On lifting poorly-scaled flux balance analysis problems

Yuekai Sun*[1], Ronan M. T. Fleming[2,3], Ines Thiele[2,3], Michael A. Saunders[4]

[1] Institute for Computational and Mathematical Engineering, Stanford University, Stanford, California
[2] Center for Systems Biology, University of Iceland, Reykjavik, Iceland
[3] Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Campus Belval, Luxembourg
[4] Department of Management Science and Engineering, Stanford University, Stanford, California

Email: Yuekai Sun*- yuekai@stanford.edu; Ronan M. T. Fleming - ronan.mt.fleming@gmail.com; Ines Thiele - ines.thiele@gmail.com; Michael A. Saunders - saunders@stanford.edu;

*Corresponding author

Abstract

Background
Biological processes such as metabolism, signaling, and macromolecular synthesis can be modeled as large networks of biochemical reactions. Large and comprehensive networks, like Metabolic-Expression networks that explicitly accounts for the demands of macromolecular synthesis at single nucleotide resolution, are inherently multiscale because reaction rates can vary over many orders of magnitude. Naively using standard optimization systems to conduct flux balance analysis can produce inaccurate or infeasible results.

Results
We describe lifting techniques that enable off-the-shelf optimization software to compute accurate solutions to the poorly scaled optimization problems arising from flux balance analysis of multiscale biochemical reaction networks. We demonstrate the benefits of lifting using the first integrated reconstruction of metabolism and macromolecular synthesis for *E. coli*.

Conclusion
Lifting techniques enable accurate flux balance analysis of multiscale networks using off-the-shelf optimization software. Although we describe these techniques in the context of flux balance analysis, our methods can be used to handle a variety of optimization problems arising from analysis of multiscale network reconstructions.

Technical report SOL 2013-2
Department of Management Science and Engineering, Stanford University, Stanford, California, July 30, 2013

Flux balance analysis of multiscale biochemical reaction networks

Flux balance analysis (FBA) predicts steady state reaction rates (fluxes) of a biochemical network by solving the linear program

$$\begin{array}{ll} \underset{v}{\text{maximize}} & c^T v \\ \text{subject to} & Sv = 0, \\ & v_l \le v \le v_u, \end{array} \tag{1}$$

where $S \in \mathbf{R}^{m \times n}$ is a stoichiometric matrix that represents a network consisting of m species interacting via n reactions, $v_l, v_u \in \mathbf{R}^n$ are lower and upper bounds on the fluxes, and c represents a biologically motivated objective function. We refer to [1] for details about FBA.

Recently, Thiele et al. [2] developed the first genome-scale network of *E. coli* metabolism and macromolecular synthesis that represents the function of almost 2000 genes. This Metabolic-Expression network explicitly accounts for the demands of macromolecular synthesis at single nucleotide resolution. To enforce consistency between the state of metabolism and macromolecular synthesis, Thiele et al. introduce *coupling constraints* on certain pairs of fluxes (for example, the fluxes for a metabolic reaction and the reaction responsible for synthesizing the enzyme that catalyzes the metabolic reaction [3]):

$$c_{\min} \le \frac{v_1}{v_2} \le c_{\max}, \tag{2}$$

where $c_{\min}, c_{\max} > 0$. A coupling constraint can be formulated as a pair of linear inequality constraints:

$$\begin{aligned} -v_1 + c_{\min} v_2 &\le 0 \\ v_1 - c_{\max} v_2 &\le 0. \end{aligned}$$

We predict the steady state reaction rates of such integrated networks by solving the linear program

$$\begin{array}{ll} \underset{v}{\text{maximize}} & c^T v \\ \text{subject to} & Sv = 0, \\ & Cv \le d, \\ & v_l \le v \le v_u, \end{array} \tag{3}$$

where $Cv \le d$ includes coupling constraints of the form (2) for many pairs of fluxes.

