An exhaustive analysis of LCP solver performance on randomly generated rigid body contact problems

Evan Drumwright
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON DC

Dylan Shell
TEXAS A&M UNIVERSITY

Motivation and goal

The Linear Complementarity Problem (LCP) arises in rigid body contact problems. Simulation robustness and performance are directly affected by the particular solver employed: important considerations include running-time, solution quality, and reliability. In this work we seek a systematic, comparative understanding of the available solvers, and ultimately hope to identify trade-offs between solvers and underlying contact methods.

Generating random contact problems

One challenge is that the underlying physical problem imposes constraints on the LCP inputs. A meaningful analysis must consider these aspects.

Copositive LCPs

Definition: Copositivity is a class of matrices that includes those that are positive definite: $\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v} \geq \boldsymbol{0}$, for all non-negative vectors $\boldsymbol{v}$.
Significance: Both [5] and [6] formulate copositive LCPs for contact problems.

Generating Random Copositive LCPs:

1. Randomly pick the number of generalized coordinates (nc) in range [2,11].
2. Randomly pick the number of generalized coordinates (ngc) in range [6,24].
3. Randomly pick the number of polygon edges in the friction cone (nk) in 4, 6, 8, ..., 40.
4. Generate a $gc\times 1$ vector ($\boldsymbol{\mu}$) with each component uniformly randomly selected from range [0,1].
5. Generate a $ngc\times 1$ vector ($\boldsymbol{v}$) with each component uniformly randomly selected from range [-1,1].
6. Generate a $ngc\times n$ matrix ($\boldsymbol{N}$) with each element uniformly randomly selected from range [-1, 1].
7. Generate a $ngc\times(n*nk)$ matrix ($\boldsymbol{D}$) with elements uniformly randomly selected from range [-1, 1]; $\boldsymbol{D}$ is constructed such that, for every column $\boldsymbol{d}$, the column $\boldsymbol{-d}$ also appears in $\boldsymbol{D}$.
8. Randomly make some columns of $\boldsymbol{N}$, and the corresponding columns of $\boldsymbol{D}$, linearly dependent.
9. Generate $ngc\times ngc$ symmetric generalized inertia matrix ($\boldsymbol{G}$) either:
   (a) the identity matrix,
   (b) a PD matrix generated through summing ngc rank-1 updates.
10. Generate the LCP as in Anitescu-Potra [5].

The solvers

The following four methods were employed:

Lemke
A C++ version of the LEMKE Matlab library produced by Fackler and Miranda [1].

PATH
An interface to Ferris and Munson's commercial grade LCP solver [2].

SOR scheme
An implementation of the projected symmetric successive over-relaxation scheme [3].

Interior point method
An implementation of the primal-dual interior point method for solving convex LCPs described in [4].

LCPs as Convex and Strictly Convex QPs

Definition: These so called monotone problems arise for an LCP ($\boldsymbol{q}$,$\boldsymbol{M}$) where $\boldsymbol{M}$ is positive semi-definite. When $\boldsymbol{q}$ is in the range of $\boldsymbol{M}$, the LCP always has a solution.
Significance: LCPs with PD $\boldsymbol{M}$ arise in [7].

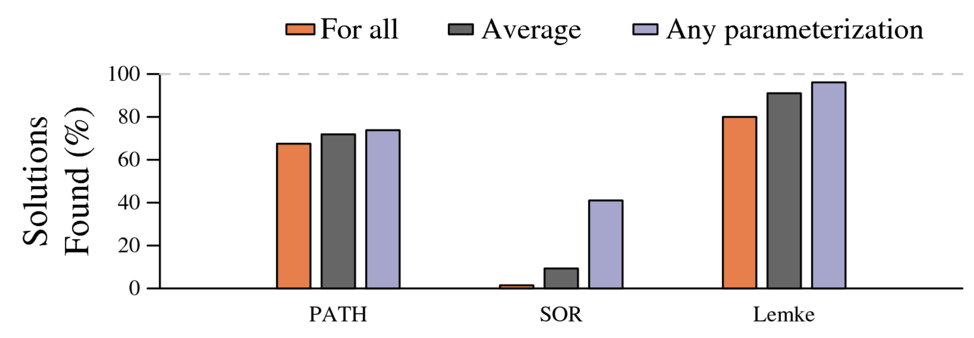
Generating Random Convex QP LCPs:

1. Randomly pick the number of generalized coordinates (n = ngc) in [2, 101].
2. Generate a $ngc\times 1$ vector ($\boldsymbol{v}$) with each component uniformly randomly selected from [-1, 1].
3. Generate a $ngc\times n$ matrix ($\boldsymbol{N}$) with components uniformly randomly selected from [-1, 1].
4. Randomly introduce linearly dependant cols. of $\boldsymbol{N}$
5. Generate $ngc\times ngc$ symmetric generalized inertia matrix ($\boldsymbol{G}$), either:
   (a) the identity matrix,
   (b) a PD matrix generated through summing ngc rank-1 updates,
   (c) a PSD matrix generated through summing k (< ngc) rank-1 updates.
6. Generate the LCP ($\boldsymbol{N}^T\boldsymbol{v}$, $\boldsymbol{N}^T\boldsymbol{G}^{-1}\boldsymbol{N}$), where $[]^{-1}$ denotes a SVD-regularized inverse.

