

TIMO BERTHOLD¹, MICHAEL PERREGAARD², CSABA
MÉSZÁROS³

Four good reasons to use an Interior Point solver within a MIP solver

¹ Fair Isaac Corporation (FICO), Zuse Institute Berlin & Forschungscampus MODAL, Takustr. 7, 14195 Berlin, Germany, timoberthold@fico.com

² FICO, Birmingham, UK, michaelperregaard@fico.com

³ FICO, Budapest, Hungary, csabameszaros@fico.com

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Four good reasons to use an Interior Point solver within a MIP solver

Timo Berthold¹, Michael Perregaard², and Csaba Mészáros³

¹ FICO (Fair Isaac Corporation), Takustr 7, 14195 Berlin, Germany
timoberthold@fico.com

² FICO, Birmingham, UK, michaelperregaard@fico.com

³ FICO, Budapest, Hungary, csabameszaros@fico.com

Abstract. “Interior point algorithms are a good choice for solving pure LPs or QPs, but when you solve MIPs, all you need is a dual simplex.” This is the common conception which disregards that an interior point solution provides some unique structural insight into the problem at hand. In this paper, we will discuss some of the benefits that an interior point solver brings to the solution of difficult MIPs within FICO Xpress. This includes many different components of the MIP solver such as branching variable selection, primal heuristics, preprocessing, and of course the solution of the LP relaxation.

Keywords: mixed integer programming, interior point, analytic center, barrier method

1 Introduction: MIP and the Analytic Center

Mixed integer programming (MIP) is one of the most important techniques in Operations Research and Discrete Optimization. A mixed integer program is an optimization problem of the form:

$$\min\{c^t x : Ax = b, x \geq 0, x_I \in \mathbb{Z}^I\}, \quad (1)$$

with matrix $A \in \mathbb{R}^{m \times n}$, vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$, and a subset $I \subseteq N := \{1, \dots, n\}$. The LP relaxation of a MIP is the continuous optimization problem which we get by dropping the integrality requirements of (1). The feasible region of the LP relaxation is a polyhedron. For an introduction to MIP, see [19].

The analytic center x^{ac} of a bounded polyhedron given in equality form ($Ax = b, x \geq 0$) has been introduced by Sonnevend [21] and is defined as

$$x^{\text{ac}} = \operatorname{argmin}\left\{-\sum_{j \in I} \ln x_j : Ax = b\right\}. \quad (2)$$

The analytic center can be efficiently computed by using a barrier algorithm, see next section. Note that the strong convexity of the logarithm implies that the analytic center of a bounded polyhedron is indeed uniquely defined. Furthermore,

it maximizes the distance to the boundary due to the logarithm going towards minus infinity when going to zero. If the polyhedron is a simplex, the analytic center is also the barycenter of the polyhedron [22].

The FICO Xpress Optimization Suite is a toolbox for mathematical optimization [6, 15]. It features software tools used to model and solve linear, integer, quadratic, nonlinear, and robust optimization problems. The core solver of this suite is the FICO Xpress-Optimizer (from here on: Xpress), a state-of-the-art MIP solver which combines ease of use with speed and flexibility. All computational results mentioned in this paper have been conducted with Xpress.

MIP solvers like Xpress feature a variety of algorithmic components which extend the basic branch-and-bound search and which are the reason that modern MIP solvers can solve some of the most complex optimization problems. These include primal heuristics to find feasible solutions, presolving techniques to reduce the problem size and branching strategies to efficiently split the problem into disjoint subproblems. All of the named techniques will be addressed in the present paper.

2 Impact of Barrier for LP solving

Unlike the simplex algorithm which iterates over extremal vertex solutions, the Newton barrier method iterates through solutions in the interior of the feasible region of the LP [13]. The name barrier comes from replacing the non-negativity constraints in the LP by logarithmic penalty terms, as in 2, and solving the problem

$$\min\{c^t x - \mu \sum_{j \in N} \ln x_j : Ax = b\}. \quad (3)$$

When μ converges to zero, the solution of the barrier problem converges to an optimal solution of the original LP (the barrier solution x^{bar}). The set of solutions for different μ describes the so-called central path which connects x^{ac} with x^{bar} . Note that x^{bar} is the analytic center of the optimal face of the LP. The barrier algorithm can be generalized to convex programming and thereby in particular to convex quadratic programming. Besides the nice property of being of polynomial complexity, the barrier method excels by its practical running time in particular on sparse problems. For an overview on interior point methods, see [20]. The barrier algorithm in Xpress is based on the solver described in [16]. It is a primal-dual algorithm extended by predictor-corrector and target-following techniques.

The observation that there is no clear dominance between simplex and barrier algorithm in terms of solving speed leads to the first, probably most obvious, application of an interior point solver within a MIP solver. For solving the initial LP relaxation, it is the default behavior of Xpress to run primal simplex, dual simplex and barrier (with a subsequent crossover) side-by-side in separate threads. Primal and dual each use one thread, all other threads are occupied by barrier. This method is known as concurrent LP solving. It will be interrupted when one of the algorithm solves the LP relaxation to proven optimality.

