Proceedings of the 14th International Symposium on Process Systems Engineering – PSE 2021+
June 19-23, 2022, Kyoto, Japan

Process Superstructure Optimization through Discrete Steepest Descent Optimization: a GDP Analysis and Applications in Process Intensification

David E. Bernal[a*], Daniel Ovalle[a,c*], David A. Liñán[b], Luis A. Ricardez-Sandoval[b], Jorge M. Gómez[c], and Ignacio E. Grossmann[a]

[a] Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213, United States of America

[b] Department of Chemical Engineering, University of Waterloo, Ontario N2L 3G1, Canada

[c] Departamento de Ingeniería Química y de Alimentos, Universidad de Los Andes, Bogotá 111711, Colombia

[*] These authors contributed equally to this work.

grossmann@cmu.edu

Abstract

This manuscript introduces a Logic-based Discrete-Steepest Descent Algorithm (LD-SDA) to tackle problems arising from process superstructure optimization. These problems often appear in Process Systems Engineering and become challenging when trying to address Process Intensification applications. The current algorithm considers a disjunctive interpretation of these optimization problems through Generalized Disjunctive Programming (GDP). This formulation allows further analysis of the solution method as a tailored approach for GDP and results in a general open-source implementation of the method relying on the modeling paradigm Pyomo.GDP. Complementing our previous studies in the subject, we compare the LD-SDA against other well-known GDP solution methods and previous versions of D-SDA, not considering the disjunctive nature of these problems showcasing its advantages when dealing with superstructure problems arising from process intensification.

Keywords: superstructure optimization; process intensification; convex discrete analysis.

1. Introduction

The optimal design of processes is a challenge faced by the Process Systems Engineering (PSE) community. To remain competitive, chemical processes require a systematic procedure to find such optimal design. Recent developments from Process Intensification (PI) have shown to be promising alternatives to traditional processes by integrating and interconnecting units and achieving superior processes in terms of economic, environmental, and efficiency objectives (Sitter et al., 2019). Different process flowsheets can be integrated into a single process superstructure, where potential units and interconnections are considered. Superstructure models allow for the units and interconnections' equations to be constraints in optimization problems.

Since these equations can involve nonlinear functions and depend on both continuous (e.g., flowrates or temperatures) and discrete variables (e.g., equipment choice, interconnection location), the mathematical models become Mixed-Integer Nonlinear Programs (MINLP). The solution to these optimization problems is challenging given their combinatorial and nonconvex nature. Generalized Disjunctive Programming (GDP) has been proposed to tackle specific modeling and solution challenges of MINLP. In GDP, the modeling capabilities of traditional mathematical programming are extended by introducing Boolean variables involved in propositions and disjunctions.

The novelty of this work is to frame a Discrete-Steepest Descent Algorithm (D-SDA) for the solution of discrete nonlinear problems within the scope of GDP and use it to address process superstructure problems with ordered interconnections. Such problems arise in PI applications, such as studying a series of units with interunit refluxes, e.g., the tray-by-tray models in distillation columns.

2. Generalized Disjunctive Programming

In general, a GDP problem can be written as

$$\begin{gathered}\min_{\mathbf{x},\mathbf{Y},\mathbf{z}} f(\mathbf{x},\mathbf{z}) + \sum_{k\in K} c_k \\ \text{s.t. } \mathbf{g}(\mathbf{x},\mathbf{z}) \leq 0;\ \mathbf{\Omega}(\mathbf{Y}) = \text{True} \\ \bigvee_{i\in D_k} \begin{bmatrix} Y_{ik} \\ \mathbf{r}_{ik}(\mathbf{x},\mathbf{z}) \leq 0;\ c_k = \gamma_{ik} \end{bmatrix} \forall\, k \in K \\ \mathbf{x} \in X \subseteq \mathbb{R}^{n_x}; \mathbf{Y} \in \{\text{True},\text{False}\}^{n_y}; \mathbf{z} \in Z \subseteq \mathbb{Z}^{n_z}; \mathbf{c} \subseteq \mathbb{R}^{|K|}\end{gathered} \tag{1}$$

