Paramotopy: Parameter Homotopy through Bertini

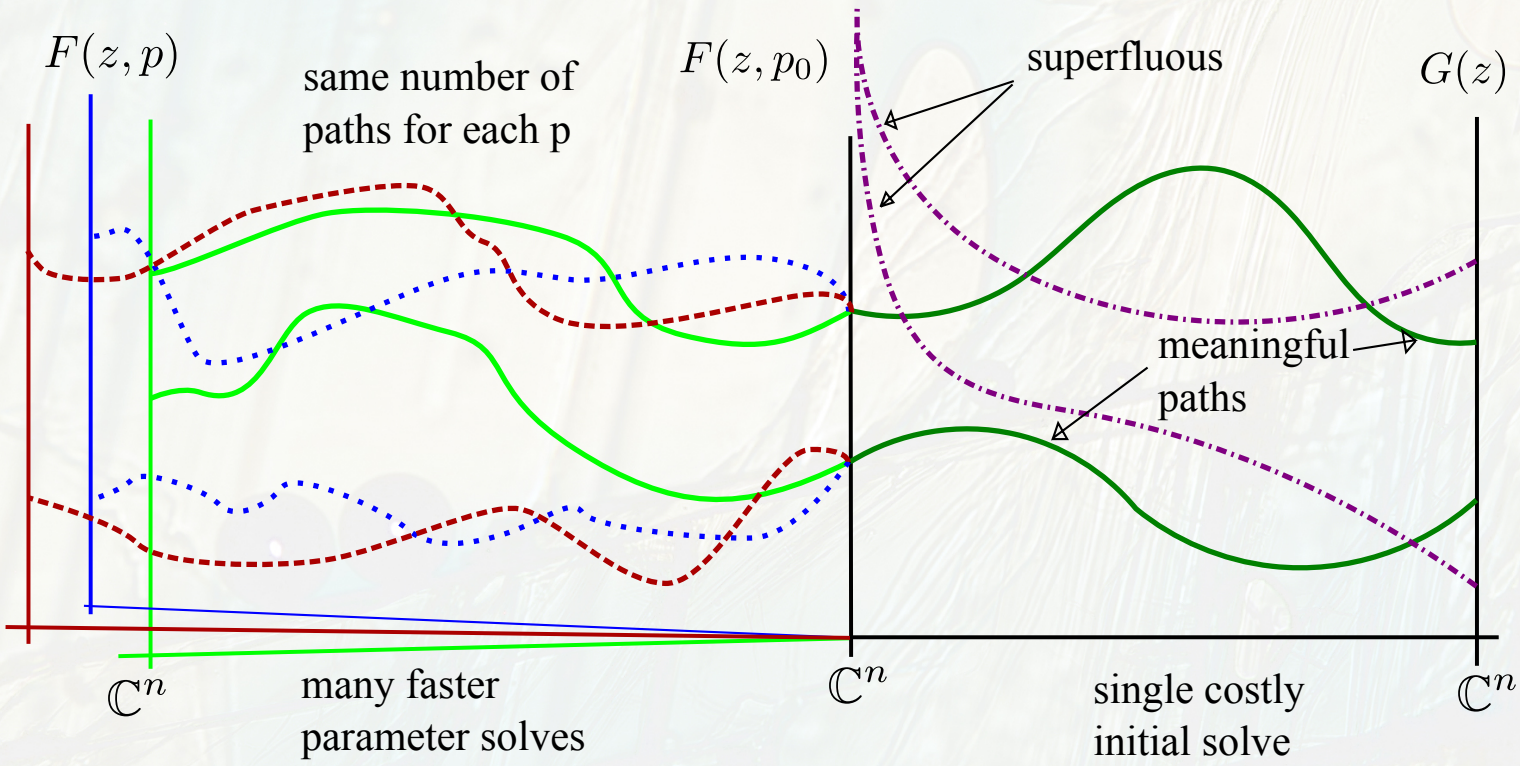
Daniel Brake and Matt Niemerg with Dan Bates

03 August, 2013

```
*******************************
 Welcome to Paramotopy.
     parametrized system analysis software by
     daniel brake and matthew niemerg, with dan bates.

     www.paramotopy.com     danielthebrake@gmail.com  matthew.niemerg@gmail.com
*************************
```

Parameter Homotopy



A reminder about the parameter homotopy.

