OPTIMIZATION OF HYBRID DYNAMIC/STEADY-STATE PROCESSES USING PROCESS INTEGRATION

A Dissertation

by

DANIEL DOUGLAS GROOMS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2006

Major Subject: Chemical Engineering

OPTIMIZATION OF HYBRID DYNAMIC/STEADY-STATE PROCESSES USING PROCESS INTEGRATION

A Dissertation

by

DANIEL DOUGLAS GROOMS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Mahmoud M. El-Halwagi
Committee Members,	Juergen Hahn
	M. Sam Mannan
	Guy L. Curry
Head of Department,	Kenneth R. Hall

August 2006

Major Subject: Chemical Engineering

ABSTRACT

Optimization of Hybrid Dynamic/Steady-State Processes Using Process Integration.
(August 2006)
Daniel Douglas Grooms, B.S., Lamar University
Chair of Advisory Committee: Dr. Mahmoud M. El-Halwagi

Much research in the area of process integration has focused on steady-state processes. However, there are many process units that are inherently unsteady-state or perform best when operated in an unsteady-state manner. Unsteady-state units are vital to chemical processes but are unable to be included in current process optimization methods. Previous methods to optimize processes containing unsteady-state units place restrictions or constraints on their use. This optimization still does not give the best system design because the solution found will only be the best out of the available options which likely excludes the true optimal design. To remedy this, a methodology was created to incorporate unsteady-state process units into process optimization analysis. This methodology is as general as possible. Unlike many existing unsteady-state optimization methods, it determines all three main components of process design: the network configuration, sizes of units, and operation schedule. This generality ensures that the truly optimal process design will be found.

Three problems were solved to illustrate the solution methodology. First, a general mass exchange network was optimized. The optimization formulation resulted in a mixed-integer nonlinear program, and linearization techniques were used to find the global solution. A property interception network was also optimized, the first work done using property integration for systems with unsteady-state behavior. Finally, an industrial semi-batch water purification system was optimized. This problem showed how process integration could be used to optimize a hybrid system and gain insights into the process under many different operating conditions.

DEDICATION

To my family, in particular my parents. For raising myself, my brother and sisters in a loving household and for always encouraging us no matter how silly we were at the time. For showing us how to love and care and to enjoy learning about the world. Most importantly, for teaching us to have faith, to believe in ourselves, and to believe in others.

ACKNOWLEDGMENTS

I would like to express my profound gratitude to my advisor, Dr. Mahmoud El-Halwagi for his wise guidance and council. There have been so many times when he has inspired me to see things I wouldn't otherwise have and helped me discover paths through roadblocks. It has been a great pleasure to work with a gentle, honorable, and insightful mentor. I can say that I have become a better person because of Dr. El-Halwagi and for that I am very grateful.

I am thankful to my committee members, Dr. Hahn, Dr. Mannan, and Dr. Curry for their helpful and insightful suggestions. Having had the pleasure of taken classes from every member of my committee, it has been a privilege to learn from and get to know each of them.

I would like to thank my fellow research group members who have helped me both professionally and personally on this journey. I would like to especially thank Eva, Vasiliki, and Qin, who have helped me better understand everything from salsa dancing and world politics to property integration.

I would also like to thank my church and my friends that I have met there who have helped me not take myself too seriously and encouraged me when I needed it. I've met too many special people to name them all here, but Linda, Lance, Karl, LuBecca, and Ruth have been particularly great blessings in my life.

Finally, I would like to express appreciation to the Department of Education for providing funding during my time here.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE Page

NOMENCLATURE

ADMI	American Dye Manufacturers Index
B	Flowrate of impurity salt/complex
BOD	Biological Oxygen Demand
c	Component index
c_{in}	Impurity mass fraction in incoming contaminated water
c_{out}	Impurity mass fraction in purified water
c^{max}	Maximum component mass fraction
c^{min}	Minimum component mass fraction
c^{int}	Interceptor outlet component mass fraction
C_D	Discharge cost factor
C_L	MSA cost factor
C_R	Regenerant cost factor
C_V	Process vessel cost factor
C_W	Wash material cost factor
$COMPONENTS$	Set of components
d	Vector of design variables
D	Flowrate of total system discharge
D_{Pur}	Diameter of main purification vessel
D_R	Diameter of regeneration vessel
D_W	Diameter of wash vessel
D^{col}	Column diameter
ELT	Capital equipment expected lifetime
F	Sink flowrate
F_{water}	Water flowrate
g	Flowrate from source to interceptor
G	Source flowrate
h	Flowrate from interceptor to sink/recycle stream
H	Interceptor flowrate
H_{Reg}	Flowrate of regenerant complex

$({}^{H}/_{D})_{Spec}$	Process unit height-to-diameter specification
i	Source index
I	Flowrate of impurity
INTERCEPTORS	Set of interceptors
j	Sink index
k	Interceptor index
L	MSA flowrate
L_{Flux}	MSA flux specification
$L_{Make\text{-}up}$	Flowrate of make-up MSA
Lo	Variable lower bound
M	Number of wash units
M_{Na}	Sodium atomic weight
M_{NaCl}	Sodium chloride molecular weight
MEN	Mass Exchange Network
MILP	Mixed Integer Linear Program
MINLP	Mixed Integer Nonlinear Program
MLD	Minimum Liquid Discharge
MSA	Mass Separating Agent
n	Time interval index
N	Number of regenerating units
N_c	Number of components
N_{Int}	Number of interceptors
N_p	Number of properties
N_{sinks}	Number of sinks
$N_{sources}$	Number of sources
o	Vector of operating variables
p	Property index
p_{max}	Sink property requirement upper limit
p_{min}	Sink property requirement lower limit
p^{in}	Interceptor inlet property value
p^{int}	Interceptor outlet property value

PIN	Property Interception Network
PROPERTIES	Set of properties
q	Measure of unsteady-state unit saturation
r	Flowrate from recycle stream to interceptor
R	Flowrate of regenerant
R_{Flow}	Flowrate of recycle stream
$(^{R}/_{L})_{\min}$	Minimum regenerant-to-MSA ratio
R_{Flux}	Regenerant flux specification
R_{Fresh}	Flowrate of purchased regenerant
R_{ratio}	Regenerant-to-MSA ratio
R_{XS}	Flowrate of regenerant in excess of stoichiometric amount
Regen	Binary regeneration variable
RelaxationDeviation	Integer relaxation penalty term
SINKS	Set of sinks
SOURCES	Set of sources
SS	Subset of steady-state sources
t	Time variable
T	Flowrate of regenerant dilution
T_R	Time required for regeneration
T_W	Time required for washing
TAC	Total Annualized Cost
TIME	Set of time intervals
TOC	Total Organic Carbon
u	Source component value
U	Variable upper bound
v	Recycle stream index
V_R	Volume of regeneration vessel
V_{Total}	Total regeneration system volume
V_W	Volume of wash vessel
w	Recycle stream component value
W	Flowrate of wash material

W_{ratio}	Wash-to-MSA ratio
$Water$	Flowrate of process water discharged
x_{clean}	Impurity mass fraction in regenerated MSA
x_{used}	Impurity mass fraction in saturated MSA
x^s	MSA supply component value
x^t	MSA target component value
y_{MSA}	MSA deactivation fraction
y^{in}	Interceptor entering stream component value
y^{out}	Interceptor exiting stream component value
z	Sink stream component value
Δ	Change in variable
ε	MSA fluidization void fraction
τ	Length of system cycle
$\psi(p)$	Property mixing operator

CHAPTER I
INTRODUCTION

The past two decades have witnessed key research contributions in the field of process integration. Nonetheless, much of the research has a common limitation: it is based on optimizing steady-state systems. Steady-state units are convenient in that once the operating conditions of the unit are specified, the behavior is also specified for as long as the system variables remain unchanged. Thus, optimizing a steady-state process is a matter of optimizing a system for a single set of conditions and the solution will be optimum for as long as the process conditions remain constant. However, in many cases, process units are inherently unsteady-state or are best operated in a dynamic mode. So at best, optimizing a dynamic process is similar to optimizing several sets of steady-state problems. Reality is much more complicated, however, because of the interactions the separate problems must have to ensure continuity. Examples of dynamic units include fixed-bed adsorbers, ion exchange beds, filters, etc. Because unsteady-state units are vital to chemical processes, it is necessary to incorporate these dynamic units in the analysis and develop a systematic procedure for the optimum design and operation of the system.

