An Improved Oracle Adaption for Bilevel Programs

Daniel Jungen,a Alexander Mitsosb,a,c,\*

aProcess Systems Engineering (AVT.SVT) RWTH Aachen University, 52074 Aachen, Germany

bJARA-CSD, 52056 Aachen, Germany

cInstitute of Energy and Climate Research: Energy Systems Engineering (IEK-10),

Forschungszentrum Jülich GmbH, 52425 Jülich, Germany.

amitsos@alum.mit.edu

Abstract

Bilevel programs with nonconvex lower levels occur in many applications in engineering but are notoriously challenging: A global optimization problem must be solved even to check the feasibility of a given candidate solution point. We present an adaption of the approach of Tsoukalas et al. [J. Glob. Optim. 44, 235–250 (2009)]. Our algorithm adaption changes the oracle to minimize directly the lower-level objective, with the target objective value inscribed into its constraints. With this formulation, we aim to obtain lower-level-optimal solution points to the oracle, and thus faster generation of good upper bounds. We implement and compare our approach to the original approach and the state-of-the-art solvers of Mitsos et al. [J. Glob. Optim. 42, 475–513 (2008)] and Djelassi et al. [J. Glob. Optim. 75, 341–392 (2019)] using a comprehensive benchmark test set comprising more than 160 problem instances. Our approach outperforms the original oracle algorithm and the solver of Mitsos et al. but not the one of Djelassie et al.

**Keywords**: bilevel programming, benchmark, oracle.

* 1. Introduction

Bilevel programs (BLPs) are optimization problems where a so-called lower-level problem is embedded into an upper-level problem. We consider optimistic BLPs of the form

|  |  |
| --- | --- |
|  | (BLP) |

with the upper-level objective function , the lower-level objective function *h*, the coupling upper- and lower-level inequality function and , respectively, and the non-coupling lower-level inequality function . A reliable and fast solution to such optimization problems is paramount due to their many applications, e.g., chemical engineering (Clark and Westerberg, 1990; Mitsos et al., 2009) or even gemstone cutting (Küfer et al., 2008). However, the solution of (BLP) is very challenging: the lower-level optimization problem must be solved globally even to check the feasibility of a given candidate pair . (BLP) can be reformulated as a single-level problem if the lower-level problem is convex and satisfies some regularity conditions. However, convexity of the lower level can, in many applications, not be assumed.

A prominent approach for solving (BLP) absent any convexity assumptions is the adaptive discretization approach. Over the last decades, discretization-based algorithms have been published for the global solution of (BLP), e.g., (Mitsos et al., 2008; Tsoukalas et al., 2009; Djelassi et al., 2019). We are especially interested in the approach Tsoukalas et al. (2009) proposed, which we implemented as the *BLP-Oracle* solver in libDIPS (Jungen et al., 2023). This approach performs a bisection search in the objective space by iteratively solving an oracle problem to determine whether a target objective value is attainable or not. We believe the *BLP-Oracle* is promising as it might inherit strong convergence guarantees through the bisection approach. The oracle problem is an unconstrained minmax problem, where the objective function combines the upper- and lower-level objectives and constraints. Tsoukalas et al. (2009) solve the minmax problem through a smoothing technique based on the Laplace method.

Our algorithm (*BLP-AdaptOracle*) adapts the concept of Tsoukalas et al. (2009) by letting the oracle directly minimize the lower-level objective, with the target objective value inscribed into its constraints. With this formulation, we aim to enhance the chance that the computed candidate point is lower-level optimal, which may lead to a faster generation of good upper bounds (UBD). Although unconstrained subproblems, as proposed by Tsoukalas et al. (2009), might be easier to solve, we do not expect any performance losses since we require a global solution of the subproblems.

