A multi-commodity flow model for managing selective catalytic reduction for coal-fired power plants.

Abstract

Selective Catalytic Reduction (SCR) reduces emissions of oxides of nitrogen (NO_x) in coal-fired power plants. However, to maintain SCR performance, layers of catalyst within an SCR reactor need to be added or changed periodically, which is expensive. In order to optimize an SCR management plan given a calendar of outages, we develop a binary multi-commodity network flow problem (MCFP) with two sets of side constraints. The first set of side constraints controls peak NO_x emissions, while the second set limits average daily NO_x emissions. The first constraint set can be linearized using reactor potential. However, average daily NO_x emissions can only be evaluated with a given schedule. Consequently, we develop a cutting plane method to eliminate infeasible schedules dynamically, referred to as MCFPwSEC. We then reduce the computational time further with the introduction of a multi-cut MCFPwSEC method, which eliminates infeasible solutions based on a heuristic algorithm. We provide computational results, conclusions, and future research topics.

Keywords: Integer programming, Selective catalytic reduction, Multi-commodity network flow, Energy management, Coal-fired power plants.

1. Introduction

This paper aims to optimize a management plan of Selective Catalytic Reduction (SCR) in a coal-fired power plant given a calendar of outages. This problem is addressed by Phananiramai *et al.* (2011), but now we propose a multi-commodity flow model that obtains an optimal management plan for one plant.

Coal-fired power plants generate nearly half of the electricity in the United States (Rubin *et al.*, 1997). They pulverize the coal and transport it to a boiler where the coal is burned and water steam is created. Later water steam exerts a force to rotate a turbine shaft, and the turbine generates electricity. Burning coal creates oxides of nitrogen (NO_x) in two ways: the natural reaction of molecules of nitrogen and oxygen of the air at high temperatures (thermal NO_x) and the oxidation of nitrogen that are in the coal (fuel NO_x) (The U.S. Department of Energy and Southern Company Services, Inc., 1997).

The emissions of NO_x are regulated by the Environmental Protection Agency (EPA). In order to comply with these regulations, some power plants use Selective Catalytic Reduction (Tennessee Valley Authority, n.d.). SCR uses layers of catalyst bed where added ammonia (NH_3) and flue gases with NO_x react to produce water and nitrogen. The NO_x that enters to the catalyst is named *inlet* NO_x, and the NH_3 that enters to the catalyst is named NH_3 *injection*. On the other hand, the NO_x and the NH_3 that do not react and exit of the catalyst are named *outlet* NO_x and NH_3 *slip*, respectively. Because the *reactor potential* (RP) of the catalyst decreases over time, NH_3 injection increases to limit the outlet NO_x, but this action also increases the NH_3 slip. NH_3 slip is undesirable because NH_3 is expensive, hazardous, and damages the catalyst and other equipment (Cichanowicz *et al*., 2006). For this reason, catalyst layers are added or replaced during scheduled maintenance outages for the plant to maintain good SCR performance.

In order to optimize catalyst usage, it is necessary to minimize the catalyst cost and also to minimize the operation cost of the facility (Staudt and Engelmeyer, 2003). Three major factors affect catalyst performance: sintering of the catalyst due to high temperature; catalyst plugging; and alkaline metals, earth metal masking, and/or arsenic oxide (Pritchard *et al*., 1995 and Muzio *et al*., 2002).

SCR reactors have four slots for layers of catalyst. For a new SCR reactor, three of these slots are usually filled with catalyst layers while one slot layer will be start empty. This empty slot is filled when SCR performance becomes unacceptable (Staudt and Engelmeyer, 2003). The catalyst can be added (when a slot layer is empty) or replaced with a new catalyst layer, a regenerated layer, or a cleaned layer. Regenerated catalyst is less expensive but also has less reactor potential than a new catalyst layer, and cleaned catalyst is the least expensive but also has the least reactor potential (Cichanowicz and Muzio, 2003).

The remainder of this paper is organized as follows. Section 2 reviews related literature. Section 3 shows the problem formulation. Section 4 presents computational experiments, and Section 5 presents conclusions and future research.

2. Literature Review

SCR cost and NO_x reduction can be optimized using stochastic programming and stochastic optimization, but Chen and Frey (2004) showed that these optimization techniques are most effective if they consider the difference between variability and uncertainty. The company Fossil Energy Research Corporation (FERCo) offers the software CatalysTraK®, which helps companies optimize the management of SCR (FERCo, 2013).

The proposed model in Phananiramai *et al.* (2011) first generates all the schedules that comply with NO_x reduction and later optimizes in order to select a schedule with a minimum cost. We propose a multi-commodity flow model (MCFP) that first selects the edges with minimum cost that comply with the minimum Reactor Potential and later checks if this selection complies with NO_x reduction.

Network flow problems and multi-commodity flow problems have been studied for many decades. Example research on network flow problems include Hitchcock (1941), Koopmans (1947), Dantzig (1951), Ford and Fulkerson (1956), Gomory and Hu (1961), Kennington (1978), Bixby and Cunningham (1980) Ahuja *et al.* (1993), and Fontes *et al.* (2006). McBride (1998) discussed advances over the years in solving the multi-commodity flow problem, and Goffin *et al.* (1996) demonstrate that analytic center cutting planes methods can solve large nonlinear multi-commodity flow problems. There are numerous real-world applications such as transportation problems (Hu, 1963; Farvolden *et al.*, 1993; Milano and Hentenryck, 2010), communication networks (Hu, 1963; Cheng *et al.*, 2006), military logistics (Bellmore and Ratliff, 1971), airline fleet assignment (Hane *et al.*, 1995; Barnhart *et al.*, 1998; Rosenberger *et al.*, 2004; Pilla *et al.*, 2008), unit commitment (Kjeldsen and Chiarandini, 2012), and fleet routing and flight scheduling (Yang and Tseng, 2002).

2.1 Contribution

While the multi-commodity network flow model has been studied extensively in the current literature and has been applied to numerous real world applications, work on applying the multi-commodity network flow model as well as mathematical optimization techniques in general for SCR management has not yet been studied in any literature.

The contribution of this research is the development of a new model that optimizes SCR management for a single plant. Phananiramai *et al.* (2011) use set partitioning to optimize SCR for multiple plants. Our model uses a binary multi-commodity network flow problem (MCFP) with two sets of side constraints. The first set of side constraints controls peak NO_x emissions, while the second set limits average daily NO_x emissions. The first constraint set can be linearized using reactor potential. However, average daily NO_x emissions can only be evaluated with a given schedule. Consequently, we develop a cutting plane method to eliminate infeasible schedules dynamically, referred to as MCFPwSEC. We then reduce the computational time further with the introduction of a multi-cut MCFPwSEC method. Multi-cut MCFPwSEC eliminates infeasible solutions on each iteration based on a heuristic algorithm. We provide computational results that are faster than the model used by Phananiramai *et al.* (2011).