Given the inherent multiscale nature of biological systems, the linear programs (1) and (3) are *poorly scaled*, *i.e.* the constraint matrices (and possibly the objective function) contain entries that vary over many orders of magnitude. Conducting FBA on these multiscale networks has been unsatisfactory because even state-of-the-art linear programming solvers can produce inaccurate (or infeasible) results. In particular, for the *E. coli* Metabolic-Expression network, using CPLEX [4] and Gurobi [5] to solve (3) with default settings (scaling enabled) has produced results with large constraint violations.

Handling poorly-scaled problems

Linear (equality and inequality constraints) describe a polytope in $\mathbf{R}^n$. The condition of a basis associated with a vertex of the polytope provides a quantitative measure of either the "sharpness" or the "flatness" of the vertex. Poorly scaled problems tend to create a polytope with very sharp and/or very flat vertices, creating very ill-conditioned bases and numerical difficulties for solvers.

Scaling techniques

To handle poorly scaled problems, solvers typically compute row and column scaling matrices $D_{\text{row}} \in \mathbf{R}^{m \times m}$ and $D_{\text{col}} \in \mathbf{R}^{n \times n}$ such that the nonzero entries of the scaled constraint matrix $D_{\text{row}} S D_{\text{col}}$ are of order one. Scaling can improve the condition of many bases, but it may be at the expense of making some bases ill-conditioned (including the optimal basis). For some problems, such as (3), the scaled constraints $D_r S D_c \bar{v} = 0$ may be satisfied accurately by the scaled solution $\bar{v}$, but when the solution is unscaled, $v = D_{\text{col}} \bar{v}$ may violate $Sv = 0$ significantly. We refer to [6] for a comprehensive study of scaling and its effects on the performance of the simplex method.

Iterative refinement

After a solver has returned an (allegedly) optimal solution, the accuracy of satisfying the general linear constraints ($Sv = 0$ and $Cv \le d$ in (3)) could be improved by applying a single step of classical iterative refinement [7], especially if extended precision were available. However, the refined basic solution could well lie outside its bounds, and further simplex iterations would be necessary. Ideally this difficulty would be handled by the simplex solver itself.

We note that more elaborate forms of iterative refinement have been used to improve the accuracy of linear programming solutions. Gleixner et al. [8] describe an incremental precision-boosting procedure that solves a sequence of linear programs, each attempting to correct the error in the previous optimal solution. The Zoom procedure of Saunders and Tenenblat [9] is an analogous strategy for interior point methods.

Lifting techniques

Lifting techniques are commonly used in optimization to create an efficient representation of a feasible set. By using auxiliary variables to "lift" the feasible set into a higher-dimensional space, lifting can dramatically reduce the computational expense (e.g., see Albersmeyer and Diehl [10], Gouveira et al. [11]). Here we apply lifting techniques to poorly scaled constraints to make the vertices of the "lifted" polytope more regular. We seek to reduce the *largest* entries in the constraint matrix.[1]

Mass balance constraints

In problem (1), the mass balance constraints $Sv = 0$ often contain poorly scaled reactions such as

$$A + 10^6 B \rightarrow C + D, \tag{4}$$

which may represent the synthesis of a macromolecule from subunits. We can reformulate such reactions into sequences of reactions involving dummy metabolites with reasonably scaled coefficients, thereby "lifting" the mass balance constraint to a higher dimensional space. For example, we can formulate (4) as two reactions involving a dummy metabolite $\hat{B}$:

$$A + 1000\hat{B} \rightarrow C + D \tag{5}$$

$$1000B \rightarrow \hat{B}. \tag{6}$$

Coupling constraints

In problem (3), the coupling constraints $Cx \leq d$ are often poorly scaled because reaction rates can vary over many orders of magnitude. For example, two fluxes could be constrained to satisfy

$$0.0001 \leq \frac{v_1}{v_2} \leq 10000. \tag{7}$$

We can also reformulate these constraints into sequences of constraints involving auxiliary variables with smaller coefficients, thereby "lifting" the coupling constraints to a higher dimensional space. For example, the second inequality in (7) is equivalent to two coupling constraints involving an auxiliary variable s_1:

$$v_1 \leq 100 s_1 \tag{8}$$

$$s_1 \leq 100 v_2. \tag{9}$$

Unlike scaling, lifting transforms poorly scaled constraints without affecting other constraints. The linear program does become larger (more constraints and variables), but the added constraints are usually very sparse (see Figure 1) and should not affect the performance of a solver. The time per iteration for the simplex method could well decrease because well-conditioned bases typically lead to sparser basis factorizations.

