15.094 Homework 2

Due: 03/17/2021

1 RO and Deterministic Optimization

1.1

Let

$$h(z) = \sum_{i=1}^{r} \frac{\alpha_i}{p_i} |(d^i)^T z - \beta_i|^{p_i},$$

where $\alpha_i > 0$ and $p_i > 1$, for all i.

Derive the robust counterpart of the constraint

$$(\overline{a} + Pz)^T x \leq b, \quad \forall z \in \mathcal{Z} = \{z : h(z) \leq \Gamma\}.$$

(Hint: You may use the results from the textbook without proof).

Solution:

The main ingredients of this problem are Theorem 2.1, result (2.45), and Table 2.2 from the textbook. Based upon these results, this problem boils down to simply reshaping conjugate of a function. The notations used in this solution are directly connected to the notations in the textbook, specifically the notation from the result (2.45). We can see that $h(z) = g(D^T z) = \sum_{i=1}^{r} g_i((d^i)^T z)$, where $g_i(t) = \frac{\alpha_i}{p_i}|t - \beta_i|^{p_i} - \frac{\Gamma}{r}$. Now using the result from Theorem 2.1, we can see that if $\tilde{f} = \alpha f(x - \beta) - \gamma$, then $\tilde{f}^*(y) = \alpha f^*(\frac{y}{\alpha}) + \beta^T y + \gamma$. For more detailed step-by-step derivation, you can see that

$$\alpha f \to \alpha f^*(\frac{y}{\alpha})$$

$$\alpha f(x - \beta) \to \alpha f^*(\frac{y}{\alpha}) + \beta^T y$$

$$\alpha f(x - \beta) - \gamma \to \alpha f^*(\frac{y}{\alpha}) + \beta^T y + \gamma$$

In Table 2.2 you can find the conjugate of the function $\frac{|t|^p}{p}$. Now starting from this conjugate, and plugging in the appropriate terms will give you the following result:

$$\overline{a}^T x + u \sum_{i=1}^{r} \frac{\alpha_i}{q_i} \left| \frac{z_i}{\alpha_i u} \right|^{q_i} + \boldsymbol{\beta}^T z + \Gamma u \leq b,$$
$$Dz = P^T x,$$
$$u \geq 0,$$

1.2

The so-called entropy uncertainty set is defined as

$$\mathcal{Z} = \left\{ z \in \mathbb{R}^L \;\middle|\; \|z\|_\infty \leq 1,\ \sum_{i=1}^{L} \{(1+z_i)\log(1+z_i) + (1-z_i)\log(1-z_i)\} \leq \beta \right\},$$

where log denotes the natural logarithm or logarithm in base e.

Derive the robust counterpart of the constraint

$$(\overline{a} + Pz)^T x \leq b, \quad \forall z \in \mathcal{Z}$$

Solution:

The solution goes exactly the same with 1.1, but this time with intersection and using the separable case from Lemma 2.1. You can define $\mathcal{Z}_1 = \{z \mid \|z\|_\infty \leq 1\}$, and $\mathcal{Z}_2 = \{z \mid \sum_{\ell=1}^{L} \{(1+z_\ell)\log(1+z_\ell) + (1-z_\ell)\log(1-z_\ell)\} \leq \beta\}$. Here we can define $h = h_1 + h_2$, where $h_1(z_l) = (1+z_\ell)\log(1+z_\ell)$ and $h_2(z_l) = (1-z_\ell)\log(1-z_\ell) - \frac{\beta}{L}$. Starting from the conjugate of $t \log t$, we can again simply plug in the right terms. This will give you the following result:

$$\overline{a}^T x + \|w\|_1 + \sum_{i=1}^{L} \left[t_i - s_i + u[e^{s_i/u - 1} + e^{-t_i/u - 1}] \right] + \beta u \leq b,$$
$$w + s + t = P^T x,$$
$$u \geq 0,$$

2 Cloud computing system

(Please use the code provided in `HW2_CloudComputing.jl/ipynb` for this question. Data generation functions, cost vectors and plotting tools are provided.)

