Available online at www.tjnsa.com
J. Nonlinear Sci. Appl. 8 (2015), 695–709
Research Article

A global optimization approach for a class of MINLP problems with applications to crude oil scheduling problem

Qianqian Duan[a], Genke Yang[a], Guanglin Xu[b,*], Xueyan Duan[c]

[a]*Department of Automation, Shanghai Jiao Tong University, Dongchuan Road 800, Shanghai, China.*
[b]*College of Mathematics and Information, Shanghai Lixin University of Commerce, China.*
[c]*School of Economics and Management, Shanghai Maritime University, China.*

Communicated by Chen Yong-Zhuo

Abstract

A global optimization algorithm is proposed to solve the crude oil schedule problem. We first developed a lower and upper bounding model by using a multiparametric disaggregation method. Secondly, the lower and the upper bounding models combined with finite state method (FSM) are incorporated to solve the bilinear programing problem jointly. The advantage of using FSM is that we can generate promising substructure and partial solution. Furthermore, the FSM can guarantee that the entire solution space is uniformly covered. Therefore, the algorithm has better global performance than some existing algorithms. Finally, a real-life crude oil scheduling problem from the literature is used for conducting simulation. The experimental results validate that the proposed method outperforms commercial solvers.

Keywords: MINLP, finite state method, hybrid optimization.
2010 MSC: 90C11, 90C26, 90C56.

1. Introduction

Planning and scheduling of the flow of crude oil is a very important problem in petroleum refineries due to the potential realization of large cost savings and improved feeds. Detailed scheduling models often require

*Corresponding author
Email addresses: dqq1019@sjtu.edu.cn (Qianqian Duan), sjtu1019@163.com (Genke Yang), glxu@outlook.com (Guanglin Xu), 328236505@qq.com (Xueyan Duan)

Received 2015-3-15

a continuous time representation that usually includes nonlinear equations, as well as binary variables to model discrete decisions, which gives rise to mixed-integer nonlinear programming (MINLP) models.

The MINLP corresponding to the scheduling problem is no-convex due to the presence of bilinear terms in some of the mass balance constraints, and hence the standard methods for solving MINLPs [5] may fail to converge to a solution or may lead to sub-optimal solutions. The global optimization of general no-convex bilinear programs has received significant attention in the literature [9, 17, 10, 16, 1, 6, 18, 13, 3]. The convex McCormick envelopes [2] coupled with spatial branch and bound search frameworks have been the basis for many of these global optimization techniques, with piecewise McCormick envelopes being a more recent development. Variations of this approach have been suggested, generalizing the convex envelopes to piecewise over- and under-estimators [2, 22] . Some researchers have proposed techniques to linearize these bilinear constraints. Sherali and Alameddine [17] proposed a new reformulation-linearization technique (RLT) and imbedded it within a provably convergent branch-and-bound algorithm. This RLT process yields a LP problem whose optimal value provides a tight lower bound on the optimal value of the bilinear programming problem. Pan [15] set up an MINLP formulation for the crude oil scheduling problem and proposed some heuristic rules collected form expert experience to linearize bilinear terms and fix some binary variables in the MINLP model, resulting into an MIP model with fewer binary variables.

As discussed above, bilinear term is the most important feature in optimizing crude oil scheduling. With this issue, scheduling problems have to be described as MINLP models with both discrete and continuous time representations, thus making a difficult procedure of finding feasible solution. The existing approaches used lots of continuous variables and constraints to linearize bilinear terms. However, when time horizon became longer, they needed to solve more complex models, which increased overall computational time and affected final objective values.

A global optimization algorithm is proposed in this paper. Firstly, we describe the multiparametric disaggregation method for the nonlinear term, which gives the lower bounding and upper bounding problem, then a global optimization algorithm combing the FSM is described to solve the problem. The algorithm based FSM builds the initial points complying with sequencing rules and operation condition. Thus, the search space of the algorithm is substantially reduced as only legal initial condition is explored.

This paper is organized as follows. Section 2 presents the upper and lower bounding formulations of the crude scheduling problem, while Section 3 provides the deterministic finite automation (DFA) model of the schedule using the FSM . A discussion of the algorithm is given in Section 4. Section 5 presents the different examples on which the algorithm was applied, and finally, Section 6 summarizes some conclusions.

2. Mathematic Model

In this section, we present the derivation of the multiparametric disaggregation technique (MDT) by Teles [20] for solving no-convex bilinear programs. Both upper and lower bounding formulations ,which corresponding to the refinery crude oil scheduling problem described in Appendix A, are derived using disjunctive programming and exact linearization.

2.1. Radix-Based Discretization

We start with reformulating the bilinear terms $L_{irc} = f_c \cdot L_{ir}$ and $V_{irc} = f_c \cdot V_{iv}$. For simplicity, we will denote the bilinear term $y = f \cdot x$. The technique known as radix-based discretization described by Kolodziej [14] and Castro [25] is used to discretize the bilinear term. The product $y = f \cdot x$ can be represented exactly with the following constraints and disjunction:

$$y = f \cdot x \tag{2.1}$$

$$f = \sum_{l \in \mathbb{Z}} \lambda_\ell \tag{2.2}$$

$$\vee_{k=0}^{9} [\lambda_\ell = 10^\ell \cdot k]. \tag{2.3}$$

where f is discretized through the disjunction in (2.3) that selects one digit k for each power $l \in$. Considering the convex hull reformulation of the disjunction[18] in (2.3) by introducing the disaggregated variables, leads to the following equations where $K = \{k|k = 0, 1, \dots 9\}$:

$$\lambda_\ell = \sum_{k=0}^{9} \widehat{\lambda}_{k\ell} \tag{2.4}$$

$$\widehat{\lambda}_{k\ell} = 10^\ell \cdot k \cdot z_{k\ell} \tag{2.5}$$

$$\sum_{k=0}^{9} z_{k\ell} = 1, \tag{2.6}$$

$$z_{k\ell} \in \{0, 1\}. \tag{2.7}$$

Substituting equations (2.5) into equation (2.4) and substituting equation (2.4) into equation (2.2) leads to the fully discretized of f:

$$f = \sum_{\ell\in\mathbb{Z}} \sum_{k=0}^{9} 10^\ell \cdot k \cdot z_{k\ell}. \tag{2.8}$$