Implementation in the openCOBRA toolbox

We implemented lifting to handle poorly scaled stoichiometric and coupling constraints in the openCOBRA toolbox [12], a MATLAB package for constraint-based reconstruction and analysis of biochemical networks. Our implementation accepts a parameter τ and lifts constraints with entries larger than τ. Very large entries might require more than one auxiliary variable and constraint. In these cases, we choose the lifted constraint coefficients to be equally spaced in logarithmic scale. For example, the poorly scaled reaction

$$A + 10^9 B \rightarrow C + D$$

(with $10^9 > \tau$) would be reformulated as

$$\begin{aligned} A + 1000B_1 &\rightarrow C + D, \\ 1000B_2 &\rightarrow B_1, \\ 1000B &\rightarrow B_2 \end{aligned}$$

(with $1000 \leq \tau$). Our implementation also makes efficient use of auxiliary variables by reusing them if possible. If a species participates in more than one reaction with large stoichiometric coefficients, then we can use the same auxiliary variables to reformulate these reaction, thereby keeping problem size to a minimum. Finally, users should disable any "presolve" option after lifting to prevent reaggregation of

[1] Small entries in the constraint matrix do not constitute poor scaling unless all entries in a row or column are small.

auxiliary constraints and variables. Our implementation automatically disables presolve for CPLEX and Gurobi

Results and discussion

We use our implementation of lifting techniques conduct FBA on the aforementioned Metabolic-Expression network of *E. coli* [2]. The network involves 62212 metabolites, 76664 reactions, with 6087 coupling constraints. The network has about 41,000 large matrix entries (exceeding $\tau = 1024$), with the biggest entry being 8×10^5. Because of the dependencies between pairs of metabolic reactions and macromolecular synthesis reactions, the resulting flux balanced steady states have reaction rates that vary by four orders of magnitude [2].

Conducting FBA on this network using CPLEX or Gurobi with default settings (scaling enabled) either cause CPLEX or Gurobi to stop due to unrecoverable numerical difficulties or produced "optimal" solutions that were infeasible. Our own simplex solver SQOPT [13] with scaling activated would solve the scaled problem well, but unscaling would magnify the infeasibilities. Conducting flux variability analysis (FVA) on this network has also produced inaccurate or infeasible results.

Lifting the problem and conducted FBA and FVA using CPLEX (simplex solver). Lifting reduces the infeasibilities of the computed steady states and also stabilizes the number of simplex iterations (see Table 1). With their more comprehensive set of constraints, we also expect Metabolic-Expression networks to shrink the solution space, and we observe this in our results (see Figure 2).

Conclusions

We described techniques that enable off-the-shelf optimization software to be applied to multiscale network reconstructions, such as integrated networks that represent both metabolism and macromolecular synthesis. The techniques enable accurate FBA and FVA of an integrated network of metabolism and macromolecular synthesis in *E. coli*, previously impossible because of numerical difficulties encountered by solvers.

As *in silico* biologists create increasingly complex networks that capture more of the multiscale nature of biological systems [15], the optimization problems that arise during the analysis of these networks will also become increasingly poorly scaled. We are aware of researchers resorting to specialized packages such as [16] that rely upon rational arithmetic to obtain exact solutions to the FBA and FVA linear programs. Such solvers are likely to be prohibitively slow for analyzing larger, more comprehensive reconstructed networks. A more practical approach is to employ quadruple-precision arithmetic, which is increasingly available in Fortran and C compilers and is valuable even when implemented in software. In the meantime, our techniques enable the constraint-based modeling community to analyze increasingly sophisticated and comprehensive networks of biological systems with improved efficiency and reliability.