Consider a cloud computing system over a finite time horizon $t \in \{1, \ldots, T\}$, with a set of servers, $s \in \{1, \ldots, S\}$. To lower verbosity, we will use S interchangeably to refer to set $\{1, \ldots, S\}$. Each server $s \in S$ has a finite amount of resources indexed by $i \in \{1, \ldots, I\}$.

$r_{i,s} \in \mathbb{R}^+,\ \forall\ i \in I,\ s \in S$ is the fixed capacity of resource i at server s. It costs a constant F_i to maintain one unit of resource i at server s in each time

period. The optimization problem is to find the optimal value of fixed resources $r_{i,s}$ subject to uncertain demand.

The demand is represented in time period t with $D_{i,s,t}$ jobs arriving at the facility, requesting resource i from server s to complete. We must never fail to satisfy demand, since folks need their Netflix! There are three ways to complete each job:

- Serve $D_{i,s,t}$ in the current server using fixed capacity $r_{i,s}$.
- Serve $D_{i,s,t}$ in the current server by temporarily expanding $r_{i,s}$ of server s in time t by $e_{i,s,t}$, while incurring an additional cost of C_i per unit.
- Transfer some or part of the job ($u_{i,s_1,s_2,t}$) from server s_1 to server s_2 with spare capacity in resource i, costing V_i for a transfer from server s_1 to s_2.

There are some additional constraints on the system. Expansions $e_{i,s,t}$ result in increases in the temperature of each server. Our temperature surrogate is $h_{s,t}$, which is the three time step moving average of expansions in all resources (moving average of $\sum_{i \in I} e_{i,s,t}$) for each server s at each time t. $h_{s,t}$ cannot exceed 1. Assume that the temperature of each server is zero at all time periods before $t = 1$.

To keep it simple, we will assume that servers may complete fractional jobs, and capacities of resources are continuously variable.

2.1

Please formulate the nominal linear optimization problem.

Solution:
To be able to formulate, we first write out our free variables:

$$r_{i,s},\ e_{i,s,t} \geq 0,\ \forall\ i \in I,\ s \in S,\ t \in T$$
$$u_{i,s_1,s_2,t} \geq 0,\ \forall\ i \in I,\ s_1 \in S,\ s_2 \in S,\ t \in T$$

In this case, to keep it simple, we have a variable for $u_{i,s_1,s_2,t}$ as well as $u_{i,s_2,s_1,t}$, even though theoretically only one of these is required. Both formulations are valid, but this one is simpler to express.

To formulate the objective function, we recognize that there are three sources of cost, which are the fixed, expansion and reallocation costs. The objective is simply the sum of all free variables over all indices times their associated cost coefficients. Note that the fixed cost is multiplied by T since $r_{i,s}$ is the cost of the fixed capacity in one time period.

$$\text{minimize } T \times \sum_{i \in I,\ s \in S} F_i r_{i,s} + \sum_{i \in I,\ s \in S,\ t \in T} V_i e_{i,s,t} + \sum_{i \in I,\ s_1 \in S,\ s_2 \in S,\ t \in T} C_i u_{i,s_1,s_2,t}$$

There are two kinds of constraints on the problem. The first is a simple demand satisfaction constraint:

$$D_{i,s,t} \leq r_{i,s} + e_{i,s,t} + \sum_{\alpha \in S} u_{i,s,\alpha,t} - u_{i,\alpha,s,t},\ \forall\ i \in I,\ s \in S,\ t \in T$$

where we must satisfy the demand in one of three ways. The second is the temperature constraint, where we have to write a three-time-period moving average. This is done as follows, given that we know the temperature of the system is zero at all $t < 1$:

$$\begin{aligned}
h_{s,1} &= \frac{1}{3} \sum_{i \in I} e_{i,s,1}, \forall s \in S \\
h_{s,2} &= \frac{1}{3} \sum_{i \in I} e_{i,s,2} + \frac{1}{3} \sum_{i \in I} e_{i,s,1}, \forall s \in S \\
h_{s,t} &= \frac{1}{3} \sum_{i \in I} e_{i,s,t} + \frac{1}{3} \sum_{i \in I} e_{i,s,t-1} + \frac{1}{3} \sum_{i \in I} e_{i,s,t-2}, \forall s \in S, t = 3, \ldots, T
\end{aligned}$$