Considering the product $y = f \cdot x$ by substituting equation (2.8) into equation (2.1) leads to:

$$y = x \cdot [\sum_{\ell\in\mathbb{Z}} \sum_{k=0}^{9} 10^\ell \cdot k \cdot z_{k\ell}] = \sum_{\ell\in\mathbb{Z}} \sum_{k=0}^{9} 10^\ell \cdot k \cdot x \cdot z_{k\ell},$$

Which involves the nonlinear term $x \cdot z_{k\ell}$. A new continuous variable, $\widehat{x}_{k\ell} = x \cdot z_{k\ell}$ is performed an exact linearization [14]:

$$y = \sum_{\ell\in\mathbb{Z}} \sum_{k=0}^{9} 10^\ell \cdot k \cdot \widehat{x}_{k\ell}. \tag{2.9}$$

Finally, multiplying equation (2.6) by x ,and results in (2.10).

$$\sum_{k=0}^{9} \widehat{x}_{k\ell} = x. \tag{2.10}$$

2.2. Lower bound model

The variable f is presented to a finite level of precision $f^{'}$, which leading to approximate representation of the bilinear term $y^{'}$. The constraints in (2.8) (2.9) are modified in (2.11) (2.12) to allow for a maximum power of $10(P)$ and a minimum power of $10(p)$. For the remaining constraints, it suffices to replace $l \in \mathbb{Z}$ with $l \in \{p, p+1, , P\}$. These sets of constraints correspond to the bilinear terms proposed by Teles et.al.[20]:

$$f^{'} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^\ell \cdot k \cdot z_{k\ell}, \tag{2.11}$$

$$y^{'} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^\ell \cdot k \cdot \widehat{x}_{k\ell}. \tag{2.12}$$

This problem (P) in Appendix A can be reformulated by discretizing the bilinear terms $L_{irc} = f_c \cdot L_{ir}$ and $V_{irc} = f_c \cdot V_{iv}$ .We can then obtain the following MILP approximation:

$$\mathbf{P}^{'} \qquad max \quad G^{'} = \sum_{i\in T}\sum_{r\in R_D}\sum_{v\in I_r}\sum_{c\in I_C} G_c \cdot V_{ivc}$$

$$\begin{cases} L_{irc} = \sum_{\ell=p}^{P}\sum_{k=0}^{9} 10^{\ell}\cdot k\cdot \widehat{L}_{irck\ell} \\ V_{irc} = \sum_{\ell=p}^{P}\sum_{k=0}^{9} 10^{\ell}\cdot k\cdot \widehat{V}_{ivck\ell} \\ f_c = \sum_{\ell=p}^{P}\sum_{k=0}^{9} 10^{\ell}\cdot k\cdot z_{ck\ell} \\ L_{ir} = \sum_{k=0}^{9} \widehat{L}_{irck\ell} \\ V_{ir} = \sum_{k=0}^{9} \widehat{V}_{ivck\ell} \\ \sum_{k=0}^{9} z_{ck\ell} = 1 \\ z_{ck\ell} \in \{0,1\} \\ (A2-A18) \end{cases}$$

Because the problem $(P^{'})$ is a maximization problem, problem $(P^{'})$ yields a lower bound for the original problem (P).Because the current feasible region is the approximated one of the original problem. By analogy, we can derive problem (P^R), which yields a relaxation (and thus an upper bound) of the original problem.

2.3. Upper bound model

By adding a slack term $\triangle f$ to the above discretized variable $f^{'}$,the continuous representation f^R is derived by Kolodziej [20]:

$$f^R = f^{'} + \triangle f \tag{2.13}$$

$$f^R = \sum_{\ell=P}^{P}\sum_{k=0}^{9} 10^{\ell}\cdot k\cdot z_{k\ell} + \triangle f \tag{2.14}$$

$$0 \le \triangle f \le 10^{p}. \tag{2.15}$$

For the continuous representation of the bilinear term y^R, note that:

$$y^R = f^R \cdot x = (f^{'} + \triangle f)\cdot x = y^{'} + \triangle y. \tag{2.16}$$

The bilinear term $\triangle y$ can be relaxed using the McCormick envelop, which coincides with the RLT bound factor products $x^L \le x \le x^U$ and $0 \le \triangle f \le 10^p$:

$$x^L \cdot \triangle f \le \triangle y \le x^U \cdot \triangle f \tag{2.17}$$

$$\triangle\, y \ge (x - x^U)\cdot 10^p + x^U \cdot \triangle f \tag{2.18}$$

$$\triangle\, y \le (x - x^L)\cdot 10^p + x^L \cdot \triangle f \tag{2.19}$$

Introducing these constraints into problem (P) in Appendix A, we can derive a relaxation of the original problem (P) that is also an MILP problem (P^R):

$$\mathbf{P}^R \qquad max \quad G^R = \sum_{i\in T}\sum_{r\in R_D}\sum_{v\in I_r}\sum_{c\in I_C} G_c \cdot V_{ivc}$$

$$\begin{cases} L^R_{irc} = \sum_{\ell=p}^{P}\sum_{k=0}^{9} 10^{\ell}\cdot k\cdot \widehat{L}_{irck\ell} + \triangle L_{irc} \\ V^R_{irc} = \sum_{\ell=p}^{P}\sum_{k=0}^{9} 10^{\ell}\cdot k\cdot \widehat{V}_{ivck\ell} + \triangle V_{irc} \\ L^L_{ir}\cdot \triangle f_c \le \triangle L_{irc} \le L^U_{ir}\cdot \triangle f_c \\ \triangle L_{irc} \ge (L_{ir} - L^U_{ir})\cdot 10^p + L^U_{ir}\cdot \triangle f_c \\ \triangle L_{irc} \le (L_{ir} - L^L_{ir})\cdot 10^p + L^L_{ir}\cdot \triangle f_c \end{cases}$$

$$\begin{cases} V_{iv}^L \cdot \triangle f_c \le \triangle V_{ivc} \le V_{iv}^U \cdot \triangle f_c \\ \triangle V_{ivc} \ge (V_{iv} - V_{iv}^U) \cdot 10^p + V_{iv}^U \cdot \triangle f_c \\ \triangle V_{iv} \le (V_{iv} - V_{iv}^L) \cdot 10^p + V_{iv}^L \cdot \triangle f_c \\ L_{ir} = \sum_{k=0}^{9} \widehat{L}_{irck\ell} \\ V_{ir} = \sum_{k=0}^{9} \widehat{V}_{ivck\ell} \\ \sum_{k=0}^{9} z_{ck\ell} = 1 \\ 0 \le \triangle f_c \le 10^p \\ z_{ck\ell} \in \{0, 1\} \\ (A2 - A18) \end{cases}$$

This problem (P^R) is a relaxation of problem (P) because it includes at least (and in most cases more than) the entire feasible region of problem (P).