3. Model Formulation

3.1 Binary MCFP model

3.1.1 Model Assumptions

The assumptions for the model are as follows:

- As stated previously, the NO_x reduction involves catalyst causing the reaction of NH_3 and NO_x to reduce outlet NO_x as well as ammonia slip.
- As reactor potential decreases NO_x reduction decreases and ammonia slip increases. In general, NO_x reduction is a function of allowance from ammonia slip and reactor potential from the catalyst. In this research, we assume that ammonia slip is kept constant and therefore NO_x reduction is strictly a function of reactor potential. Although the details of this function are proprietary to a corporation that provided us the exact equation, it is a function in which NO_x

reduction increases with reactor potential. This function is the same as in Phananiramai *et al.* (2011).

- As mentioned previously, regenerated catalyst is assumed to be less expensive but also has less reactor potential than a new catalyst, and cleaned catalyst is assumed to be the least expensive but also has the least reactor potential.
- As stated previously, we assume we have a model that can correctly predict the values of outlet NO_x, inlet NO_x, NH_3 injection, and NH_3 slip.
- We assume we have all the data needed for the formulas of the model and we always have layer assets available.
- We assume that the only way to obtain the average daily NO_x reduction of a schedule involves integrating NO_x reduction over the time horizon when the schedule is given in its entirety.
- Due to time constraints on outages, we assume that only one layer may be changed or added in each outage.
- We assume that the cost difference between adding a catalyst layer to an empty SCR slot and changing a catalyst layer is *disposal cost* of the existing layer, which is the same for all layers.

All of the aforementioned assumptions are consistent with conversations that the authors have had with domain experts and found in commercial SCR management software, as well as discussions of SCR within the literature.

3.1.2 Notation and Model

Now, we describe the formulation of the binary MCFP optimization model with side constraints to solve the SCR management problem. We formulate edges to represent the flow of SCR catalyst layers that can be up to four layers per plant. Essentially, we would like to find a path from the start of time horizon (source) to the end of time horizon (sink) for each layer of the plant throughout the time horizon. Like time, the flow of the edges can only be forward. A path represents a sequence of outages used for that particular layer. The edges in the path also determine the actions used in the outages. Layer actions consist of adding a new layer of catalyst (AddNew), a regenerated layer (AddRegenerated), or a cleaned layer (AddCleaned) and replacing a layer with a new layer (ChangeNew), a regenerated layer (ChangeRegenerated), or a cleaned layer (ChangeCleaned). As mentioned in Phananiramai *et al.* (2011), only one action can be taken during each outage due to time and cost restrictions. First, we define the variable vector *x* that represents edges that flow from one outage to the next as well as from the start (source node) of the time horizon and to the end (sink node) of the time horizon. The edge information also includes which layer it represents and the layer action that was taken at the previous outage. After each edge is generated, we calculate the corresponding RP and cost associated with that particular edge. Consequently, for each edge, a 0-1 decision variable determines whether the edge is used in the solution plan. The following formulation describes the MCFP variables.

Let x_{ija}^{l} be 1 if two consecutive outages *i* and *j* are used and action *a* is taken on layer *l* in outage *i*, and 0 otherwise. Given a set of scheduled outages, consider a set of SCR catalyst layers, where up to four layers can be filled at the start of the time horizon. Edges are generated based upon any pair of consecutive outages *i* and *j* and include the start of the time horizon (source node) and the end of time horizon (sink node). Edges are also generated for the slots that are empty at the start of the time horizon for potential additions. If a layer at a given slot is already filled prior to the start of the time horizon, all subsequent outages for that particular slot can only consist of changes that can be either a new layer, a regenerated layer, or a cleaned layer. Conversely, if a given slot is unfilled before the start of time horizon, subsequent layer actions can only be additions that can also be either a new layer, a regenerated layer, or a cleaned layer. Furthermore, after an addition of a particular layer has been made, the following actions can only consist of changes. Similarly, at each outage (node), we determine whether that particular slot has been filled or not, which determines what set of actions that can be applied at the particular outage. Consequently, after each edge is generated, we calculate its corresponding RP and cost for that particular edge, where RP_{ija}^{l} is the reactor potential between two consecutive outages *i* and *j* where action a is taken on layer l in outage i, and C_{ija}^{l} is similarly the cost incurred between two consecutive outages i and j where action a is taken on layer l in outage i.

The binary multi-commodity network flow model can be constructed as follows:

Nodes:

1) Create a sink node for each slot of the plant at the end of time horizon.

2) Create a source node for each slot of the plant at the start of the time horizon.

3) Create an intermediate node for each slot of the plant at all the possible outages in the time horizon in chronological order.

Arcs:

1) Create an arc from the start node of a slot to the sink node of the same slot. After the arc is generated, we calculate the corresponding reactor potential RP_{ija}^{l} and cost C_{ija}^{l} for the arc.

2) Create an arc from the start node of a slot to each intermediate node of the same slot. After the arc is generated, we calculate the corresponding reactor potential RP_{ija}^{l} and cost C_{ija}^{l} for the arc.

3) Create an arc for each of three actions from each intermediate node of a layer to each intermediate node of the same layer with reactor potential RP_{ija}^{l} and cost C_{ija}^{l} if and only if the tail of the arcs start in a node with a date prior to the date of the node where the head of the arcs arrives.

4) Create an arc for each of three actions from each intermediate node of a layer to the sink node of the same layer with reactor potential RP_{ija}^{l} and cost C_{ija}^{l}.

The three possible actions if the slot has a layer are ChangeNew, ChangeRegenerated, ChangeCleaned. The three possible actions if the slot does not have a layer are AddNew, AddRegenerated, AddCleaned.

List of parameters used:

- d = minimum value of average daily NO_x reduction to obtain.

- O = set of all outages, indexed by o.
- A = set of all actions, indexed by a.
- L = set of all layers, indexed by l.
- $E_l(i)$ = set of edges from the node outage i in the sub-network layer l.
- S = source node.
- T = sink node.
- RP_{ija}^{l} = reactor potential between two consecutive outages i and j and action a is taken on layer l in outage i.
- C_{ija}^{l}= cost incurred between two consecutive outages i and j and action a is taken on layer l in outage i.
- $f(x)$ = average daily NO_x reduction for a decision variable x.

List of variables used:

- x_{ija}^{l} = 1 if two consecutive outages i and j are used and action a is taken on layer l in outage i, and 0 otherwise.

To illustrate the model, consider a simple example of a layer from a single plant where there are two outages, o_1 and o_2. Figure 1 demonstrates the flow of a catalyst layer from the start of the time horizon (S) to the end of time horizon (T).