Currently when optimizing most chemical processes, dynamic units are either not used at all or restrictions are placed on their behavior because of these complications. However, optimization done on a system with restrictions placed on it is not true optimization. An optimum solution found under these conditions will only be the best out of the available options. For example, consider a constraint that is often used in environmental applications: A steady-state pollutant-removal process is optimized save for an unsteady-state mass separator that is used at the end of a process to remove the final amount of a pollutant. The separator’s effluent concentration is not constant, but it is regenerated often enough so that the concentration is always below the environmental

This dissertation follows the style and format of *Chemical Engineering Communications*.

regulation limit. Because of the inability to include the unsteady-state unit in the optimization, its place in the network is fixed regardless of whether this is the optimum location. As seen from this example, if a method were developed to include unsteady-state units in the optimization of the process, it would be possible to optimize processes more generally and be confident that the resulting solution is truly the best solution.

Some of the challenges of dealing with unsteady-state units have already been mentioned. The primary difficulty is that unlike existing methods for steady-state systems, not only must the system design be formulated, but the optimal design and operation profile must be found simultaneously for all of the unique system conditions. Also, in the same way that the performance of steady-state units is modeled, the behavior of the dynamic units must also be described. Naturally, this is more difficult because the performance of unsteady-state units varies with time. Finally, unsteady-state units are usually unsteady-state because they operate in some kind of batch routine. That is, the unit typically operates for a period of time, then stops operating and undergoes some kind of regeneration or reloading. This behavior introduces the complication of determining the lengths of time the unit operates and regenerates because the degradation of the unit's performance with time must be balanced with the cost of regenerating the unit and the loss of the unit's availability during that time. This scheduling must be done so that the needs of the process are met.

Early work on scheduling dynamic systems involved a mixed integer linear programming (MILP) formulation for short-range production planning of multipurpose batch plants (Rich and Prokopakis, 1987), which allowed the use of different production routings and facilitated preventive maintenance. Later work from Kondili et al. (1993) described a (MILP) formulation of a general framework for handling multipurpose batch plants using the state–task network (STN) notion under discrete time representation. Reformulation of this work demonstrated improvement of the computational efficiency (Shah et al., 1993). Both publications from Kondili et al. and Shah et al. have led to many applications and extensions based on the STN representation (Pinto and Grossmann, 1994, Ierapetritou and Floudas, 1998a and 1998b, Lee et al., 2001, Maravelias and Grossmann 2003 and 2004). Although useful, these procedures optimize processes containing solely unsteady-state operations which is a special case of the problem being addressed here.

There has also been research into methods to optimize networks containing units with performance decay and how to schedule their operation around cleaning and replacement. Pinto and Grossmann (1994) first developed a general model (MINLP) addressing the cyclic scheduling of multistage multiproduct continuous plants where the structure of the network is specified and Alle et al. (2004) extended this work of simultaneously optimizing the production and cleaning operations to include units with performance decay. The problem of optimizing the operation of multiproduct plants with parallel units that exhibit exponential performance decay was addressed by Jain and Grossmann (1998) for a single stage process. The resulting model was again applicable only for networks with known structures. The present work addresses optimization of not only production schedules, but also has the capability of designing the network structure and individual unit sizes.

Soileau (1998) introduced the idea of optimizing the production scheduling of processes containing unsteady-state units by including unsteady-state mass exchange units in a mass exchange network with the restriction that the dynamic units are grouped together rather than intermixed with the steady-state units. He optimized a group of columns using pressure swing adsorption (PSA) or temperature swing adsorption (TSA) such that the unsteady-state units were grouped and operated together in such a way as to produce steady-state output. Thus, the group of unsteady-state units behaved as a single steady-state unit and could be placed anywhere in the network so that existing methods for optimizing steady-state systems could be used. This method placed a major restriction on the network, grouping the unsteady-state units together. So the optimum design found using this method was only the best design that met this restriction.

There has been a substantial amount of work that has been done on a general problem of optimizing an unsteady-state process or unit based on differential and algebraic equation (DAE) models (Cervantes et al., 2000; Jockenhovel et al., 2003). However, this approach is very detailed and requires a large amount of process and modeling information. The approach in this paper requires much less detailed information about the process and is to be used in a more general design phase. This follows the process integration philosophy of developing the overall process design first, and calculating the specific profiles of the units later. This approach is useful because it enables one to formulate an optimum overall process design without requiring detailed information.

The works described above are important background for the present problem but are significantly different. The method presented here simultaneously optimizes the operation and structure of the network and the sizes of the units. Here, the term "network structure" refers to the configuration of the process (units in series vs. parallel or some combination of the two) while the operation of the process consists of running times and distribution of flows to the units of the process. Previous work optimized the network structure but assumed the processing times and unit sizes were known (as in Kondili et al. (1993). Alternatively, the processing time was left to be optimized, but the network structure is fixed (as in Jain and Grossmann (1998) and Alle et al. (2004)). As will be described in more detail later, this work describes a solution to the problem of optimizing the three areas of network structure, process operation, and unit sizing in a chemical or other manufacturing process containing both dynamic and steady-state unit operations.

The objective of this work is to introduce a novel method of synthesizing an optimal interception and allocation network structure, physical unit sizes, and also the operation schedule of the hybrid (steady-state and dynamic) process. This method may be used for many diverse applications, as will be shown. Case studies and examples are used to illustrate the usefulness of the method.

CHAPTER II
PROBLEM STATEMENT

This dissertation presents a solution formulation to the problem that is formally introduced and described below.

Given a process with:

- A set of sources: SOURCES = $\{i \mid i = 1,2, \ldots, N_{sources}\}$ composed of process streams, which are available to be allocated. Each source has a given flowrate, G_i, and is characterized as containing a set of components: COMPONENTS = $\{c \mid c = 1,2, \ldots, N_c\}$. The mass fractions of the components of the sources are also given and designated by $c_{i,c}^{in}$.

- A set of process sinks: SINKS = $\{j \mid j = 1,2, \ldots, N_{sinks}\}$. Sinks are process units or other discharge locations that can accept the sources. Each sink requires a given flowrate, F_j, and mass fractions, $c_{j,c}^{in}$, that satisfy the following constraints:

$$c_{j,c}^{\min} \le c_{j,c}^{in} \le c_{j,c}^{\max} \qquad j \in \text{SINKS}, c \in \text{COMPONENTS} \qquad (2.1)$$

where $c_{j,c}^{\min}$ and $c_{j,c}^{\max}$ are given lower and upper bounds on acceptable mass fractions of component c in unit *j*.

- A set of interception units: INTERCEPTORS = $\{k \mid k = 1,2, \ldots, N_{Int}\}$. These units can be added to the process to adjust the mass fractions of the sources to meet the sink requirements. Some of the interceptors are operated dynamically with the following performance equations:

$$c_{i,c}^{\text{int}} = f\left(c_{i,c}^{in}, d_k, o_k, t\right) \qquad (2.2)$$

where $c_{i,c}^{\text{int}}$ is the mass fraction of the c^{th} component leaving the k^{th} interceptor, d_k and o_k are design and operating variables of the k^{th} interceptor, and t is time of operation. Clearly, if the unit is steady-state, its intercepted components are independent of *t*.

The problem statement is schematically represented in Figure 2.1.

Some examples of types of sinks in reality are: product storage tanks, biological treatment ponds, and downstream processing areas. Typical sources are reactors or raw material suppliers. Interceptors may be any mass exchange units such as fixed bed adsorbers, distillation columns, ion exchange beds, countercurrent absorbers, etc. The process streams typically will be product streams that require purification or waste streams containing environmentally harmful compounds to be removed before the streams are discharged.