3. Model of the schedule

In the MINLP model [12], it has an obvious and direct correspondence between binary solution and schedule. So it is important to construct a model which builds legal schedules complying with sequencing rules and operation condition. Using the finite state method, the model of the legal schedule constitutes a reasonable framework, capturing all possible schedules and removing many redundant sequences of operations.

3.1. Schedule of Operations

In the SOS representation [12], at most one operation can be assigned to each priority. It is therefore possible to represent the scheduling solution by a schedule. We used an example in Figure 1 to show how binary solution can be mapped to a schedule. The schedule $S = [7683513762]$ where 7 stands for the specific operation 7 to assign to position 1 corresponding to the binary decisions $Z_{17} = 1$.

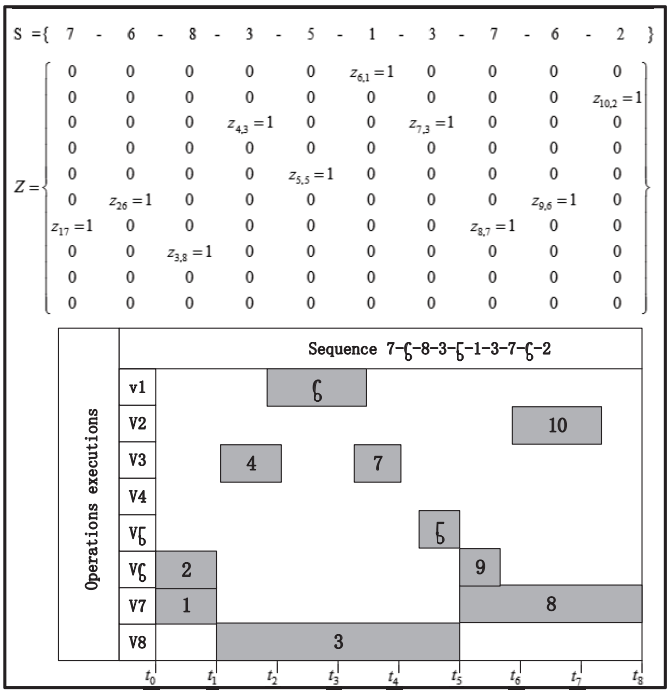

Figure 1: An example to indicate the relationship between variable and schedule

3.2. The rule of the schedule

A feasible schedule S should comply with a number of rules. The rules in this problem [12] usually fall into the following classes:

–Sequencing rules:

Specify the allowed and forbidden sequences of operations.

–Valid Length rule:

The total number of operations executed in the schedule is postulated a prior.

3.2.1. The Sequencing Rule

Here, we still use the instance with 8 operations from Mouret et al. [12] to describe an efficient sequencing rule by using a regular expression. A feasible schedule S can be described by the following:

$$sequenceL_a = (\varepsilon + L_7)(L_8 \cdot L_7)^*(\varepsilon + L_8)$$
$$L_7 = 7(\varepsilon + 4)(\varepsilon + 6)(\varepsilon + 1 + 14)(\varepsilon + 2 + 26)$$
$$L_8 = 8(\varepsilon + 3)(\varepsilon + 5)(\varepsilon + 1 + 13)(\varepsilon + 2 + 25),$$

Where $1 - 8 \in W$, W is the set of all operations.

The automation $M_1 = (Q_1, \sum, \sigma_1, q_0^1, F_1)$ in Figure 2 3 on the basis of this regular expression sequence L_a , captures all possible schedule and removes many redundant schedules. For example, in Figure 4, schedule (a) is accepted by the automation M_1 while the other schedule is rejected and thus not explored during the search. However, as to correctness, this automation M_1 suffers from a serious problem of over generation. For example: the short length of the sequence may lead to infeasibility, while the long length of the sequence may result in an unsolvable model. It is an interesting challenge for finite state syntactic description to specify a sublanguage that contains all and only the sequences of valid length.

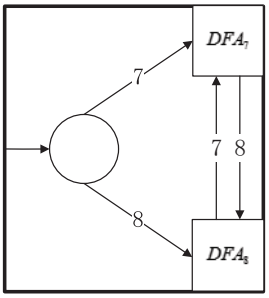

Figure 2: The automation M_1 on the basis of this regular expression sequence L_a

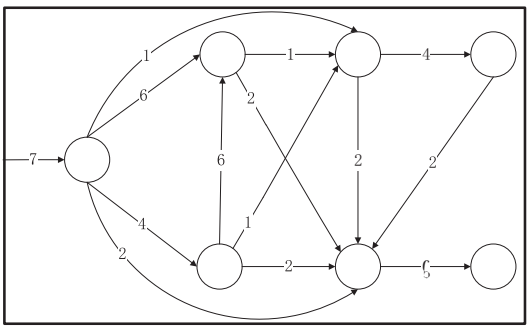

Figure 3: The automation DFA_7 on the basis of this regular expression sequence L_7

