

ARMIN FÜGENSCHUH, STEFAN N. GRÖSSER, INGMAR  
VIERHAUS

**A Global Approach to the Control of an  
Industry Structure System Dynamics  
Model**

Herausgegeben vom  
Konrad-Zuse-Zentrum für Informationstechnik Berlin  
Takustraße 7  
D-14195 Berlin-Dahlem

Telefon: 030-84185-0  
Telefax: 030-84185-125

e-mail: [bibliothek@zib.de](mailto:bibliothek@zib.de)  
URL: <http://www.zib.de>

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

# A Global Approach to the Control of an Industry Structure System Dynamics Model

Armin Fügenschuh  
Helmut Schmidt University Hamburg  
Department of Mechanical Engineering  
Email: fuegenschuh@hsu-hh.de

Stefan N. Grösser  
Bern University of Applied Sciences  
School of Business  
Email: stefan.groesser@bfh.ch

Ingmar Vierhaus  
Zuse Institute Berlin  
Department Optimization  
Email: vierhaus@zib.de

**Abstract**—We consider a system dynamics model that describes the effect of human activity on natural resources. The central stocks are the accumulated profit, the industry structures, and the water resources. The model can be controlled through two time-dependent parameters. The goal in this paper is to find a parameter setting that leads to a maximization of a performance index, which reflects both environmental and economic aspects. Thus, the goal is to identify the most sustainable stock of industry structures within the model's constraints and assumptions. In order to find a proven global optimal parameter set, we formulate the System Dynamics Optimization model as a mixed-integer nonlinear problem that is accessible for numerical solvers. Due to the dynamic structure of the model, certain steps of the solution process must be handled with greater care, compared to standard non-dynamic problems. We describe our approach of solving the industry structure model and present computational results. In addition, we discuss the limitations of the approach and next steps.

## I. INTRODUCTION

System dynamics (SD) is a computer-aided approach to theory-building, policy analysis, and strategic decision support which emerges from an endogenous point of view [1], [2], [3]. SD models are formal, structural models based on ordinary differential equations which incorporate hypotheses about the causal connections of parameters and variables as functional units, and the outcomes of their interactions. SD is applied to organizational, economic, social, and ecological situations which are characterized by information feedback, time delays between cause and effect, interdependence between elements, and nonlinearities [4].

SD models, as described in the previous section, are simulation models. That is, besides the internal structure of the model described by ordinary differential equations, the user specifies parameter values that reflect his or her knowledge on the initial state of the system at some time  $t_0$ . The forward simulation of the model, numerically carried out by a suitable discretization of the continuous time into a finite number of time steps, describes the development of each model variable over time.

We assume that the user is interested in optimizing a certain goal. This is typically a final state of the system, for example, a certain variable should be as high or as low as possible at the final time of the simulation horizon  $t_{\max}$ . Another example would be the maximization (or minimization) of the integral

of some function. Mathematically, this goal is described by an objective function. The maximization (or minimization) of such an objective function is subject to the model's constraints (the mathematical equations) and a range in which certain designated parameters are allowed to vary. We distinguish a continuous control, where each varying parameter is an unknown, to be determined function over time. In other words, the function may take a different value at each discrete time. The second case is that of a constant control, where a single value is sought that is used for the complete time horizon of the problem. In the application, we will focus on the first case, a control varying over time.

An SD model that is extended by an objective function and a range of potential values for the set of control parameters is what we call a system dynamics optimization (SDO) problem.

The question arises how to solve an SDO problem. Several methods are described in the literature to solve optimization problems. To provide an overview, they can be grouped into heuristic, local, or global methods. Heuristic methods (such as genetic algorithms [5], [6], tabu search [7], or simulated annealing [8], [9]) do not give any quality measure on the solution quality. That is, they do not describe if the solution is optimal or not. Local methods end up with a local optimal solution, i.e., they guarantee that within some neighborhood around the computed solution, no better solution exists [10], [11]. Finally, a global method provides a quality certificate for the computed solution. That is, they either prove that the solution is globally optimal, or they provide an estimation of how far the solution is away from optimality. The disadvantage of global methods, as compared to heuristic or local methods, usually is that they are slower. Solving SDO problems is currently carried out by heuristic or local methods [12]. To the best of our knowledge there are no global approaches for the solution of SDO problems in the literature. Details of our approach are described in Section IV.