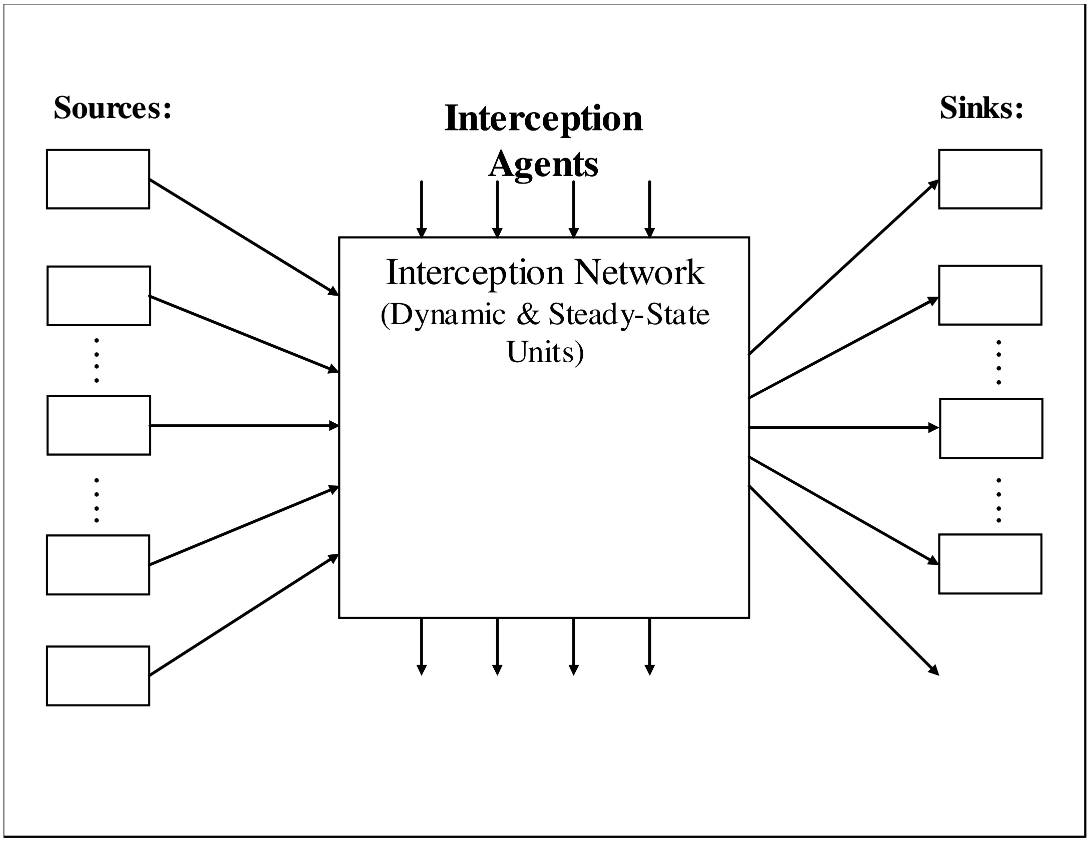

Figure 2.1. Problem Statement Schematic

The objective is to synthesize and schedule a network of sources-interceptors-sinks that provides optimal allocation and interception of sources while satisfying the constraints of the sinks at minimum cost. In particular, the following are some design challenges associated with the problem:

- Which types of interception units should be selected? What sizes?
- How should the sources be allocated to the interception devices? Should sources be segregated, mixed, or split prior to interception? What should be the extent of interception of each source?
- How should each intercepted source be allocated? Should it be mixed with other sources, intercepted or otherwise?
- How should the interception and allocation schemes be scheduled so as to cope with the time-varying performance of the interceptors while satisfying the steady-state requirements of the sinks?

The following chapters will describe problems demonstrating these characteristics and will address the above questions when solving the problems.

CHAPTER III
OPTIMAL SYNTHESIS AND SCHEDULING OF HYBRID DYNAMIC/STEADY-STATE MASS EXCHANGE NETWORKS

3.1 Introduction

While unsteady-state units could be incorporated into many aspects of chemical processes, a mass exchange network (MEN) was chosen to be a subject of this research for two main reasons. First, much research has been done on mass integration in general and optimizing mass exchange networks in particular. Because of this, there is a wealth of work to compare results with and many opportunities to extend this concept. Second, mass exchange is a very important aspect of a production process. Due to the increased regulation and emphasis on minimizing industry’s impact on the environment, cleaning up products and recovering harmful chemicals has become a vital part of petrochemical processes and mass exchange operations are one of the primary means to achieving this objective. In fact, one mass exchange operation, fixed-bed adsorption, is a very useful tool in mass separation but it is an intrinsically unsteady-state operation and this fact has discouraged its use in mass integration research because of the reasons outlined above.

This chapter introduces a novel approach for optimizing the design and operation of mass exchange networks (MENs) utilizing both dynamic and steady-state units. These networks reduce the mass of a targeted species in a given set of rich streams from known supply concentrations to target concentrations. Although the network contains a combination of both dynamic and steady-state units, the output variables of the network are constant with time. Design tasks include selecting the optimal mass exchange units, the optimal mass separating agents (MSAs), the optimal network configuration, and the best operation schedule. A flexible structural representation is used to include all possible configurations of the system. Physical constraints are then obtained from the structural representation and the performance of the mass exchange units. Different time scales are used for each of the units and related to the overall timeline to optimize the operation time and schedule for each unit. Also, a backward-moving horizon is employed to represent regeneration of the dynamic units. The formulation yields a mixed-integer nonlinear program (MINLP) which is linearized to obtain a mixed-integer linear program (MILP). The MILP provides a lower bound and initial point that are used to solve the MINLP and optimize the design of the system as well as its operation. A sample application of this methodology is also included.

The structural representation used here was based in part on the mathematical optimization of segregation, mixing, and recycle for steady-state processes (El-Halwagi 1997, p.176). However, previously it was used to calculate only the lean stream flows for a system structure that was already given.

There has also been work done on optimizing operation and scheduling for existing batch processes (Pinto and Grossmann, 1994; Rich and Prokopakis, 1987) and minimizing waste from batch plants (Sharratt, 1993; Wang and Smith, 1995). Both of these areas are more restrictive than the more general optimization that is the goal here.

Grossmann and Beigler (2004) and Floudas et al. (2005) give a good review of global optimization techniques currently in use. McCormick (1976) and Al-Khayyal and Falk (1983) developed a general algorithm for globally solving nonlinear programs using convex underestimators, which Quesada and Grossmann (1995) used in part for their spatial branch and bound global optimization algorithm. Szitkai et al. (2006) used a relaxation method with a penalty term in the objective function for integer variables. The relaxation allows solutions that otherwise would be infeasible to become feasible but with a penalty. This allows more solutions to be in the search region, reducing the chance of the algorithm becoming trapped at a local solution.

This work is not primarily concerned with global optimization techniques but it was necessary to explore this area because the formulation produced a MINLP. The procedure used to solve the MINLP was similar to the procedure set forth by Quesada and Grossmann (1995). The nonlinear constraints in the MINLP were approximated with linear inequalities and a MILP was obtained. The MILP was solved and this gave a lower bound for the MINLP and also an initial point for solving the MINLP. The

MINLP was then solved and the solution was compared to the lower bound. The global optimization process will be described in more detail later in this paper.

3.2 Solution Approach

3.2.1 Representation and Terminology

To simplify the mathematical formulation, we focus on a single key component, *c*. Using a source-interception-sink representation, each source is first segregated and split into fractions of unknown flow rate (to be optimized). Those fractions are allocated to the interception network. A combination of steady- and unsteady-state mass exchange units are available for use and are numbered $k = 1,2,\ldots,N_{Int}$ inside the mass exchange network (MEN). Within the MEN, the component mass fractions of each stream are altered if an interception device is used. The optimum extent of interception for each component and stream is to be determined. A stream may also pass through the MEN unchanged indicating that the stream is not intercepted. The streams leaving the interception network are allowed to mix and are fed to the process sinks. The optimum mixing ratios are to be determined. An additional sink is used to account for the waste material (unrecycled/unreused material), and it is designated as the “waste” sink. Also, an external fresh stream may be used in process sinks but not the waste sink. The network representation is shown in Figure 3.1.

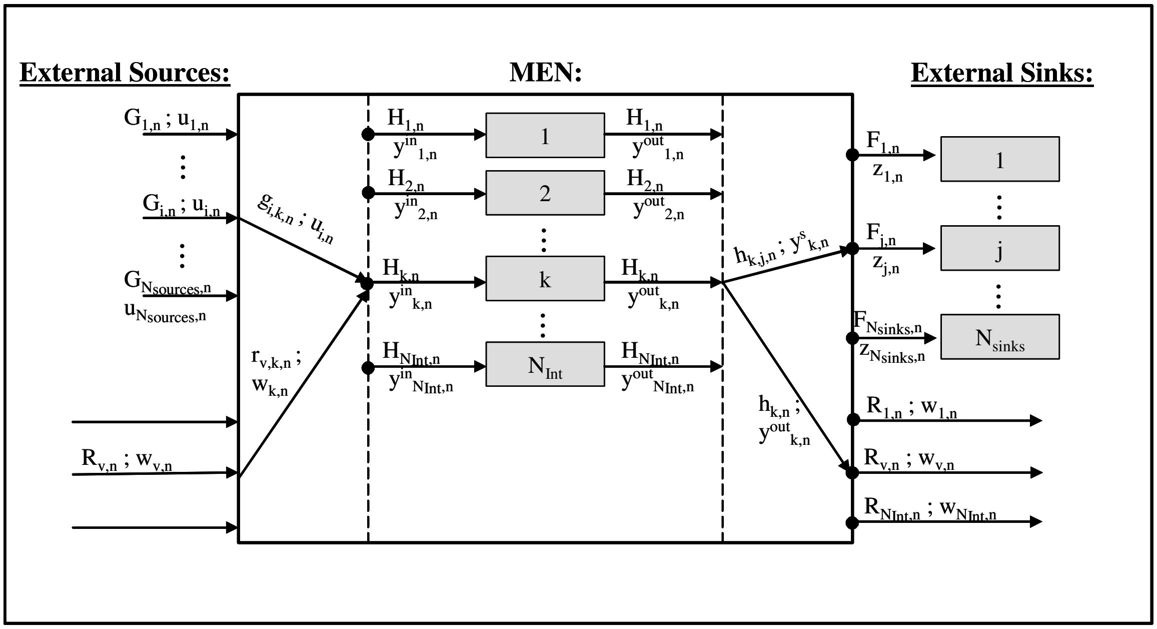

Figure 3.1 Source-Interception-Sink Representation

In order to develop a scheduling scheme, we start first by describing the process's operating time through a number of repeating and identical episodes. The duration of the episode corresponds to the cycle time. Because of the repeating nature of the episodes, it is sufficient to optimize the process over an episode and the process will then be optimized into infinity. To insure continuity of the episodes, constraints are added such that the state of the process at the end of the episode is identical to the state at the beginning of the episode. This episode time is discretized into a number of time intervals over which the process is optimized. Structural changes (e.g., starting or ending the operation of a unit, starting or ending the regeneration of a unit, changing the destination of a stream) are performed at the beginning (or end) of a time interval. The index *n* is used to denote a time interval.

