)\ \epsilon_{NE}(t)\ \epsilon_{NW}(t)\ s(t)\right]$$

where $\epsilon_i(t)$ denotes the cumulative number of exogenous-arriving calls by the time t ($i \in \{PS, PE, IN, NE, NW\}$) and $s(t)$ denotes the cumulative number of net-flow servers by the time t.

- Activity-duration: G_j is the cumulative distribution function of the duration time of type j activity; $\bar{G_j} = 1 - G_j$

$$G^T = \left[G_{PS}(t)\ G_{PE}(t)\ G_{IN}(t)\ G_{NE}(t)\ G_{NW}(t)\right]$$

- Consumption matrix $C_{l,j}$ = amount of sub-resource l consumed by a single type-j activity:

$$C = \begin{bmatrix} C_\epsilon \\ C_s \end{bmatrix} = \begin{bmatrix} I_{5\times5} \\ 1_{1\times5} \end{bmatrix}$$

where the partition C_ϵ denotes the consumption matrix regarding the 5 types of arriving customers and partition C_s denotes the consumption matrix regarding the servers. Similar subscripts apply to the production matrix

- Production matrix: $P = P^+ + P^-$:
where $P^+_{l,j}$ ($P^-_{l,j}$) = amount of sub-resource l that remains (leave) in the system, upon completion of a single activity j

$$P^+ = \begin{bmatrix} P^+_\epsilon \\ P^+_s \end{bmatrix} = \begin{bmatrix} 0_{5\times5} \\ 1_{1\times5} \end{bmatrix}$$

$$P^- = \begin{bmatrix} P^-_\epsilon \\ P^-_s \end{bmatrix} = \begin{bmatrix} I_{5\times5} \\ 0_{1\times5} \end{bmatrix}$$

- Plan $X = X(t)$ is a five-dimensional function of time $X_j(t)$ = total number (amount) of type j activities that started during $[0, t]$

$$X = \left[X_{PS}\ X_{PE}\ ...\ X_{NW}\right]^T$$

Note that in the example of call centre we analysed, $X(t)$, which shows the planned total number of five different services during $[0, t]$, is the key decision variable of the manager to manage the volumes of call services for a period of $[0, t]$.

RAN Master Inequality(RMI)

RAN Master Inequality(RMI)

To make a given plan $X(t)$ feasible, we require enough resources to make it happen. This is mathematically captured by the RAN Master Inequality. This inequality means that the amount of sub-resources that need to be consumed must be no more than the amount of current available sub-resources at any time: $CX(t) \le P^+(GX)(t) + \Lambda(t)$ at every $t \ge 0$. This specifies the minimum amount of resources required to implement the plan $X(t)$.

For customers:

$$C_\epsilon X(t) \le P^+_\epsilon GX(t) + \epsilon(t) \Rightarrow X(t) \le \epsilon(t)$$

For servers:

$$C_s X(t) \le P^+_s (GX)(t) + s(t) \Rightarrow X(t) \le G * X(t) + s(t)$$

RMI: $X \in D^1_\uparrow$ s.t. $X(t) \le \epsilon(t)$ and $X(t) \le G * X(t) + s(t), \forall t \ge 0$

Optimisation goal

Let $Q^{\leftarrow}_t(> r)$ be the number of queueing customers at time t, whose time-in-queue is over r seconds, so $Q^{\leftarrow}_t(> r)$ can be seen as one type of queue management plan made by the manager. In the RAN model, $Q^{\leftarrow}_t(> r) = [\epsilon(t - r) - X(t)]^+$. We can see that a given queue management plan ($Q^{\leftarrow}_t(> r)$) specifies a specific plan $X(t)$.

To formalize the manager's plan mentioned in the introduction part: minimizing the percentage of the calls that are answered beyond 30 seconds waiting, we can write down the following optimisation plan: use minimum amount of business resource (e.g. $s(t)$ a shift structure or number of server in our simulation case) to achieve $\frac{Q^{\leftarrow}_{61200}(>30)}{\sum \epsilon_i(61200)} \le 0.2$ (there are 61200 seconds per working day and $i \in \{PS, PE, IN, NE, NW\}$). Next, we apply the simulation plan to solve with this in a more detailed way.

Simulation Process

Before implementing the simulation, we first did some data analysis mainly on the arrival process and service time of different types. For convenience, we first exclude two types of calls: IN, since its calls are answered by a different group of servers, and PE since it is very rare in our data set.

The arriving patterns of different types, which are shown in the following graphs, show that there exists time-dependent patterns (e.g. there exists some arriving peaks during stock trading time). Therefore, we decide to simulate different types' arriving process by using different piece-wise homogeneous Poisson processes, i.e. a higher Poisson rate will be associated with time periods that are busier.