As a demonstrator, we apply our techniques to an SDO problem that asks for the optimal setting of some parameters in order to find the most performant industry structure with respect to its water consumption. Fresh water is a scarce resource, in particular, in arid areas of the world, and cannot be replaced or substituted by other substances for all living organisms. An efficient use of this natural resource is of great importance for a sustainable economy. For natural systems, sustainability can be understood as the capacity to

remain productive over time. Today, sustainability is discussed especially regarding natural resources such as oil, gas, and fresh water [13]. In the paper, we use an SD model which addresses the water consumption of industry structures. It is obvious that industrial development strives only when a mix of resources are available in sufficient amount. However, today the challenge arising from limited availability of water resources is largely neglected compared to the issues discussed, e.g., in relation to consumption of oil and gas resources.

## II. INDUSTRY STRUCTURE SD MODEL

The industry structure model was developed by Mojta-hedzadeh and has been analyzed later in several publications (Ford and Flynn [14], Kampmann and Oliva, [15], and Mojta-hedzadeh [16]). The purpose of the model was to study economic development in arid areas such as Yazd in Iran, that heavily depend on underground water. In such areas, the adequate supply of water is a core element for industrial development and societal prosperity. The idea of water adequacy was implemented as the ratio of water consumption and water demand. The model emphasizes on the role of market signals in making decisions for developing new industry structures in a country. Even though the model explicitly accounts for water resources, the range of applications of the model is much wider: Water resources can be conceived as a generic placeholder for any resource which is scarce in a specific context. As such, the model can be perceived as a theory of growth under conditions of constrained resources.

The model being analyzed and optimized in this paper generates an overshoot pattern of behavior. Figure 1 shows a classic structure that is able to explain how industrial structures prosper over time until they reach a limit imposed by nature. The model, even though it can be considered as small, accounts for four essential processes of reality:

- 1) Industry Structures ( $S$ ) grow with new industry structures ( $i$ ) through a reinforcing mechanism (R1) and demolished with demolition rate ( $d$ ) through a balancing mechanism (B3).
- 2) Industry Structures ( $S$ ) consume Water Resources ( $W$ ) by water consumption ( $c$ ) which in turn drains available Water Resources ( $W$ ).
- 3) A shortage in Water Resources ( $W$ ) reduces the growth in new industry structures ( $i$ ). This is captured by the effect of water adequacy on new industry structures ( $n$ ). Initially, the growth of Industry Structures ( $S$ ) increases the water consumption ( $c$ ) by a reinforcing mechanism (R2). For low levels of Water Resources ( $W$ ), the water adequacy ( $a$ ) is controlled by the balancing mechanism (B1).
- 4) The availability of water controls the water consumption ( $c$ ) through a balancing mechanism (B4). This is captured by the effect of water availability on water consumption ( $e$ ).

As has been detailed before, the concept of water adequacy ( $a$ ) is central to the model structure and its behavior. The concept is the ratio of water demand ( $w$ ) and water consumption ( $c$ ). The final objective in the model is formulated as a performance index which weights the Accumulated Profit ( $P$ ) with the weight of accumulated profit and the remaining Water Resource ( $W$ ) with the weight of water resources.

Parameter	Value
$p_{profit}$	0.01
$p_{water}$	0.1
$p_{init}$	10000
$p_{ind}$	100
$p_{demand}$	10

TABLE I. PARAMETER VALUES USED IN THE OPTIMIZATION.

## III. INDUSTRY STRUCTURE AS SDO PROBLEM

We transform the SD model from the previous section into an SDO problem. The model has three stocks: accumulated profit  $P(t)$ , industry structures  $S(t)$  and water resources  $W(t)$ . The initial values at  $t = 0$  are  $P(0) = 0, S(0) = 10, W(0) = p_{init}$ . There are five parameters that remain constant throughout the timeframe: the weight of accumulated profit  $p_{profit}$  and the weight of remaining water resources  $p_{water}$  in the performance index, the initial water resources  $p_{init}$ , the profit from industry per month  $p_{ind}$  and the water demand per industry  $p_{demand}$ . The values used for this parameters are summarized in Table I.