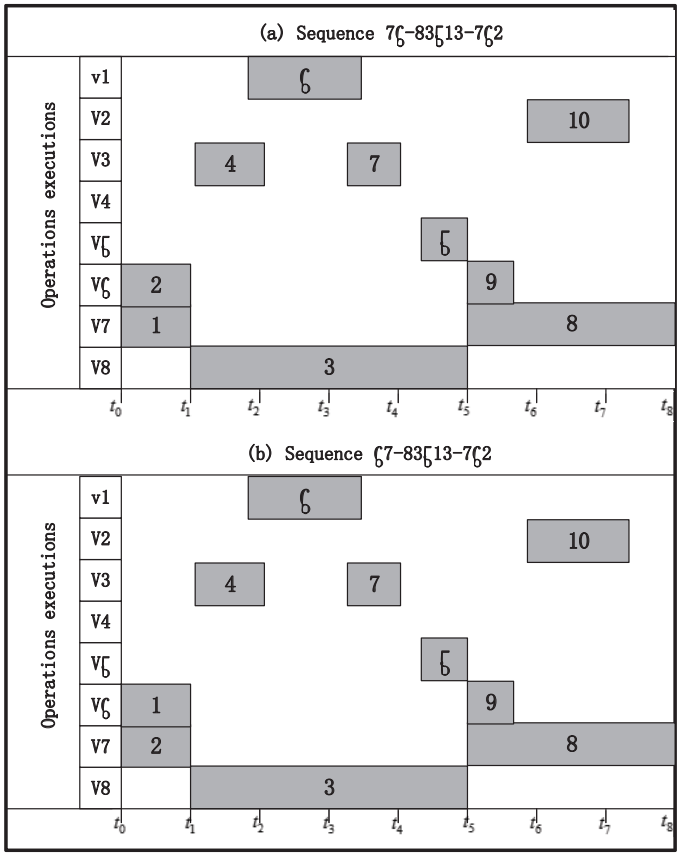

Figure 4: The example of schedule according the automation M_1

3.2.2. The Valid Length rule

One particular benefit of this model [12] is that the only parameter that needs to be postulated a prior is the length of the schedule. In order to satisfy the condition which the length of the schedule has to be postulated a priori, The valid length rule is defined to constraint the schedule S:

$$ValidLengthL_b = [1+2+3+4+5+6+7+8]^T,$$

Where $1-8 \in W$, W is the set of all operations.

The automation $M_2 = (Q_2, \sum, \sigma_2, q_0^2, F_2)$ in Figure 5 on the basis of this regular expression L_b , accepts only schedule sequences of length T but excludes all others. Then, the number of executions of operations is fixed and known in advanced, guaranteeing global optimality of the solution obtained. For example, in Figure 6, given the length 10, schedule (a) is accepted by the automation M_2 while the other schedule is rejected and thus not explored during the search.

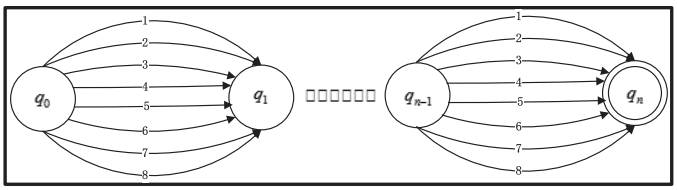

Figure 5: The automation M_2 on the basis of this regular expression sequence L_b

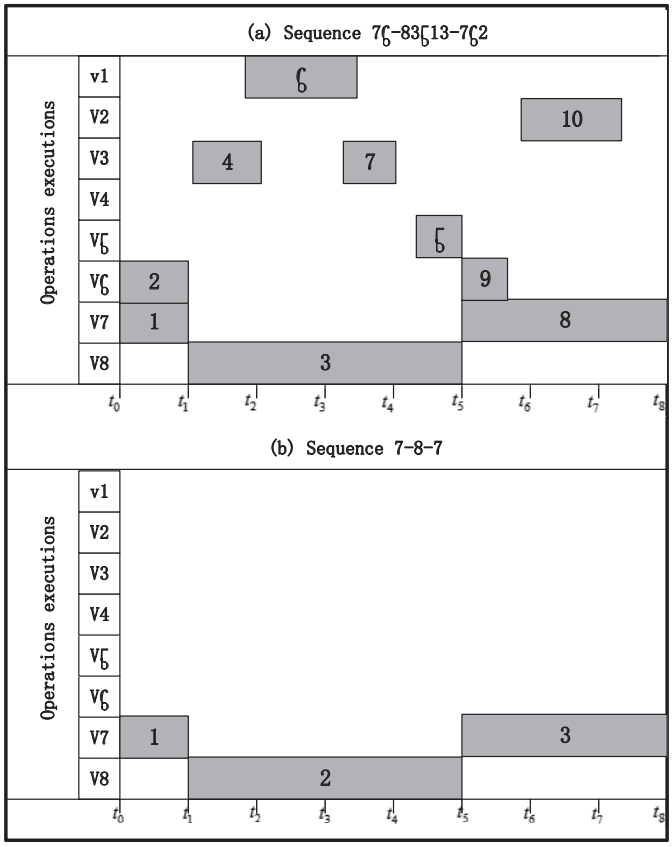

Figure 6: The example of schedule according the automation M_2

3.3. Model of the legal schedule

The substructure of this problem is to build legal schedule complying with a number of ordering or cardinality rules. The sequencing rules and length rules should be satisfied simultaneously. One may envisage sometimes to the intersection operation of regular language, with an automata, in order to model all sequencing and length rules.

The desired effect (model/ legal sequence) can be obtained by intersecting M_1 describing the sequencing rule with the M_2 describing the valid length rule.The intersection of the two models which satisfying the sequencing and length rules simultaneously:

$$leglalscheduleS = SequenceM_1 \cap ValidLengthM_2.$$

We could design a DFA $M = (Q, \sum, \sigma, q_0, F)$, that recognizes the leglal schedule S. The steps are as following:

Step1: Given the DFA $M_1 = (Q_1, \sum, \sigma_1, q_0^1, F_1)$,which recognize the regular expression sequence L_a .

Step2: Given the DFA $M_2 = (Q_2, \sum, \sigma_2, q_0^2, F_2)$,which recognize the regular expression sequence L_b .

Step3: Design a DFA $M = (Q, \sum, \sigma, q_0, F)$,which obtained by intersecting M_1 with M_2.The DFA M accepts strings if both M_1 and M_2 accepts.

We have now completed the task of describing the language of valid sequences from the set of possible sequence expressions. It is also able to create an automation, and then generate all possible schedules S accepted by the automaton. The processes are implemented using FSA Utilities [21] that is a package for implementing and manipulating DFA.

4. Global Algorithm

Large-scale MINLPs such as problem [12] require specialized solution algorithm. We propose a specialized global algorithm for solving the no-convex model to global optimality within a specified tolerance. In the proposed technique, the upper and lower bounding schemes described in Section 2 can be combined into a global optimization algorithm. First, the following property can be established:

4.1. Property

As p approaches $-\infty$,$G^{'}$ approaches G^R.

