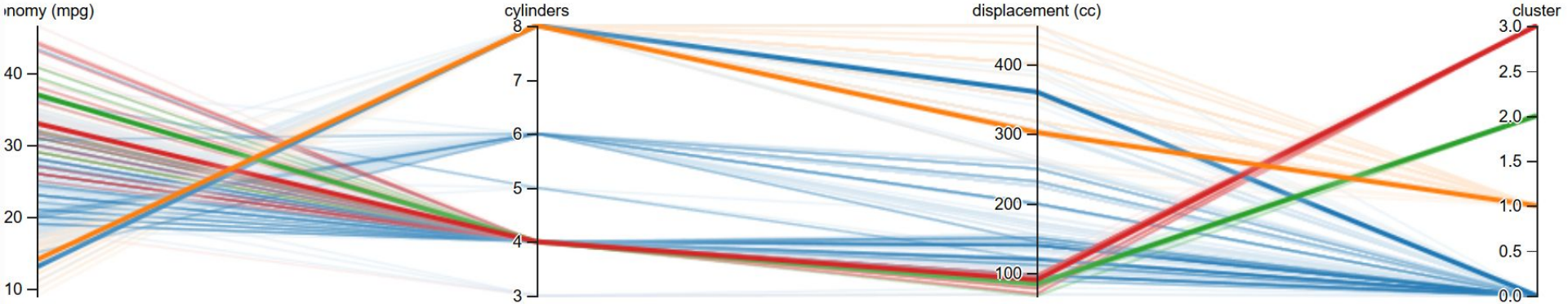
Interactive visualizations for multiobjective optimization problems



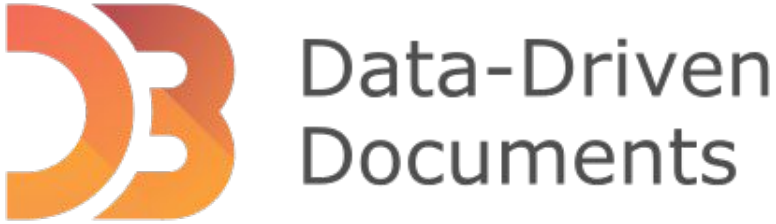
William Raseman[1], Josh Jacobson[2], and Prof. Joseph Kaspryzk[1]

1. Civil, Environmental & Architectural Engineering, University of Colorado Boulder
2. Applied Mathematics, University of Colorado Boulder

parasol: an open source interactive visualization JavaScript library for multi-objective decision making

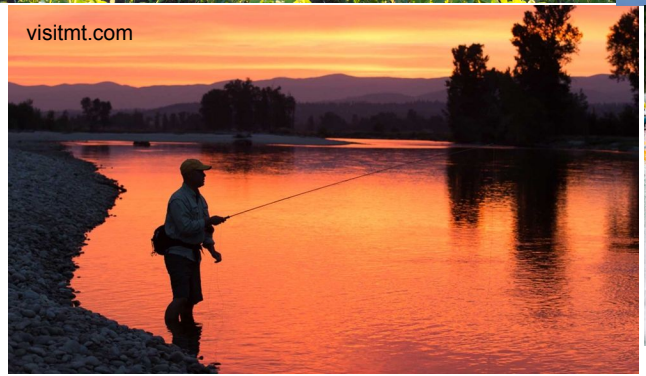
JavaScript Web Application

Interactive visualizations



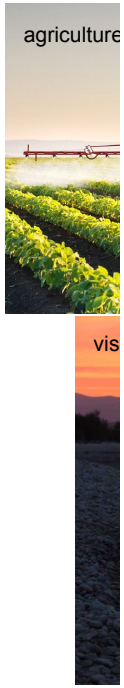
Multi-objective optimization is a popular way to explore tradeoffs in environmental management problems