Two parameter homotopies:

1. 'Cheater's Homotopy', which moves the entire function. [1]
2. 'Coefficient Parameter Homotopy', which moves only the coefficients under study. [2]

We practice the second.

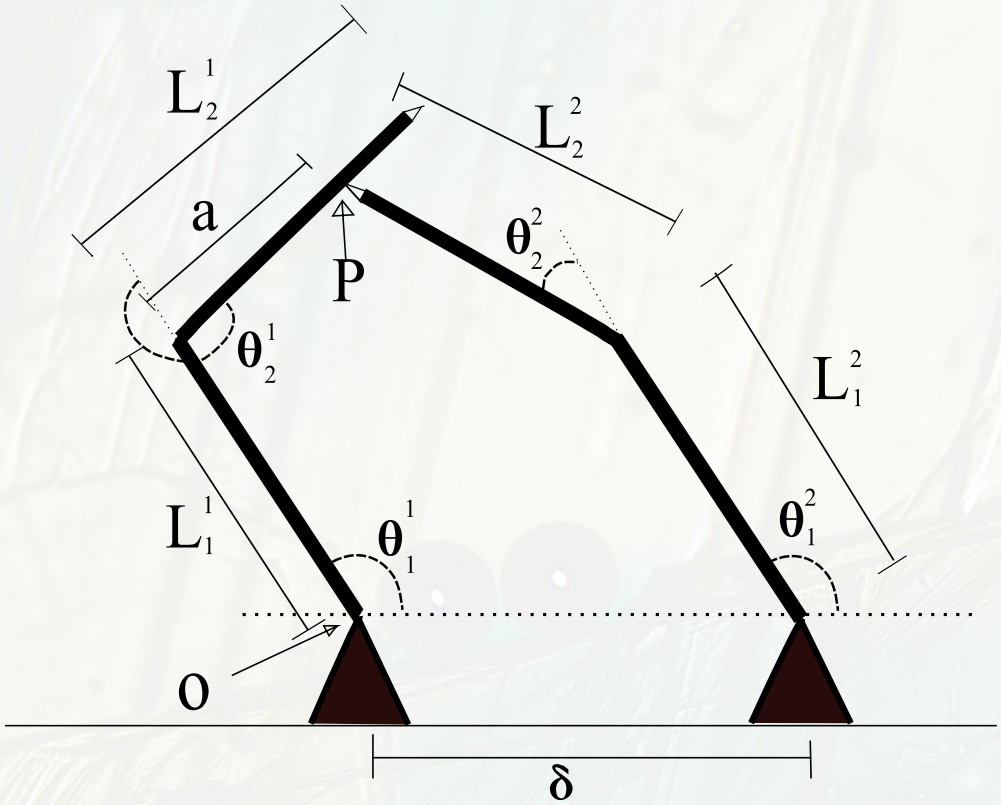
[1] Li, T., Sauer, T., and Yorke, J. "The Cheaters Homotopy: An Efficient Procedure for Solving Systems of Polynomial Equations." SIAM Journal on Numerical Analysis 26:5 (1989): 1241-1251.

[2] Morgan, Alexander P., and Andrew J. Sommese. "Coefficient-parameter polynomial continuation." Applied Mathematics and Computation 29.2 (1989): 123-160.

Paramotopy

- Command-line software which enables rapid investigations into parametrized polynomial systems.
- Uses *Bertini* to perform solves, under the hood.
- Numeric-menu driven interface which simplifies and automates Bertini functions.
- Open source, free. `www.paramotopy.com`

Motivation



2D Example of problem.

Optimize to determine a and δ:

- Robots separated by distance δ,
- Right robot (#2) grasps left (#1) at point P, at distance a from the second joint of robot 1,

so that some function of the post-failure workspace is maximized.

Demo

And now, an active demo.