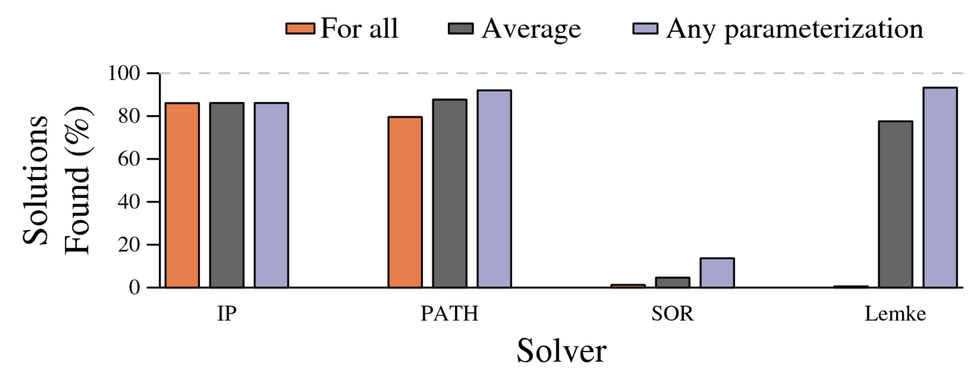
Difference between solvers

The following summarizes of the solution frequency of the tested solvers.

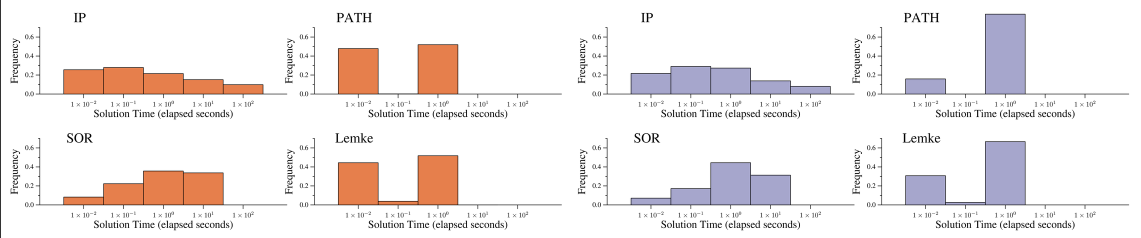
Copositive LCPs



Monotone LCPs

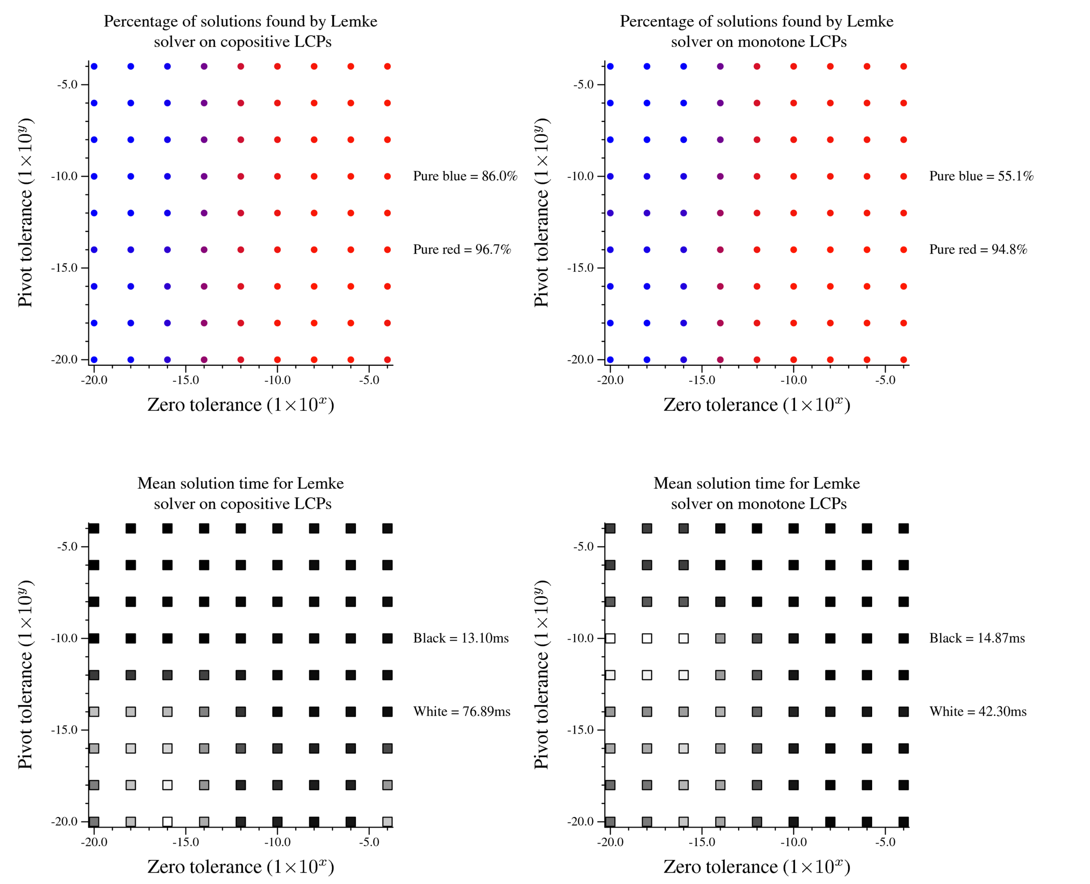


The following graphs summarize the processing time used by the solvers across random monotone LCPs.