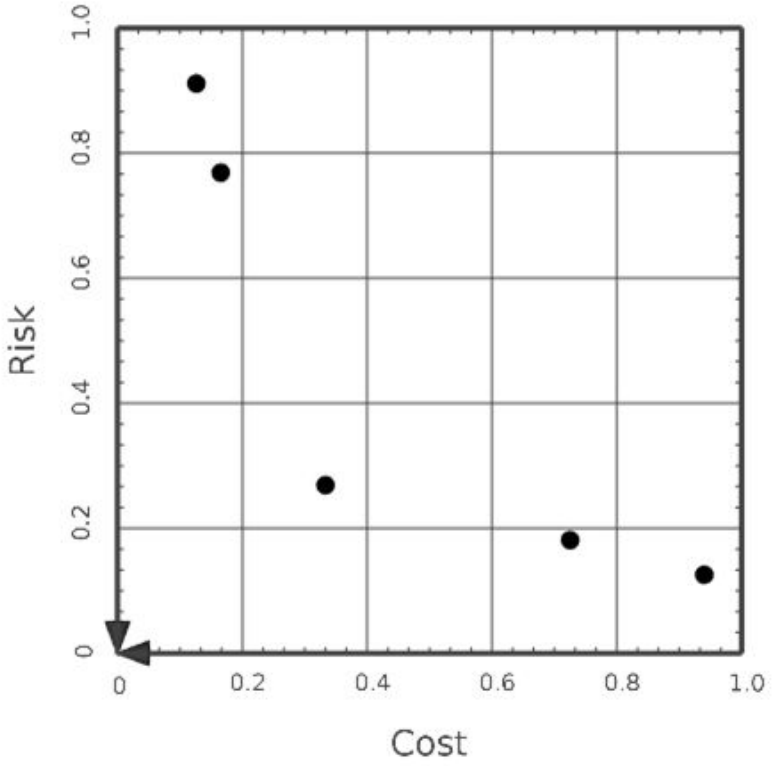
For example:
competing interests in water resources management

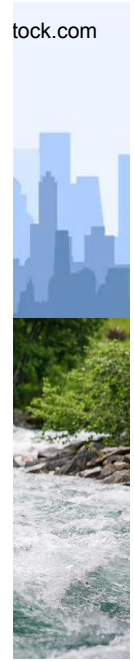


Multi-objective optimization is a popular way to explore tradeoffs in environmental management problems

