

## Accelerating the Simplex Algorithm Via Novel Crash Procedures

Nikolaos Ploskas, Carnegie Mellon University, Pittsburgh, PA, Nikolaos Samaras, University of Macedonia, Thessaloniki, Greece and Nick Sahinidis, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA

### Abstract Text:

Dantzig's simplex algorithm was initially proposed in the late 1940s [1]. Since then, variants of this algorithm have been studied extensively by numerous researchers [2] and have become core routines in many optimization algorithms for linear, integer and nonlinear programming [3, 4]. Leading implementations such as CPLEX [5] are currently capable of routinely solving linear programs with millions of constraints and variables using the simplex algorithm [3, 4]. The algorithm is in use by most Fortune 500 companies and was recently included in the list of the top-10 algorithms of the twentieth century [6]. Given this level of activity in the field and presumed maturity of the simplex algorithm, one would think that it would be very difficult, if at all possible, to speed up current implementations of the simplex algorithm. In this paper, we show that a novel initialization strategy provides a means of considerable acceleration of the simplex algorithm.

We propose novel crash procedures for finding an initial basis for the simplex algorithm. The main idea is to obtain a starting basis that is sparse and will reduce the fill-ins in LU factorization that forms the core operation of simplex iterations. We present extensive computational comparisons on standard benchmarks by supplying our initial basis to the state-of-the-art linear programming code CPLEX. The computations demonstrate that, in comparison to the CPLEX default crash procedure, our initialization strategy on average reduces the running time of CPLEX by 60% in the case of primal simplex and by 27% in the case of dual simplex. For large and difficult instances, our algorithm reduces the running time of CPLEX by an order of magnitude.

### References cited:

- [1] Dantzig, G. B. (1963). Linear programming and extensions. Princeton, NJ: Princeton University Press.
- [2] D. Bertsimas and J. Tsitsiklis (1997). Introduction to Linear Optimization. Athena Scientific, Boston, MA.
- [3] Bixby, R. E. (2002). Solving real-world linear programs: A decade and more of progress. *Operations research*, 50, 3–15.
- [4] Bixby, R. E. (2012). A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, 107–121.
- [5] IBM (2016). IBM ILOG CPLEX V12.6. Available online: [http://www.ibm.com/support/knowledgecenter/SSSA5P\\_12.6.0/ilog.odms.cplex.help/CPLEX/homepages/CPLEX.html](http://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.0/ilog.odms.cplex.help/CPLEX/homepages/CPLEX.html)
- [6] Dongarra, J., Sullivan, F. (2000). Guest editors' introduction: The top 10 algorithms. *Computing in Science & Engineering*, 2(1), 22-23.