For MIP solving, the impact of concurrent LP solving on the overall running time is limited, about 2% speedup on MIPLIB [14]. However, the time for solving the initial LP relaxation improves by 36%. As a side effect of solving the initial LP faster, the time to find a first MIP solution improves by about 4%.

Note that even if barrier plus crossover is not the concurrent winner, barrier alone might have finished before the winning simplex algorithm. In this case, we might use x^{bar} without the need to compute it from scratch.

3 Using the Analytic Center for Presolving

The barrier solution is in the center of the optimal face, hence it minimizes the number of variables which are at their bound. In particular, it maximizes the number of fractional binaries. Similar to simplex solutions, the barrier solution comes with a dual solution, from which a set of reduced costs can be computed. These can be used for reduced cost fixing, a bound tightening algorithm.

The analytic center of the optimal face provides a maximal set of non-zero dual values and hence of non-zero reduced costs. If a variable has a zero reduced cost in this solution, it must have a zero reduced cost in any optimal LP solution. Note that this is different from, but related to a recent work by Bajgiran et al. [3] who compute a set of reduced costs s.t. a maximal set of variables can be fixed w.r.t. a given primal MIP solution value. Conversely, if we wanted to find an integer solution in the optimal face, we could safely fix all variables with non-zero duals to their current bounds. This could, also be used in the context of pump-reduce [1] to reduce the size of the auxiliary LP.

Besides using the dual values, there is a direct, primal way to use the analytic center for fixing variables. Assume for the remainder of this section, that we have an analytic center solution x^{ac} for the whole feasible region. By definition, this solution is strictly in the relative interior of the feasible region. Hence, any variable that is at its bound in this solution must therefore be at its bound in any feasible solution. This allows us to fix and remove such variables from any further consideration during a MIP solve. Principally, the same holds for slack variables. This is potentially beneficial for ranged rows, where fixing the slack will result in the ranged row to be tightened to an equation.

Note that the fixed variables would have been at their bound in any LP or MIP solution anyway. So the direct impact of this presolving step is limited. There is a certain benefit from the sheer reduction in the problem size if many variables can be fixed that way. Additional benefit might occur indirectly from extra presolving that has been enabled by the analytic-center-based fixings. In our computational experiments, we observed about 2% improvement from fixing variables w.r.t. the analytic center of the whole problem. The impact of using reduced costs from barrier solutions was performance neutral.

4 An Analytic Center Heuristic

Naoum-Sawaya [18] introduced (*recursive*) *central rounding*, which is based on the idea to round the analytic center to the nearest integer vector. Their intuition was that for general integer variables, a point in the “middle” of the relaxation’s feasible region is more likely to have an integer feasible solution in its vicinity as compared to an extremal solution of the relaxation.

We suggest a different way to interpret the analytic center of the whole problem in a heuristic context. It can be seen as an indicator for the direction into which a variable is likely to move when going towards feasibility. This is particularly interesting for variables that are close to zero or one in a pure binary problem. However, not all of them might be simultaneously set to their corresponding bound value; thus a pure rounding approach might not be sufficient. We propose to rather apply a soft rounding in a large neighborhood search fashion, compare, e.g., proximity search [9].

Thus, we set up an auxiliary objective function, whose coefficients are proportional to the analytic center solution values. That is, the closer a binary is to one in the analytic center solution, the more the objective will try to push it towards one. Further, we tentatively fix some variables that are very close to one of their bounds and finally apply an auxiliary MIP solve with strict working limits on the modified problem. Note that this heuristic, similar to the feasibility pump [8], completely disregards the original objective function of the MIP. Thus, it makes most sense as a start heuristic to find a first feasible solution, not so much as an improvement heuristic.

This heuristic is relatively expensive, as it involves computing the analytic center x^{ac} and solving at least some nodes of a MIP of similar size as the original. Thus, Xpress only uses it in rare cases. For instances for which it is particularly cumbersome to find a feasible solution, however, it makes a big difference. We observed an overall speedup of 40% on the Feasibility benchmark of Hans Mittelmann [17]. This big difference is due to the fact that there are a few instances for which this heuristic is the only one finding a solution within the time limit of that benchmark.

5 Branching w.r.t. Analytic Centers

Finally, we present a branching strategy that makes use of the analytic center of the whole problem. As argued in the previous section, the analytic center can be understood as an indicator in which direction variables are easiest to move while maintaining feasibility. The analytic center branching in Xpress branches on binary variables that are close to one. Additionally, it searches the subtree resulting from the up-branch first.

This is applied only for extremely dual degenerate MIPs and only on the top levels of the branch-and-bound tree. This follows the idea that when the analytic center has only a few binaries close to one, then it is likely that those, or at least most of those, should be one in any optimal MIP solution. While for

the variable fixing procedure in Section 3 it is important that we use the actual analytic center, for the heuristic in Section 4 or for branching, other interior points might work similarly well.