The 'Alicia' system

$$
\begin{aligned}
x0 + x1 + x2 + y1 + y2 + y3 + y4 &= k1,\\
e + y1 + y2 &= k2,\\
f + y3 + y4 &= k3,\\
-a1 * x0 * e + b1 * y1 + c4 * y4 &= 0,\\
c2 * y2 - a3 * x2 * f + b3 * y3 &= 0,\\
a1 * x0 * e - (b1 + c1) * y1 &= 0,\\
a2 * x1 * e - (b2 + c2) * y2 &= 0,\\
a3 * x2 * f - (b3 + c3) * y3 &= 0,\\
a4 * x1 * f - (b4 + c4) * y4 &= 0.
\end{aligned}
$$

The system has 9 variables, 9 equations, and 15 parameters. The parameters in the problem are a_i, b_i, c_i, k_j, for $i \in \{1, 2, 3, 4\}$, and $j \in \{1, 2, 3\}$.

Starting Point

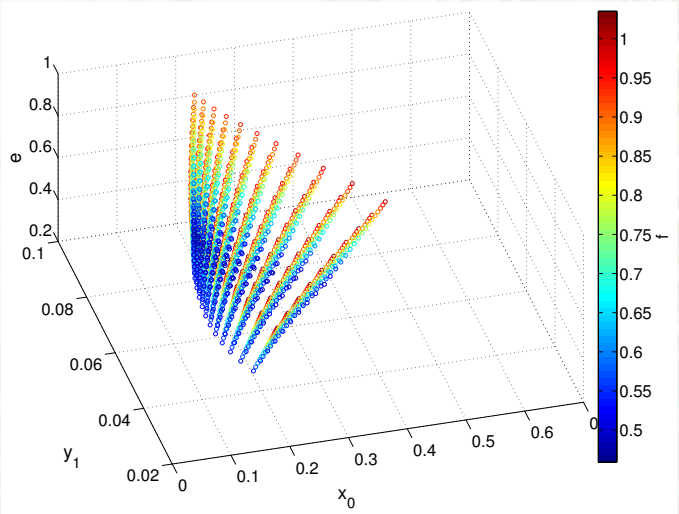
[The parameters] are all assumed to be positive. For some of them, the system has only one positive solution (i.e., all coordinates are real and positive) and for the others, it has three. How can numerical methods help us determining values, regions, etc., for this to happen?

Initial answer

To start answering questions about this system:

- Perform grid-sample solves across the unit cube $[0, 1]^{15}$, to find the number of solutions

  Found only points with a single positive real equilibrium.

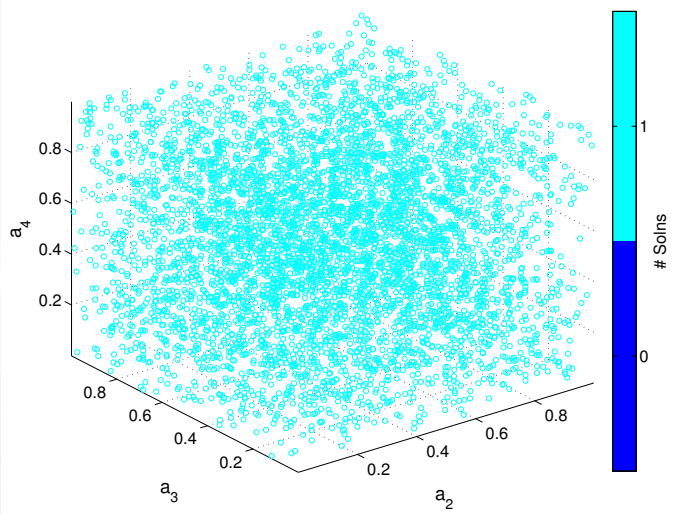


- Perform positive-dimensional numerical irreducible decomposition for random point in parameter space

  Only solutions are ‘isolated’ solutions – just points.