For example:
competing interests in water resources management

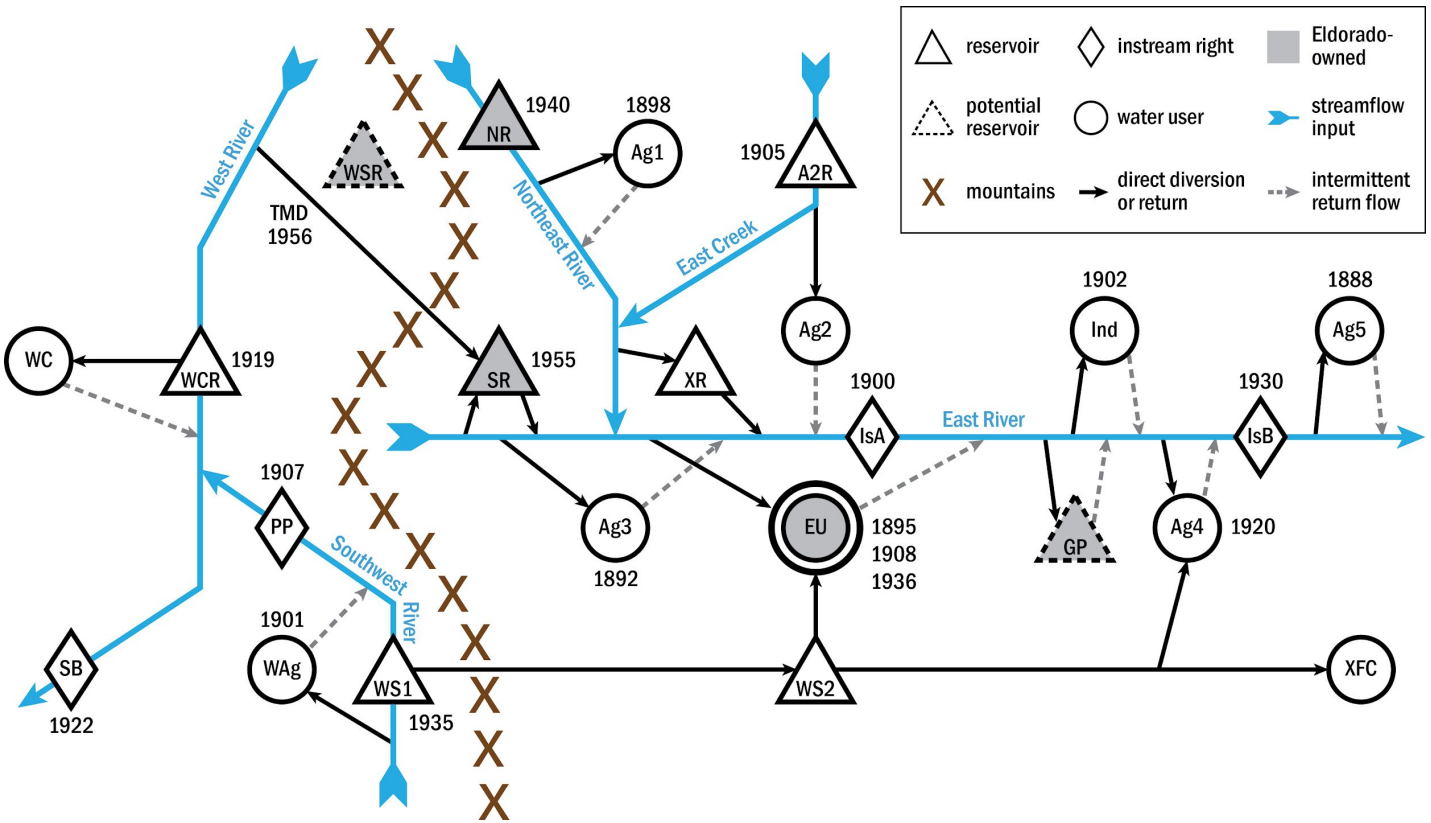






What is the best way to manage this system and prepare for drought?