Proof: As p approaches $-\infty$ in (P^R)

$lim_{p\to-\infty} \triangle f = lim_{p\to-\infty} 10^p = 0$

$lim_{p\to-\infty} \triangle y = x^U \cdot lim_{p\to-\infty} \triangle f = (x - x^L) \cdot lim_{p\to-\infty} 10^p + x^L \cdot lim_{p\to-\infty} \triangle f = 0$

Thus, since the variables $\triangle f$ and $\triangle y$ are eliminated,this yields $G^{'}$ approaches G^R.

From Property [7], we can establish that as precision is increased (i.e. p approaches $-\infty$), both $P^{'}$ and P^R converge to the same value. Assuming P is large enough such that $10^p \geq f^U$,we can further state that $P^{'}$ and P^R converge such that $G^{'} = G^R = G$.

4.2. Global Algorithm

From property, it is noted that the relaxation problems derived from multiparametric disaggregation, convergence to the global optimal solution of the original problem can only be guaranteed for an infinite number of discretization points .Thus, the DFA model in section 3 which guarantee that the entire solution space is uniformly covered is combined to solve the MINLP, The flowchart for the algorithm is given in Figure 7, and the steps for our algorithm are given as follows:

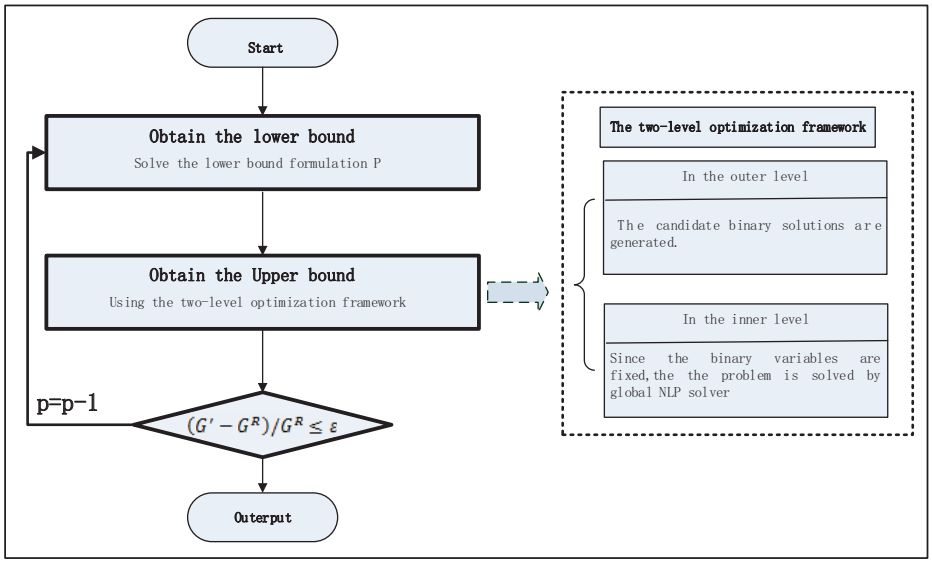

Figure 7: Flowchart for the global algorithm

Step1: Initialization.

Choose $p = P = \lfloor log_{10} f^U \rfloor$.

Step2: Obtain the lower bound.

Solve (P^R) to obtain the lower bound G^R.

Step3: Obtain the upper bound.

It is implemented in a two-level optimization framework. In outer level, a population of candidate binary solutions is generated by randomly selecting from the population of the all possible binary decisions (see **Optimization Algorithm**). In inner level, since the binary variables are fixed, the problem $(P^{'})$ becomes a no convex NLP. This is solved by using any local or global NLP solver. At last, we update the candidate binary solutions, until the best solution is found in the outer level.

Step4: Update.

If $(G^{'} - G^{R})/G^{R} \leq \varepsilon$,STOP,the solution is globally optimal.Otherwise,set $p = p - 1$,and return to step 1.

4.3. Optimization Algorithm

In the global algorithm, optimization in the space of binary variables is described in this section. Finite state method is used to generate the feasible binary variable for the inner NLP. Each candidate or individual then is sent to an inner loop for optimization, which is very important in the two-level optimization. It is a combinatorial optimization problem, since their computation time increases exponentially with problem size.

Step1: Determines the model of structure.

Using the method in Section 3, the automation of valid schedules is created.

Step2: Generate all possible schedules.

Based on the automation, and it is possible to generate all possible schedules accepted by the automaton. The processes are implemented using FSA Utilities [21] that is a package for implementing and manipulating DFA.

Step3: Determine the binary variable.

When all possible schedules accepted by the automaton are generated, the according possible binary decisions are generated.

The method is based on a reasonable structure, capturing all possible schedules and removing many redundant sequences of operations. Initial values of binary variables satisfy the equality constraints and operation conditions, and therefore represent a feasible operating point.

5. Experimental study

To illustrate the performance of the proposed global algorithm, we consider 13 examples, in which three examples are taken from the Mouret et al. [12] and the other ten examples are extension of the former three examples. Tables 15 provide the complete data of these examples. Two state of the art global solvers GloMIQO [11] and BARON [19] have also been used for comparison. For the global optimization algorithm, the relative tolerance was set to $\varepsilon = 10^{-4}$ and the maximal runtime is set to $CPU^{MAX} = 3600CPUs$. The hardware consisted on an Intel Core i5-4300(1.9 GHz) processor, with 8 GB of RAM running, Windows 8 Professional.

For interpretation of the results ,we use one of the standard metrics given in Dolan and More [4] that can be used to generate performance profiles, a widely used tool for benchmarking and comparing optimization software

$t_{p,s} \equiv$ Performance metric for problem p by solver $s \in S$

$$r_{p,s} \equiv \frac{t_{p,s}}{min\{t_{p,s':s' \in S}\}}$$

$$\rho_s(\tau) = \frac{1}{n_p} size\{p \in P : log_2(r_{p,s}) \leq \tau\}$$

The performance profile for total computational time is given in Figure 8. The results for $\tau = 0$ indicate that the proposed algorithm has the most wins with a probability of 53.8% that is the fastest solver on given problems. For $\tau \neq 0$,GloMIQO and BARON are always beaten by the proposed algorithm.