Figure 1 about here

From Figure 1, we observe that there were 12 edges created that are denoted by their indices with their corresponding x_{ija}^{l}, RP_{ija}^{l}, and C_{ija}^{l}. Note that edges must originate from the source to a future outage. Therefore, edge 1 goes from the source node to o_1, edge 2 goes from the source node to o_2, and edge 3 goes from the source node to the sink node, which means that no action was taken for this particular layer. We refer to these types of edges as *from source* and *source to sink* edges. Next, edges must end in the sink node (T), so there are edges 3, 7, 8, 9, 10, 11, 12 that go to the sink, and we refer to these types of edges as *source to sink* and *to sink* edges. Observe that from o_1 to T there are three edges to sink. These three edges represent changing either a new layer, a regenerated layer, or a cleaned layer in outage o_1. Whether it was an addition or a change depends upon whether that particular layer slot was filled or not at the start of o_1. Figure 2 summarizes an example of having a filled layer at the source node S. Edges 4, 5, and 6 are referred to as *intermediate* edges. These are edges that flow between two consecutive outages that do not include the source node and/or the sink node as an endpoint. Similarly, there are three possible actions that can be done during the previous outage.

In Figure 2, we assumed that the layer was originally filled at the start of the time horizon. Observe that not all edges are drawn in Figure 2. For a complete list of edges, refer to Figure 1. Since the layer is already filled in the particular slot, it cannot be added, so the only option is to change layers. Consequently, decisions are either to do a change in o_1, do a change in o_2, or do nothing. Changes in o_1 and o_2 can be either with a new layer, a regenerated layer, or a cleaned layer. Suppose the edge that flows from S to o_2 was chosen. There are three edges from o_2 to T representing the three possible change actions in outage o_2. Therefore in this example, the layer was filled at the start of time horizon, then, a change was made at o_2.

In Figure 3, we assume that the layer slot was not filled at the start of time horizon. Since the layer slot is empty, the options are to add a layer in o_1, add in o_2, or do nothing. Suppose that the edge that flows from S to o_1 is chosen and the action in o_1 is to add a cleaned layer. Then, since the layer has been added already, meaning the slot is no longer empty, the remaining options are to either do a change in o_2 or go to sink (T), and in this case we choose the edge that goes from o_1 to T. In summary, this particular layer slot started out as empty, and then we added a cleaned layer to that slot in o_1.

Figure 2 about here

Figure 3 about here

The cost difference between adding and changing the layer is the cost of disposing the used layer as showed in Figure 3. When we add the layer the cost is zero because the layer slot is empty and does not have a used layer to dispose. However, when we change a layer, the disposal cost is positive because we need to pay to dispose the used layer. For this reason, it is less expensive to add a layer than change it. In the case of Figure 3, the cost of disposal is subtracted from the cost of the from source arcs in slots that are empty at the beginning of the time horizon.

In the formulation, recall that there is a corresponding reactor potential RP_{ija}^{l} and a cost C_{ija}^{l} for each edge. Since RP is directly proportional to NO_x reduction, we would like to ensure that before each outage o_i across all layers for the plant, a certain RP value is met to limit peak NO_x emissions. Consequently, at the start of each outage, we add a constraint where the instantaneous RP value meets at least a pre-specified minimum value. Furthermore, at the end of the time horizon, power plants are not shut down. They still must run for a certain period until the next outage beyond the time horizon. Therefore, we would like to impose a constraint that specifies how many months we would like the plant to run without an outage before the next outage beyond the time horizon. The number of months is then converted to a minimum RP value needed. Figure 4 illustrates an example.

Figure 4 about here

From Figure 4, we expand the example to include an additional layer slot to illustrate the RP constraints. Notice that Figure 4 still represents a single plant with two outages, but we apply minimum RP constraints at the start of each outage, o_1 and o_2. These minimum RP constraints are derived from a pre-specified minimum NO_x reduction requirement for the plant. This minimum RP value implies that at constraint is added to ensure that at the end of time horizon, the NO_x reduction for the plant can still be maintained until the next outage after the time horizon.

The 0-1 integer program to solve the SCR management problem is given by equations (1) to (8).

$$min \sum_{l \epsilon L} \sum_{a \epsilon A} \sum_{(i,j) \epsilon E} C_{ija}^{l} x_{ija}^{l} \tag{1}$$

s.t.

$$\sum_{l \epsilon L} \sum_{a \epsilon A} \sum_{(j,k) \epsilon E_l(i)} RP_{jka}^{l} x_{jka}^{l} \geq min\, RP \qquad \forall i \in O \tag{2}$$

$$\sum_{l \epsilon L} \sum_{a \epsilon A} \sum_{j | \exists (i,j) \epsilon E} x_{ija}^{l} \leq 1 \qquad \forall i \in O \tag{3}$$

$$\sum_{a \epsilon A} \sum_{j | \exists (i,j) \epsilon E} x_{ija}^{l} = \sum_{a \epsilon A} \sum_{j | \exists (j,i) \epsilon E} x_{jia}^{l} \qquad \forall i \in O, l \in L \tag{4}$$

$$\sum_{a \epsilon A} \sum_{j | \exists (s,j) \epsilon E} x_{sja}^{l} = 1 \qquad \forall l \in L \tag{5}$$

$$\sum_{a \epsilon A} \sum_{j | \exists (j,t) \epsilon E} x_{jta}^{l} = 1 \qquad \forall l \in L \tag{6}$$

$$x_{ija}^{l} \in \{0,1\} \qquad \forall (i,j) \in E,\ l \in L,\ a \in A \tag{7}$$

$$f(x) \geq d \tag{8}$$

The problem is to minimize the total costs across all edges in the plant subject to flow constraints. Constraints (1) and (3) to (7) are traditional binary MCFP constraints, where edges flow from sources to sinks in the layer sub-networks. Constraint set (2) states that RP is over certain RP value, limits peak NO_x emissions. Constraint (8) control average daily NO_x reduction, where $f(x)$ can only be obtained once we have the schedule. In section 3.2 we discuss schedule elimination constraints.

3.2 Schedule Elimination Constraints

We can construct a relaxed binary MCFP (1)-(7), because they limit the instantaneous RP value at a certain point in time and are based upon the minimum NO_x reduction (upper limit on peak NO_x

emissions). Even though we can obtain the RP value of each edge and then sum them up to get the total RP for the schedule, all of the information here cannot directly derive the average daily NO_x reduction.

Because the relaxed binary MCFP can only limit the peak NO_x emissions but not the average daily NO_x emissions, we introduce MCFP with schedule elimination constraints (MCFPwSEC). The only way to obtain the average daily NO_x reduction of a schedule involves integrating NO_x reduction over the time horizon when the schedule is given in its entirety. Therefore, after a schedule is found, we determine whether or not it violates the average daily minimum NO_x constraint (8). If it does not, then it is an optimal solution. If it does, then we generate a constraint to make that solution infeasible and then re-optimize the problem.

Consider the following set elimination constraints. For a vector x, let schedule $s(x)$ be the set of edges in which x_e=1. Let F be the set of all feasible flows in MCFP that relax the average daily NO_x constraint (8); that is, $F = \{s(x) \mid x$ satisfies equations (1) to (7)$\}$. Let $G = \{s(x) \in F \mid f(x) \geq d\}$ be the set of all feasible schedules. Let $G^c = F \backslash G$ be the set of flows in MCFP that are infeasible schedules; that is, schedule $s \in G^c$ violates the average daily NO_x constraint (8). Then, the set of schedule elimination constraints is given by (8').

$$\sum_{e \in s} x_e \leq |s| - 1 \qquad \forall s \in G^c \qquad (8')$$

Figure 5 about here

Figure 5 and Algorithm 1 summarize the MCFPwSEC algorithm. From Algorithm 1, G^c can be a potentially huge set. Therefore, we can generate G^c dynamically through MCFPwSEC. From Figure 5, we can observe that a cut is added after a schedule is found that violates the minimum average daily NO_x constraint. Cuts are added one by one until an optimal solution is found when the minimum average daily NO_x constraint is met.