* 1. The *BLP-AdaptOracle* Approach
     1. Algorithm Description and Subproblem Formulations

For the bisection search in the objective space, we need an initial lower bound (LBD) and UBD on the objective. We compute an initial LBD for (BLP) by

|  |  |
| --- | --- |
|  | (LBP) |

and an initial UBD by

|  |  |
| --- | --- |
|  | (UBP) |

Once an initial LBD and UBD are computed, the target objective value is set to . We formulate our adapted oracle problem as

|  |  |
| --- | --- |
| . | (ORA) |

If the optimal objective value of (ORA) is or infeasible, then is not attainable. Then, the current is set to and (ORA) is solved again. Whenever (ORA) is feasible, first, feasibility of the computed candidate point is checked by solving the lower-level problem for fixed , which reads

|  |  |
| --- | --- |
| . | (LLP) |

If the candidate point is -feasible, i.e., , or the corresponding upper-level objective value is better than the incumbent *,* is updated, and (ORA) is solved again. If the candidate point is infeasible, the discretization is populated with a suitable discretization point, and (ORA) is solved again. An auxiliary problem needs to be solved to compute a valid discretization point, similar to the algorithm of Mitsos and Tsoukalas (2015) for generalized semi-infinite programs. The auxiliary problem is given by

|  |  |
| --- | --- |
|  | (AUX) |

with . If no valid discretization point is computed, is decreased and (AUX) is resolved. The complete algorithm description of *BLP-AdaptOracle* is given in Algorithm 1.

* + 1. Assumptions and Proof of Convergences

The following assumptions are primarily standard in global optimization and very similar to the assumptions used in Mitsos et al. (2008) of the solver *BLP-Box*.

*Assumption 1 (Compactness of Sets)*: The sets and are compact.

*Assumption 2 (Continuous Functions)*: All occurring functions are continuous on their respective host sets.

*Assumption 3 (Global solution of subproblems):* All subproblems are solved to global optimality.

Algorithm 1: Algorithm description of *BLP-AdaptOracle*. User inputs are the initial finite discretization set , an initial value for , a decreasing rule for , and the termination tolerances and

|  |  |  |
| --- | --- | --- |
| **1** | solve (LBD) to obtain | |
| **2** | **if** (LBD) is infeasible **then** | |
| **3** |  | set and terminate; |
| **4** | **end** | |
| **5** | Set ; | |
| **6** | solve (UBD) to obtain ; | |
| **7** | set ; | |
| **8** | set and ; | |
| **10** | **if**  **then** | |
| **11** |  | terminate; |
| **12** | **end** | |
| **13** | solve (ORA) to obtain and ; | |
| **14** | **if** (ORA) is infeasible **then** | |
| **15** |  | set and ; |
| **16** |  | got to line 13; |
| **17** | **end** | |
| **18** | solve (LLP) for fixed to obtain and | |
| **19** | **if**  and  **then** | |
| **20** |  | set , , and ; |
| **21** |  | go to line 10; |
| **22** | **end** | |
| **23** | **If**  and  **then** | |
| **24** |  | set , and ; |
| **25** |  | go to line 10; |
| **26** | **end** | |
| **27** | solve (AUX) for fixed to obtain and ; | |
| **28** | **if** **then** | |
| **29** |  | decrease |
| **30** |  | go to line 27; |
| **31** | **end** | |
| **32** | set ; | |
| **33** | go to line 10; | |

*Assumption 4 (Inner Problem):* For each point with and it holds: For any there exists a point such that

|  |  |
| --- | --- |
|  | (A3) |

Note that Assumption 4 is stronger than Assumption 4 in Mitsos et al. (2008).

* + 1. Outline of Proof of Convergence

First, we show that if , i.e., the target objective value is not attainable, (ORA) will become infeasible after finitely many iterations. The only interesting iterates are the pairs satisfying all lower- and upper-level constraints but not satisfying optimality in (LLP). Suppose a pair is found, which exhibits this. The discretization point computed by (AUX), which is added, forbids in all following iterations to revisit (, and a neighborhood around it. Therefore, after finitely many iterations, (ORA) becomes infeasible, and it is proven that is not attainable. Second, we show that if , i.e., the target objective value is attainable, (ORA) will furnish a feasible point after finitely many iterations. Like in the first case above, we exclude all pairs that do not satisfy lower-level optimality after a finite number of iterations. Because at least one pair fulfills all lower- and upper-level constraints and is optimal in the lower level, we must obtain such a point after finitely many iterations. Whenever the target value is proven to be not attainable or a feasible pair is found satisfying , the target objective value is updated via bisection. Through bisection, we terminate finitely with proof of -optimality.