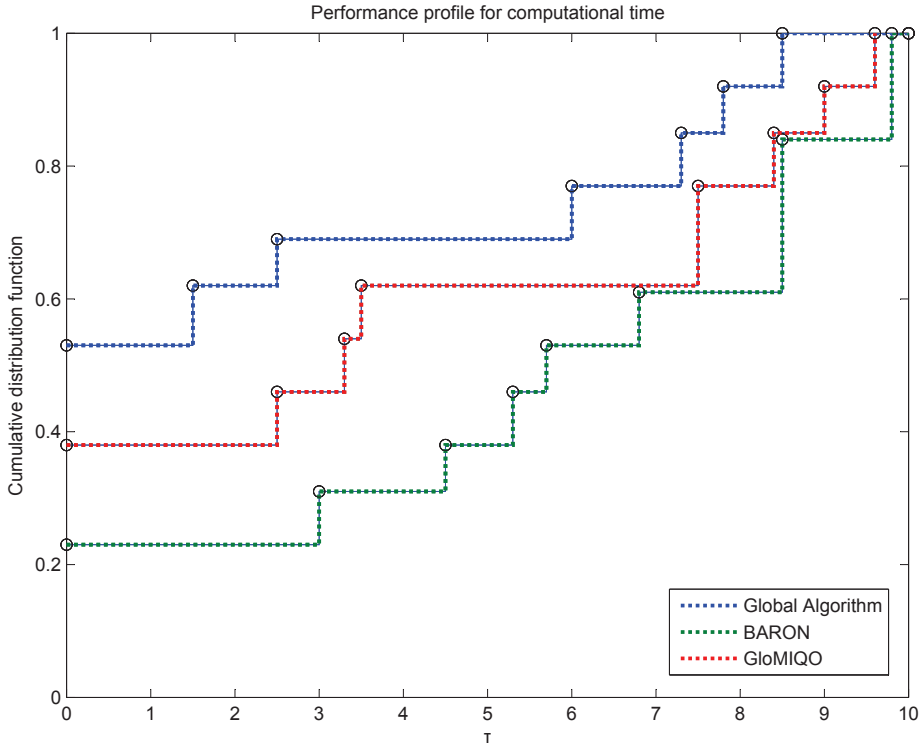

Figure 8: Performance profile for computational time

The success of the proposed algorithm lies in a comprehensive analysis of the region of the search space and its capacity to focus the search on the regions with the partial solution. One of the good merits of the hybrid algorithm is that each candidate solution guaranteed to be feasible by using the DFM method, while in existing algorithms the procedure to generate feasible solution under complex process constraints is very time costive. The deterministic finite automata (DFA) can easily represent this kind of structure. Furthermore, the complex process constraints can be very difficult to express with mixed integer programming.

6. Conclusion

This paper has proposed global optimization algorithm for the refinery scheduling problem. The real advantage of algorithm is based on the DFA model to ensure the satisfaction of individual constraints and termination in the optimization process, which is translated into an ability to reach considerably lower optimality gaps and a clearly better computational performance. When compared to commercial solvers, the results on a set of example problems from the literature have shown that the algorithm exhibited a considerably better computational performance.

Appendix A

In this section, the MINLP model[12] of refinery scheduling problem is discussed .This problem has been widely studied from the optimization viewpoint since the work of Lee [8].

Objective Function
The objective is to maximize the gross margins of the distilled crude blends. Let $G_c$be the individual gross margin of crude c.

$$MaxG = \sum_{i\in T}\sum_{r\in R_D}\sum_{v\in I_r}\sum_{c\in C} G_c \cdot V_{ivc} \tag{A1}$$

General Constraints

$$\sum_{v\in W} Z_{iv} = 1 \tag{A2}$$

$$S_{iv} + D_{iv} \le H \cdot Z_{iv} \tag{A3}$$

$$V_{iv}^t \le \overline{V_v^t} \cdot Z_{iv} \tag{A4}$$

$$V_{iv}^t \ge \underline{V_v^t} \cdot Z_{iv} \tag{A5}$$

$$\sum_{c\in C} V_{ivc} = V_{iv}^t \tag{A6}$$

$$L_{ir}^t = L_{0r}^t + \sum_{j\in T, j<i}\sum_{v\in I_r} V_{iv}^t - \sum_{j\in T, j<i}\sum_{v\in O_r} V_{iv}^t \tag{A7}$$

$$L_{irc} = L_{0rc} + \sum_{j\in T, j<i}\sum_{v\in I_r} V_{ivc} - \sum_{j\in T, j<i}\sum_{v\in O_r} V_{ivc} \tag{A8}$$

$$\underline{N_D} \le \sum_{i\in T}\sum_{v\in W_D} \le \overline{N_D} \tag{A9}$$

$$S_{iv} + D_{iv} \le S_{jv} + H \cdot (1 - Z_{jv}) \tag{A10}$$

$$\sum_{i\in T}\sum_{v\in I_r} D_{iv} = H \tag{A11}$$

$$S_{iv} \ge S_r \cdot Z_{iv} \tag{A12}$$

$$\underline{FR_v} \cdot D_{iv} \le V_{iv}^t \le \overline{FR_v} \cdot D_{iv} \tag{A13}$$

$$\underline{y_{vk}} \cdot V_{iv}^t \le \sum_{c\in C} y_{ck} V_{ivc} \le \overline{y_{vk}} \cdot V_{iv}^t \tag{A14}$$

$$0 \le L_{irc} \le \overline{L_r^t} \tag{A15}$$

$$\underline{L_r^t} \le L_{0r}^t + \sum_{i\in T}\sum_{v\in I_r} V_{iv}^t - \sum_{i\in T}\sum_{v\in O_r} V_{iv}^t \le \overline{L_r^t} \tag{A16}$$

$$\underline{D_r} \le \sum_{i\in T}\sum_{v\in O_r} V_{iv}^t \le \overline{D_r} \tag{A17}$$