Algorithm 1: MCFPwSEC

Initialization step: Let $\bar{G}^c \subset G^c$

Relaxed problem step: Solve a relaxed MCFP (1)-(7) with schedule elimination constraints (8') from $\bar{G}^c$ to obtain x*.

Feasibility check: If $f(x^*) \geq d$, then return schedule $s(x^*)$.

Cut generation step: Set $\bar{G}^c \leftarrow \bar{G}^c \cup \{s(x^*)\}$, and go to the Relaxed problem step.

3.3 Multi-cut MCFP with Schedule Elimination Constraints

In this section, we introduce multi-cut MCFPwSEC that generates multiple constraints per iteration based upon heuristics to improve algorithmic performance. Recall that a schedule uses actions on layers during outages, and certain actions improve reactor potential and NO_x reduction at higher costs; such alternative actions are represented by parallel arcs in the binary MCFP. Consequently even though a schedule is found to be infeasible, another schedule with different actions on the same layers during the same outages may be feasible. However, if a schedule is found to be infeasible, then any other schedule with actions that reduce reactor potential on the same layers during the same outages must be infeasible. Consider a schedule vector x, let $H(x)$ be a set of schedule vectors that use actions with lower reactor potential, which also decreases NO_x reduction.

$$H(x) = \{\tilde{x} | \, \tilde{x} \text{ satisfies } (3)-(7), \tilde{x}^l_{e\tilde{a}} = 1 \text{ if and only if } \exists\, a \in A, x^l_{ea} = 1, RP^l_{e\tilde{a}} \leq RP^l_{ea}\}$$

Because the NO_x reduction function f is increasing reactor potential, Property 1 is true.

Property 1: If $f(x) < d$, then $f(\tilde{x}) < d, \forall \tilde{x} \in H(x)$.

In multi-cut MCFPwSEC, we search for an infeasible schedule vector $\bar{x}$ and then make use of Property 1 to generate multiple cuts for each vector $H(\bar{x})$. Although the schedule vector x* from the relaxed problem step in Algorithm 1 may be infeasible, it is not beneficial to generate multiple cuts from $H(x^*)$. Comparing actions involving new layers, regenerated layers, and cleaned layers, actions with higher reactor potential have a higher cost. Specifically, new layers have the highest reactor potential, associated NO_x reduction, and cost, while cleaned layers have the least in all three measures. Hence, the following Property 2 is also true.

Property 2: $C\tilde{x} < Cx, \forall \tilde{x} \in H(x) \backslash \{x\}$.

Consider a schedule vector x^* from the relaxed problem step in Algorithm 1 that optimizes the relaxed MCFP (1)-(7), (8'). We can prove the following Proposition 1.

Proposition 1: $H(x^*)\backslash\{x^*\} \cap \{x|x \text{ satisfying } (3)-(7),(8')\} = \emptyset$.

Proof: Assume to the contrary that there exists schedule vector $\tilde{x} \in H(x^*)\backslash\{x^*\}$ that satisfies (3)-(7), (8'). By Property 2, $\tilde{x}$ has better objective value (1) than x^*, which contradicts that assumption that x^* optimizes the relaxed MCFP (1)-(7), (8'). □

Proposition 1 implies that adding schedule elimination constraints from $H(x^*)$ will not reduce the number of iterations of the algorithm. Consequently, multi-cut MCFPwSEC seeks to add constraints based upon a schedule with higher reactor potential than that of x^*. For a schedule vector x and a *targeted action* $a \in A$, let h_a: $x \rightarrow \bar{x}$ be a mapping defined by the following.

$$\bar{x}^l_{ea} = \begin{cases} 1 & \text{if } \exists\ \tilde{a} \in A \text{ such that } x^l_{e\tilde{a}} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 2 summarizes the multi-cut MCFPwSEC.

Algorithm 2 Multi-cut MCFPwSEC

Initialization step: Let $\bar{G}^c \subset G^c$, and let a be the targeted action.

Relaxed problem step: Solve a relaxed MCFP (1)-(7) with schedule elimination constraints (8') from $\bar{G}^c$ to obtain x^*.

Feasibility check: If $f(x^*) \geq d$, then return schedule $s(x^*)$.

Multi-cut feasibility check: If $h_a(x^*) \geq d$, then set $\bar{G}^c \leftarrow \bar{G}^c \cup \{s(x^*)\}$, and go to the Relaxed problem step.

Multi-cut generation step: Set $\bar{G}^c \leftarrow \bar{G}^c \cup \{s(x) |\ x \in H(h_a(x^*))\}$, and go to the Relaxed problem step.

Because regenerated or new layers both improve reactor potential over cleaned layers, we consider two variants multi-cut MCFPwSEC: one in which the targeted action is a new layer, and another with regenerated targeted actions.

3.3.1 New Layer Targeted Action

In the multi-cut feasibility check step, the targeted action can be changing or adding a new layer of catalyst. Consider a single cut example. If the sequence of actions in the schedule $s(x^*)$ were ChangeCleaned, AddCleaned, and ChangeCleaned were found to be infeasible on one iteration, then the schedule $s(x^*)$ on the next iteration will likely be the same set of outages but with actions ChangeCleaned, AddCleaned, and ChangeRegenerated. In order to find multiple cuts in each iteration, multi-cut MCFPwSEC checks the feasibility of the schedule $s(h_{new}(x^*))$ with actions ChangeNew, AddNew, and ChangeNew, which replaces the layers with new ones while keeping the outage dates the same. If this new schedule $s(h_{new}(x^*))$ does not satisfy the minimum average daily NO_x constraint (8), multi-cut MCFPwSEC eliminates 27 schedules immediately, which reduces the number of optimization iterations and potentially reduces computational time. Figure 6 describes the algorithm when the targeted action is changing or adding a new layer of catalyst, and Table 1 illustrates the 27 sequences of actions that are cut in the multi-cut generation step of the algorithm when the targeted action is changing or adding a new layer of catalyst.

Figure 6 about here

Table 1 about here

From Table 1, we observe that multiple cuts may be added per iteration instead of just a single cut as in Section 3.2.

3.3.2 Regenerated Layer Targeted Action

One possible problem with targeting changing and adding new catalyst is that it potentially creates too many new cuts, many of which are unused in the relaxed problem step. The reason for this is that, for example, the constraint ChangeNew, AddNew and ChangeNew in Table 1 is very expensive, and the relaxed problem step will never explicitly consider this schedule before MCFPwSEC finds an optimal solution. To reduce the number of new cuts created in multi-cut MCFPwSEC, now we consider targeting regenerated layers instead of new layers. For example, in Figure 6, we substitute the dashed rounded rectangle with: "Replace each Cleaned in the sequence with Regenerated layer". Similarly, in Table 1, we only add cuts for the 8 sequences that include Cleaned and Regenerated actions.