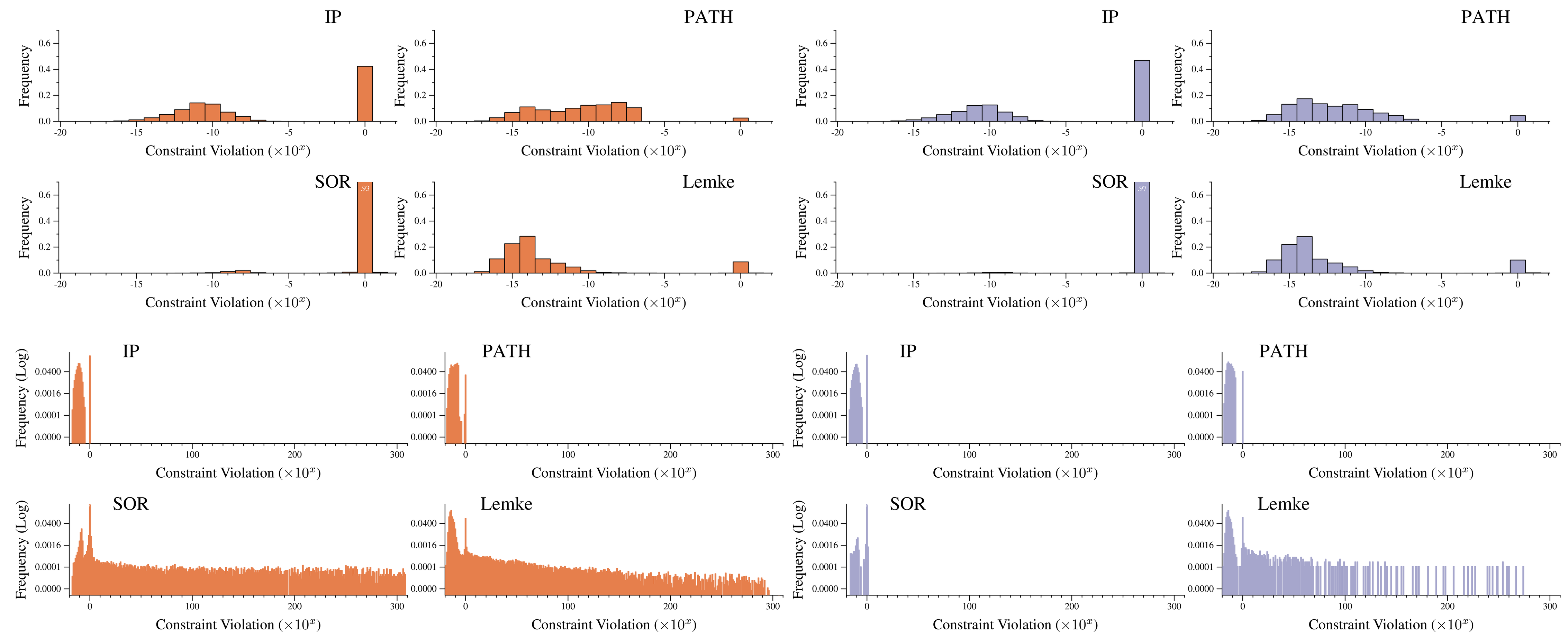


Solver sensitivity to parameters

The following summarizes of the solution frequency and quality of the tested solvers.



The following plots summarize aggregate solution quality.



[1] P. L. Fackler and M. J. Miranda, "LEMKE," http://people.sc.fsu.edu/ burkardt/m src/lemke/lemke.m
[2] M. C. Ferris and T. S. Munson, "Complementarity problems in GAMS and the PATH solver," J. of Economic Dynamics and Control, 24(2):165–188, Feb 2000.
[3] K. G. Murty, "Linear complementarity, linear and nonlinear programming," Sigma Series in Applied Mathematics 3. Berlin: Heldermann-Verlag. (Pg. 373).
[4] S. Boyd and L. Vandenberghe, "Convex Optimization," Cambridge University Press, 2004.
[5] M. Anitescu and F. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," Nonlinear Dynamics, vol. 14, pp. 231–247, 1997.
[6] D.E. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. Int. Journal of Numerical Methods in Engineering, 39:2673-2691, 1996..
[7] E. Drumwright and D. A. Shell, "Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation," in WAFR, 2010.