

An Optimization Framework for Resource Allocation in Multi-Tenant Communication Networks

CCAAW 2021

Kyle Polich (Data Skeptic)
Brad Bode (Atlas Space Operations)
Alex Reeves (Data Skeptic)

Problem

- Satellite operators need to communicate with spacecraft
- Dedicated ground stations are inefficient



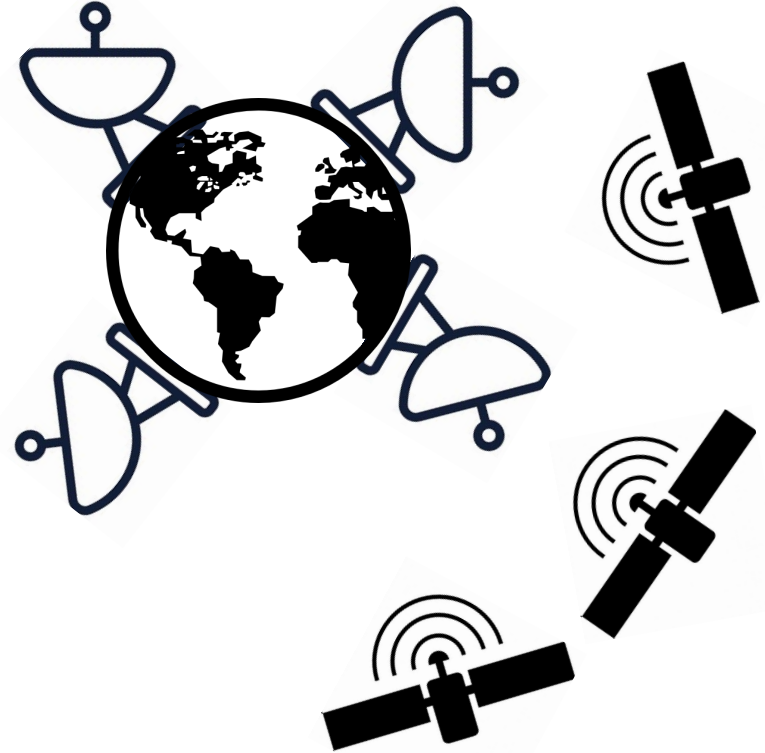
Problem

- Satellite operators need to communicate with spacecraft
- Dedicated ground stations are inefficient
- An industry of “ground station as a service” providers need ways to allocate resources amongst many tenants



Problem

- Satellite operators need to communicate with spacecraft
- Dedicated ground stations are inefficient
- An industry of “ground station as a service” providers need ways to allocate resources amongst many tenants
- **Over-subscribed networks need to be de-conflicted**



State of the Art

- **Under-utilized networks** can guarantee allocation
- Human operators perform **manually configuration** often via email
- **Proprietary / custom scheduling** (e.g. Atlas's Flex Scheduling solution)
- The path toward a **federation networks** is not clear

Two Tenant System (Toy Example)

Tenant **A** - 6 minute Visibilities every 12 minutes

Tenant **B** - 4 minute Visibilities every 7 minutes

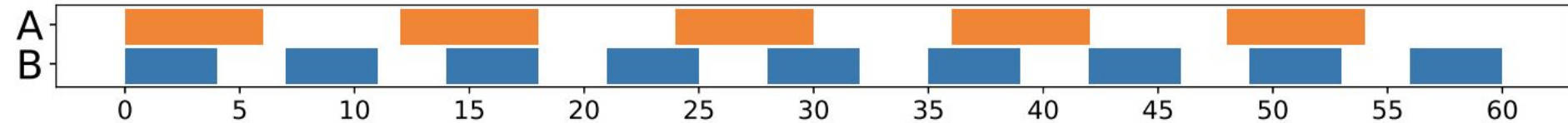


Fig. 5. Simple Use Case with Overlapping Visibilities

Two Tenant System (Toy Example)