Each source is split into fractions that are passed to the interception devices. The flowrate from source *i* to interceptor *k* is called $g_{i,k,n}$, while the flowrate from interceptor *k* to sink *j* is labeled $h_{k,j,n}$. As mentioned earlier, the flowrate through an interceptor *k* is denoted $H_{k,n}$ while the component mass fraction entering and exiting the interceptor is designated by $y^{in}_{k,n}$ and $y^{out}_{k,n}$, respectively. It is also possible to recycle streams leaving interception devices and, in turn, they become new sources. The index for these recycled streams is called *v*. The flowrate and mass fraction values of a recycle stream are represented as $R_{v,n}$ and $w_{v,n}$.

Now that the terms have been defined, let us revisit the structural representation. A number of process and fresh source streams which have certain properties that need to be modified are numbered 1 through $N_{sources}$ and are given with flowrates and mass fraction values ($G_{i,n}$ and $u_{i,n}$). Also, a number (N_{sinks}) of process sinks are known along with their flowrate and component mass fraction requirements ($F_{j,n}$ & $z_{j,n}$). A mixture of steady- and unsteady-state mass exchange units that are of interest are selected and numbered 1,2,…,N_{Int} inside the mass exchange network (MEN). The flow from source *i* to unit *k* is represented as $g_{i,k,n}$ while the flow from unit *k* to sink *j* is labeled $h_{k,j,n}$. The flow rate through a unit is denoted $H_{k,n}$ while the mass fractions entering and exiting the unit are $y^{in}_{k,n}$ and $y^{out}_{k,n}$ respectively. Finally, flowrates and component mass fraction values of possible recycle streams for a mass exchange unit *v* are represented as $R_{v,n}$ and $w_{v,n}$. These recycle streams serve both as sinks and as sources. The indices *i, j*, and *k* refer to sources, sinks, and mass exchange units respectively, while the index *n* specifies the time interval in question. The indices *k* and *v* are similar, both referring to mass exchange units; the only difference is that they are used in different contexts – *k* for describing the units and *v* for describing the recycle streams coming from those units.

3.2.2 Solution Formulation:

Performing a material balance on the splitting of the source streams yields the equation

$$G_{i,n} = \sum_k g_{i,k,n} \quad \forall i \in SOURCES,\ n \in TIME \tag{3.1}$$

where *TIME* is the set of time intervals. Total and component mass balances at the mixing of the source-to-interceptor nodes yields:

$$H_{k,n} = \sum_i g_{i,k,n} + \sum_v r_{v,k,n} \quad \forall k \in INTERCEPTORS,\ n \in TIME \tag{3.2}$$

$$y_{k,n}^{in} \cdot H_{k,n} = \sum_i u_{i,n} \cdot g_{i,k,n} + \sum_v w_{v,n} \cdot r_{v,k,n} \quad \forall k \in INTERCEPTORS,\ n \in TIME \tag{3.3}$$

A balance on the splitting of the interceptor outlet streams gives the equation

$$H_{k,n} = \sum_j h_{k,j,n} + h_{k,n} \quad \forall k \in INTERCEPTORS,\ n \in TIME \tag{3.4}$$

where $h_{k,n}$ refers to the flow from interceptor *k* to its recycle stream.

Balances on the total flow and the components at the mixing of the streams to the process sinks give the following equations:

$$F_{j,n} = \sum_k h_{k,j,n} \quad \forall j \in SINKS,\ n \in TIME \tag{3.5}$$

$$z_{j,n} \cdot F_{j,n} = \sum_k y_{k,n}^{out} \cdot h_{k,j,n} \quad \forall j \in SINKS,\ n \in TIME \tag{3.6}$$

Finally, each mass exchange unit has its own recycle stream. The flowrate and concentrations of the recycle streams are specified by the equations

$$R_{v,n} = h_{k,n} \quad \forall v = k \in INTERCEPTORS,\ n \in TIME \tag{3.7}$$

$$w_{v,n} = y_{k,n}^{out} \quad \forall v = k \in INTERCEPTORS,\ n \in TIME \tag{3.8}$$

To account for the mass lost to the mass separating agents (MSAs), mass balances are also performed on each unit. Figure 3.2 illustrates the flows in a steady-state unit.

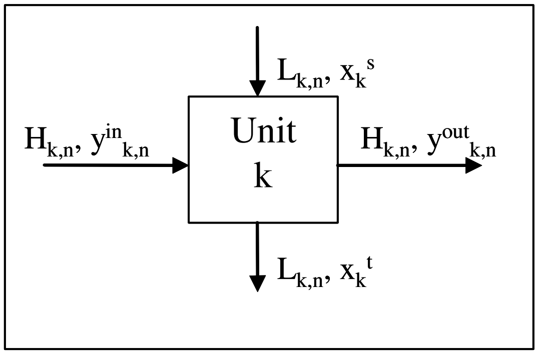

Figure 3.2. Steady-state Unit Balance

For each steady-state interceptor, a supply concentration, x_k^s, and target concentration, x_k^t is given and L_k represents the optimizable mass flowrate of the mass separating agent (MSA) through unit *k*. In time interval *n*, the mass exchanged from the rich stream to the lean MSA stream is given by integrating over t_n, the time in interval *n*.

$$\int_{t_n} L_k(x_k^t - x_k^s)dt = \int_{t_n} H_k(y_k^{in} - y_k^{out})dt \tag{3.9}$$

where L_k and H_k are the instantaneous flowrates of the lean and rich streams and y_k^{in} and y_k^{out} are the instantaneous concentrations in the rich stream. If we let $L_{k,n}$ be the average lean stream flowrate in interval *n*, $H_{k,n}$ be the average rich stream flowrate, and $y_{k,n}^{in}$ and $y_{k,n}^{out}$ be the average rich stream concentrations in interval *n*, this implies that

$$\frac{\partial L_{k,n}}{\partial t_n} = \frac{\partial H_{k,n}}{\partial t_n} = \frac{\partial y_{k,n}^{in}}{\partial t_n} = \frac{\partial y_{k,n}^{out}}{\partial t_n} = 0 \tag{3.10}$$

So, the mass balance equation becomes

$$L_{k,n}(x_k^t - x_k^s)\int_{t_n} dt = H_{k,n}(y_{k,n}^{in} - y_{k,n}^{out})\int_{t_n} dt \tag{3.11}$$

which simplifies to:

$$L_{k,n}\left(x_k^t - x_k^s\right) = H_{k,n}\left(y_{k,n}^{in} - y_{k,n}^{out}\right) \quad \forall k \in SS \subset INTERCEPTORS, n \in TIME \tag{3.12}$$

where *SS* is the subset of steady-state units in the set *INTERCEPTORS*.

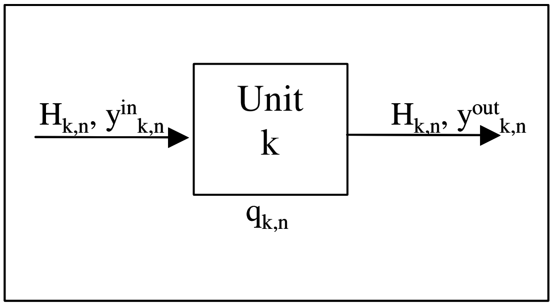

Figure 3.3. Unsteady-state Unit Balance

Figure 3.3 is a diagram of an unsteady-state unit where $q_{k,n}$ represents the amount of accumulated mass contained in the unit over time interval n. This load can be calculated by the following equation:

$$\int_{t_n} H_k (y_k^{in} - y_k^{out}) dt = \int_{t_n} q_k dt \tag{3.13}$$

where H_k is the instantaneous flowrate of the rich stream through unit k, y_k^{in} and y_k^{out} are the instantaneous component mass fractions of the rich stream, and q_k is the instantaneous load of unit k. To simplify the formulation, we assume that within a time interval, the interceptor operates in a steady-state mode so its flowrate and inlet and outlet mass fractions are taken to be constant over the time interval and are designated by $H_{k,n}$, $y_{k,n}^{in}$, and $y_{k,n}^{out}$. These values change from one interval to another leading to a step-wise approximation of the nonlinear changes in flowrate, and inlet and outlet concentrations. Therefore, equation 3.13 becomes

$$H_{k,n} (y_{k,n}^{in} - y_{k,n}^{out}) \int_{t_n} dt = (q_{k,n+1} - q_{k,n}) \int_{t_n} dt \tag{3.14}$$

where $q_{k,n+1}$ is the load of unit k at the beginning of interval $n+1$. This simplifies to:

$$H_{k,n} (y_{k,n}^{in} - y_{k,n}^{out}) = \Delta q_{k,n} \tag{3.15}$$

where $\Delta q_{k,n}$ is the net load accumulation for unit k in time interval n.