4. Experiments

In this section, we show computational experiments of single cut MCFPwSEC and multi-cut MCFPwSEC with both targeted actions. To conduct the experiments, we used the C++ programming language with CPLEX version 12.5.1 callable library (IBM, 2013) on a workstation with an Intel(r) Xeon(r) X3450 2.67GHz processor.

4.1 Example Problem Instances

We consider three examples that are from a modified version of a real-world problem instance. Example 1 includes a plant with five scheduled outages during a five-year planning horizon. From the available four catalyst layer slots, two layers were installed before the start of the time horizon while two slots were empty. Layers are indexed 1, 2, 3, and 4 where layer 4 is the one closest to the inlet. The pre-specified conditions are that NO_x reduction is at least 75% for the entire time horizon as well as the 8 subsequent months after the horizon. Because layers 1 and 2 have exactly the same parameters, we always select the layer closest to the inlet as the first option to add a layer when it is needed. A summary of the outages of example 1 is shown in Table 2. Example 2 is similar to example 1, except that the outage plan and the dates when the two filled layers were filled are different. Example 3 has four scheduled outages during a five-year planning horizon, and the pre-specified conditions are that NO_x reduction is at least 80% during the entire time horizon.

Table 2 about here

4.2 Results for MCFPwSEC

MCFPwSEC found an optimal solution for example 1 with SCR maintenance plan shown in Table 3 and a total cost of $13.64 million. Single cut MCFPwSEC required 231 cuts and 26 seconds of

CPU time. Using multi-cut MCFPwSEC targeting new catalyst actions needed only 8 seconds but 874 cuts were generated. Targeting regenerated actions used 11 seconds and 520 cuts.

Table 3 about here

For example 2, MCFPwSEC found an optimal solution with a total cost of $14.75 million and a maintenance plan that adds cleaned layers in slots 1 and 2 during outages 2 and 1, respectively, and changes in cleaned layers in slots 3 and 4 during outages 5 and 3. Single cut MCFPwSEC found this solution in 29 seconds after 364 cuts. Targeting new layers of catalyst, multi-cut MCFPwSEC obtained the solution in 22 seconds but with 2360 cuts. Multi-cut MCFPwSEC targeting regenerated layers required 14 seconds and generated 909 cuts. For example 2, targeting regenerated layers avoids 1451 cuts and reduces the CPU time with respect to targeting new layers.

For example 3, MCFPwSEC found an optimal solution with a total cost of $6.95 million and an SCR maintenance plan that adds a cleaned layer on slot number 2 on outage 2. Because of the simplicity of the maintenance plan, all three versions of MCFPwSEC required only 1 cut and less than 1 second to solve.

As described in Section 2, the algorithm in Phananiramai *et al.* (2011) includes two major procedures. The schedule generation procedure enumerates all feasibly schedules for the plants, while the optimization procedure solves a set partitioning problem. In Table 4, we compare the CPU time taken by each variant MCFPwSEC with the algorithm in Phananiramai *et al.* (2011). In these results, the schedule generation procedure was conducted on the same workstation as the one used for the MCFPwSEC results, but a workstation with an Intel(R) Pentium(R) P6100 2.00GHz processor performed the optimization procedure. As shown in the table, MCFPwSEC was considerably faster than the algorithm in Phananiramai *et al.* (2011). Single cut MCFPwSEC was 83% faster on average and 82% faster solving example 2, which was the least improved in CPU time. Multi-cut MCFPwSEC targeting new catalyst layers had an improvement ranging between 86% to 95% and 91% on average, while targeting regenerated layers improved CPU time between 91% and 93%.

Table 4 about here

5. Conclusions and future research

In this paper, we proposed a multi-commodity network flow problem (MCFP) with side constraints to solve a Selective Catalytic Reduction (SCR) management problem. We introduced schedule elimination constraints to ensure average daily NO_x reduction is above a predetermined level, and we proposed a cutting plane algorithm (MCFPwSEC) to solve the model. We then introduced a multi-cut MCFPwSEC algorithm where we considered two different targeted actions to add multiple cuts per iteration. Finally, we provided results on three examples and showed that multi-cut MCFPwSEC performs better than single cut MCFPwSEC, and both are considerably more computationally efficient than the algorithm in Phananiramai *et al.* (2011). However, MCFPwSEC is for only a single plant, while Phananiramai *et al.* (2011) can optimize an SCR management plan for a fleet of plants. Consequently, future research includes extending MCFPwSEC to optimize a fleet of plants.

References

Ahuja, R., Magnanti, T. and Orlin, J. (1993). *Network flows: theory, algorithms, and applications.* Prentice-Hall, Inc.

Barnhart, C., Boland, N., Clarke, L., Johnson, E., Nemhauser, G. and Shenoi, R. (1998). Flight string models for aircraft fleeting and routing. *Transportation Science,* 32(3), 208–220.

Bellmore, H. and Ratliff, H. (1971). Optimal defense of multi-commodity networks. *Management Science,* 18(4), B174–B185.

Bixby, R. and Cunningham, W. (1980). Converting linear programs to network problems. *Mathematics of Operations Research,* 5(3), 321–357.

Chen, J. and Frey, H. (2004). Optimization under variability and uncertainty: A case study for NOx emissions control for a gasification system. *Environmental Science & Technology*, 38(24), 6741-6747.

Cheng, M. X., Li, Y. and Du, D. (2006). *Combinatorial Optimization in Communication Networks.* Springer-Verlag New York, Inc.

Cichanowicz, J. and Muzio, L. (2003). Factors affecting selection of a catalyst management strategy. Power Plant Air Pollutant Control Mega Symposium: DOE/EPRI/EPA/AWMA. Washington, DC: *Proceedings of the DOE-EPRI-EPA-AWMA.*

Cichanowicz, J. E., Muzio, L. J., and Hein, M. C. The First 100 GW of SCR in the U.S. - What Have We Learned? Paper # 129. Mega Symposium, Baltimore, Maryland, 08/06. http://www.ferco.com/Files/Paper129.pdf

Dantzig, G. (1951). Application of the simplex method to a transportation problem. In *Activity Analysis and Production and Allocation*, T. C. Koopmans (pp. 359–373). New York: Wiley.

Farvolden, J., Powell, W. and Lustig, I. (1993). A primal partitioning solution for the arc-chain formulation of a multicommodity network flow problem. *Operations Research*, 41(4), 669–693.

FERCo. (2013). CatalysTraK® - Manage: SCR Catalyst Management Software. Retrieved 02 25, 2013, from *Specialists in Environmental and Energy Technologies*: http://www.ferco.com/SCR-Catalyst-Management-CatalysTraK-Manage.html

Fontes, D., Hadjiconstantinou, E. and Christofides, N. (2006). A branch-and-bound algorithm for concave network flow problems. *Journal of Global Optimization*, 34(1), 127–155.