Credit: Rebecca Smith



Decision Space

lease water

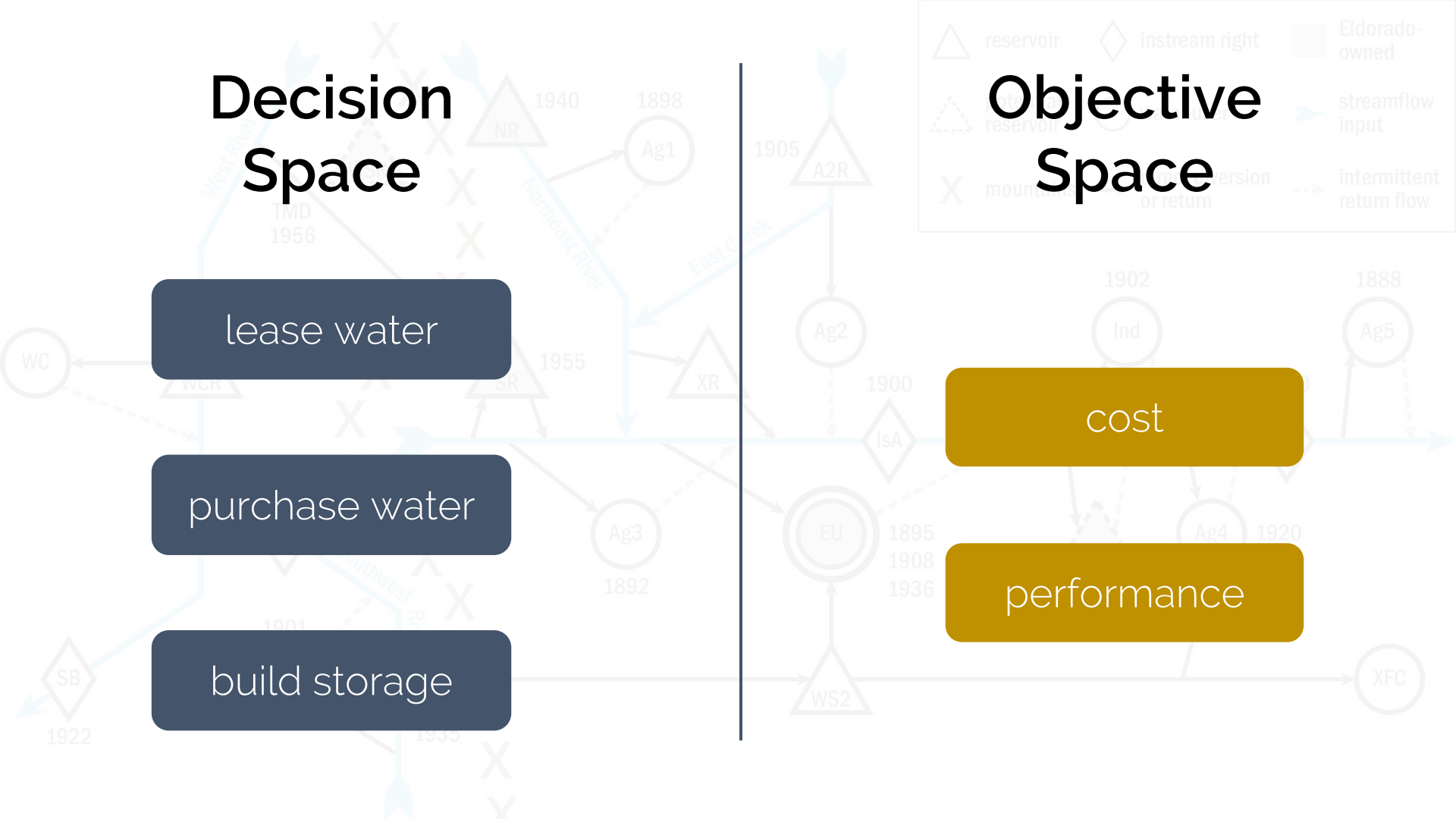
purchase water

build storage

Objective Space

cost

performance



...possible actions we could take found by an optimization algorithm

decisions							objectives			
water is leased			water is purchased long-term		new reservoir is constructed		minimize	minimize	minimize	maximize
lease_1	lease_2	lease_3	ag_rights	indust_rights	reservoir_1	reservoir_2	missed_opportunity	new_supply	new_storage	min_spring
5990	800	9600	0.01	0.11	6900	400	1057.075	20049.259	9500	-1.261
5890	6300	10000	0.05	0.17	10000	300	2051.91	23861.129	11200	-1.385
6000	5500	9900	0.06	0.14	9700	100	1662.625	22696.44	11100	-1.363
5930	9800	9800	0.01	0.15	7900	300	2261.214	21930.235	9500	-1.299
5980	2800	9800	0.01	0.12	7700	0	1317.336	20540.628	9500	-1.281
5990	800	9600	0.07	0.18	6900	400	1481.632	24845.76	9500	-1.311
5880	200	9600	0	0.11	7300	400	1104.785	19729.028	9900	-1.273
5930	7200	10000	0.01	0.12	7900	300	1874.251	20526.268	10700	-1.332
5820	1300	9600	0.04	0.17	4700	800	1565.252	23553.561	7400	-1.211
5880	200	9600	0.06	0.11	7300	400	998.983	21165.235	8200	-1.203
5930	7200	10000	0.01	0.18	9100	300	2196.304	23427.303	11100	-1.386
5930	2600	9800	0	0.11	9100	400	1363.54	19793.249	11500	-1.341
5780	3300	9900	0.02	0.17	7300	300	1777.684	23097.213	8400	-1.263
5920	3100	10000	0.01	0.11	8900	0	1328.71	20089.201	10700	-1.319