Acknowledgements

This work was supported by the Department of Energy (Offices of Advanced Scientific Computing Research and Biological and Environmental Research) as part of the Scientific Discovery Through Advanced Computing program, grant DE-FG02-09ER25917, by the National Institute of General Medical Sciences of the National Institutes of Health, award number U01GM102098, and by the Office of Naval Research, grant N00014-11-1-0067. The content is solely the responsibility of the authors and does not necessarily represent the official views of DOE, NIH, or ONR.

References

1. Orth JD, Thiele I, Palsson BØ: **What is flux balance analysis?** *Nature Biotechnology* 2010, **28**(3):245–248.
2. Thiele I, Fleming RMT, Que R, Bordbar A, Diep D, Palsson BØ: **Multiscale modeling of metabolism and macromolecular synthesis in *E. coli* and its application to the evolution of codon usage**. *PLoS One* 2012, **7(9):e45635**, [doi:10.1371/journal.pone.0045635].
3. Thiele I, Fleming RMT, Bordbar A, Schellenberger J, Palsson BØ: **Functional characterization of alternate optimal solutions of Escherichia coli's transcriptional and translational machinery**. *Biophysical Journal* 2010, **98**(10):2072–2081.
4. **CPLEX mathematical programming solver**. *http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/*.
5. **Gurobi mathematical programming solver**. *http://www.gurobi.com/*.
6. Elble J, Sahinidis N: **Scaling linear optimization problems prior to application of the simplex method**. *Computational Optimization and Applications* 2012, **52**:345–371.
7. Moler CB: **Iterative refinement in floating point**. *Journal of the ACM* 1967, **14**(2):316–321.

8. Gleixner A, Steffy D, Wolter K: **Improving the accuracy of linear programming solvers with iterative refinement**. ZIB-Report 12-19, Zuse Institute Berlin 2012. [Proc. of ISSAC 2012: 37th International Symposium on Symbolic and Algebraic Computation, to appear].

9. Saunders MA, Tenenblat L: **The Zoom strategy for accelerating and warm-starting interior methods**. *Presented at INFORMS Annual Meeting, Pittsburgh, PA* Nov 5–8, 2006. [http://www.stanford.edu/group/SOL/talks/saunders-tenenblat-INFORMS2006.pdf].

10. Albersmeyer J, Diehl M: **The lifted Newton method and its application in optimization**. *SIAM J. Optim.* 2010, **20**(3):1655–1684.

11. Gouveia J, Parrilo PA, Thomas R: **Lifts of convex sets and cone factorizations**. *ArXiv:1111.3164* 2011.

12. Schellenberger J, Que R, Fleming RMT, Thiele I, Orth JD, Feist AM, Zielinski DC, Bordbar A, Lewis NE, Rahmanian S, et al.: **Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0**. *Nature Protocols* 2011, **6**(9):1290–1307, [http://opencobra.sourceforge.net].

13. Gill PE, Murray W, Saunders MA: **SNOPT: An SQP algorithm for large-scale constrained optimization**. *SIAM Review* 2005, **47**:99–131. [SIGEST article].

14. Savinell JM, Palsson BØ: **Network analysis of intermediary metabolism using linear optimization. I. Development of mathematical formalism**. *Journal of Theoretical Biology* 1992, **154**(4):421–454.

15. Thiele I, Heinken A, Fleming RMT: **A systems biology approach to studying the role of microbes in human health**. *Current opinion in biotechnology* 2012.

16. Cook W, Koch T, Daniel E, Wolter K: **An exact rational mixed-integer programming solver**. In *Proceedings of the 15th international conference on Integer Programming and Combinatorial Optimization*, IPCO'11, Berlin, Heidelberg: Springer-Verlag 2011:104–116, [http://dl.acm.org/citation.cfm?id=2018158.2018167].

Figures

Figure 1 - E. coli Metabolic-Expression matrix before and after lifting

Spy plot of the *E. coli* Metabolic-Expression matrix before and after lifting. The red areas were added by the lifting and are very sparse.