Where the continuous variables are denoted by the n_x-dimensional vector **x** bounded by the finite set X, and the discrete variables are denoted by the n_z-dimensional vector $\mathbf{z}$, bounded by the finite set Z. The function $f: \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \to \mathbb{R}$ is the objective function, and the vector function $\mathbf{g}: \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \to \mathbb{R}^{n_l}$ denotes the global inequality constraints. **Y** is a n_y-dimensional vector of logic variables, where for each disjunct $i \in D_k$ of each disjunction $k \in K$ the individual logic variable Y_{ik} enforce the set of inequalities $\mathbf{r}_{ik}: \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \to \mathbb{R}^{n_{ik}}$ and the fixed cost γ_{ik}. Logical constraints $\mathbf{\Omega}: \{\text{True},\text{False}\}^{n_y} \to \{\text{True},\text{False}\}$ encode logical relationships among the logical variables.

Besides offering a more intuitive modeling paradigm of discrete problems through disjunctions, a GDP model can inform computational solution tools of the original problem's underlying structure of the original problem, thus leading to improved solving performance. The GDP framework has successfully addressed problems derived from process superstructure optimization (Chen et al., 2021).

GDP problems are often solved by reformulating them as MINLP problems, by adding a binary variable y_{ik} for each Boolean variable Y_{ik}, and reformulating the constraints $\mathbf{r}_{ik}$ within the disjunctions to be enforced when the corresponding variable $y_{ik} = 1$ or trivially satisfied otherwise. The two best-known cases are the Big-M and the Hull reformulation, for which the Big-M case requires fewer continuous variables while the Hull reformulation is always at least as tight as the Big-M reformulation.

The tailored solution methods for GDP are usually based on generalizing algorithms for MINLP. The optimization problems are decomposed in a way where the discrete variables are fixed into what we call a discrete combination and allow to solve the problem only in terms of the continuous variables. Different methods are used to select the combination of these discrete variables, including branching across the different values the discrete variables can take (i.e., Branch-and-Bound) or solving a linear approximation of the original problem (Kronqvist et al., 2019). For GDP algorithms, contrary to the case in MINLP, these (possibly Mixed-Integer) Nonlinear Programming (NLP) subproblems that arise when fixing a particular discrete combination, now including the logical variables, only include the constraints that concern the logical variables within each combination. Namely, for a given logical combination $\hat{\mathbf{Y}}$ the subproblem becomes

$$\begin{gathered}\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x},\mathbf{z}) + \sum_{ik \text{ if } \hat{Y}_{ik}=\text{True}} \gamma_{ik} \\ \text{s.t. } \mathbf{g}(\mathbf{x},\mathbf{z}) \le 0 \\ \mathbf{r}_{ik}(\mathbf{x},\mathbf{z}) \le 0 \text{ if } \hat{Y}_{ik} = \text{True } \forall i \in D_k, k \in K \\ \mathbf{x} \in X \subseteq \mathbb{R}^{n_x}, \mathbf{z} \in Z \subseteq \mathbb{Z}^{n_z}\end{gathered} \tag{2}$$

Notice that in the most general case, the problem in Eq.(2) is an MINLP, although in most applications, $n_z = 0$, leading to the problem being an NLP. This problem avoids evaluating numerically challenging nonlinear equations whenever its corresponding logical variables are irrelevant (i.e., "zero-flow" issues).

The different tailored algorithms for GDP are defined in the strategy to find the logical combination $\hat{\mathbf{Y}}$ such that subproblems as in Eq.(2) solve to the optimal solution Eq.(1). One alternative is using gradient-based linearizations of the nonlinear constraints at the optimal solution of Eq.(2) to approximate the original problem feasible region. This defines a Mixed-Integer Linear Program (MILP) whose optimal solution returns values for the integer combinations. This method is known as the Logic-based Outer-Approximation (LOA) method. One can also systematically explore the values of the Boolean variables in a search tree where the nodes correspond to partial fixations of these variables, whose solutions provide bounds to the optimal solution, in a method called the Logic-based Branch-and-Bound (LBB) method (Chen et al., 2021). Both methods seek to find potentially optimal combinations of logical variables efficiently.