Ford, L. and Fulkerson, D. (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, 8, 399–404.

Goffin, J.-L., Gondzio, J., Sarkissian, R. and Vial, J.-P. (1996). Solving nonlinear multicommodity flow problems by the analytic center cutting plane method. *Mathematical Programming,* 76, 131-154.

Gomory, R. and Hu, T. (1961). Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics,* 9(4), 551–570.

Gomory, R. and Hu, T. (1962). An application of generalized linear programming to network flows. *Journal of the Society for Industrial and Applied Mathematics,* 10(2), 260–283.

Hane, C., Barnhart, C., Johnson, E., Marsten, R., Nemhauser, G. and Sigismondi, G. (1995). The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming,* 70(1), 211–232.

Hitchcock, F. (1941). The distribution of a product from several sources to numerous facilities. *Journal of Mathematical physics,* 20, 224–230.

Hu, T. (1963). Multi-commodity network flows. *Operations Research,* 11(3), 344–360.

IBM (2013). CPLEX callable library version 12.5.1. Retrieved August1, 2014, CPLEX > User's Manual for CPLEX > Languages and APIs: http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r5/index.jsp

Kennington, J. (1978). A survey of linear cost multicommodity network flows. *Operations Research,* 26(2), 209–236.

Kjeldsen, N. H. and Chiarandini, M. (2012). Heuristic solutions to the long-term unit commitment problem with cogeneration plants. *Computers & Operations Research,* 39(2), 269-282. DOI:10.1016/j.cor.2011.03.018

Koopmans, T. C. (1947). Optimum utilization of the transportation system. *Proceedings of the International Statistical Conference.* Washington, DC.

McBride, R. (1998). Advances in solving the multicommodity-flow problem. *Interfaces,* 28(2), 32–41.

Milano, M. and Hentenryck, P. V. (2010). *Hybrid Optimization: The Ten Years of CPAIOR.* Springer.

Muzio, L., Quartucy, G. and Cichanowicz, J. (2002). Overview of status of post-combustion NOx control: SNCR, SCR, and hybrid technologies. *International Journal of Environment and Pollution,* 17(1-2), 4-30.

Phananiramai, P., Rosenberger, J. M., Chen, V. C.P., Kim, S. B. and Sattler, M. L. (2011). A mathematical optimization technique for managing selective catalytic reduction for coal-fired power plants. *Energy Systems,* 2(2), 171-188.

Pilla, V., Rosenberger, J., Chen, V. and Smith, B. (2008). A statistical computer experiments approach to airline fleet assignment. *IIE Transactions,* 40(5), 524–537.

Pritchard, S., DiFrancesco, C., Kaneko, S., Kobayashi, N., Suyrgama, K. and Eda, K. (1995). Optimizing SCR catalyst design and performance for coal-fired boilers. Symposium on Stationary Combustion NO Control. Kansas City, MO: *Proceedings of the EPRI/EPA* 1995.

Rosenberger, J., Johnson, E., and Nemhauser, G. (2004). A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science,* 38(3), 357–368.

Rubin, E. S., Kalagnanam, J. R., Frey, H. C. and Berkenpas, M. B. (1997). Integrated Environmental Control Modeling of Coal-Fired Power Systems. *Journal of the Air & Waste Management Association,* 47(11), 1180-1188.

Staudt, J. and Engelmeyer, A. (2003). SCR catalyst management strategies-modeling and experience. Power Plant Air Pollutant Control Mega Symposium: DOE/EPRI/EPA/AWMA. Washington, DC: *Proceedings of the DOE-EPRI-EPA-AWMA.*

Tennessee Valley Authority. (n.d.). Nitrogen Oxides. http://www.tva.com/environment/air/nox.htm

The U.S. Department of Energy and Southern Company Services, Inc. (1997, July). Topical report number 9, control of nitrogen oxide emissions: Selective catalyst reduction (SCR).

Yan, S., Tseng, C.-H. (2002). A passenger demand model for airline flight scheduling and fleet routing. *Computers & Operations Research*, 29(11), 1559-1581.

Figure 1. Example of the MCFP variable creation.

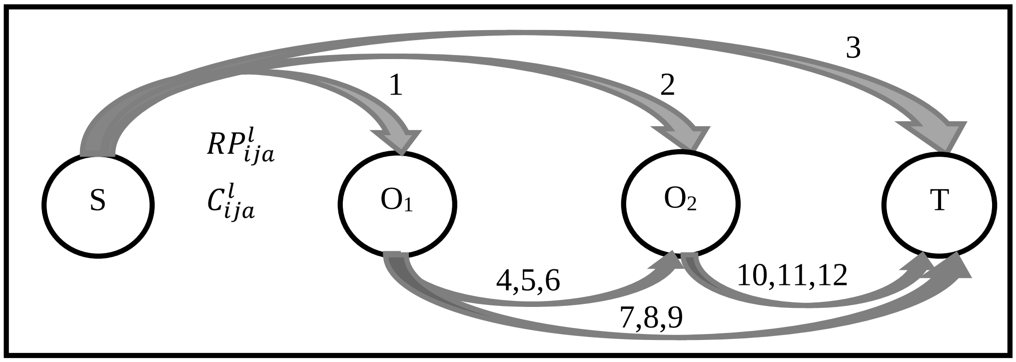

Figure 2. Example with filled layer at source.

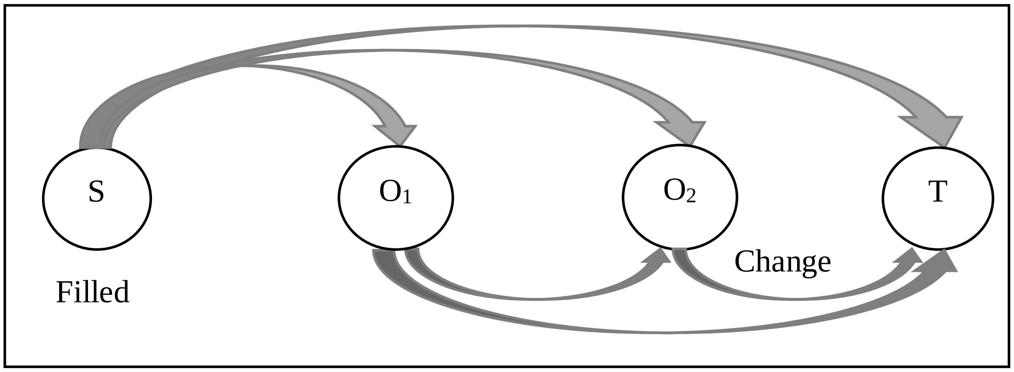

Figure 3. Example with empty layer at source.