The final formulation is the following:

$$\begin{aligned}
\text{minimize } & T \times \sum_{i \in I,\ s \in S} F_i r_{i,s} + \sum_{i \in I,\ s \in S,\ t \in T} V_i e_{i,s,t} \\
& + \sum_{i \in I,\ s_1 \in S,\ s_2 \in S,\ t \in T} C_i u_{i,s_1,s_2,t} \\
\text{s.t. } & D_{i,s,t} \leq r_{i,s} + e_{i,s,t} + \sum_{\alpha \in S} u_{i,s,\alpha,t} - u_{i,\alpha,s,t},\ \forall\ i \in I,\ s \in S,\ t \in T \\
& h_{s,1} = \frac{1}{3} \sum_{i \in I} e_{i,s,1}, \forall\ s \in S \\
& h_{s,2} = \frac{1}{3} \sum_{i \in I} e_{i,s,2} + \frac{1}{3} \sum_{i \in I} e_{i,s,1}, \forall\ s \in S \\
& h_{s,t} = \frac{1}{3} \sum_{i \in I} e_{i,s,t} + \frac{1}{3} \sum_{i \in I} e_{i,s,t-1} + \frac{1}{3} \sum_{i \in I} e_{i,s,t-2}, \forall\ s \in S, t = 3, \ldots, T \\
& h_{s,t} \leq 1,\ \forall\ s \in S,\ t \in T \\
& r_{i,s},\ e_{i,s,t} \geq 0,\ \forall\ i \in I,\ s \in S,\ t \in T \\
& u_{i,s_1,s_2,t} \geq 0,\ \forall\ i \in I,\ s_1 \in S,\ s_2 \in S,\ t \in T
\end{aligned}$$

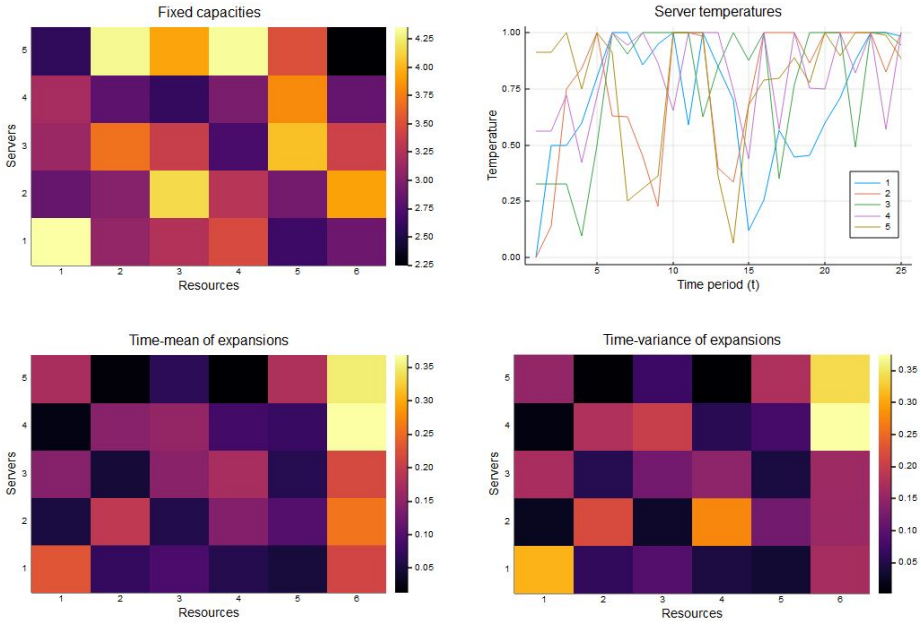
2.2

Use JuMP/JuMPeR to solve the nominal capacity allocation problem, using the `generate_data` function provided, with $I = 6$ resources, $S = 5$ servers,

over $T = 25$ periods. Please include the plot for matrix r (your fixed capacity allocation), your optimal cost, the time-statistics plots of expansions e, as well as the temperature plot. (Please use the plotting functions provided for your convenience. There should be four plots total.) Examine the temperature plot. Does your system anticipate peaks? How can you tell?