The load of the unit may be tracked using the equation

$$q_{k,n+1} = q_{k,n} + \Delta q_{k,n} \quad n \in TIME \tag{3.16}$$

where $q_{k,0}$ =0 when starting the time period with a clean unit.

Another equation that will be needed is an equation for the outlet concentration of the unsteady-state unit.

$$y_{k,n}^{out} = f_n (q_{k,n-1}, t, H_{k,n}, y_{k,n}^{in}, d_k, o_{k,n}, \Delta t_n) \tag{3.17}$$

where d_k is the vector of design variables for the k^{th} interceptor, $o_{k,n}$ is the vector of operating variables of the kth interceptor during the n^{th} time interval, and Δt_n is the duration of the n^{th} interval. Clearly, for the steady-state interceptors, there is no need to incorporate dependence on time intervals or duration.

The unsteady-state interceptors must be frequently taken offline and regenerated. A binary integer variable *Regen*$_{k,n}$ defined as follows:

Regen$_{k,n}$ = 1 if interceptor *k* is being regenerated during the n^{th} time interval
= 0 otherwise (3.18)

Therefore,

$$y^{out}_{k,n} = (1 - Regen_{k,n}) \cdot f_n(q_{k,n-1}, H_{k,n}, y^{in}_{k,n}, d_k, o_{k,n}, \Delta t_n) + Regen_{k,n} \cdot y^{in}_{k,n} \tag{3.19}$$

and the flowrate entering the k^{th} unit during the n^{th} time period is given by a revised form of equation 3.2:

$$H_{k,n} = (1 - Regen_{k,n}) \cdot (\sum_i g_{i,k,n} + \sum_v r_{v,k,n}) \tag{3.20}$$

Additionally, equation 3.16 can now be extended to include the effect of regeneration as follows:

$$q_{k,n} = (1 - Regen_{k,n}) \cdot (q_{k,n-1} + \Delta q_{k,n}) + Regen_{k,n} \cdot F_{k,n}(q_{k,n-1}, d_k, o_{k,n}, \Delta t_n) \tag{3.21}$$

where $F_{k,n}$ is the functional form of the regeneration period which provides the component load in unit *k* resulting after regeneration for a duration Δt_n.

The total annualized cost per episode (or per operating cycle) "*TAC*" of the interception network is given by:

$$TAC = \sum_n \sum_k [(1 - Regen_{k,n}) \cdot Cost\_Interception_{k,n} \cdot Regen_{k,n} \cdot Cost\_Regen_{k,n}] \tag{3.22}$$

The form and arguments of the cost equation depend on the characteristics of the interceptors and their regeneration.

The foregoing formulation is a mixed-integer nonlinear program (MINLP) whose solution provides a synthesized interception and allocation network as well as the schedule to operate the system.

Several assumptions were implicitly made in formulating this problem. One is that the flowrates and species concentration in the process source and sink streams are constant with time. This is normally the case, but it can be generalized to unsteady-state streams without much additional effort. Also, the component that is being removed from the rich streams is assumed to be in small concentrations, so the mass flowrates across the mass exchange units are taken to be constant. The other main assumptions are that steady-state MSA streams are not mixed in a unit and that MSA streams are not reused in one unit after being used in another unit. Based on the latter assumption, there is one type of MSA specified for each steady-state unit. If this specification were not made, two units could use the same MSA, but it would be cheaper to simply have a large unit that can accommodate the flow from both units. Thus, in this formulation, each steady-state unit utilizes a unique MSA stream.

3.3 Case Study

To illustrate these concepts, let us consider an environmental application with one process source (a wastewater effluent) and one sink in which the objective is to reduce the concentration of a pollutant in a stream to be discharged into the environment. The effluent flowrate is 10 $^{kg}/_{min}$ and has a mass fraction of 0.0050. The sink requires a flowrate of 10 $^{kg}/_{min}$ with a mass fraction of no more than 0.0002. One steady-state absorber and two unsteady-state fixed-bed adsorbers were used in the MEN. There was no waste sink and no fresh supply stream was available. Figure 3.4 shows the stream representation that was used for the example problem. Unit 1 is the steady-state absorber and units 2 & 3 are the unsteady-state adsorbers.

An episode of 4 hours and 10 minutes was used and this was divided into 25 intervals, giving an interval size of 10 minutes. So, changes to the process were allowed every 10 minutes and the performance of the process cycled every 250 minutes. The states of the process at the end of the episode were specified to be equal to the states of the process at the beginning of the episode.

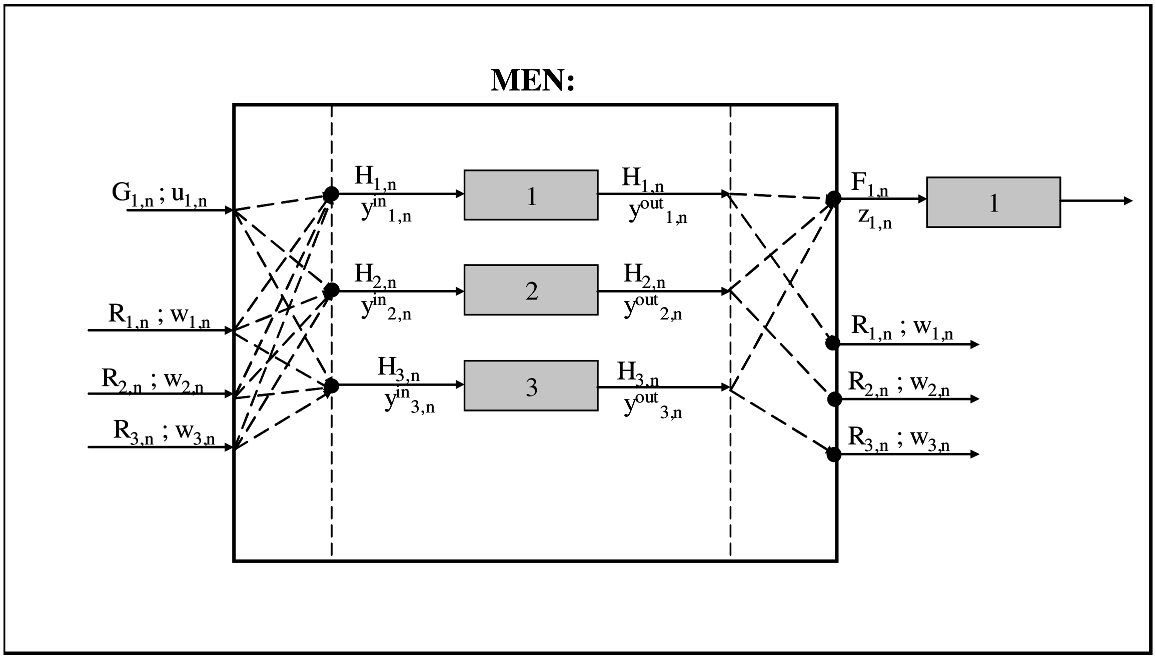

Figure 3.4. Case Study Structural Representation

A very simple model was used for the adsorbers. The overall performance of the units is represented in Figure 3.5, but as will be explained later, only the initial performance range of the units was considered. Thus, the units were restricted to operate no more than 250 minutes consecutively.