Tenant **A** - 6 minute Visibilities every 12 minutes, **4 Contacts / hour**

Tenant **B** - 4 minute Visibilities every 7 minutes, **18 Minutes / hour**

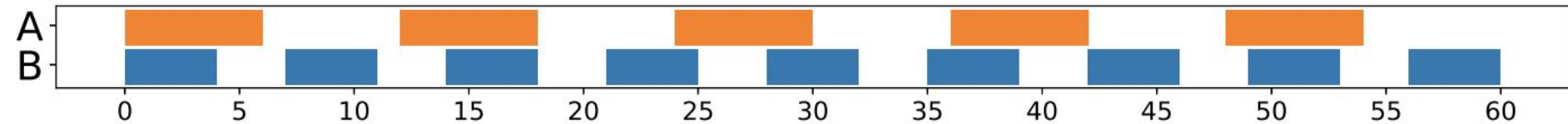


Fig. 5. Simple Use Case with Overlapping Visibilities

Cognitive Scheduling

- Capable of operating with minimal or no human interaction
- Adapting to changing conditions
- Extending beyond it's pre-programmed knowledge

User specifies:

1. *an Objective*
2. *a Utility Function*
3. *Parameters* for the function

Objectives

Primary

- Contacts per Day
- Average / min / max Contact Duration
- Average / min / max Minutes Between Contacts

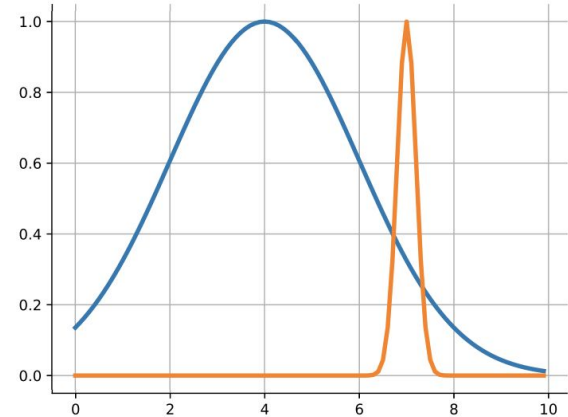
Extended

- Minimize expected wait time
- Maximum buffer fill size
- Percentage of communication by band

Utility Functions

$$\text{Score} = \text{Utility}(\text{value}, \text{UserPreference})$$

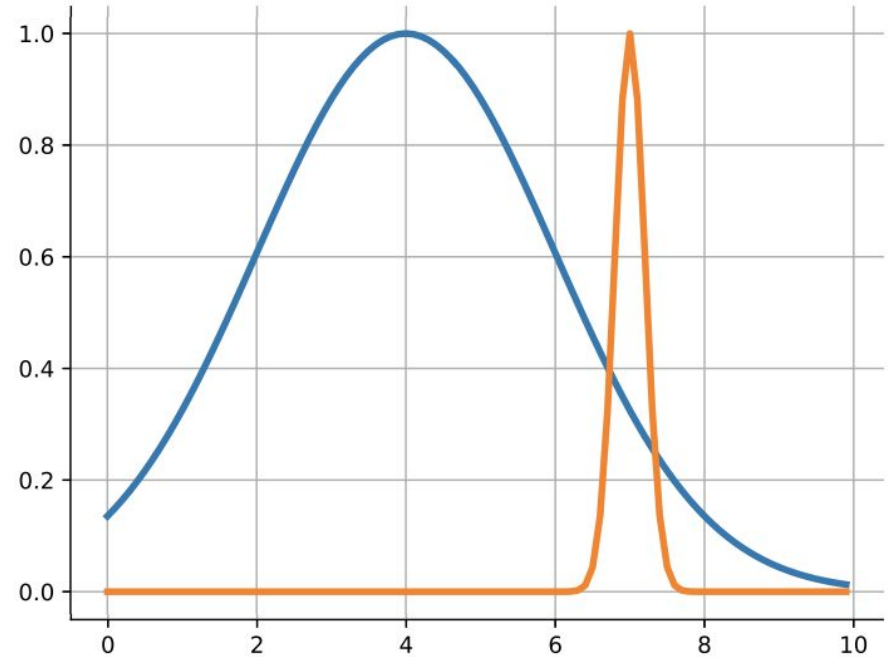
- Allow users to describe their preferences in terms of parameterized utility functions
- Optimize toward maximizing utility
- Enabled “set and forget” scheduling
- Resilient to interruptions (e.g. LEOP)
- Finds solutions rather than executing solutions