Solution: Please refer to `HW2_CloudComputing_Solution.ipynb` to generate the plots.

For our attempt, the optimal cost was 3411. The plots are as follows:



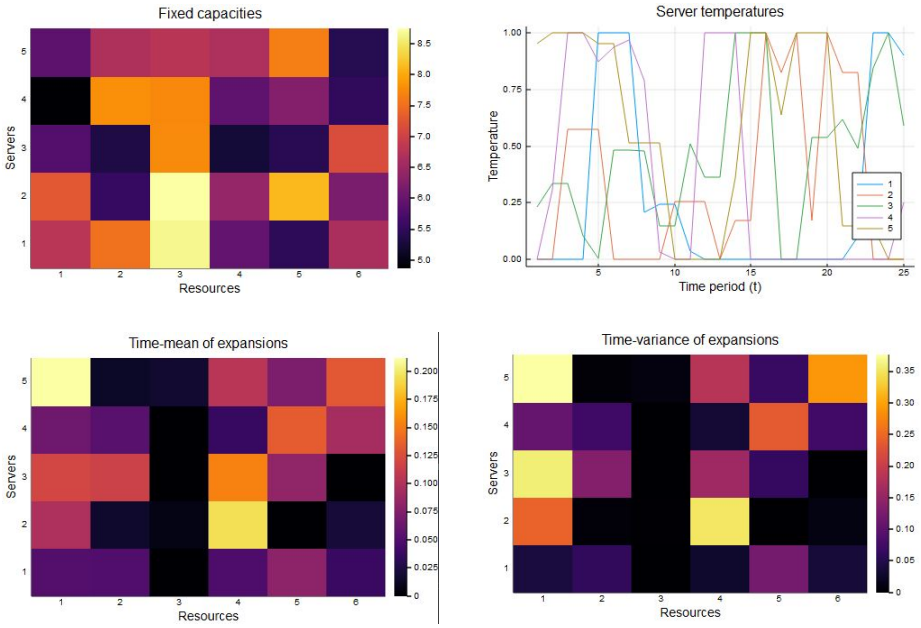
As far as the temperature plots, it is clear that the system can anticipate peak demands. The server temperatures go down (often together) ahead of periods of high demand. It's clear that there is simultaneous optimization of system temperatures to buffer the peaks.

2.3

Solve the nominal problem, but without the ability to transfer jobs ($u_{i,s,t} = 0, \ \forall \ i, s, t$). Provide the same four plots as above. How does this affect the cost and the allocation of r and e? What is the intuition behind it? As far as the system temperature, can the system anticipate peak demand? How is the system temperature response different compared to the problem with transfers?

Solution: Our new cost is 5806, which is %70 higher than the system with transfers. The plots are as follows:

The allocation of r increases dramatically, while the expansions tend to decrease, with large swaths of servers not ever expanding capacity. This is because



servers can only expand to buffer their own demand, which is oftentimes limited. The servers end up sitting idle a lot. The result is that server temperatures are down dramatically. The system can no longer anticipate peaks, because the system no longer has the ability to diffuse the demand among the servers.

2.4

Write the robust version of the capacity allocation problem by adding uncertainty d to demands $(D_{i,s,t} + d_{i,s,t})$ in your optimization problem. The uncertainty in demand is decoupled for each server; assume that uncertain variables $d_{1:I,s,1:T}$ for each server s lie in a budget uncertainty with bounded support $[-1, 1]$ so that we are guaranteed to be feasible for %95 of uncertain outcomes. Please provide the form of your uncertainty sets as well as computations for the safety factors.