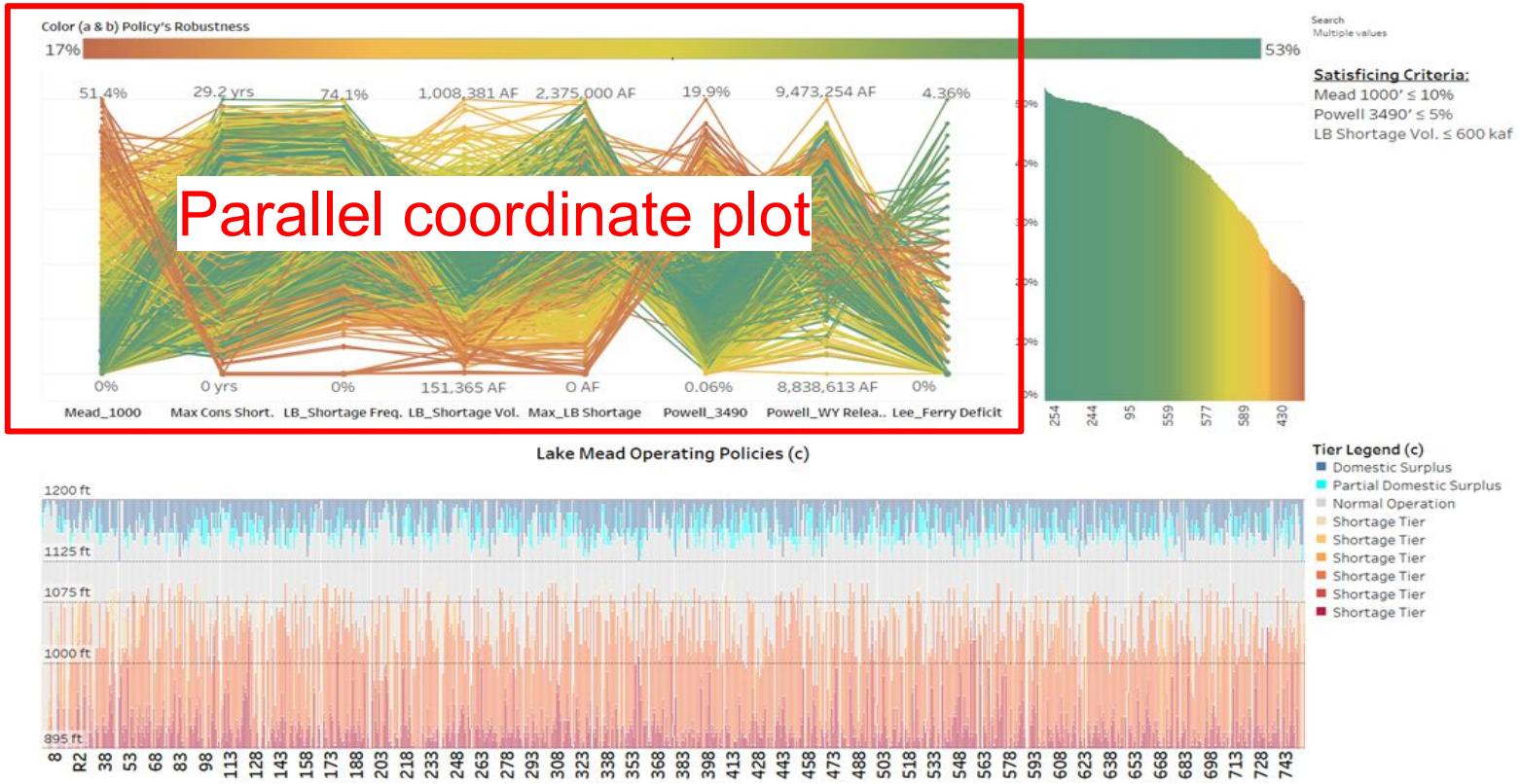
ooo

[plus a few thousand more solutions]

How should we present solution data such that the user can productively explore relevant solutions and make better informed decisions?

Multi-objective visual analytics facilitate insight discovery via an interactive, iterative experience

Credit: Elliot Alexander



Multi-objective visual analytics facilitate insight discovery via an interactive, iterative experience