Utility Functions: Normalized Gaussian

User defined parameters

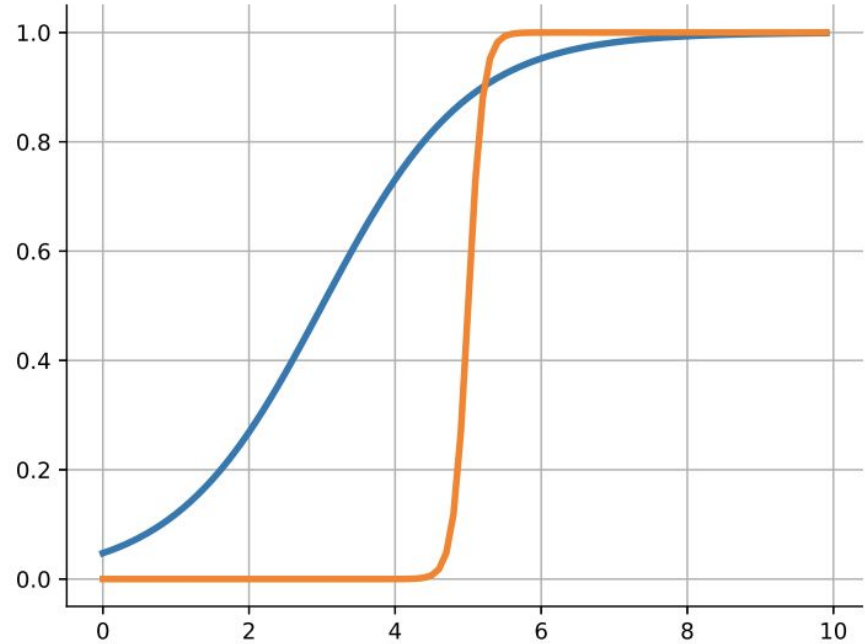
- Mean
- Standard deviation
- Min
- Max



Utility Functions: Logistic

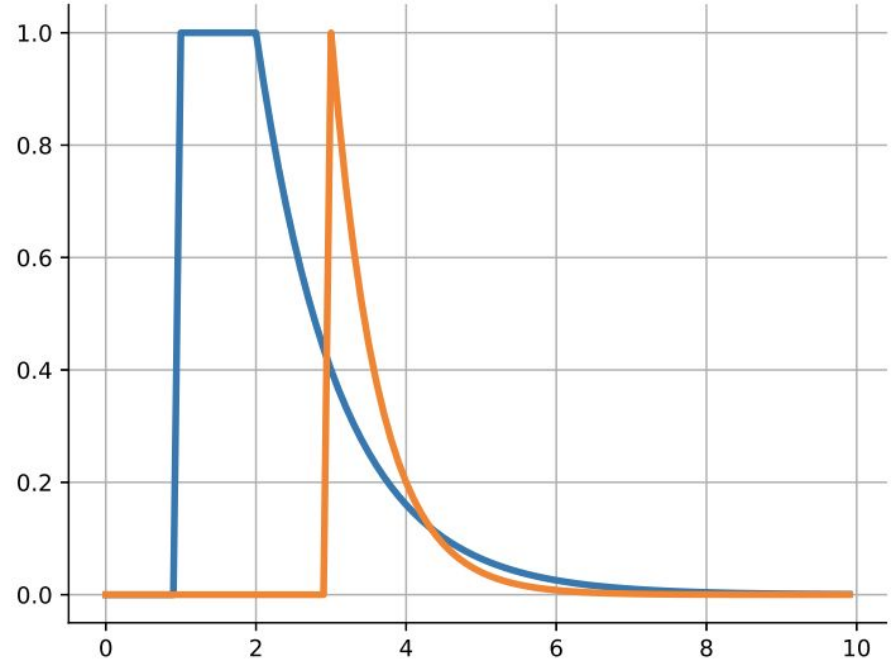
- Bias
- Shape

Often useful for “soft floor” or “soft ceiling”