As said, this strategy is only applied in a few cases, but relatively efficient on those. Overall, branching w.r.t. analytic centers gives 2% speedup on MIPLIB. This comes from few instances on which this branching strategy improves performance by orders of magnitude.

6 Conclusion

Taking all the uses of the barrier solver and analytic center solutions together, having a barrier available makes up for 10% speedup in solving MIPs to proven optimality and three more MIPLIB2010 problems being solved by Xpress. The number of branch-and-bound nodes reduces by 6% and the primal-dual integral [5] by 9%.

Within Xpress, the proposed presolving, heuristic, and branching strategy all improve performance, but come with the computational burden of having to compute the analytic center first. For each of the individual procedures, this is a rather big overhead. However given that the analytic center only needs to be computed once to enable the application of all of them, it seems worthwhile to consider further applications of the analytic center within MIP, even if a single application will not justify the computational cost. This includes, e.g., extensions of the feasibility pump which make use of the analytic center [2, 7]. In the present paper, we did not discuss possibilities to use the analytic center for generating cutting planes, as it is done in convex programming [12, 11] or for filtering cuts. Compare also [4, 10] for the use of interior points for cutting plane separation.

References

1. Tobias Achterberg. LP basis selection and cutting planes, 2010.
2. Daniel Baena and Jordi Castro. Using the analytic center in the feasibility pump. *Operations Research Letters*, 39(5):310–317, 2011.
3. Omid Sanei Bajgiran, Andre A Cire, and Louis-Martin Rousseau. A first look at picking dual variables for maximizing reduced cost fixing. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 221–228. Springer, 2017.
4. Walid Ben-Ameur and José Neto. Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, 49(1):3–17, 2007.
5. Timo Berthold. Measuring the impact of primal heuristics. *Operations Research Letters*, 41(6):611–614, 2013.
6. Timo Berthold, James Farmer, Stefan Heinz, and Michael Perregaard. Parallelization of the FICO Xpress-Optimizer. *Optimization Methods and Software*, pages 1–12, 2017.
7. Natasha L Boland, Andrew C Eberhard, Faramroze G Engineer, Matteo Fischetti, Martin WP Savelsbergh, and Angelos Tsoukalas. Boosting the feasibility pump. *Mathematical Programming Computation*, 6(3):255–279, 2014.

8. Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005.
9. Matteo Fischetti and Michele Monaci. Proximity search for 0-1 mixed-integer convex programming. *Journal of Heuristics*, 20(6):709–731, 2014.
10. Matteo Fischetti and Domenico Salvagnin. An in-out approach to disjunctive optimization. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, pages 136–140. Springer, 2010.
11. Jean-Louis Goffin and Jean-Philippe Vial. Convex nondifferentiable optimization: A survey focused on the analytic center cutting plane method. *Optimization Methods and Software*, 17(5):805–867, 2002.
12. Jacek Gondzio, O Du Merle, Robert Sarkissian, and J-P Vial. ACCPM—a library for convex optimization based on an analytic center cutting plane method. *European Journal of Operational Research*, 94(1):206–211, 1996.
13. Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM, 1984.
14. Thorsten Koch, Tobias Achterberg, Erling Andersen, Oliver Bastert, Timo Berthold, Robert E. Bixby, Emilie Danna, Gerald Gamrath, Ambros M. Gleixner, Stefan Heinz, Andrea Lodi, Hans Mittelmann, Ted Ralphs, Domenico Salvagnin, Daniel E. Steffy, and Kati Wolter. MIPLIB 2010. *Mathematical Programming Computation*, 3(2):103–163, 2011.
15. Richard Laundy, Michael Perregaard, Gabriel Tavares, Horia Tipi, and Alkis Vazacopoulos. Solving hard mixed-integer programming problems with Xpress-MP: a MIPLIB 2003 case study. *INFORMS Journal on Computing*, 21(2):304–313, 2009.
16. Csaba Mészáros. The bmpd interior point solver for convex quadratic problems. *Optimization Methods and Software*, 11(1-4):431–449, 1999.
17. Hans Mittelmann. Benchmarks for optimization software: Feasibility benchmark. <http://plato.asu.edu/bench.html>.
18. Joe Naoum-Sawaya. Recursive central rounding for mixed integer programs. *Computers & Operations Research*, 2013.
19. George L. Nemhauser and Laurence A. Wolsey. *Integer and combinatorial optimization*. Wiley, 1988.
20. Yuri Nesterov and Arkadi Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. Studies in Applied and Numerical Mathematics. Society for Industrial and Applied Mathematics, 1994.
21. György Sonnevend. An “analytical centre” for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In A. Prékopa, J. Szelezsáan, and B. Strazicky, editors, *System Modelling and Optimization*, volume 84 of *Lecture Notes in Control and Information Sciences*, pages 866–875. Springer Berlin Heidelberg, 1986.
22. György Sonnevend. Applications of the notion of analytic center in approximation (estimation) problems. *Journal of Computational and Applied Mathematics*, 28:349–358, 1989.