* 1. Numerical Experiments

We use our library libDIPS (Jungen et al., 2023) to compare *BLP-AdaptOracle* to our implementation of the approach of Tsoukalas et al. (2009), i.e., *BLP-Oracle,* and the solvers *BLP-Box* (Mitsos et al., 2008) and *BLP-noBox* (Djelassie et al. 2019). To enable a fair comparison to the underlying algorithm idea of Tsoukalas et al. (2009), we do not apply the smoothing method proposed in the original application but rather directly solve the discrete minmax problem, and all other occurring subproblems, using the subsolver MAiNGO v0.7.1. (Bongartz et al., 2018) with its default settings. We use the bilevel benchmark test set provided by libDIPS (Jungen et al. 2023) for the numerical experiments, which consists of 167 problem instances. All calculations were conducted on the RWTH High-Performance Computing cluster running Rocky Linux 8 on a single core with an Intel Xeon Platinum 8160 Processor ‘SkyLake’ running at 2.1 GHz. We use the ‘best-working’ hyperparameters of *BLP-Box*, as reported in Jungen et al. (2023). For BLP-AdaptOracle we use an initial and decrease it by 1.5 whenever (AUX) fails to produce a discretization point.

Figure 1 shows the time factor performance plot for the different solvers. It can be seen that *BLP-AdaptOracle* exceeds *BLP-Oracle* as well as slightly outperforms *BLP-Box*. However, *BLP-noBox* is still by far the best solver.

* 1. Conclusion

We presented an adaptation of the BLP algorithm proposed by Tsoukalas et al. (2009). While our adaptation outperforms our implementation of the original approach and the state-of-the-art solver of Mitsos et al. (2008), it is superseded by the approach of Djelassi et al. (2019). However, our results (likely) depend on the composition of the benchmark test set, the subproblem formulation, and the used subsolver. The bilevel community might benefits from our algorithm adaption and its integration into libDIPS because the seamless transition between the bilevel solvers within libDIPS enhances accessibility, and our algorithm adaptation might be advantageous in specific applications. Furthermore, our contribution shows that there is likely some undetected potential in the existing approaches, as a simple adaptation resulted in significant performance improvements.