The outlet concentration was related to time rather than the unit load in order to reduce the amount of data needed for the problem. The other option was to link the outlet concentration to the load through some modification of the Langmuir equation but this would have required Langmuir constants, which can be difficult to obtain. Using time as the variable is an approximation, but a reasonable one and one that simplifies the problem. Regeneration of the bed completely cleaned the unit and thus reset the unit's

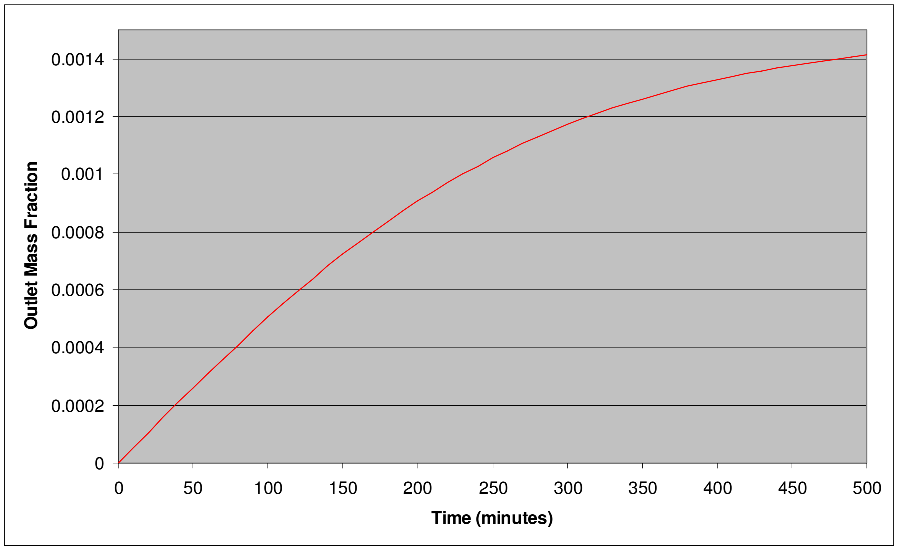

Figure 3.5. Adsorber Performance Curve

internal "clock" to zero. If the binary variable specified that the unit ran during a time interval, the unit's clock increased by the size of the time interval to reflect the bed becoming saturated:

$$t_{k,n} = (1 - Regen) \cdot (t_{k,n-1} + \Delta t) \tag{3.23}$$

where $t_{k,n}$ is the "unit time" of unit k at time interval n and Δt is the length of the time intervals.

It was observed in the previous section that the problem formulation results in a MINLP, which does not give a globally optimum solution. To resolve this, a procedure was adopted that was similar to that put forth in Quesada & Grossmann (1995). A mixed-integer linear program (MILP) was obtained from the original MINLP and this was solved to give a lower bound and also an initial solution point for the MINLP. The MINLP was then solved and the optimal solution was compared to that given by the MILP. Several steps had to be taken to linearize the original program.

First, the binary regeneration variables were relaxed and were permitted to take continuous values, based on the technique described in Szitkai et al. (2006). However, additional binary variables were added and compared with the regeneration variables in the following manner:

$$Regen_{k,n} = Integ_{k,n} + pos_{k,n} - neg_{k,n} \quad \forall n, \; k = 2,3 \tag{3.24}$$

where *Regen* is the now continuous regeneration variable, *Integ* is the added binary variable, and *pos* and *neg* are positive continuous variables that give the difference between *Regen* and *Integ*. A penalty term was added to the objective function to penalize any deviation of *Regen* from a value of either 0 or 1. The penalty term was calculated in the following way:

$$RelaxationDeviation = \sum_{n} pos_{k,n} + neg_{k,n} \quad k = 2,3 \tag{3.25}$$

Since *pos* ≥ 0 and *neg* ≥ 0, then any difference of the continuous *Regen* variable from the binary *Integ* will result in a larger objective value. The result of this relaxation is that the optimization procedure can search over solutions that would otherwise be infeasible and more easily escape an inferior local optimal solution. But the inclusion of the penalty term in the objective function causes the relaxed variable to take binary values for an optimum solution. This relaxation technique also simplifies the remainder of the linearization procedure. In the problem formulation there were constraints in which the regeneration variable was multiplied by another optimizable variable, so with the binary relaxation those become bilinear terms with two continuous variables instead of one continuous and one binary variable.

The bilinear terms were linearized using the linearization method outlined by McCormick (1976) and later by Quesada and Grossmann (1995). With two continuous variables, *x* and *F*, possessing upper and lower bounds

$$x^{Lo} \le x \le x^{U} \tag{3.26}$$

$$F^{Lo} \le F \le F^{U} \tag{3.27}$$

The bilinear product of the two variables,

$$f = x \cdot F \tag{3.28}$$

may be approximated by the four linear inequalities (Al-Khayyal and Falk, 1983 and McCormick, 1976):

$$f \geq x \cdot F^{Lo} + x^{Lo} \cdot F - x^{Lo} \cdot F^{Lo} \quad (3.29)$$

$$f \geq x \cdot F^{U} + x^{U} \cdot F - x^{U} \cdot F^{U} \quad (3.20)$$

$$f \leq x \cdot F^{U} + x^{Lo} \cdot F - x^{Lo} \cdot F^{U} \quad (3.31)$$

$$f \leq x \cdot F^{Lo} + x^{U} \cdot F - x^{U} \cdot F^{Lo} \quad (3.32)$$

So, in the original MINLP, component flowrate equations 3.28 were added. These can be seen in equations 3.3 and 3.6. Also, similar equations were added for the bilinear terms involving the now continuous regeneration variables (equation 3.23). When linearizing the MINLP, those bilinear constraints were removed and replaced with the linear constraints 3.29-3.32. The bounds chosen for the variables will affect the tightness of the approximation. Knowledge of the process and the likely optimal solution must be used in selecting the bounds. The variable bounds used for the case study are given in Table 3.1.

Table 3.1 – Variable Bounds Used in MILP

Variable	**Lower Bound**	**Upper Bound**
H	0	10
y^{in}	0.0002	0.005
y^{out}	0	0.5
R	0	10
w	0	0.5
t	0	240
Regen	0	1

As mentioned before, only the initial section of the adsorber unit performance curve was used. This was because it could be approximated by a linear function and to obtain a linear optimization program, a linear unit performance equation must be used. The equation used was:

$$y_{k,n}^{out} = 4\times10^{-6}\cdot t_{k,n} \quad \forall n\,,\; k = 2,3 \tag{3.33}$$

where y^{out} is the outlet component mass fraction and t is the "unit time."

The cost functions used in this case study consisted of both operating costs and capital costs. The operating cost was calculated by adding the cost of the steady-state MSA used in the absorber and the cost of regenerating the unsteady-state adsorbers. The steady-state MSA cost $1.00 per kg MSA used and regeneration of the fixed beds cost $1.00 for each regeneration. The annualized fixed capital cost for all three columns was calculated based on the diameter of the columns according to the following equation:

$$\text{Column Cost} = f\cdot D^{col} \tag{3.34}$$

where f is a cost factor for the type of column divided by the ratio of the optimization time horizon to the life expectancy of the equipment (equal to 1 in this case study). The column diameter was calculated based on the maximum flowrate passing through the column. The constraint that was used was:

$$D_k^{col} \geq s\cdot G_{k,n} \;\; \forall k,n \tag{3.35}$$

where s is a sizing coefficient, equal to 3 in this case.

The objective function used in the case study was:

$$Cost_{MSA} + Regen_1Cost + Regen_2Cost + D_1^{col} + D_2^{col} + D_3^{col} + 10^{11}\cdot RelaxationDeviation \tag{3.36}$$

where $Cost_{MSA}$ is the cost of the total flowrate of the MSA stream in the steady state unit, $Regen_1Cost$ is the regeneration cost for the first unsteady-state adsorber, $Regen_2Cost$ is the regeneration cost for the second unsteady-state adsorber, D_1^{col}, D_2^{col}, and D_3^{col} are the diameters of the three units, and *RelaxationDeviation* is the penalty term for the integer variable relaxation.

The MILP was run on Hyper LINGO version 8.0, from LINDO Systems Inc. using branch-and-bound. The solution required less than one second using a desktop computer with a 2.52 GHz Pentium 4 processor and 512 MB of RAM. The MILP gave a weekly cost of $32.33. The MINLP was also solved using branch-and-bound on Hyper

LINGO. Its solution required three seconds and gave a weekly cost of $32.33. Because this is identical to the solution given by the MILP, we know that this is the global optimum solution for this problem. The MINLP solution will rarely equal the MILP solution, so the confidence in the globality of the solution must come from the closeness of the solution to that given by the MILP. As stated previously, the MILP solution will be influenced by the bounds chosen for the variables. Thus, it is important to have a feel for the process and be able to anticipate certain aspects of the optimal solution.

3.4 Conclusions

A novel representation and solution method was given for the problem of optimizing the network structure, operation, schedule, and unit sizes for a mass exchange network. No other solution approach formulated has optimized this type of process in such a general way. The problem formulation creates a mixed-integer nonlinear program (MINLP) whose nonlinear constraints may be approximated to give a mixed-integer linear program (MILP). Although introducing a global optimization algorithm was not the purpose of this research, a method that could locate the global solution was illustrated. The variable bounds should be chosen wisely for the MILP since they have a significant impact on the solution obtained. A case study illustrating an application of the problem formulation and some techniques of global optimization were demonstrated.