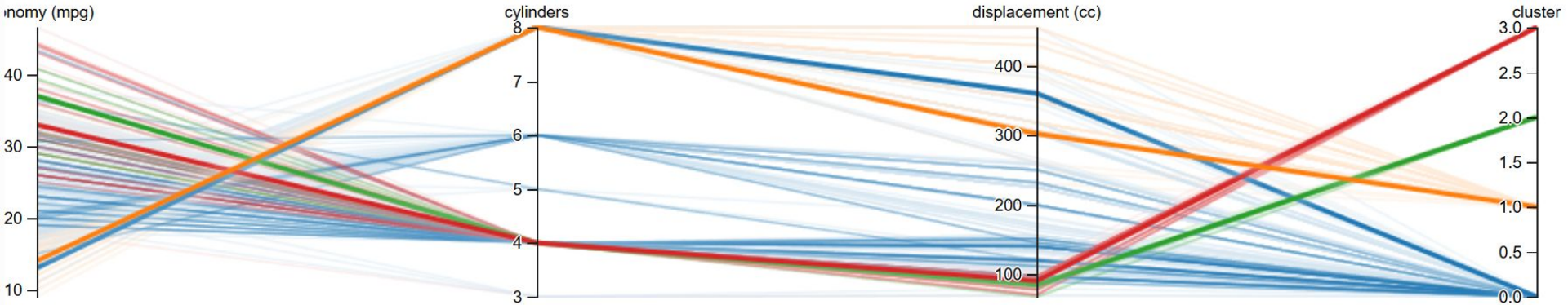
Credit: Elliot Alexander

Limitations with current software

- Limited support/flexibility for parallel coordinate plots
- Difficult to share with stakeholders
- High learning curve for developers

parasol library

The library developed in this work gives users the building blocks to easily create shareable, parallel coordinate plot web applications for multi-objective decision making



Library features

The *parasol* application programming interface (API) includes

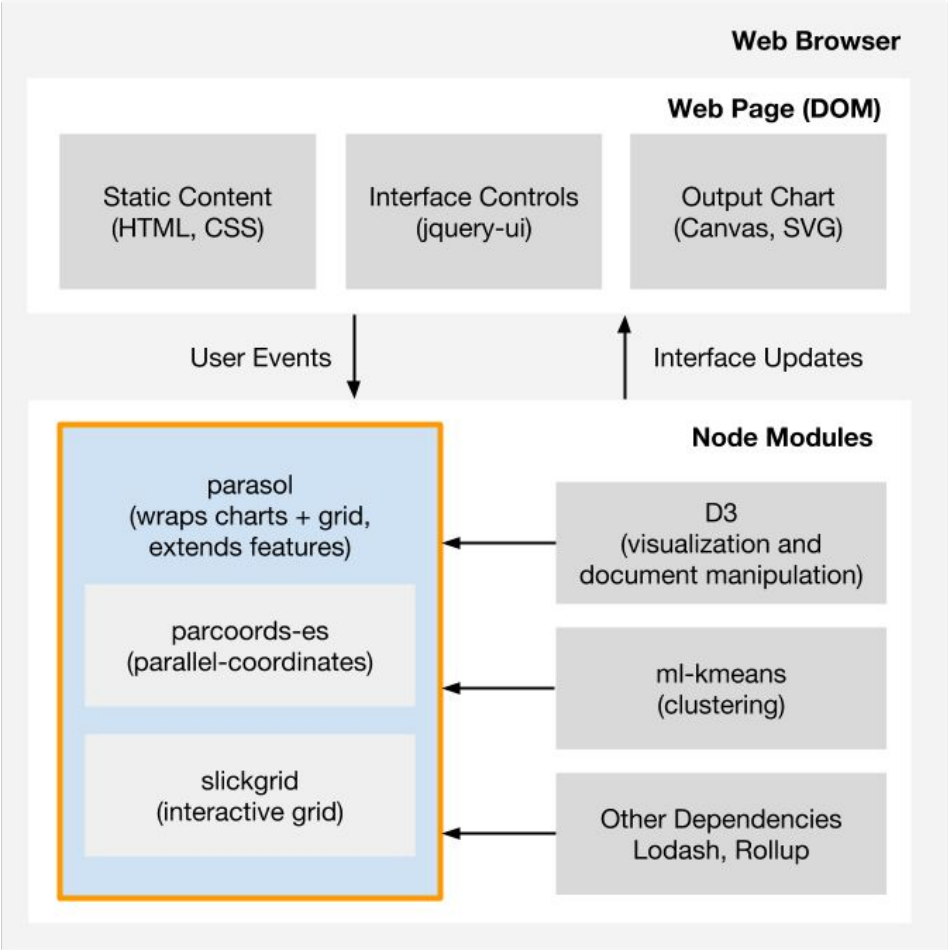
- lightning fast, interactive, customizable grid/spreadsheet
- linked charts
- objective weighting
- k-means clustering
- hide / show axes
- keep / remove / export data
- compatible with all features in *parcoords* library (brushing, marking, coloring etc.)