Monte Carlo sampling

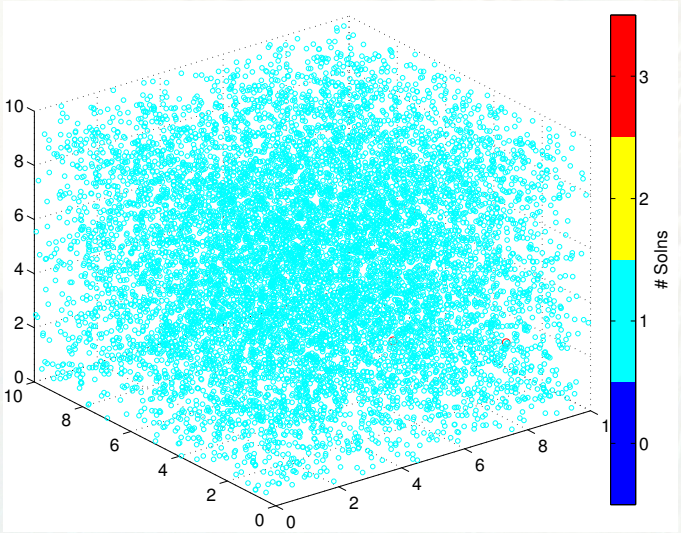
Grid-sampling the unit cube provided only single-equilibria points – maybe Monte Carlo sampling will help? Randomly sampling 10^5 times in all parameters yielded the same picture:



Monte carlo sampling of the unit cube seemed to lend itself to the conjecture that this portion of parameter space is devoid of points at which multiple positive equilibria exist. Perhaps the unit cube has only unique equilibria?

Beyond the Unit Cube

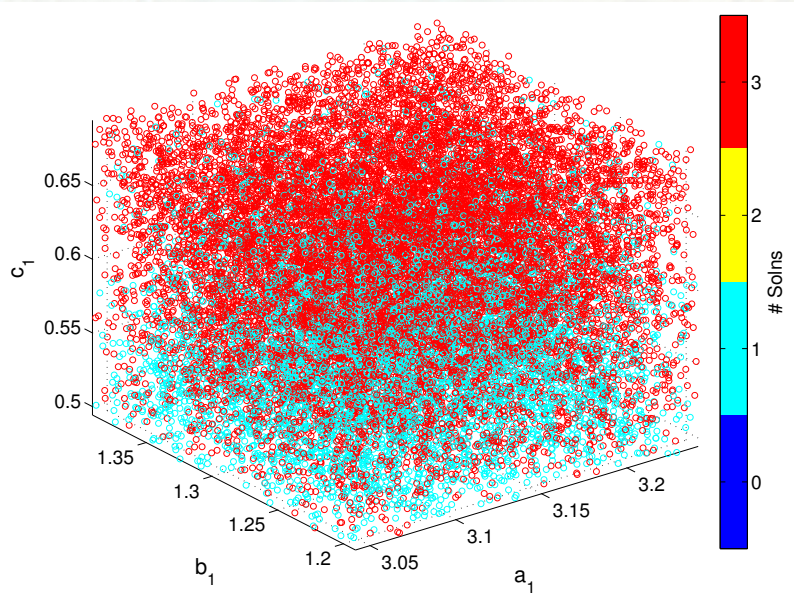
Let's move out a 10 units – the 10-cube, $[0, 10]^{15}$
Randomly sampling can be quite fruitful for system with large numbers of parameters, with estimates on numbers (such as volumes) converging as the square root.



My first pass at sampling, using only 10^4 points, produced *two* parameter points with non-unique equilibria. See the red points?