$$\underline{L_r^t} \le L_{ir}^t \le \overline{L_r^t} \tag{A18}$$

$$L_{irc} = f_c \cdot L_{ir} \tag{A19}$$

$$V_{ivc} = f_c \cdot V_{iv} \tag{A20}$$

Table 1: Vessel Arrival Data for Examples 1-13

	Arrival time			**composition**			**Amount of crude**		
	Vessel 1	Vessel 2	Vessel 3	Vessel 1	Vessel 2	Vessel 3	Vessel 1	Vessel 2	Vessel 3
E1	0	3	6	100%A	100%B	100%C	1000	1000	1000
E2	0	4	7	100%A	100%B	100%C	300	300	300
E3	0	4	8	100%A	100%B	100%C	500	500	500
E4	0	3	7	100%A	100%B	100%C	400	400	400
E5	0	4	7	100%A	100%B	100%C	300	300	300
E6	0	4	8	100%A	100%B	100%C	600	600	600
E7	0	3	6	100%A	100%B	100%C	450	450	450
E8	0	5	10	100%A	100%B	100%C	600	600	600
E9	0	5	9	100%A	100%B	100%C	500	500	500
E10	0	4	10	100%A	100%B	100%C	600	600	600
E11	0	5	9	100%A	100%B	100%C	550	550	550
E12	0	6	10	100%A	100%B	100%C	650	650	650
E13	0	6	11	100%A	100%B	100%C	700	700	700

Table 2: Scheduling horizon,flowrate limit,and number of distillations for Examples 1-13

Example	**Scheduling horizon**	**Unloading flowrate**	**Transfer flowrate**	**Distillation flowrate**	**Number of distillations**
E1	10	[0,500]	[0,500]	[50,500]	5
E2	11	[0,500]	[0,500]	[50,500]	5
E3	12	[0,500]	[0,500]	[50,500]	5
E4	11	[0,500]	[0,500]	[50,500]	5
E5	11	[0,500]	[0,500]	[50,500]	5
E6	13	[0,500]	[0,500]	[20,500]	6
E7	12	[0,500]	[0,500]	[50,500]	5
E8	15	[0,500]	[0,500]	[20,500]	7
E9	13	[0,500]	[0,500]	[50,500]	5
E10	15	[0,500]	[0,500]	[20,500]	7
E11	14	[0,500]	[0,500]	[50,500]	5
E12	17	[0,500]	[0,500]	[10,500]	8
E13	19	[0,500]	[0,500]	[10,500]	9

Table 3: Storage tank capacity , and initial inventories for example 1-13

	Capacity			**Initial composition**			**Initial amount of crude**		
	Storage Tank 1	Storage Tank 2	Storage Tank 3	Storage Tank 1	Storage Tank 2	Storage Tank 3	Storage Tank 1	Storage Tank 2	Storage Tank 3
E1	[0,1000]	[0,1000]	[0,1000]	100%A	100%B	100%C	200	500	700
E2	[0,1000]	[0,1000]	[0,1000]	100%A	100%C	100%F	300	500	400
E3	[0,1000]	[0,1000]	[0,1000]	100%D	100%E	100%F	200	200	200
E4	[0,1000]	[0,1000]	[0,1000]	100%D	100%A	100%C	300	500	300
E5	[0,1000]	[0,1000]	[0,1000]	100%A	100%E	100%A	500	500	500
E6	[0,1000]	[0,1000]	[0,1000]	100%A	100%B	100%F	500	500	500
E7	[0,1000]	[0,1000]	[0,1000]	100%A	100%E	100%F	500	500	500
E8	[0,900]	[0,1100]	[0,1100]	100%D	100%A	100%B	600	100	500
E9	[0,1000]	[0,1000]	[0,1000]	100%C	100%A	100%E	400	500	600
E10	[0,1000]	[0,1000]	[0,1000]	100%A	100%B	100%C	500	300	500
E11	[0,900]	[0,1000]	[0,1000]	100%A	100%E	100%F	400	500	200
E12	[0,1000]	[0,1000]	[0,1000]	100%D	100%A	100%B	300	300	500
E13	[0,1000]	[0,1100]	[0,1100]	100%A	100%B	100%C	200	600	500

Table 4: Storage tank capacity , and initial inventories for example 8-13

	Capacity			**Initial composition**			**Initial amount of crude**		
	Storage Tank 4	Storage Tank 5	Storage Tank 6	Storage Tank 4	Storage Tank 5	Storage Tank 6	Storage Tank 4	Storage Tank 5	Storage Tank 6
E8	[0,1100]	[0,900]	[0,900]	100%C	100%E	100%E	400	300	600
E9	[0,1000]	[0,1000]	[0,900]	100%E	100%F	100%C	500	400	400
E10	[0,1000]	[0,900]	[0,1000]	100%B	100%A	100%C	300	600	500
E11	[0,1000]	[0,1000]	[0,1000]	100%A	100%B	100%E	400	500	400
E12	[0,1000]	[0,1100]	[0,1000]	100%B	100%A	100%B	200	300	500
E13	[0,900]	[0,1000]	[0,1100]	100%A	100%E	100%F	500	500	400

Table 5: Charging tank capacity, and initial inventories for example 1-13

	Capacity				**Initial composition**			
	Charging Tank 1	Charging Tank 2	Charging Tank 3	Charging Tank 4	Charging Tank 1	Charging Tank 2	Charging Tank 3	Charging Tank 4
E1	[0,1000]	[0,1000]	[0,1000]	–	100%D	100%E	100%F	–
E2	[0,1000]	[0,1000]	[0,1000]	–	100%D	100%E	100%G	–
E3	[0,1000]	[0,1000]	[0,1000]	–	100%G	100%E	100%F	–
E4	[0,1000]	[0,1000]	[0,1000]	–	100%E	100%D	100%F	–
E5	[0,1000]	[0,1000]	[0,1000]	–	100%F	100%G	100%H	–
E6	[0,1000]	[0,1000]	[0,1000]	–	100%F	100%D	100%E	–
E7	[0,1000]	[0,1000]	[0,1000]	–	100%F	100%E	100%G	–
E8	[0,800]	[0,800]	[0,800]	[0,800]	100%F	100%G	100%H	100%E
E9	[0,1000]	[0,1000]	[0,1000]	[0,1000]	100%G	100%E	100%H	100%F
E10	[0,1000]	[0,1000]	[0,1000]	[0,1000]	100%E	100%G	100%D	100%H
E11	[0,900]	[0,900]	[0,900]	[0,900]	100%D	100%F	100%G	100%E
E12	[0,1000]	[0,1000]	[0,1000]	[0,1000]	100%E	100%D	100%G	100%F
E13	[0,800]	[0,800]	[0,800]	[0,800]	100%F	100%G	100%H	100%E