Software architecture

Learn more about:

Data Visualization in Web Browsers with JavaScript APIs
http://sched.co/EbOO

Thursday @ 3:00 PM



Usage

Module

1.Install library in your project

```
npm install parasol
```

2.Import module

```
import 'parasol/parcoords.css';
import Parasol from 'parasol';

const ps = Parasol(data)(selection)....
```

Standalone

```
<link rel="stylesheet" type="text/css" href="./parcoords.css">
<script src="./parasol.standalone.js"></script>

var ps = Parasol(data)("#example")....
```

HTML template script

```
<script>
d3.csv('filepath/to/data').then(function(data) {
  // tell Parasol which charts each variable should be included in
  config.partition = {...};              // {"variable name": [chart list]} pairs

```

```
<script>
d3.csv('filepath/to/data').then(function(data) {
  // tell Parasol which charts each variable should be included in
  config.partition = {...};              // {"variable name": [chart list]} pairs

  // initialize chart closures in all divs with class 'parcoords'
  var ps = Parasol(data, config)('.parcoords')
            .attachGrid()('#grid')       // place grid in div with id #grid
            .linked()                    // link all charts and grid by default
            .aggregate({...})            // {"variable name": weight value} pairs
            .cluster(3, colors='Dark2') // axis hidden by default

```

```
<script>
d3.csv('filepath/to/data').then(function(data) {
  // tell Parasol which charts each variable should be included in
  config.partition = {...};              // {"variable name": [chart list]} pairs

  // initialize chart closures in all divs with class 'parcoords'
  var ps = Parasol(data, config)('.parcoords')
             .attachGrid()('#grid')      // place grid in div with id #grid
             .linked()                   // link all charts and grid by default
             .aggregate({...})           // {"variable name": weight value} pairs
             .cluster(3, colors='Dark2') // axis hidden by default

  // polish charts with parcoords methods
  ps.charts.forEach(
    (pc) => {
      pc.alpha(0.4).shadows().reorderable().render();
    }
  )
});
</script>
```

Workflow demo

ParaSol.js

Links to info and tutorials: Example Example

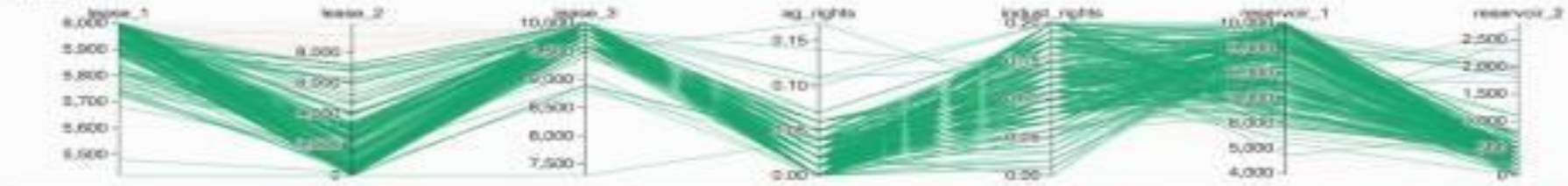
Keep Remove Export Reset Brushes Clear Selections Show Clusters Hide Clusters Set Axes Limits Explore Selection

Objective Space

Save Image

Decision Space

Save Image



lease_1	lease_2	lease_3	ag_rig...	indust_...	reservo...	reservo...	missed...	new_s...	new_st...	min_sp...	cluster
5990	800	9600	0.01	0.11	6900	400	1057.08	20049.26	9500	-1.26	0

Follow along

parasol is still in an early stage of development, so new features are being added on a regular basis. If you'd like to participate in development, or know of a feature that could be beneficial to include, let us know!

GitHub

https://github.com/joshhjacobson/parasol

Thanks for your attention!

Contact:

william.raseman@colorado.edu

josh.jacobson@colorado.edu

joseph.kasprzyk@colorado.edu

Helpful resources:

NPM: https://www.npmjs.com/

D3: https://d3js.org/