CHAPTER IV
OPTIMAL SYNTHESIS AND SCHEDULING OF HYBRID DYNAMIC/STEADY-STATE PROPERTY INTEGRATION NETWORKS*

4.1 Introduction

The process industries are characterized by the use of enormous quantities of mass and energy. Therefore, the economic performance of the processing facilities is typically tied to their effective usage and transformation of mass and energy. As such, process synthesis, integration, and scheduling activities have traditionally been based on tracking and optimizing the allocation of mass and/or energy. For instance, mass-integration techniques have provided systematic approaches that can identify rigorous targets and implementation strategies for allocation, transformation, and separation of chemical species (e.g. El-Halwagi, 1997, Dunn and El-Halwagi, 2003). A common feature of mass-integration techniques is that they are “chemo-centric”; they are based on tracking individual chemical species. It is important to note that there are many industrial applications that are not directly governed by the chemical nature of the streams. For instance, the selection of solvents is typically based on properties such as equilibrium distribution coefficients, viscosity and volatility. It is also worth noting that the quality of many products and intermediates is described in terms of properties. Hence, in recycling/reusing process and waste streams the properties of those streams must satisfy the requirements of the processing units that can accept them. Additionally, a large number of environmental problems are associated with properties of the discharges. For instance, the extent of environmental emissions is typically linked to properties of the pollutants (e.g. volatility, solubility, etc.). Furthermore, the environmental regulations involve limits on properties (e.g., pH, color, toxicity, TOC, BOD, ozone-depleting ability). Similarly, ecological consequences of the discharged wastes are dependent on

* Reprinted from *Computers & Chemical Engineering*, **29**, Grooms et al. “Optimal synthesis and scheduling of hybrid dynamic/steady-state property integration networks,” 2318-2325, Copyright (2005) with permission from Elsevier.

the properties of the pollutants.

In response to this need, the new paradigm of *property integration* has been recently introduced. Property integration is “a functionality-based, holistic approach to the allocation and manipulation of streams and processing units, which is based on the tracking, adjustment, assignment, and matching of functionalities throughout the process” (El-Halwagi et al., 2004). Shelley and El-Halwagi (2000) developed the concept of property-based clusters as tailored quantities that satisfy intra- and inter-stream conservation and used clusters to synthesize property-based separation for mixtures with numerous components. Graphical as well as algebraic techniques have been developed to foster material reuse and process modification using property-based approaches. Single-property pinch analysis techniques have been developed to optimize the allocation of process streams to units based on property constraints (Kazantzi and El-Halwagi, 2005). For systems with up to three properties of concern, graphical tools may be used to guide the synthesis and analysis tasks (Shelley and El-Halwagi, 2000; El-Halwagi et al., 2004). When there are more than three properties, algebraic tools can be employed (Qin et al., 2004). Process modification based on properties can be addressed using property-based models incorporated in a process synthesis framework (Kazantzi et al., 2004 a,b). Moreover, the clustering concept has also been extended to address simultaneous process and molecular design problems (Eden et al., 2002, 2004).

While the foregoing research activities on property integration have provided many useful techniques and insights, they have a common limitation: they are based on synthesizing a steady-state system. As discussed previously, process units are, in many cases, inherently unsteady-state or are best operated in a dynamic mode. As such, it is necessary to incorporate these dynamic units in the analysis and develop a systematic procedure for the synthesis and scheduling of the system. This work introduces the problem of developing a systematic procedure to optimize a process containing property-modifying units. The units may be steady-state or dynamic units whose performance is expressed in terms of input–output relations for designated properties. A source–interception–sink representation is developed to embed structural configurations of interest. This representation is used to optimize the network structure, unit capacities, and operation schedule.

4.2 Problem Description

Given is a process with:

- A set of sources: SOURCES = $\{i \mid i = 1,2, \ldots, N_{sources}\}$ composed of process streams, which are available to be allocated. Each source has a given flowrate, G_i, and is characterized by a set of properties: PROPERTIES = $\{p \mid p = 1,2, \ldots, N_p\}$. The values of the properties of the sources are also given and designated by $p_{i,p}^{in}$.

- A set of process sinks (units): SINKS = $\{j \mid j = 1,2, \ldots, N_{sinks}\}$. Sinks are process units that can accept the sources. Each sink requires a given flowrate, F_j, and property values, $p_{j,p}^{in}$, that satisfy the following constraints:

$$p_{j,p}^{\min} \le p_{j,p}^{in} \le p_{j,p}^{\max} \quad j \in \text{SINKS, p } \in \text{PROPERTIES} \tag{4.1}$$

where $p_{j,p}^{\min}$ and $p_{j,p}^{\max}$ are given lower and upper bounds on acceptable properties to unit j.

- A set of interception units: INTERCEPTORS = $\{k \mid k = 1,2, \ldots, N_{Int}\}$. These units can be added to the process to adjust the properties of the sources to meet the sink requirements. Some of the interceptors are operated dynamically with the following performance equations:

$$p_{i,p}^{\text{int}} = f(p_{i,p}^{in}, d_k, o_k, t) \tag{4.2}$$

where $p_{i,p}^{\text{int}}$ is the value of the intercepted property leaving the kth interceptor, d_k and o_k are design and operating variables of the kth interceptor, and t is time of operation. Clearly, if the unit is operated in a steady-state mode, its intercepted properties are independent of t.

The problem statement is schematically represented in Figure 4.1.

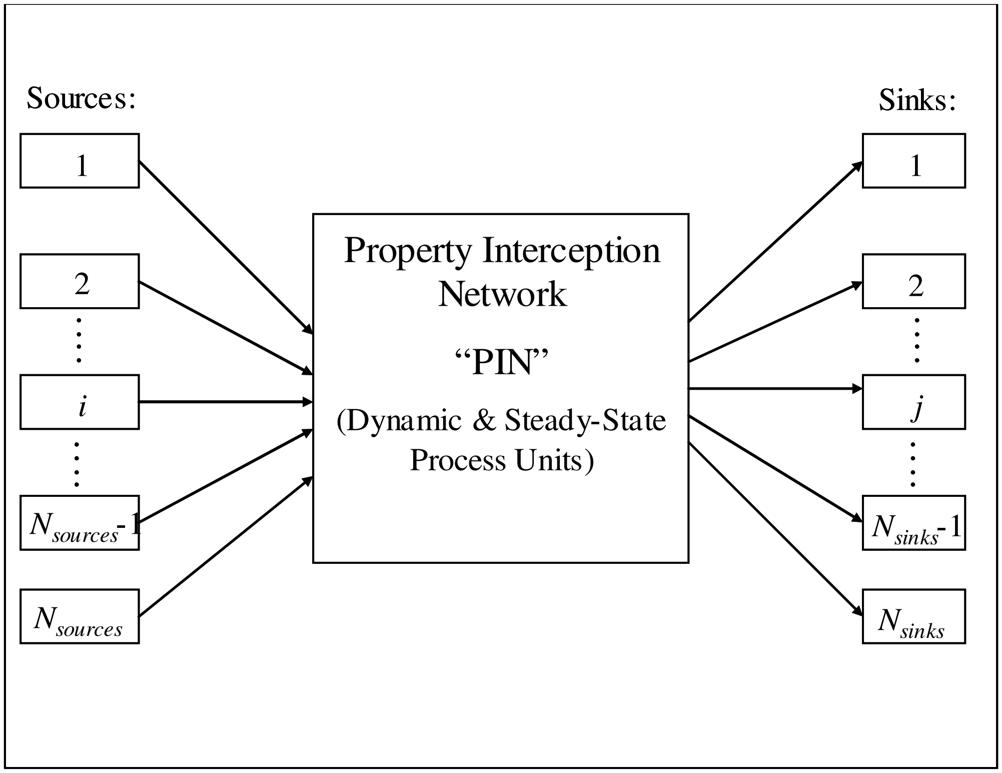

Figure 4.1. Schematic Representation of Problem

The objective is to synthesize and schedule a network of sources-interceptors-sinks that provides optimal allocation and interception of sources while satisfying the constraints of the sinks at minimum cost. In particular, the following are design challenges associated with problem:

- Which interception units should be selected? To do what?
- How should the sources be allocated to the interception devices? Should sources be segregated, mixed, or split prior to interception? What should be the extent of interception of each source?
- How should each intercepted source be allocated? Should it be mixed with other sources?
- How should the interception and allocation schemes be scheduled so as to cope with the time-varying performance of the interceptors while satisfying the steady-state requirements of the sinks?

In order to address these challenging tasks, we start by developing a generic representation for the problem and derive a mathematical optimization formulation.

