1 Repository

The repository Julia contains four main files:

- HM.jl: implementation of the Hotelling model.
- LDM.jl: implementation of the general affine demand model.
- utilities.jl: contains a series of methods to compute demands, virtual costs, and check assumptions.
- exmaples.jl: tests solutions with close forms to contrast with the results from optimization.

In what follows I describe some details that are relevant in each file.

2 HM.jl

This file implements the Hotelling demand model. Given $\Theta = (\theta_1, \ldots, \theta_m)$ and n suppliers, the structure of the variables is

$$x(\theta) = \left[x_1(\theta_1), \ldots, x_n(\theta_1), x_1(\theta_2), \ldots, x_n(\theta_2), \ldots, x_1(\theta_m), \ldots, x_n(\theta_m)\right].$$

The same holds for the other variables, i.e. t, p, and also for the dual variables u and v.

The formulation of the centralized problem follows directly from the paper, considering as objective function

$$\max \mathbb{E}_\theta \left[\sum_{i=1}^{n} -\frac{\delta}{2} \left[\left(l_i - \sum_{j=1}^{i-1} x_j(\theta) \right)^2 + \left(\sum_{j=1}^{i} x_j(\theta) - l_i \right)^2 \right] \right]$$

In the case with two suppliers in the extremes of [0,1] this reduces to

$$\max \mathbb{E}_\theta \left[-\frac{\delta}{2} \left[x_1(\theta)^2 + x_2(\theta)^2 \right] - t_1(\theta) - t_2(\theta) \right]$$

To implement the decentralized problem, we incorporate the KKT conditions as additional constraints to the centralized problem. In the two-supplier case these are: for each $\theta \in \Theta$ and $i \in \{1, 2\}$

$$\begin{aligned} p_i(\theta) + \delta x_i(\theta) - u_i(\theta) + v(\theta) &= 0 \\ x_i(\theta) \cdot u_i(\theta) &= 0 \\ u_i(\theta) &\geq 0 \\ t_i(\theta) - x_i(\theta) \cdot p_i(\theta) &= 0 \end{aligned}$$

Primal feasibility constraints are omitted because they are part of the centralized problem.

This file has the following methods:

- GenerateInputs: returns the matrices and vectors to construct the constraints and objective of the optimization problem. Some special elements of this method are:
  - line 12: generates the set of types Θ, i.e. all possible combinations of types of suppliers.
  - line 27-34: represents computation of $f_{-i}(\theta_{-i})$.

  The outputs of this method are:
  - nsupp: number of suppliers
  - ntypes: number of types for each supplier;
  - nvars: length of each vector of variables, i.e. $n \times m$
  - sts: number of scenarios, i.e. $|\Theta|$
  - bGx: matrix multiplying x in the inequality IC and IR constraints. This matrix is organized as follows:
    - The first nsupp $\times$ ntypes rows are IR constraints, ordered lexicographically by (i, θ_i) in increasing order, where i is the supplier and θ_i is his type. For example, in a 2 supplier-2 types example, the first row corresponds to supplier 1 with low type, then supplier 1 with high type, then supplier 2 with low type, and finally supplier 2 with high type.
    - The last nsupp $\times$ ntypes $\times$ (ntypes-1) rows correspond to IC constraints, ordered lexicographically by (i, θ_i, θ'_i) in increasing order.
  - bGt: matrix multiplying t in the inequality IC and IR constraints
  - bh: right-hand side of IR-IC constraints (equal to a vector of zeros)
  - bA: matrix multiplying x in the feasibility constraints
  - bb: right-hand side of feasibility constraints (equal to a vector of ones)
  - wqt: vector with the probabilities of each scenario in Θ to compute the expected value for objective function
- CheckFeasibility: receives the inputs of the problem and solutions x_0, t_0 and checks whether this solution is feasible for the decentralized problem. To accomplish this, I fix $x = x_0, t = t_0$ and solve the decentralized problem with objective value equal to 1.
- SolveOptimization: formulates and solves the optimization problem using as inputs the parameters of the problem and version, which is equal to centralized or decentralized.
- SimulateOptimization: solve recursively the optimization problem for different δ.

3 LDM.jl

This file implements the general affine linear demand model. The structure of the variables is equivalent to the previous case. The centralized problem follows directly from the paper, while the decentralized problem is implemented by adding the following KKT conditions as constraints (in matrix form):

$$\begin{aligned} p(\theta) - c + Dx(\theta) - u(\theta) + v(\theta) &= 0 \\ x(\theta) \cdot u(\theta) &= 0 \\ u(\theta) &\geq 0 \\ t(\theta) - x(\theta) \cdot p(\theta) &= 0 \end{aligned}$$

The methods in this file are the same as in the previous case adapted to the general affine demand model.

4 utilities.jl

This file contains a series of methods that are used in the previous files. These methods are:

- combine and combwithrep: used to generate all potential profiles in Θ.
- ComputeCumulativeDistribution: for a given dictionary of marginal distributions, returns a disctionary of marginal cumulative distributions.
- virtualcost: computes, for a given supplier i and type θ, the virtual cost.
- demandLM: computes the demand faced by supplier i in the general affine model with two suppliers (obtained from close form solution).
- assortmentHM: returns the set of suppliers in the assortment of the Hotelling model for a given price and δ.
- demandHM: computes the demand faced by supplier i in the Hoteeling model with two suppliers.
- checkvcincreasing: checks whether the virtual costs are increasing for each supplier.
- checkconditionsHM: check whether the conditions in Theorem 4.1 are satisfied; returns a list with two booleans, one for each condition.
- checkconditionsLM: check whether the conditions in Theorem 5.2 are satisfied; returns a list with two booleans, one for each condition. To find d^*, we use the following heuristic:

1. Divide Θ in two subsets Θ^S and Θ^N, such that

$$\Theta^S = \{\theta \in \Theta : \ x_i(\theta) > 0, \ \forall i\}$$
$$\Theta^N = \{\theta \in \Theta : \ \exists i \ st. \ x_i(\theta) = 0\}$$

2. For each $\theta \in \Theta^S$, compute

$$d(\theta) = \max\left\{\left|v_i(\theta) - v_i(\theta')\right| : \theta' \in \Theta^N, \ i = 1, \dots, n\right\}$$

3. Define

$$d^* = \min\left\{d(\theta) : \ \theta \in \Theta^S\right\}$$

5 examples.jl

This file tests the methods described previously.