In addition, there are two control parameters, the demolition fraction  $x_{demol}(t)$ , the normal industry growth rate  $x_{grow}(t)$ . They can assume a different value at each time step, and define the solution space of the problem.

A performance index  $q(t)$  is computed, that is proportional to the accumulated profit  $P(t)$ , the weight of accumulated profit  $p_{profit}$ , the water resources  $W(t)$ , and the weight of water resource  $p_{water}$ :

$$q(t) = P(t) \cdot p_{profit} \cdot W(t) \cdot p_{water}. \quad (1)$$

The objective is to maximize the performance index  $q(t)$  at the end of the simulated time horizon for  $t = t_{max}$ . The three stocks have the following dynamical structure: The accumulated profit is described by

$$\frac{dP}{dt}(t) = g(t), \quad (2)$$

where  $g(t)$  denotes the profit per month, the industry structures is described by

$$\frac{dS}{dt}(t) = i(t) - d(t), \quad (3)$$

where  $i(t)$  is the creation rate of new industry structures and  $d(t)$  is the demolition rate, and the water resources are described by

$$\frac{dW}{dt}(t) = -c(t), \quad (4)$$

where  $c(t)$  is the water consumption rate. In the model, the water consumption actually represents a net flow, i.e., it accounts both for the water inflow and the water consumption which is the outflow. Given the contextual circumstances, the outflow is larger than the inflow. To simplify the model formulation, a net flow rate has been used.

The profit per month  $g(t)$  is proportional to the industry structures  $S(t)$  and the profit from industry per month  $p_{ind}$ :

$$g(t) = S(t) \cdot p_{ind}. \quad (5)$$

The creation rate of new industry structures  $i(t)$  is proportional to the normal industry growth rate  $x_{grow}$ , the effect of water