Utility Functions: Halflife

- Start
- Begin Decay
- Lambda decay param



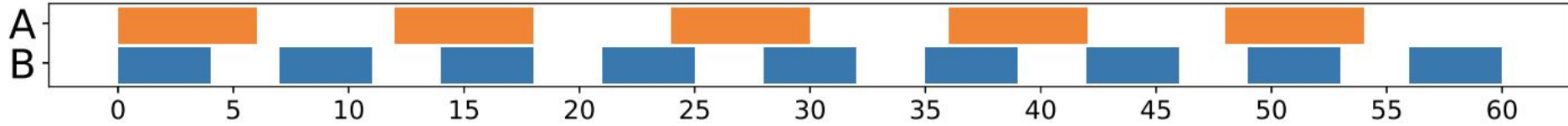
Algorithm

We present a round-robin style allocation process which iteratively adds Tasks to available Visibilities until users are satisfied or the network is fully allocated.

Algorithm 1 Cognitive Constellation Management (CCM)

```
1: procedure SCHEDULE(users, visibilities)
2:   output  $\leftarrow \emptyset$ 
3:   updated  $\leftarrow True$ 
4:   while updated do
5:     updated  $\leftarrow False$ 
6:     for user  $\in$  users do
7:       v = filter(visibilities, user)
8:       choice = SelectOptimal(user, v, output)
9:       output = output  $\cup$  choice
10:      updated  $\leftarrow True$ 
11:    end for
12:  end while
13: end procedure
```

Two Tenant System (Toy Example)



Tenant **A** - **3 Contacts / hour**

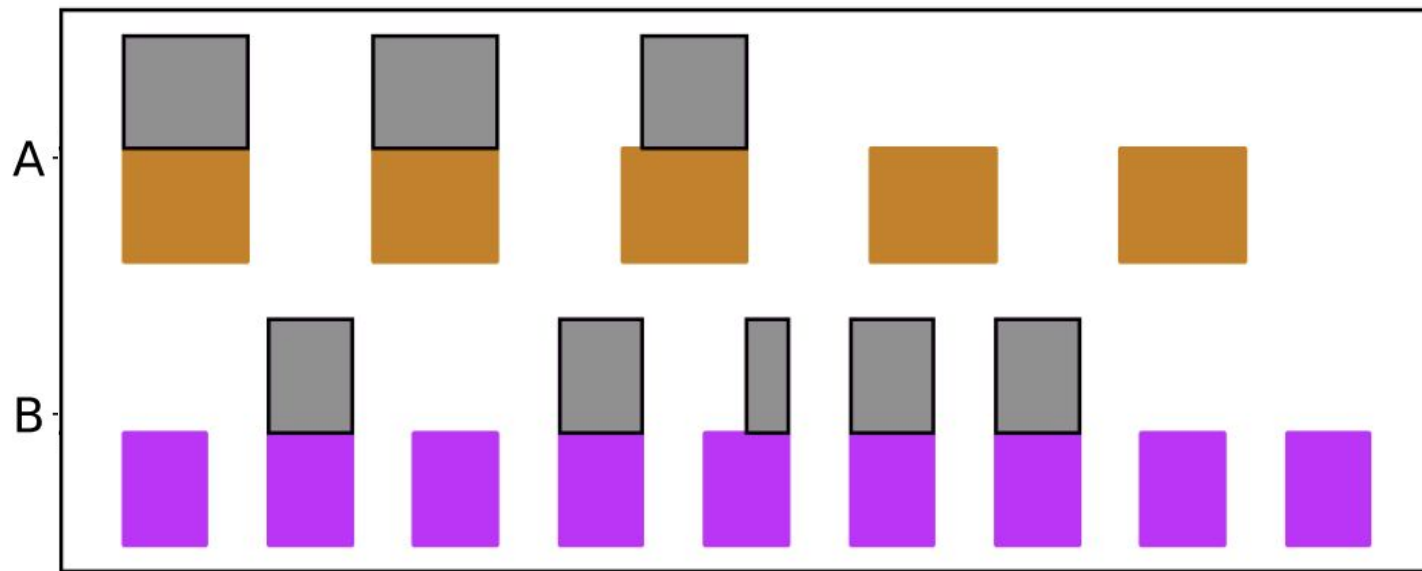
Both preferences expressed as Normalized Gaussians