Solution: The robust formulation is the same as in 2.1 with an additional uncertain term d next to each D. For the uncertainty sets, there should be $S = 5$ sets each defined as follows:

$$||d_{i\in I,s,t\in T}||_1 \leq \Gamma, \forall s \in S$$
$$|d_{i,s,t}| \leq 1, \forall\ i \in I,\ s_1 \in S,\ s_2 \in S,\ t \in T$$

In this case, all servers have the same Γ, because we ask the same probabilistic guarantees from each. (But do note that the worst case outcomes will be different for each server as was the intent.)

The computation of Γ relies on the probabilistic assumptions from Lecture 4, especially the one for the budget uncertainty set. We get

$$\Gamma = (2\ln(\frac{1}{\epsilon}))^{0.5} L^{0.5} = (2\ln(\frac{1}{0.05}))^{0.5} (T \times I)^{0.5} = 2.448 \times 12.25 \approx 30$$

2.5

Solve the RO problem in JuMPeR/JuMP with and without job transfers. (The robust counterpart will take a lot longer to solve.) Please provide plots of r and the temperature, as well as the costs. (There should be four plots). How do your r, cost and the temperature plots change? How are the robust solutions similar or different? What do the results say about the ability of server farms to buffer variable demand?

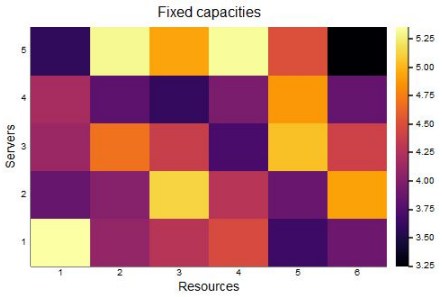
Solution:

The cost of the robust server farm with job transfers is 4286. This is an increase in cost of %25, while making us immune to all outcomes from the budget uncertainty set we constructed.

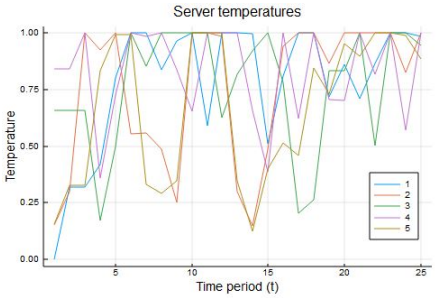
The robust solution without job transfers costs 6681, which is only a 15% increase. Why might this be? This can be linked to the idea that risk is rewarded! The collaborative server farm is running with fewer resources (i.e. leaner) which makes it incur a higher cost for robustness. But even so, the robust server farm with job transfers is 36% cheaper than the one without, for the same level of robustness!

Note that the temperatures for the robust solution with job transfers are somewhat higher, while there is almost no change in temperatures for the robust solution with no job transfers. There are two reasons for this. The worst case outcomes of the uncertainty are interdependent in the collaborative server setting, since one server can assist another in times of peak demand. In the separated server setting, the worst case temperatures end up being the same as in the nominal case, since the uncertainty is always incurred in the peak demands of the nominal case.

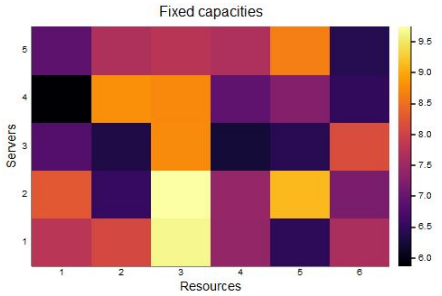
In the end, the robustification strategy for "lone" servers is to just add more capacity from the get-go, whereas collaborative server farms can buffer their joint uncertainty better by transferring jobs, squeezing even more performance from the job transfers at the same temperature limit.



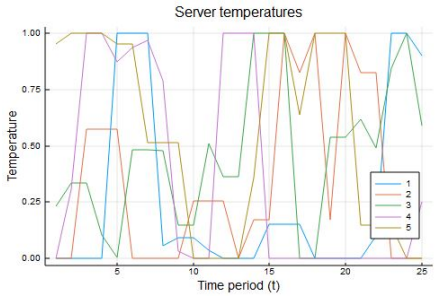
(a) Robust with transfers, fixed capacities.



(b) Robust with transfers, temperatures.



(c) Robust without transfers, fixed capacities.



(d) Robust without transfers, temperatures.