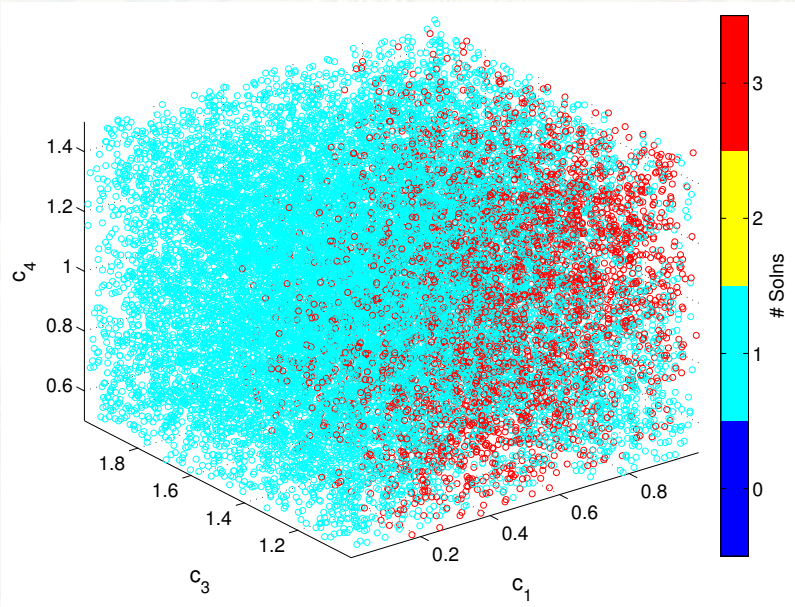
Zooming In

First zoom in around one of the special points.



Definite space containing multiple different numbers of positive real equilibria.

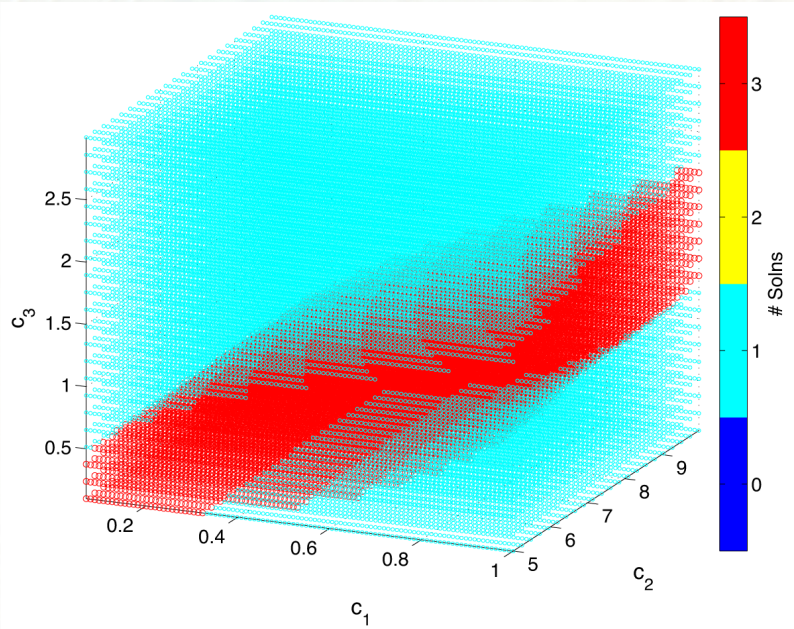
Second zoom, different projection.



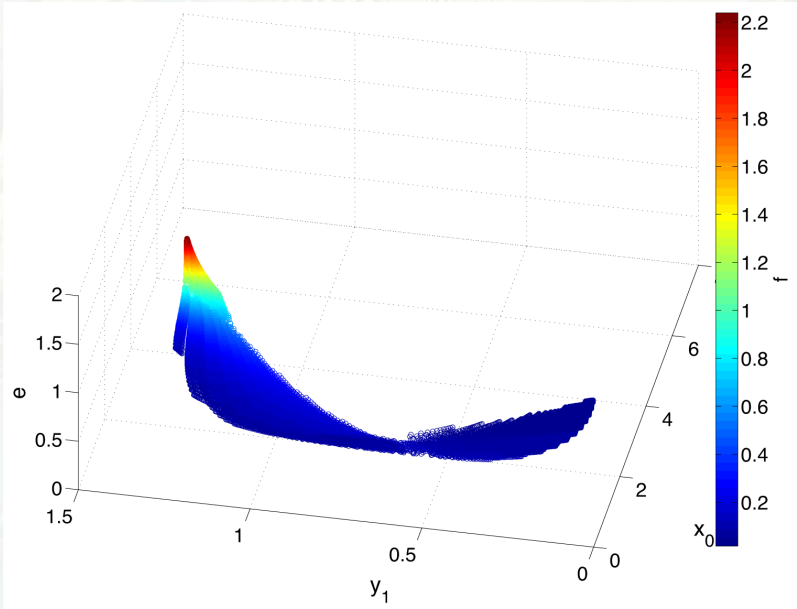
Structure begins to emerge.