Tenant **B** - **18 Minutes / hour**

$$\mu_A = 3 \quad \mu_B = 18$$

$$\sigma_A = 1 \quad \sigma_B = 2$$

Resulting Schedule

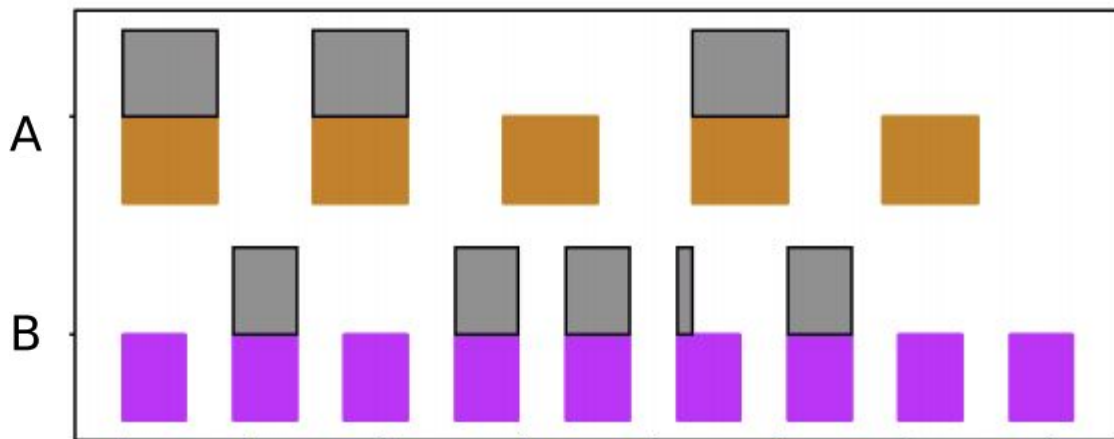
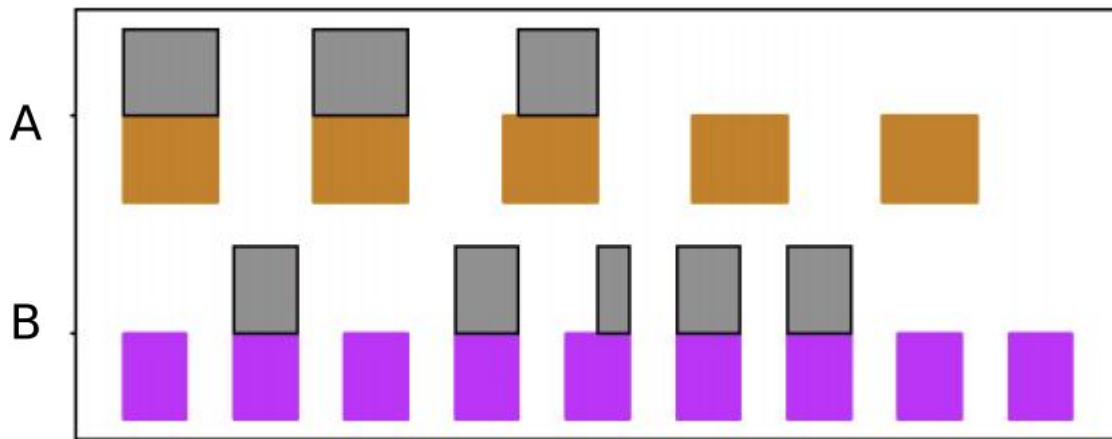