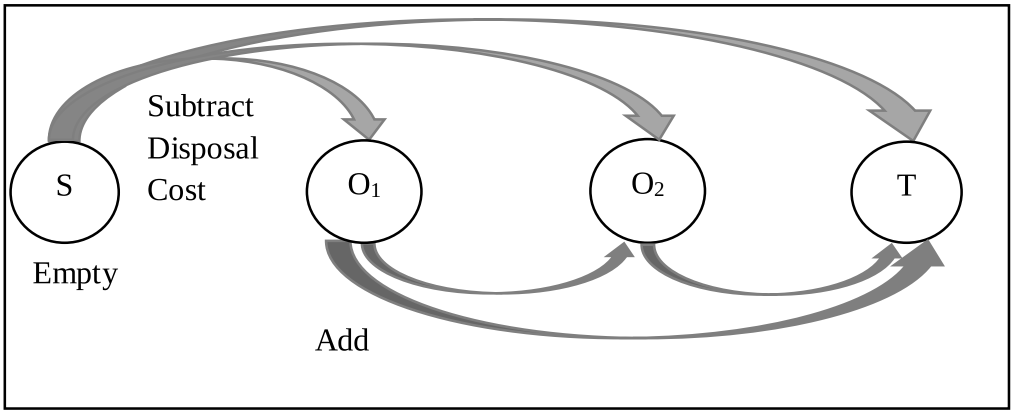

Figure 4. Example of the MCFP with instantaneous RP constraints for controlling peak NO_x emissions.

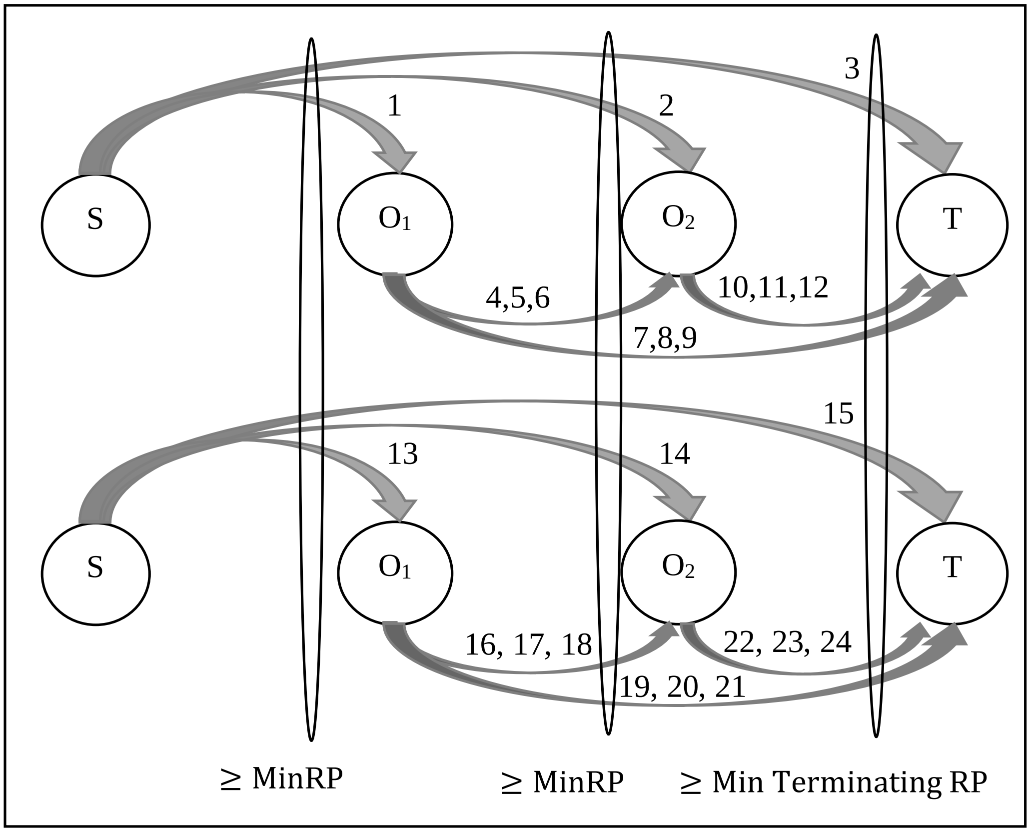

Figure 5. Overview of the MCFPwSEC algorithm.

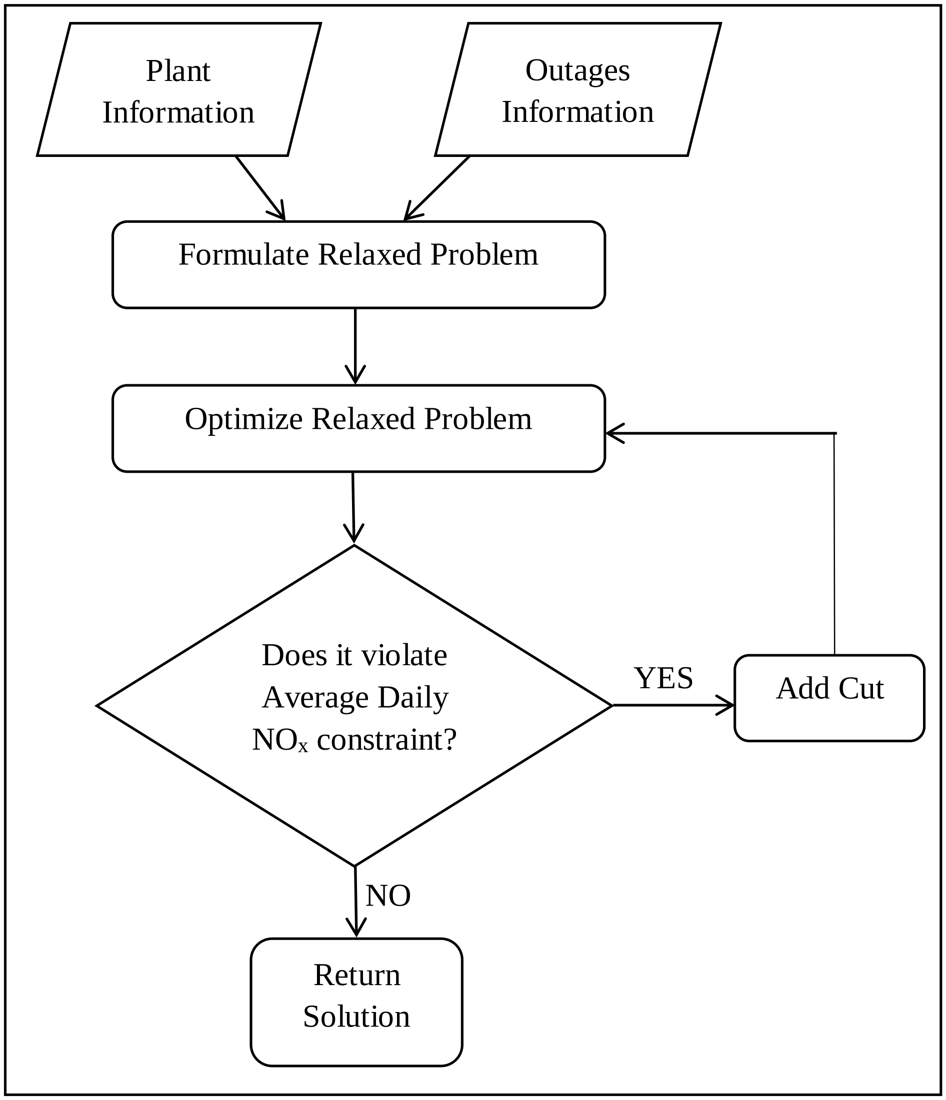

Figure 6. Overview of the multi-cut MCFPwSEC algorithm.