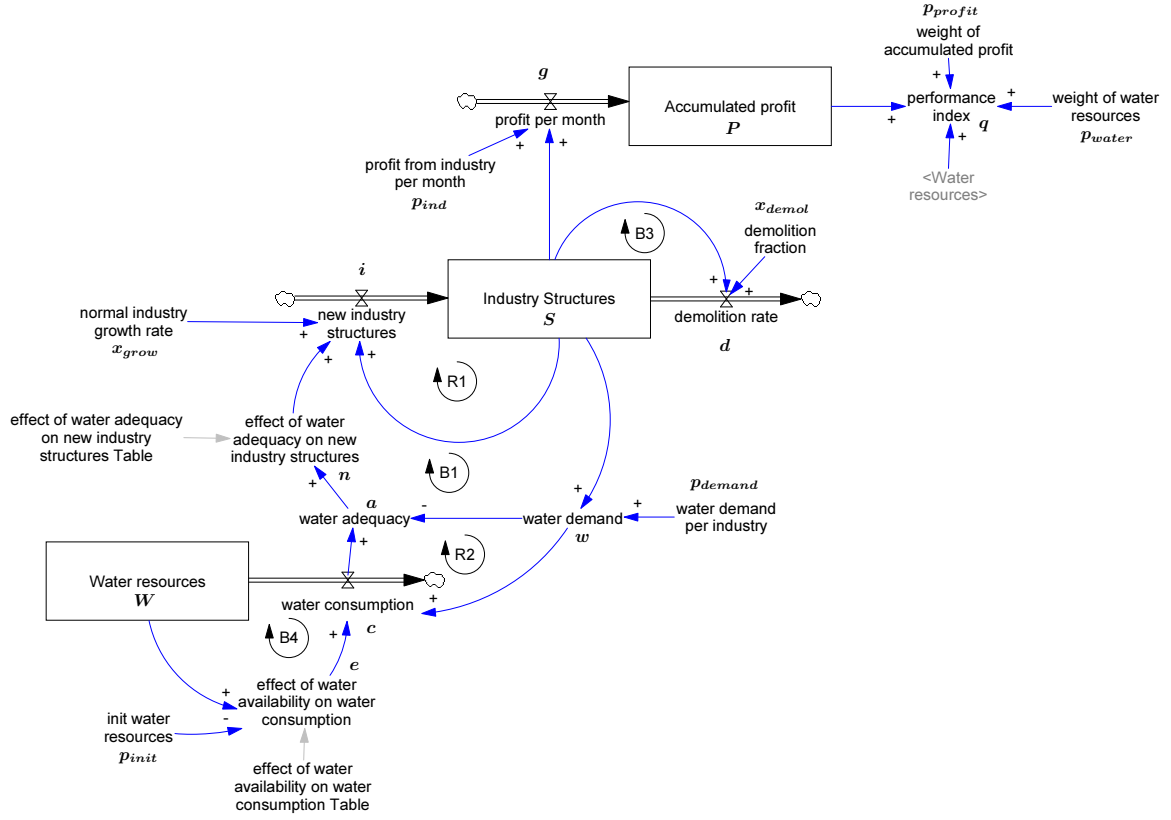


Fig. 1. The SD model.

adequacy on new industry structures  $n(t)$ , and the industry structures  $S(t)$ :

$$i(t) = x_{grow} \cdot n(t) \cdot S(t). \quad (6)$$

The demolition rate  $d(t)$  is proportional to the demolition fraction  $x_{demol}$  and the industry structures  $S(t)$ :

$$d(t) = x_{demol} \cdot S(t). \quad (7)$$

The effect of water adequacy on new industry structures  $n(t)$  is a function of the water adequacy  $a(t)$  from a linear interpolation of the data given in Table II. The water adequacy  $a(t)$  the ratio of the water consumption  $c(t)$  and the water demand  $w(t)$ :

$$a(t) = \frac{c(t)}{w(t)}. \quad (8)$$

The water demand  $w(t)$  is proportional to the industry structures  $S(t)$  and the water demand per industry  $p_{demand}$ :

$$w(t) = S(t) \cdot p_{demand}. \quad (9)$$

The water consumption  $c(t)$  is proportional to the effect of water availability on water consumption  $e(t)$  and the water demand  $w(t)$ :

$$c(t) = w(t) \cdot e(t). \quad (10)$$

The effect of water availability on water consumption  $e(t)$  is a function of the ratio  $f(t) = \frac{W(t)}{p_{init}}$  of water resource  $W(t)$  and the initial water resource which is set to  $p_{init}$  from a linear interpolation of the data given in Table II.

$\frac{a(t)}{f(t)}$	$\frac{n(t)}{e(t)}$
0	0
0.122324	0.0570175
0.2	0.14
0.3	0.25
0.4	0.395
0.5	0.535
0.6	0.685
0.7	0.825
0.8	0.92
0.9	0.975
1	1

TABLE II. EFFECT OF WATER ADEQUACY ON NEW INDUSTRY STRUCTURES  $n(t)$  AS A TABULAR FUNCTION OF THE WATER ADEQUACY  $a(t)$ . THE SAME DATA IS USED FOR THE EFFECT OF WATER AVAILABILITY ON WATER CONSUMPTION  $e(t)$  AS A FUNCTION OF  $f(t)$ , THE RATIO OF WATER RESOURCE  $W$  AND THE INITIAL WATER RESOURCE  $p_{init}$ .

The ranges for the decision variables are defined as follows:

$$0.05 \leq x_{grow}(t) \leq 0.2 \quad (11a)$$

$$10^{-4} \leq x_{demol}(t) \leq 0.1. \quad (11b)$$

Summing up, we aim to solve the following SDO problem:

$$\max\{p(t_{max}) \mid (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), x \in \mathbb{R}^2\}. \quad (12)$$

#### IV. SOLUTION TECHNIQUES

Our solution method for SDO problems is based on linear programming embedded in a branch-and-cut search procedure.

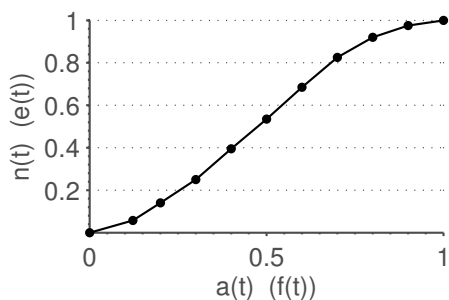


Fig. 2. Plot of piecewise linear functions  $a(t) \mapsto n(a(t))$  and  $f(t) \mapsto e(f(t))$ .