3. Discrete Steepest Descent Optimization

In a previous study, we presented the D-SDA (Liñán et al., 2020a) based on the theory of discrete convex analysis (Murota, 1998). The algorithm aims to solve Mixed-Binary Nonlinear Programs (MBNLP) and relies on reformulating the original discrete problem, in terms of binary variables, into a problem of integer choices, referred to as external variables. This reformulation was designed for binary variables defined in an ordered set constrained to an assignment constraint, meaning that only one of these ordered binary variables can be 1. These external variables, which are no longer representable in the original problem constraints, provide a concise representation of the discrete feasible region. This structure often appears in process superstructure optimization problems, e.g., when a binary variable defines the location of a reflux stream within a stages sequence, implicitly defining the existence of left-over stages after them.

Exploring discrete neighborhoods of the external variables provides the D-SDA with an efficient approach to choose which combination of the discrete variables should be considered to solve the subproblems that appear by fixing such values, NLPs in this case, thus efficiently searching the combinatorial space of the discrete variables. The D-SDA uses the integrally local optimality as a termination criterion (Murota, 1998), enabling the efficient solution of process superstructure optimization problems.

When considering a series of continuously stirred tank reactors (CSTR), the D-SDA outperforms MINLP solvers in solution time and quality (Liñán et al., 2020a). Furthermore, we applied the algorithm to the optimal design of a PI application involving reactive distillation, where we tackled the production of Ethyl tert-butyl ether (ETBE) from iso-butene and ethanol through the optimal design of a catalytic distillation column.

The D-SDA revealed a better performance against MINLP solvers when optimizing an economic objective in this problem (Liñán et al., 2020b). This allowed us to consider more complex models for this system, i.e., modeling multi-scale phenomena through a rate-based model for mass and energy transfer (Liñán et al., 2021).

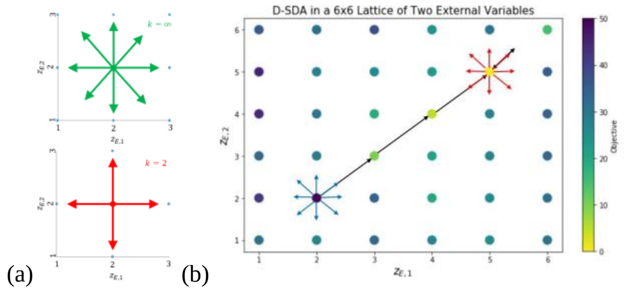

Figure 1. Pictorial representation of (a) different search neighborhoods in external variables lattice and (b) D-SDA with the neighbor and line search using $k = \infty$

4. Discrete-steepest descent optimization as a disjunctive algorithm LD-SDA

The problem in Eq. (1) suggests that the structure fitting for the D-SDA algorithm appears naturally in GDP, namely the disjunctions ($\vee_{i \in D_k} Y_{ik}$) enforce the assignment constraint, $\text{ExactlyOne}\left(Y_{1k}, \ldots, Y_{|D_k|k}\right)$ for the reformulation to be performed. This can also appear across other sets of Boolean variables $S = \{s_1, \ldots, s_{|S|}\}$ and if the set S is ordered, which is usually the case with process superstructures, then the reformulation into external variables $\mathbf{z}_E \in \{1, \ldots, |S|\} \subseteq \mathbb{Z}$ becomes $Y_{S(a)} \Leftrightarrow z_E = a$, where we denote the element of S in its a-th position as S(a). Notice that this reformulation also adds interpretability to the external variables, turning them into an indicator of the position within the set.

Such a reformulation allows us to map the Boolean variables into a lattice of integer variables, on which we can perform exploration based on ideas from discrete convex analysis (Murota, 1998). This leads to the Logic-based D-SDA (LD-SDA), which compares the objective function of each problem solved at a lattice site with its neighbors, defined by either a $k = 2$ or $k = \infty$ norm as seen in Fig.(1a), together with a line search along the direction provided by the best objective improvement after a complete neighbor search, as seen in Fig.(1b). The stopping criterion is determined by the local optimality of the solution compared to its neighbors, leading to certain convergence guarantees in the discrete convex (Murota, 1998).