Results

A was satisfied but prefers the full 6 minute contact, and thus adds:

Logistic preference on Contact Duration with

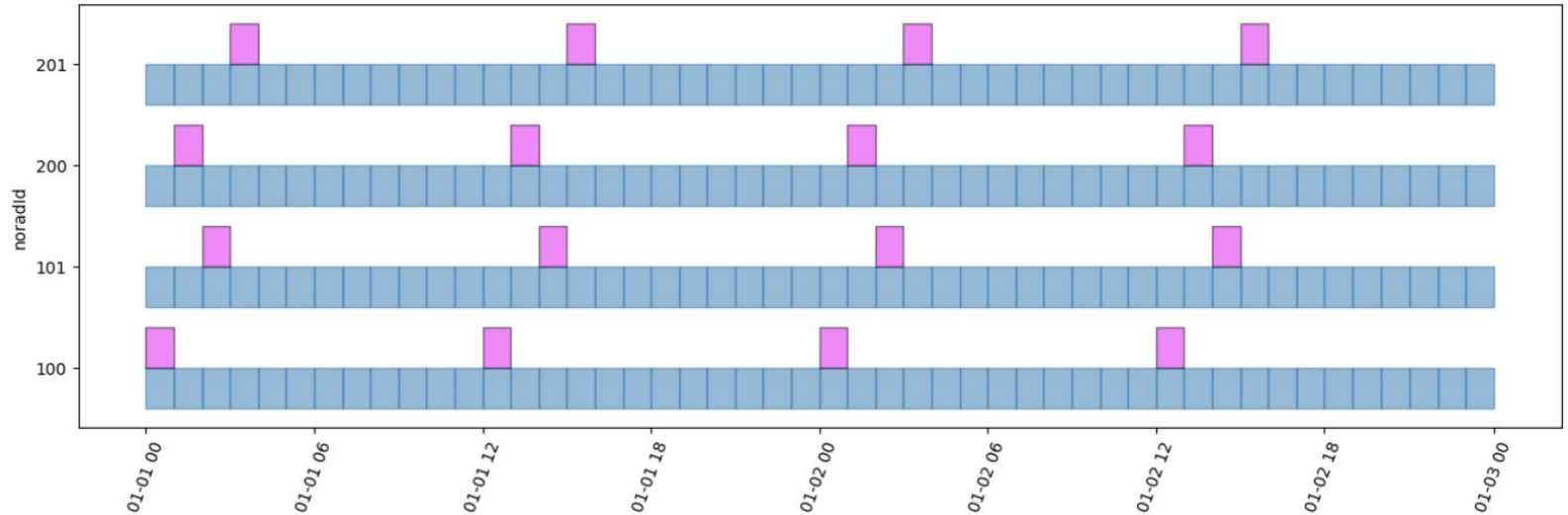
$\lambda = 5.5$ and $x_0 = 0.1$



Toy Example with high availability of Visibilities

Two users, two satellites each (100, 101 and 200, 201)

Each wants 2 contacts per day and 12 hours between contact



Conclusion

- Multi-tenant systems demand deconfliction
- Our approach will find deconflicted solutions that are optimal given utility functions
- Anecdotal evidence that our Utility functions robustly describe user needs
- Our prototype can rapidly revise solutions, making it ideal for simulation or what-if scenarios

Future Work

- Inferring User Preferences based on historical schedules
- Expanding on Objectives to meet additional requirements
- Automation of insights
- Federation of commercial and private networks

Q&A

kyle@dataskeptic.com

bbode@atlasground.com

alex@dataskept.com