The nonlinear constraints are relaxed by a linear outer approximation, see [17], [18], [19], [20]. This relaxation can easily be solved by linear programming methods; we apply Dantzig's simplex algorithm [21], [22]. Afterwards the linear relaxation is improved by adding cutting planes around the computed optimal solution of the LP relaxation [23], [24]. These either cut off fractional solutions in order to get closer to integral solutions. This case is not needed in the case study we present here, because this model does not make use of integer variables. Or cutting planes improve the linear relaxation of the nonlinear function, which yields an improved linear outer approximation. This case is needed here. If cutting planes do not help (if they cut too little or add too much numerical difficulties), then one can finally resort to branching, where two new subproblems are created out of the current problem. In the case of integer variables present, one can round up and down the variable in each of these branches, respectively. In the case of continuous variables involved in nonlinear functions, one can add further local cuts that improve the linear outer approximation, but are only valid in the subtree induced by the branch.

This approach has been successfully applied to the solution of Mixed Integer Linear Programs as well as MINLPs from a range of applications (for example, see [25], [26], [27]). However, solving a control problem derived from a discretized dynamical system with a standard branch-and-cut solver is in many cases unsuccessful, since the solver does not take into account the special structure of the MINLP, that arises from the discretization, and from the handling of non-smooth functions via integer variables. Without considering this structure even finding a single feasible solution can exceed a reasonable time budget of several hours or even days (In the context of MILP this was also observed in [28].)

In order to improve the performance of a branch-and-cut approach to the solution of control problems, we have implemented two tailored methods for the solution of such problems. The first one is a primal heuristic which addresses the primal side of the branch and bound process, and the second one is a bound propagation presolving method, that addresses the dual side [29]. We will briefly describe the two algorithms in the following.

#### A. Time Discretization

In order to transcribe the continuous model into an MINLP, we need to discretize the differentials as well as the non smooth

functions. In order to discretize the differentials, we choose a simple time discretization of 20 steps with a step size of  $\Delta = 2.5$  month. We choose an explicit Euler scheme:

$$P(t+1) = P(t) + \Delta \dot{P}(t) \quad (13)$$

and analogously for the remaining states.

#### B. Piecewise Linear Functions

The model contains two non-smooth functions, i.e. the linear data interpolations  $n(a)$  and  $e(f)$ . In order to formulate these functions in our MINLP, we use a set of constraints and introduce at each time new sets of positive variables  $\lambda_{n,k}$  and  $\lambda_{e,k}$  where  $k \in \{0, 1, \dots, 10\}$  that are each part of a special ordered set of type two (SOS2), introduced by Beale and Forrest [30]. Out of a set of SOS2-Variables, at least two can be non-zero, and the two need to be adjacent. We define two vectors  $l_x$  and  $l_y$  containing the left and the right column of Table II, respectively:

$$l_x = (0, 0.122324, 0.2, \dots, 1) \quad (14a)$$

$$l_y = (0, 0.0570175, 0.14, 0.25, \dots, 1) \quad (14b)$$

The set of constraints that we need to implement at each point in time then reads for  $n(a)$  :

$$a = \sum_k l_{x,k} \lambda_{n,k} \quad (15a)$$

$$n = \sum_k l_{y,k} \lambda_{n,k} \quad (15b)$$

$$1 = \sum_k \lambda_{n,k} \quad (15c)$$

and analogously with the same vectors  $l_x, l_y$  for  $e(f)$ .

With these two steps, we have transformed the continuous problem into an MINLP. Instead of three state variables defined over our time frame, we now have 60 variables, each reflecting the values of the states at one given time. The same holds for the algebraic variables.

#### C. Primal Heuristic

As mentioned above, finding feasible solutions is a requirement for an efficient branch-and-cut approach. To quickly produce feasible solutions, we implemented a simple heuristic, that reduces the control problem to a simulation problem, by fixing the control variables to their lower (or in a second run upper) bound. For the industry structure problem, this will always yield a feasible solution. In the case of another problem with path constraints, this might lead to an infeasible solution. The heuristic solution checks for the violation of path constraints when solving the simulation problem. If a violation is detected, the heuristic can track back a given number of time steps and switch the control to the upper (lower) bound.

#### D. Bound propagation

In a branch-and-cut algorithm, bound propagation describes the derivation of bounds from one variable to another. This is done usually along a single constraint.