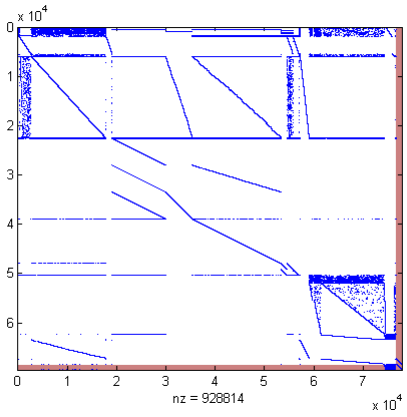
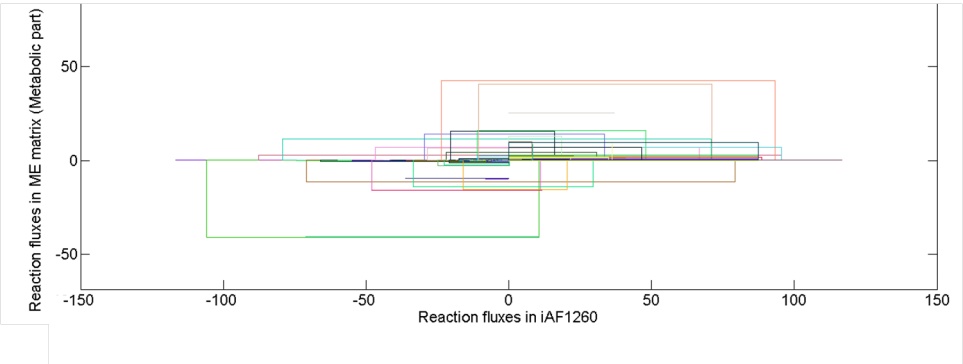


Figure 2 - Flux variability analysis of the E. coli Metabolic-Expression network
Minimum and maximum flux for iAF1260 (which only accounts for metabolic reactions) versus the minimum and maximum flux for the Metabolic-Expression model. Each colored box corresponds to a different reaction in metabolism. The boxes are always longer on the axis for the metabolic model (iAF1260) than on the axis for the Metabolic-Expression model. This demonstrates that increasing the comprehensiveness of the model toward whole cell modeling leads to a substantial shrinkage of the steady state solution space. (Fluxes are plotted in mmol $\cdot$ $g_{dw}^{-1} \cdot hr^{-1}$.)



Tables

Table 1 - FBA and FVA results (simplex solvers) for the Metabolic-Expression network before and after lifting
FBA and FVA results for the *E. coli* Metabolic-Expression network using the CPLEX simplex solver. Iterations and sum of infeasibilities before and after lifting. The first column lists which flux is being maximized.

	Simplex iterations		Infeasibility $\|b - Ax\|_1$	
	Before	After	Before	After
1	22510	44496	1.1×10^{-4}	7.6×10^{-5}
5001	30405	40318	1.5×10^{-4}	9.6×10^{-5}
10001	34963	41231	9.4×10^{-2}	8.2×10^{-5}
15001	103210	41891	4.5×10^{-5}	9.4×10^{-6}
20001	120089	40587	8.8×10^{-2}	8.3×10^{-5}
25001	30786	41161	1.7×10^{-4}	8.3×10^{-5}
30001	55177	40534	9.8×10^{-2}	8.1×10^{-5}
35001	68760	40933	1.3×10^{-4}	8.3×10^{-5}
40001	30360	40778	1.2×10^{-4}	1.6×10^{-5}
45001	107485	40905	3.1×10^{-5}	8.3×10^{-5}
50001	32553	40360	9.7×10^{-2}	8.5×10^{-5}
55001	20661	39909	9.5×10^{-5}	5.7×10^{-5}
60001	25477	39830	9.4×10^{-2}	8.6×10^{-5}
65001	139251	42230	2.9×10^{-5}	8.4×10^{-5}
70001	137611	42389	6.2×10^{-5}	8.3×10^{-5}
75001	40930	41139	4.0×10^{-5}	8.2×10^{-5}
biomass	48603	58288	1.3×10^{-4}	2.9×10^{-6}