From a GDP perspective, the external variables delineate a branching rule in the disjunctions, informing the problem structure. Notice that the more $\text{ExactlyOne}(Y_S)$ constraints the problem have, the more effective this reformulation is, with the limiting case of not having any other apart from the disjunctions, making it equivalent to LBB.

5. Numerical Experiments

We implement this method in open-source code using Python, available in https://github.com/bernalde/dsda-gdp. This code automatically transforms Pyomo.GDP (Chen et al., 2021) models, reformulates the disjunctions and the logical constraints $\text{ExactlyOne}\left(Y_{S(1)}, \ldots, Y_{S(|S|)}\right)$ automatically and solves the models using LD-SDA. We present the following two case studies after solving these problems with the solvers in GAMS 34.2 and using an Intel Core i7-7700 @ 3.6GHz PC with 16 Gb of RAM Memory.

5.1. Continuously Stirred Tank Reactors in series superstructure

We consider a superstructure of N_T CSTR in series where its total volume is minimized given an autocatalytic reaction $A + B \rightarrow 2B$ with 1st order reaction rate. This example is generalized from the one presented in (Liñán et al., 2020a). This example is illustrative given that we have an analytical solution at the limit of $N_T \rightarrow \infty$ equivalent to the Plug Flow reactor, and that we can explore the behavior in instances varying the value of N_T. Fig.(2) presents a scheme of the problem and its GDP formulation, together with its external variables reformulation.

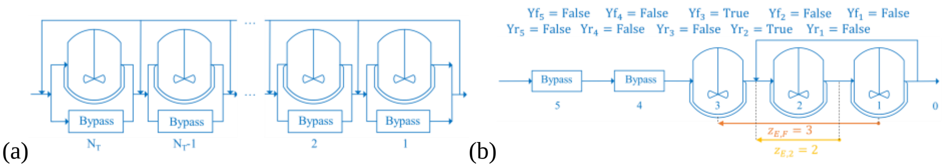

Figure 2. Scheme of (a) CSTR reactor superstructure and (b) Case with $N_T = 5$ and reformulation using external variables $z_{E,1}$ the number of reactors, $z_{E,2}$ the relative position of the reflux

For this instance of the problem, there is a locally optimal solution with five reactors and reflux before the first reactor $\mathbf{z}_E = (5,1)$.

We considered a whole set of different solver approaches to this problem with $N_T = [5, \dots, 25]$, including reformulating it into MINLP via Big-M and Hull reformulations, using LBB, LOA, and GLOA, and LD-SDA with two different norms, as seen in Fig.(3). We also include the total enumerations through the external variable reformulation.

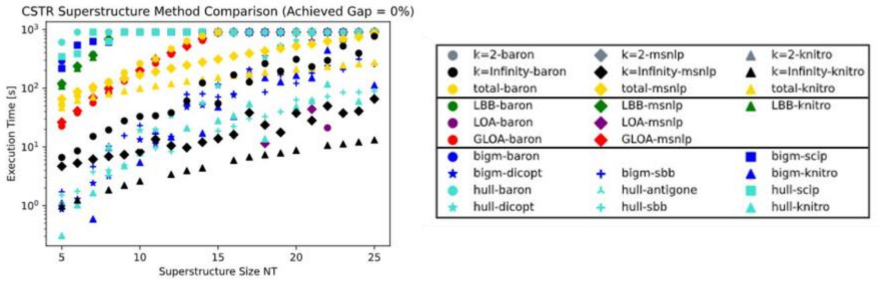

Figure 3. Execution time to achieve global minimum vs N_T for optimization of CSTR superstructure problem using different combinations of NLP solvers and reformulation methods.

From Fig.(3), one can see that LD-SDA provides the most efficient methods to solve this problem to global optimality. For this problem, the $k = 2$ norm neighborhood does not obtain the optimal solution. Notice that the external variable reformulation leads to a better search procedure, as seen when a total enumeration in this space can be more efficient than other GDP solution alternatives.