Our tailored bound propagation is executed only once as part of presolving. Its goal is to determine, which values of the

state and algebraic variables at each time are reachable with the given initial conditions and allowed control. In this context, the bound propagation can be considered a reachability analysis of the dynamic system.

In order to find the reachable values, we formulate subproblems  $s_{i,h}$  that contain all constraints and variables of the times  $t \in i-h, i-h+1, \dots, i$ . Within this subproblem we consider finding the maximal and minimal values for all states and algebraic variables again as optimization problems.

In more detail, our proposed bound propagation method iterates the following steps for each discretized time  $i$  except  $i = T$ , starting at  $i = 0$ :

- 1) Formulate the subproblem  $s_{i,h}$ .
- 2) For each algebraic variable  $v_i$  at time  $i$ :
  - a) Solve the maximization problem  $\max v_i$  subject to the constraints of  $s_{i,h}$  to optimality and set the upper bound of the variable  $\bar{v}_i$  to the solution value.
  - b) Solve the minimization problem  $\min v_i$  subject to the constraints of  $s_{i,h}$  to optimality and set the lower bound of the variable  $\bar{v}_i$  to the solution value.
- 3) For each differential variable  $w_{i+1}$  at time  $i+1$ :
  - a) Solve the maximization problem  $\max w_{i+1}$  subject to the constraints of  $s_{i,h}$  to optimality and set the upper bound of the variable  $\bar{w}_{i+1}$  to the solution value.
  - b) Solve the minimization problem  $\min w_{i+1}$  subject to the constraints of  $s_{i,h}$  to optimality and set the lower bound of the variable  $\bar{w}_{i+1}$  to the solution value.

The subproblems are solved with a time limit of 120 seconds. If the time limit is reached before the problem is solved to optimality, the considered bound is set to the best dual bound.

## V. COMPUTATIONAL RESULTS

We implemented our tailored methods as plug-ins to the branch-and-cut framework and solver SCIP [31], [32]. For the solution of the subproblems in bound propagation and for the branch-and-cut process that follows presolving, we will rely entirely on SCIP using the nonlinear solver IPOPT [33] and the linear solver SoPlex [34].

We run each of our calculations on one core of an HP machine, equipped with Intel Xeon E5-2690 2.90GHz processors and a total of 384 GB of memory.

To demonstrate the improvements resulting from our new methods, we attempted to solve the MINLP several times. As a reference run, we first tried to solve the control problem in SCIP without our plugins. Then, we started several runs, each using our primal heuristic and conducting the presolving bound propagation with a lookback parameter between 0 and 2.

The bounds that were derived from the presolving are depicted in Figure 3 for the state variable Industry Structures. We can clearly see, how with a higher parameter  $h$ , the bounds

TABLE III. COMPARISON OF SOLUTION PERFORMANCE

	Standard	$h = 0$	$h = 1$	$h = 2$
Time for presolve [m]	0.002	17	177	382
After presolving				
Primal Bound [1e5]	-	3.2	3.2	3.2
Dual Bound [1e5]	-	141.6	68.7	18.9
Gap [1]	Inf	42	20	4.8
At timelimit				
Primal Bound [1e5]	-	3.2	3.2	3.2
Dual Bound [1e5]	-	36.4	18.2	11.4
Gap [1]	Inf	10.2	4.6	2.5

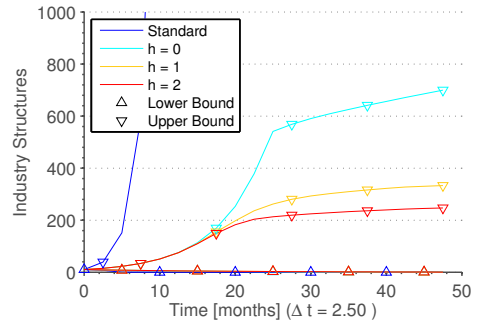


Fig. 3. Comparison of bounds calculated using standard bound propagation with bounds derived from improved presolving with several lookback parameters  $h$ . Upward pointing triangles mark lower, downward pointing triangles upper bounds.

become increasingly tighter, thereby providing better limits to the solution space.