Structure Emerges

Having seen a separation of the number of posreal equilibria depending on what looks like c_1, c_3, I chose to run a mesh around the point of interest.



Number of solutions - vs parameter point.



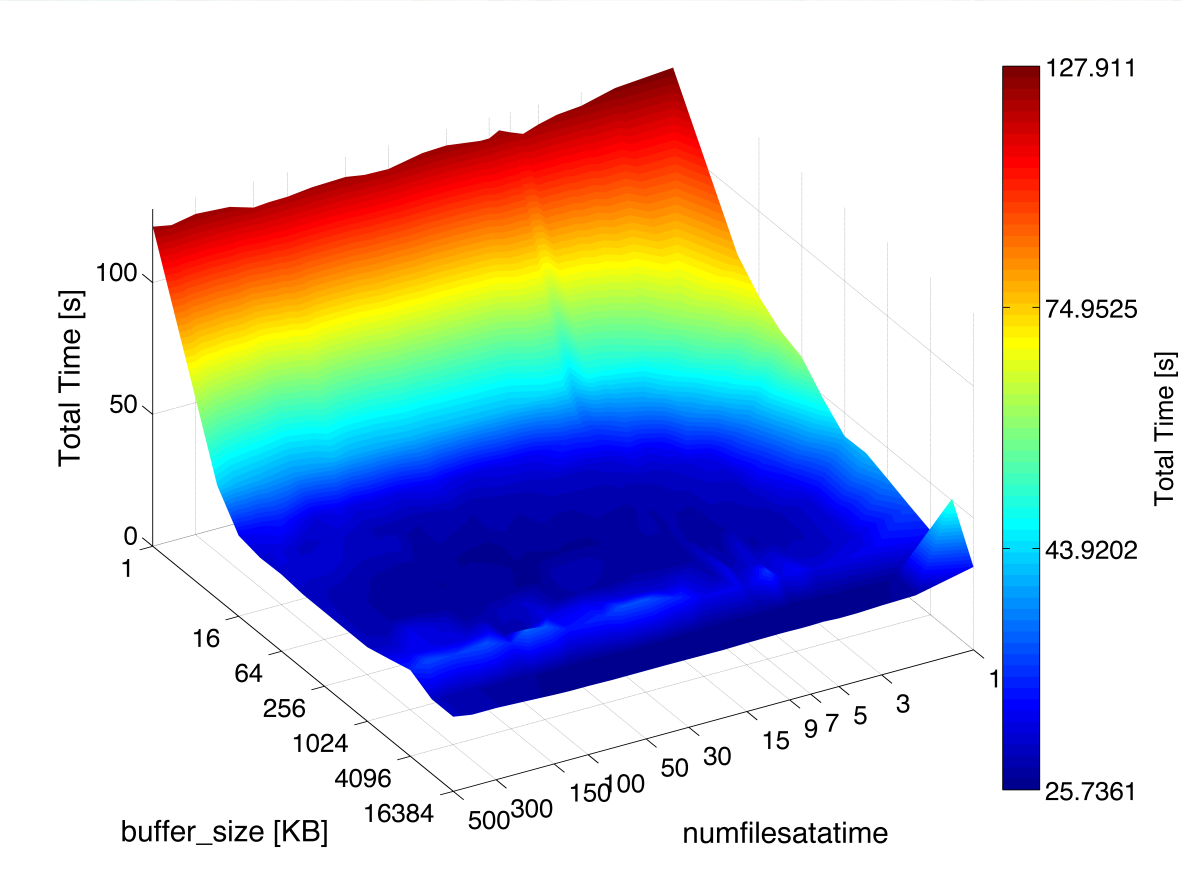
Solution set structure.

Summary

How can numerical methods help investigate a dynamical system?

- Provide numerical description to equilibria.
- Visualize to gain intuition.
- Search for anomalous points.
- Fix some parameters, vary others.
- Break down solution set in terms of dimensions and components.