5.2. Rate-based catalytic distillation column

The economic objective maximization of a catalytic distillation column to produce ETBE from butenes and methanol was solved using a D-SDA (Liñán et al., 2021). This test case is relevant since it deals with a PI design problem, where several traditional optimization methods fail even to compute feasible solutions (Liñán et al., 2021). The derivation of the D-SDA method was initially motivated to address this PI superstructure optimization problems, leading to a series of papers as seen in (Liñán et al., 2020b, 2020a, 2021)

The previous D-SDA would tackle the problem as an MBNLP, fixing and unfixing binary variables and including constraints of the form $yg(x) \leq 0$ to enforce the logic constraints.

Here we show that, when considering the problem from a disjunctive point of view as in LD-SDA, leads to the solution of subproblems as in Eq.(2) instead of including irrelevant and numerically challenging nonlinear constraints.

Table 1. Execution time of Catalytic distillation optimal design problem from (Liñán et al., 2021)

	D-SDA: (Liñán et al., 2021)				**LD-SDA: This work**			
NLP Solver	CONOPT		MSNLP		CONOPT		MSNLP	
Neighborhood	k=2	k=inf	k=2	k=inf	k=2	k=inf	k=2	k=inf
Time [s]	367.1	16880.0	3626.0	102030.7	118.7	6751.1	2000.0	38532.5

As seen in Table (1), the proposed LD-SDA method leads to speedups up to 3x in this challenging PI problem. We could obtain the same solution to all subproblems more efficiently, given that only the relevant constraints were included for each problem. Adding to the fact that the previous results using the D-SDA were already beating state-of-the-art MINLP solution methods shows the advantages of the LD-SDA.

6. Conclusions

The current manuscript presents the usage of a disjunctive discrete steepest descent optimization algorithm LD-SDA to tackle process superstructure problems. This algorithm is presented from the perspective of Generalized Disjunctive Programming solution methods, showing its relationship with existing algorithms for GDP. Moreover, this allowed for the algorithm to be implemented in Python and through the modeling paradigm of Pyomo.GDP. With this implementation, we solved problems of superstructure optimization, a series of CSTR volume minimization, and a rate-based catalytic distillation column economical design more efficiently than other proposed solution methods. These solution methods include MINLP reformulations, GDP-tailored algorithms, and a previous version aimed at MBNLP problems. The results in this manuscript show how LD-SDA becomes a valuable tool to address process superstructure problems, of which many challenges instances arise from PI applications.

References

Chen, Q., Johnson, E. S., Bernal, D. E., Valentin, R., Kale, S., Bates, J., Siirola, J. D., & Grossmann, I. E. (2021). Pyomo.GDP: an ecosystem for logic based modeling and optimization development. *Optimization and Engineering*. https://doi.org/10.1007/s11081-021-09601-7

Kronqvist, J., Bernal, D. E., Lundell, A., & Grossmann, I. E. (2019). A review and comparison of solvers for convex MINLP. *Optimization and Engineering*, 1–59. https://doi.org/10.1007/s11081-018-9411-8

Liñán, D. A., Bernal, D. E., Gómez, J. M., & Ricardez-Sandoval, L. A. (2021). Optimal synthesis and design of catalytic distillation columns: A rate-based modeling approach. *Chemical Engineering Science*, *231*, 116294. https://doi.org/10.1016/J.CES.2020.116294

Liñán, D. A., Bernal, D. E., Ricardez-Sandoval, L. A., & Gómez, J. M. (2020a). Optimal design of superstructures for placing units and streams with multiple and ordered available locations. Part I: A new mathematical framework. *Computers & Chemical Engineering*, *137*, 106794.

Liñán, D. A., Bernal, D. E., Ricardez-Sandoval, L. A., & Gómez, J. M. (2020b). Optimal design of superstructures for placing units and streams with multiple and ordered available locations. Part II: Rigorous design of catalytic distillation columns. *Computers & Chemical Engineering*, *139*, 106845.

Murota, K. (1998). Discrete convex analysis. *Mathematical Programming, Series B*. https://doi.org/10.1007/bf02680565

Sitter, S., Chen, Q., & Grossmann, I. E. (2019). An overview of process intensification methods. In *Current Opinion in Chemical Engineering*. https://doi.org/10.1016/j.coche.2018.12.006