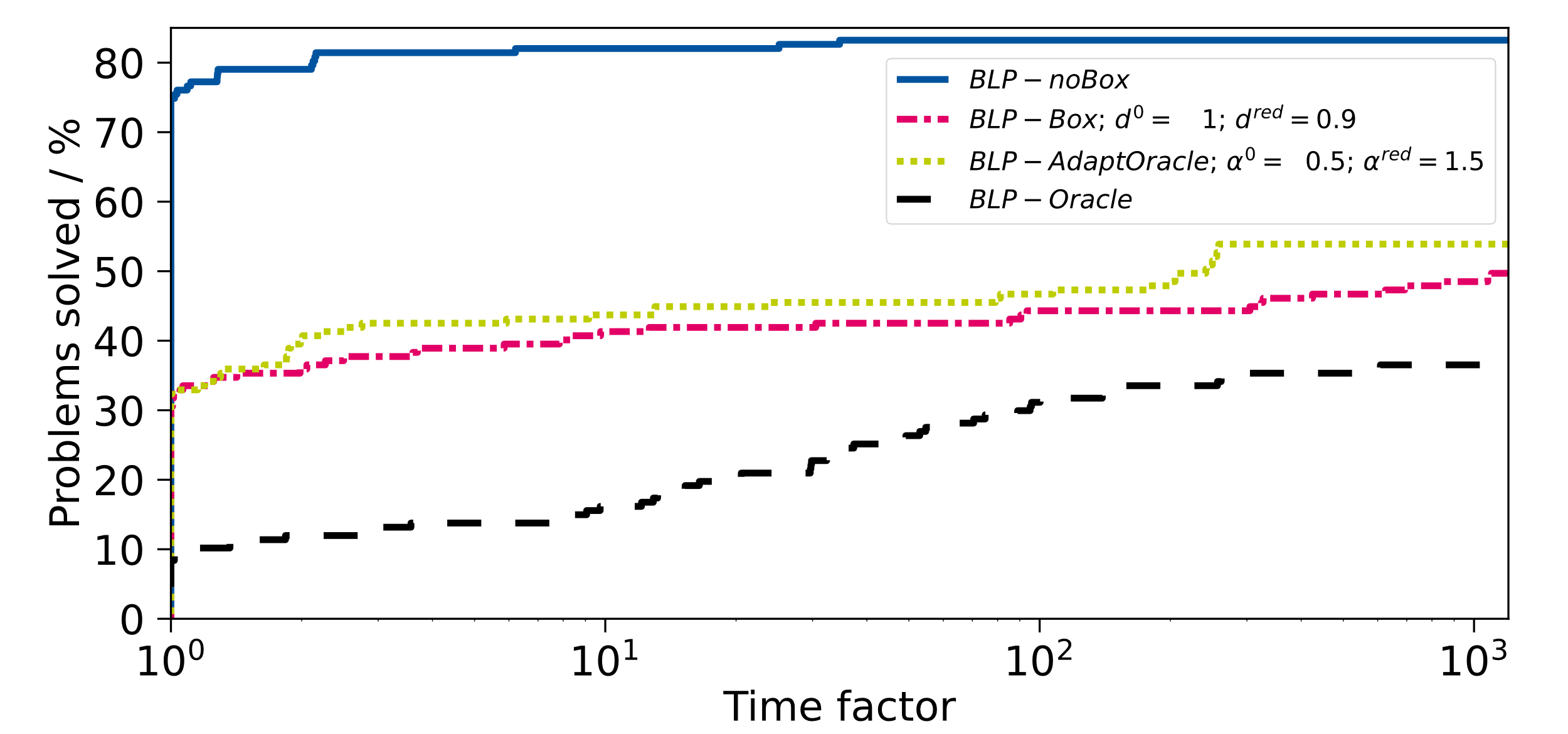


Figure 1: Time factor performance plot showing the numerical performance of the solvers *BLP-noBox* (Djelassi et al., 2019); *BLP-Box* (Mitsos et al. 2008); our algorithm adaptation *BLP-AdaptOracle*; and *BLP-Oracle,* our implementation of Tsoukalas et al. (2009).

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy – Cluster of Excellence 2186 ‘The Fuel Science Center’ – ID: 390919832. Computations were performed with computing resources granted by RWTH Aachen University.

References

Bongartz, D., Najman, J., Sass, S., & Mitsos, A. (2018). MAiNGO - **M**cCormick-based **A**lgorithm for mixed-**i**nteger **N**onlinear **G**lobal **O**ptimization. *Technical Report, Process Systems Engineering (AVT.SVT), RWTH Aachen University, Germany*. *http://permalink.avt.rwth-aachen.de/?id=729717*

Clark, P. A. & Westerberg, A. W. (1990). Bilevel programming for steady-state chemical process design—I. Fundamentals and algorithms. *Computers & Chemical. Engineering,* *14*, 87–97.

Djelassi, H., Glass, M. & Mitsos, A. (2019). Discretization-based algorithms for generalized semi-infinite and bilevel programs with coupling equality constraints. *Journal of Global Optimization,* *92*, 341–392.

Küfer, K.-H., Stein, O. & Winterfeld, A. 2008. Semi-infinite optimization meets industry: A deterministic approach to gemstone cutting. *SIAM News, 41* (8).

Jungen, D., Zingler, A., Djelassi, H., Mitsos, A. (2023; in preperation) libDIPS – Discretization-Based Semi-Infinite Programming Solvers. https://git.rwth-aachen.de/avt-svt/public/libdips.

Mitsos, A., Lemonidis, P. & Barton, P. I. (2008). Global solution of bilevel programs with a nonconvex inner program. *Journal of Global Optimization, 42*, 475–513.

Mitsos, A., Bollas, G. M. & Barton, P. I. (2009). Bilevel optimization formulation for parameter estimation in liquid–liquid phase equilibrium problems. *Chemical Engineering Science, 64*,548–559.

Mitsos, A. & Tsoukalas, A. (2015). Global optimization of generalized semi-infinite programs via restriction of the right hand side. *Journal of Global Optimization,* *61*, 1–17.

Tsoukalas, A., Rustem, B. & Pistikopoulos, E. N. (2009). A global optimization algorithm for generalized semi-infinite, continuous minimax with coupled constraints and bi-level problems. *Journal of Global Optimization, 44*, 235–250.