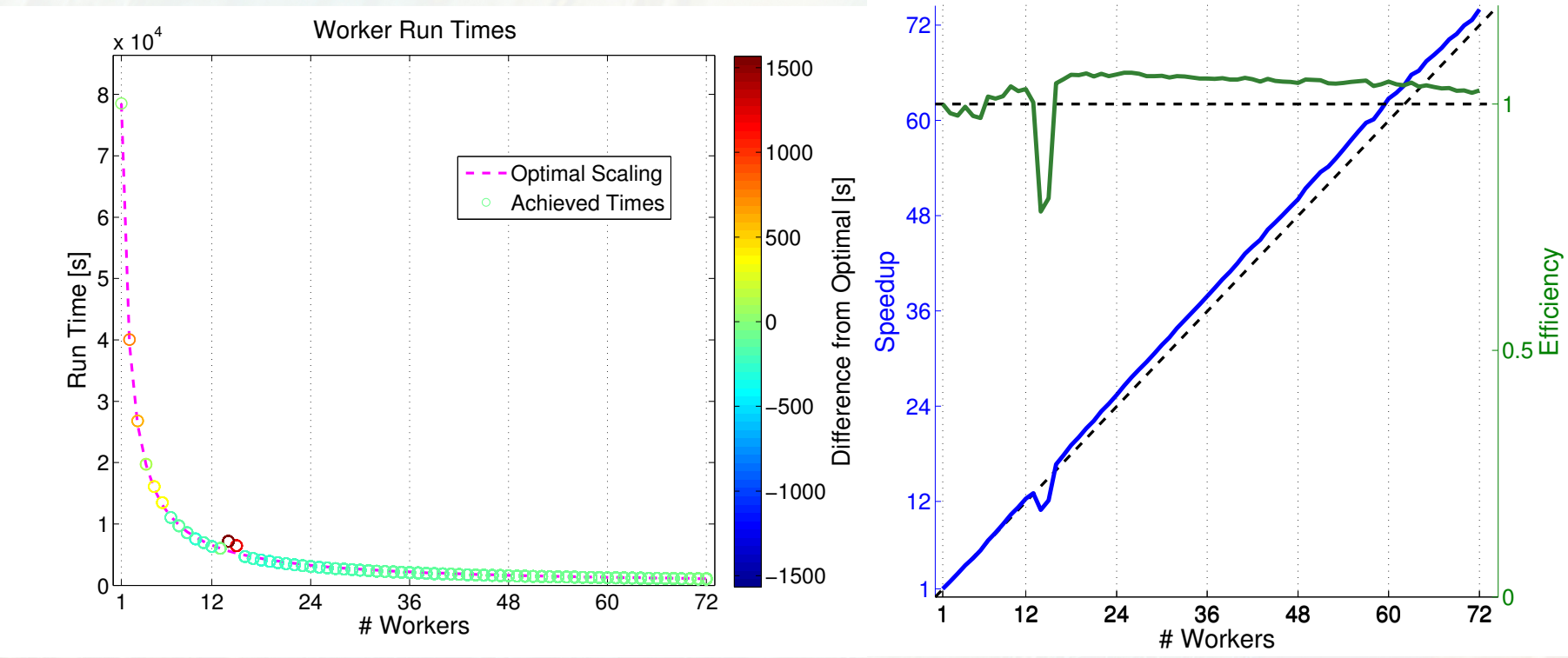
Tunable program settings



Testing Paramotopy runtimes versus generated data buffer size and file distribution packet size.

Timing example

9x9 system, 10^6 points in sample.



- Linear speedup and nice efficiency.
- Blip at 13, 14 workers where another process was using the hard drive.

Paramotopy – Recent Advancements

- Added Search method
  - Can search for a specific number of positive real solutions, or a range of values, from 0 to ∞.
  - Search a maximum number of test points.
  - Accumulate a minimum number of desired parameter points, or a target number of total solutions.
- Reorganized code, and menu structures, to allow further expansion in the future.
- Lookup table to facilitate searches for point data in large set.

Paramotopy – Future Work

- Replace Step 1 with numerical irreducible decomposition.
- In-program visualization.
- Interfaces for real decompositions, etc.
- Building databases of completed start systems, and produced data.

Thank you for your kind attention.

Acknowledgements

- `Paramotopy` is joint work with Matt Niemerg and Dan Bates
- We have been graciously funded by the NSF, and DARPA.