Table 6: Initial amount of crude for example 1-13

	Initial amount of crude			
E1	Charging Tank 1	Charging Tank 2	Charging Tank 3	Charging Tank 4
E2	300	500	300	–
E3	500	300	500	–
E4	300	500	300	–
E5	200	500	500	–
E6	500	300	200	–
E7	300	200	200	–
E8	300	500	100	–
E9	50	300	300	500
E10	200	400	500	300
E11	200	300	500	300
E12	500	300	500	300
E13	400	300	200	500
E1	200	400	500	400

Table 7: Propertys and gross margin for crudes for Examples 1-13

	Property 1				**Property 2**				**Gross margin**			
	Crude E	Crude F	Crude G	Crude H	Crude E	Crude F	Crude G	Crude H	Crude E	Crude F	Crude G	Crude H
E1	0.03	0.0433	–	–	0.23	0.133	–	–	3	4.33	–	–
E2	0.05	0.08	0.03	–	0.3	0.23	0.186	–	3	3.8	2.9	–
E3	0.05	0.08	0.03	–	–	–	–	–	5	8	3	–
E4	0.03	0.0433	–	–	0.25	0.155	–	–	4	6	–	–
E5	0.075	0.0317	0.0483	0.0633	0.31	0.1	0.15	0.133	6	5.8	4	5
E6	0.03	0.0433	–	–	–	–	–	–	5.5	3.92	–	–
E7	0.05	0.08	0.03	–					7	6	4.2	–
E8	0.075	0.0317	0.0483	0.0633	–	–	–	–	7.5	3.17	4.83	6.33
E9	0.075	0.0317	0.0483	0.0633	0.19	0.098	0.21	0.12	6	5.14	2.5	4.9
E10	0.075	0.0317	0.0483	0.0633	–	–	–	–	4.1	4.8	3	4
E11	0.05	0.08	0.03	–	–	–	–	–	8	5	3.6	–
E12	0.05	0.08	0.03	–	0.22	0.15	–	–	5.5	5.5	4	–
E13	0.075	0.0317	0.0483	0.0633	–	–	–	–	7	3	5	3

References

[1] F. A. Al-Khayyal, J. E. Falk, *Jointly Constrained Biconvex Programming*, Math. Oper. Res., **8** (1983), 273–286.1

[2] N. Adhya, M. Tawarmalani, N. V. Sahinidis, *A Lagrangian Approach to the Pooling Problem*, Indu. Eng. Chem. Res., **38** (1999), 1956–1972.1

[3] M. L. Bergamini, P. Aguirre, I. Grossmann, *Logic-based outer approximation for globally optimal synthesis of process networks*, Comput. Chem. Eng., **29** (2005), 1914–1933.1

[4] E. D. Dolan, J. J. Mor, *Benchmarking optimization software with performance profiles*, Math. Program., **91** (2002), 201–213.5

[5] I. E. Grossmann, *Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques*, Optim. Eng., **3** (2002), 227–252.1

[6] R. Horst, H. Tuy, *Global optimization: Deterministic approaches*, Springer, Berlin, (1996).1

[7] S. Kolodziej, P. M. Castro, I. E. Grossmann, *Global optimization of bilinear programs with a multiparametric disaggregation technique*, J. Global Optim., **57** (2013), 1039–1063.4.1

[8] H. Lee, J. M. Pinto, I. E. Grossmann, S. Park, *Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management*. Ind. Eng. Chem. Res., **35** (1996), 1630–1641.6

[9] L. Liberti, S. Cafieri, F. Tarissan, *Reformulations in Mathematical Programming: A Computational Approach*, Found. Comput. Intell., **203** (2009), 153–234.1

[10] L. Liberti, C. C. Pantelides, *An Exact Reformulation Algorithm for Large Nonconvex NLPs Involving Bilinear Terms*, J. Global Optim., **36** (2006), 161–189.1

[11] R. Misener, C. Floudas, *GloMIQO: Global mixed-integer quadratic optimizer*, J. Global Optim., **57** (2013), 3–50.5

[12] S. Mouret, I. E. Grossmann, P. Pestiaux, *A Novel Priority-Slot Based Continuous-Time Formulation for Crude-Oil Scheduling Problems*, Ind. Eng. Chem. Res., **48** (2009), 8515–8528. 3, 3.1, 3.2, 3.2.1, 3.2.2, 4, 5, 6
[13] Y. Nesterov, *Semidefinite relaxation and nonconvex quadratic optimization*, Optim. Meth. Softw., **9** (1998), 141–160. 1
[14] M. Oral, O. Kettani, *A Linearization Procedure for Quadratic and Cubic Mixed-Integer Problems*, Oper. Res., **40** (1992), 109–116. 2.1
[15] M. Pan, Y. Qian, X. Li, *Flexible scheduling model of crude oil operations under crude supply disturbance*, Science in China, **52** (2009), 387–400. 1
[16] H. S. Ryoo, N. V. Sahinidis, *Global optimization of nonconvex NLPs and MINLPs with applications in process design*, Comput. Chem. Eng,. **19** (1995), 551–566. 1
[17] H. Sherali, A. Alameddine, *A new reformulation-linearization technique for bilinear programming problems*, J. Global Optim., **2** (1992), 379–410. 1
[18] N. Z. Shor, *Dual quadratic estimates in polynomial and Boolean programming*, Ann. Oper. Res., **25** (1990), 163–168. 1
[19] M. Tawarmalani, N. V. Sahinidis, *A polyhedral branch-and-cut approach to global optimization*, Math. Program., **103** (2005), 225–249. 5
[20] J. P. Teles, P. M. Castro, H. A. Matos, *Multi-parametric disaggregation technique for global optimization of polynomial programming problems*, J. Global Optim., **55** (2013), 227–251. 2, 2.2, 2.3
[21] G. Van Noord, *FSA Utilities: A toolbox to manipulate finite-state automata*, Automata Implementation, **1260** (1997), 87–108. 3.3, 4.3
[22] D. S. Wicaksono, I. A. Karimi, *Piecewise MILP under- and overestimators for global optimization of bilinear programs*, AIChE J., **54** (2008), 991–1008. 1