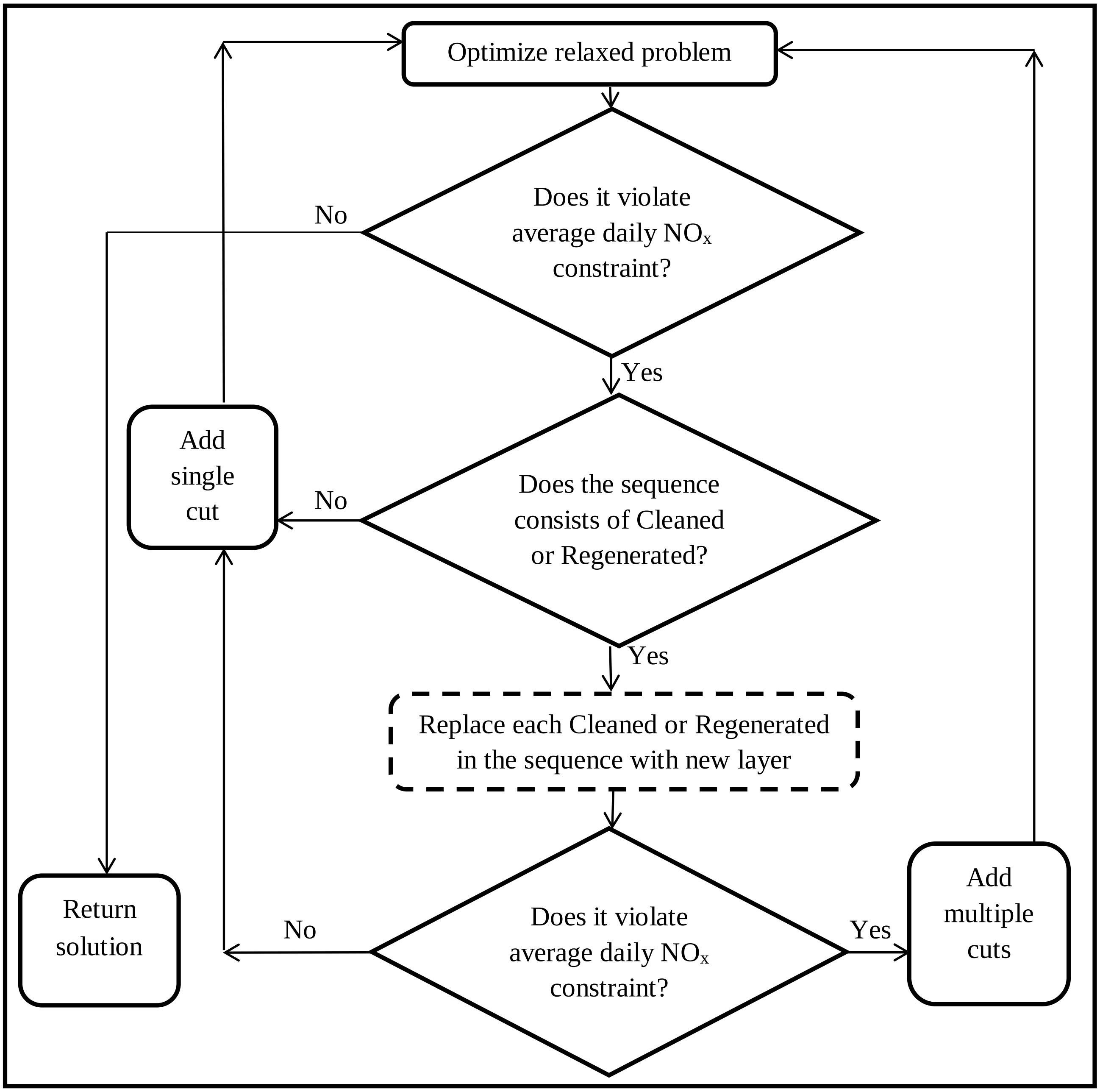

Table 1. Example of multi-cut generation step of the multi-cut MCFPwSEC algorithm when the targeted action is changing in or adding a new layer of catalyst.

ChangeNew, AddNew, ChangeNew
ChangeNew, AddNew, ChangeRegenerated
ChangeNew, AddNew, ChangeCleaned
ChangeNew, AddRegenerated, ChangeNew
ChangeNew, AddRegenerated, ChangeRegenerated
ChangeNew, AddRegenerated, ChangeCleaned
ChangeNew, AddCleaned, ChangeNew
ChangeNew, AddCleaned, ChangeRegenerated
ChangeNew, AddCleaned, ChangeCleaned
ChangeRegenerated, AddNew, ChangeNew
ChangeRegenerated, AddNew, ChangeRegenerated
ChangeRegenerated, AddNew, ChangeCleaned
ChangeRegenerated, AddRegenerated, ChangeNew
ChangeRegenerated, AddRegenerated, ChangeRegenerated
ChangeRegenerated, AddRegenerated, ChangeCleaned
ChangeRegenerated, AddCleaned, ChangeNew
ChangeRegenerated, AddCleaned, ChangeRegenerated
ChangeRegenerated, AddCleaned, ChangeCleaned
ChangeCleaned, AddNew, ChangeNew
ChangeCleaned, AddNew, ChangeRegenerated
ChangeCleaned, AddNew, ChangeCleaned
ChangeCleaned, AddRegenerated, ChangeNew
ChangeCleaned, AddRegenerated, ChangeRegenerated
ChangeCleaned, AddRegenerated, ChangeCleaned
ChangeCleaned, AddCleaned, ChangeNew
ChangeCleaned, AddCleaned, ChangeRegenerated
ChangeCleaned, AddCleaned, ChangeCleaned

Table 2. Scheduled outages for example 1.

Outage	Start Date	End Date
1	03/15/2020	03/29/2020
2	11/16/2021	11/24/2021
3	10/27/2022	11/04/2022
4	04/06/2023	05/04/2023
5	10/24/2024	11/01/2024

Table 3. Optimal solution for example 1.

Start Date	End Date	Action	Layer
03/15/2020	3/29/2020	AddRegenerated	2
11/16/2021	11/24/2021	AddRegenerated	1
4/6/2023	5/4/2023	ChangeRegenerated	4

Table 4. Algorithm comparison in CPU time (seconds).

Example	Single Cut MCFPwSEC	Multi-cut MCFPwSEC		Phananiramai *et al.* (2011)		
		Targeting New Layers	Targeting Regenerated Layers	Schedule Generation	Optimization	Total
1	26	8	11	149	8	158
2	29	22	14	155	10	165
3	1	1	1	9	4	13