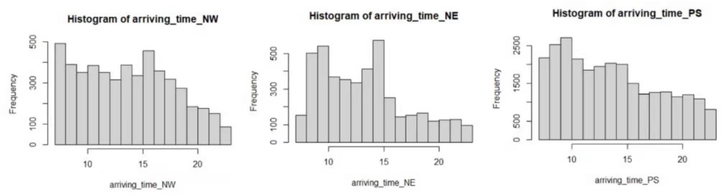


Fig. 1: Arriving process

Similarly, we simulate the service time of different types by dividing them into morning section and afternoon section, and model each type's service time using two exponential distribution with two different parameters.

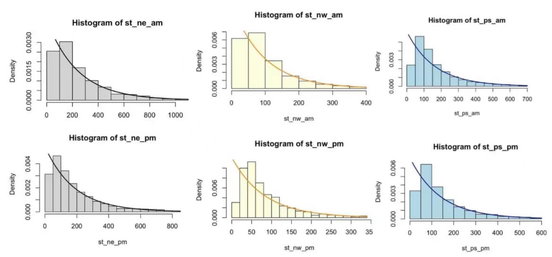


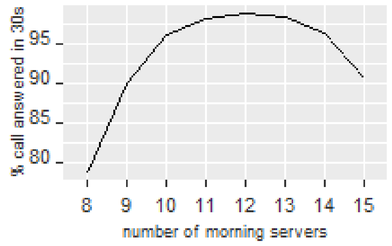
Fig. 2: Service time distribution

We now provide a brief outline of the simulation itself, which was implemented using Python.We first pre-generate calls with their arrival and service times as discussed before (piece-wise homogeneous Poisson process and exponential distribution with time-dependent parameters. This is done by type, but since all call types arrive independently to each other to the same servers, we need to merge these calls, preserving their order of arrival times.

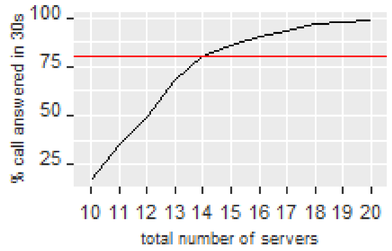
Then, we run the discrete event simulation, which moves forward in steps of one second at a time. At each of these time steps, a call may or may not arrive. Also, a previously busy server may or may not become free. By inspecting the data structures and suitably updating the variables that govern these events, the program then makes decisions of whether the call is directed to an idle server or to the queue. It also inspects whether a call waiting in the queue can be served by a newly idle server, doing so before a new call in line with the first-come-first-serve rule.

Simulation Result

The simulation was run 100 times under various input parameters. These inputs were varied with two differing methodologies, briefly described below:



1. constant number of available servers (= 20), variable composition across the two shifts. 12 servers working in the morning and 8 in the evening yielded the best results, with the median of the percentage of calls answered within 30 seconds across 100 simulations being 98.7%. Further, only about 3% of customers had to wait in the queue at all, although this was for a median of 33 seconds. Hence, most customers that did have to wait actually did so for over 30 seconds, but these were very few (most probably at narrow peak timings).



2. constant morning-evening ratio (approximately 3:2 as was found to be the best in method 1), variable no. of available servers. As expected, keeping the morning-evening composition constant, decreasing the total no. of servers decreases the percentage of calls answered within 30 seconds. However, this decrease is not very dramatic near our current strength of 20 workers. In fact, having just 14 servers with 9 in the morning and 5 in the evening yields a rate in excess of 80%, with about 27% of customers having to wait a median of 95 seconds in this case. Thus, it may be better for the call centre to release a few of its employees for better resource management.

Concluding Remarks

The RAN framework provides a powerful alternative to traditional queueing theory to model complex systems involving a large number and many classifications of interacting servers, activities, intermediate resources, and customers. Admittedly, our model makes several simplifying assumptions which do not do full justice in exhibiting the functionality of RANs, it provides a simple example of a RAN in action about how RAN can improve the operation performance of call centre system.

The next step would be to relax these assumptions and investigating the effect (eg. that each server can deal with each type of service, or introducing customer priorities and moving beyond the FCFS system). We can also explore dividing the day into more than one piece, and having more shift structures in which the servers can work to better mimic reality. Also, we can further apply the RAN paradigm to more complex systems such as hospitals.

Acknowledgements

We are grateful to the Vacation Scholarship Program that offered us a valuable opportunity to experience how mathematical research is like. We would like to thank Professor Peter Taylor and Dr. Mark Farkwell for his professional advice, inspiring guidance, and constant encouragement throughout the project.

References

[1] Avishai Mandelbaum. "Resource-Driven Activity Networks (RANs), arising from "Theompirical" Research at the Technion SEELab". In: *Data-driven Queueing Challenges* (Aug. 2021).