We set a time limit of 1000 minutes for the solution process including presolving. In Table III, we summarized the state of the solution process after presolving, and at the time limit.

None of runs solved the problem to optimality in the given time limit. We can however clearly see, how the solution process improves when more time is spend in presolving. Without our plugins, neither a primal solution or a dual bound can be computed within the time limit. Turning on the primal heuristic, we immediately find a primal solution. The bound propagation also yields a dual bound for the problem that improves with higher lookback parameter.

Starting with a primal and dual bound, the branch-and-cut process is then again able to improve the lower bound.

As a quality measurement for the current solution, we use the so called gap  $g$ , defined in terms of the calculated bounds  $b_{dual}$  and  $b_{primal}$ :

$$g = \frac{b_{dual} - b_{primal}}{b_{primal}}. \quad (16)$$

With the highest considered lookback level  $h = 2$ , we arrive at the timelimit at a gap of 250%.

## VI. CONCLUSIONS

We presented a method to solve SDO problems, which are SD simulation models, extended by controllable parameters or functions and an objective function that should be maximized or minimized. For the solution of such methods we suggested

to apply mixed-integer nonlinear programming, a framework that could potentially lead to globally optimal solutions.

As a demonstrator example, we applied our techniques to an industry structure SD model that captures the sustainability of industry with respect to their water demand and consumption. Our methods were able to identify a solution and subsequently improve the lower bound to that solution to finally reach a gap of 250%.

Our current work focuses on an improvement of the solution speed and on closing the remaining gap. We aim at solving larger models in shorter time. This means, we need to achieve an even better understanding of the special structure inherited in MINLPs coming from SDOs. A further exploitation of this structure is a key for developing better solution methods that lead to faster numerical solvers.

#### ACKNOWLEDGEMENTS

We thank Robert Gottwald (ZIB) for implementing a parser that transforms Ventana Vensim models into GAMS models. Armin Fügenschuh conducted parts of his research as a Junior Fellow at the Zuse Institute Berlin (ZIB). This work is supported by the German Research Foundation DFG, as part of the Collaborative Research Center SFB-1026.

#### REFERENCES

- [1] J. Forrester, *Industrial Dynamics*. MIT Press, 1961.
- [2] G. Richardson, *Feedback Thought in Social Science and Systems Theory*. Pegasus Communications, Waltham, MA, 1999.
- [3] G. Richardson and A. Pugh, *Introduction to System Dynamics Modeling with DYNAMO*. MIT Press, 1983. [Online]. Available: <http://books.google.ca/books?id=urrMYgEACAAJ>
- [4] J. W. Forrester, "Principles of systems : text and workbook," Cambridge, Mass. Wright Allen Press, 1968. [Online]. Available: <http://opac.inria.fr/record=b1117113>
- [5] I. Rechenberg, *Evolutionsstrategie*. Holzmann-Frobog Verlag, Stuttgart, 1973.
- [6] H. Schwefel, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie mit einer vergleichenden Einführung in die Hill-Climbing- und Zufallsstrategie*. Birkhäuser Verlag, Basel, 1977.
- [7] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers and Operations Research*, vol. 13, no. 5, pp. 533 – 549, 1986.
- [8] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671 – 680, 1983.
- [9] V. Cerny, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, pp. 41 – 51, 1985.
- [10] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, Cambridge, MA, 1999.
- [11] A. Ruszczyński, *Nonlinear Optimization*. Princeton University Press, Princeton, NJ, 2006.
- [12] C. R. Brian Dangerfield, "An overview of strategy and tactics in system dynamics optimization," p. 405–423, 1996. [Online]. Available: <http://www.palgrave-journals.com/jors/journal/v47/n3/abs/jors199640a.html>
- [13] J. Hu, C. Wang, and G. Seliger, "Sustainable consumption of water by nega-resources approach," in *Design for Innovative Value Towards a Sustainable Society*, M. Matsumoto, Y. Umeda, K. Masui, and S. Fukushima, Eds. Springer Netherlands, 2012, pp. 27–32. [Online]. Available: [http://dx.doi.org/10.1007/978-94-007-3010-6\\_7](http://dx.doi.org/10.1007/978-94-007-3010-6_7)
- [14] A. Ford and H. Flynn, "Statistical screening of system dynamics models," *System Dynamics Review*, vol. 21, no. 4, pp. 273 – 303, 2005.
- [15] C. Kampmann and R. Oliva, "Loop Eigenvalue Elasticity Analysis: Three Case Studies," *System Dynamics Review*, vol. 22, no. 2, pp. 141 – 162, 2006.
- [16] M. Mojtahedzadeh, "Do Parallel Lines Meet? How Can Pathway Participation Metrics and Eigenvalue Analysis Produce Similar Results?" *System Dynamics Review*, vol. 24, no. 4, pp. 451 – 478, 2008.
- [17] E. Smith and C. Pantelides, "A Symbolic Reformulation/Spatial Branch-and-Bound Algorithm for the Global Optimization of nonconvex MINLPs," *Computers and Chemical Engineering*, vol. 23, pp. 457–478, 1999.
- [18] M. Tawarmalani and N. Sahinidis, "Global optimization of mixed-integer nonlinear programs: A theoretical and computational study," *Mathematical Programming*, vol. 99, pp. 563–591, 2004.
- [19] —, "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming*, vol. 103, pp. 225–249, 2005.
- [20] A. Fügenschuh, H. Homfeld, H. Schülldorf, and S. Vigerske, "Mixed-Integer Nonlinear Problems in Transportation Applications," in *Proceedings of the 2nd International Conference on Engineering Optimization (+CD-rom)*, 2010, h. Rodrigues et al. (Hrsg.), ISBN 978-989-96264-3-0.
- [21] G. B. Dantzig, *Linear programming and extensions*. Princeton University Press, 1963.
- [22] R. E. Bixby, "Solving Real-World Linear Programs: A Decade and More of Progress," *Operations Research*, vol. 50, no. 1, pp. 1–13, 2002.
- [23] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*. Wiley Interscience, 1988.
- [24] A. Fügenschuh and A. Martin, "Computational Integer Programming and Cutting Planes," in *Handbook on Discrete Optimization*, R. W. K. Aardal, G. Nemhauser, Ed. Elsevier, 2005, pp. 69–122.
- [25] O. Defterli, A. Fügenschuh, and G.-W. Weber, "Modern Tools for the Time-Discrete Dynamics and Optimization of Gene-Environment Networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 12, pp. 4768 – 4779, 2011.
- [26] R. Borndörfer, A. Fügenschuh, T. Klug, T. Schang, T. Schlechte, and H. Schülldorf, "The Freight Train Routing Problem," ZIB Technical Report ZR-13-36, Tech. Rep., 2013.
- [27] J. Humpola and A. Fügenschuh, "A Unified View on Relaxations for a Nonlinear Network Flow Problem," ZIB Technical Report ZR-13-31, Tech. Rep., 2013.
- [28] A. Fügenschuh, A. Dittel, S. Göttlich, and M. Herty, "MIP Presolve Techniques for a PDE-based Supply Chain Model," *Optimization Methods and Software*, vol. 24, no. 3, pp. 427 – 445, 2009.
- [29] A. Fügenschuh and I. Vierhaus, "A Global Approach to the Optimal Control of System Dynamics Models," in *31st International Conference of the System Dynamics Society, Cambridge, Massachusetts, USA, 2013*. [Online]. Available: <http://www.systemdynamics.org/web.portal?P1434+0>
- [30] E. Beale and J. Forrest, "Global Optimization Using Special Ordered Sets," *Mathematical Programming*, vol. 10, pp. 52 – 69, 1976.
- [31] T. Achterberg, "SCIP: Solving constraint integer programs," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 1–41, 2009.
- [32] T. Berthold, S. Heinz, and S. Vigerske, "Extending a CIP framework to solve MIQCPs," in *Mixed Integer Nonlinear Programming*, ser. The IMA Volumes in Mathematics and its Applications, J. Lee and S. Leyffer, Eds. Springer Verlag, Berlin, 2012, vol. 154, part 6, pp. 427–444.
- [33] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006. [Online]. Available: <http://dx.doi.org/10.1007/s10107-004-0559-y>
- [34] R. Wunderling, "Paralleler und objektorientierter Simplex-Algorithmus," Ph.D. dissertation, Technische Universität Berlin, Fachbereich Mathematik, 2006.