4.3 Solution Approach

4.3.1 Revised Problem Representation and Property Mixing Rules

Using a source-interception-sink representation as shown in Figure 4.2, each source is first segregated and split into fractions of unknown flow rate (to be optimized). Those fractions are allocated to the interception network. A combination of steady- and unsteady-state property-intercepting units are available for use and are numbered k=1,2,…,N_{Int} inside the property interception network (PIN). Within the interception network, the properties of each stream are altered if an interception device is used. The optimum extent of interception for each property and stream is to be determined. A stream may also pass through the interception network unchanged indicating that the stream is not intercepted. The streams leaving the interception network are allowed to mix and fed to the process sinks. The optimum mixing ratios are to be determined. An

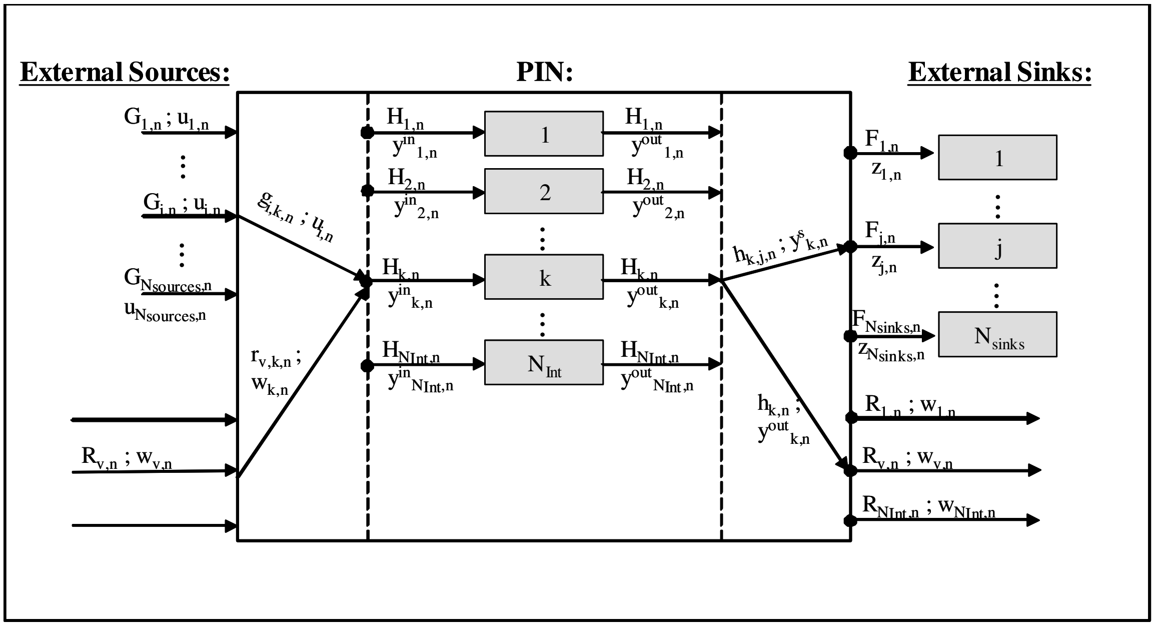

Figure 4.2. Source-Interception-Sink Representation

additional sink is used to account for the waste material (unrecycled/unreused material), and it is designated as the "waste" sink. Also, the external fresh can be used in process sinks but not the waste sink.

In order to develop a scheduling scheme, we start first by describing the process's operating time through a number of repeating and identical episodes. The duration of the episode corresponds to the cycle time. Because of the repeating nature of the episodes, it is sufficient to optimize the process over an episode and the process will then be optimized into infinity. To insure continuity of the episodes, constraints are added such that the state of the process at the end of the episode is identical to the state at the beginning of the episode. This episode time is discretized into a number of time intervals over which the process is optimized. Structural changes (e.g., starting or ending the operation of a unit, starting or ending the regeneration of a unit) are performed at the beginning (or end) of a time interval. The index n is used to denote a time interval.

In order to track properties throughout the system, let us first consider property-mixing rules. To simplify the mathematical representation, we focus on a single key property (*p*). Multiple properties can be easily incorporated by adding an extra index for properties. Suppose that several sources are mixed. Each source (i) has a property value (p_i). Consider the following mixing rule for estimating the resulting property of the mixture:

$$\overline{G} \cdot \psi(\bar{p}) = \sum_i G_i \cdot \psi(p_i) \tag{4.3}$$

where $\psi(p_i)$ is the property-mixing operator and $\overline{G}$ is the total flow rate of the mixture which is given by:

$$\overline{G} = \sum_i G_i \tag{4.4}$$

The property-mixing operators can be evaluated from first principles or estimated through empirical or semi-empirical methods.

Each source is split into fractions that are passed to the interception devices. The flowrate from source *i* to interceptor *k* is called $g_{i,k,n}$, while the flowrate from interceptor *k* to sink *j* is labeled $h_{k,j,n}$. As mentioned earlier, the flowrate through an interceptor *k* is denoted $H_{k,n}$ while the property operator value entering and exiting the interceptor is designated by $y^{in}{}_{k,n}$ and $y^{out}{}_{k,n}$, respectively. It is also possible to recycle streams leaving interception devices and, in turn, they become new sources. The index for these recycled streams is called *v*. The flowrate and property operator values of a recycle stream are represented as $R_{v,n}$ and $w_{v,n}$.

We can now rewrite the sink constraints given by equation 4.1 in terms of the property-mixing operator as:

$$z_j^{\min} \le z_j^{in} \le z_j^{\max} \qquad (4.5)$$

Now that the terms have been defined, let us revisit the structural representation shown in Figure 4.2. A number of process and fresh source streams which have certain properties that need to be modified are numbered 1 through $N_{sources}$ and are given with flowrates and property operator values ($G_{i,n}$ and $u_{i,n}$). Also, a number (N_{sinks}) of process sinks are known along with their flowrate and property operator requirements ($F_{j,n}$ & $z_{j,n}$). A mixture of steady- and unsteady-state property-intercepting units that are of interest are selected and numbered 1,2,…,N_{Int} inside the property interception network (PIN). The flow from source *i* to unit *k* is represented as $g_{i,k,n}$ while the flow from unit *k* to sink *j* is labeled $h_{k,j,n}$. The flow rate through a unit denoted $H_{k,n}$ while the property operator value entering and exiting the unit is $y^{in}{}_{k,n}$ and $y^{out}{}_{k,n}$ respectively. Finally, flowrates and property operator values of possible recycle streams for a property intercepting unit v are represented as $R_{v,n}$ and $w_{v,n}$. These recycle streams serve both as sinks and as sources. The indices *i, j,* and *k* refer to sources, sinks, and property intercepting units respectively, while the index n specifies the time interval in question. The indices *k* and *v* are similar, both referring to property intercepting units; the only difference is that they are used in different contexts – *k* for describing the units and *v* for describing the recycle streams coming from those units.

4.3.2 Solution Formulation

Performing a material balance on the splitting of the source streams yields the equation

$$G_{i,n} = \sum_k g_{i,k,n} \quad \forall i \in SOURCES,\ n \in TIME \tag{4.6}$$

where *TIME* is the set of time intervals. A total balance and a property operator balance at the mixing of the source-to-interceptor nodes yields:

$$H_{k,n} = \sum_i g_{i,k,n} + \sum_v r_{v,k,n} \quad \forall k \in INTERCEPTORS,\ n \in TIME \tag{4.7}$$

$$y_{k,n}^{in} \cdot H_{k,n} = \sum_i u_{i,n} \cdot g_{i,k,n} + \sum_v w_{v,n} \cdot r_{v,k,n} \quad \forall k \in INTERCEPTORS,\ n \in TIME \tag{4.8}$$

A balance on the splitting of the interceptor outlet streams gives the equation

$$H_{k,n} = \sum_j h_{k,j,n} + h_{k,n} \quad \forall k \in INTERCEPTORS,\ n \in TIME \tag{4.9}$$

where $h_{k,n}$ refers to the flow from interceptor *k* to its recycle stream.

Balances on the total flow and the property operators at the mixing of the streams to the process sinks give the following equations:

$$F_{j,n} = \sum_k h_{k,j,n} \quad \forall j \in SINKS,\ n \in TIME \tag{4.10}$$

$$z_{j,n} \cdot F_{j,n} = \sum_k y_{k,n}^{out} \cdot h_{k,j,n} \quad \forall j \in SINKS,\ n \in TIME \tag{4.11}$$

Finally, each interceptor has its own recycle stream. The set of recycle streams from interceptors is referred to by *U* and the index *v* is used to enumerate its members. The flowrate and property operators of the recycle streams are specified by the equations

$$R_{v,n} = h_{k,n} \quad \forall v = k \in U,\ n \in TIME \tag{4.12}$$

$$w_{v,n} = y_{k,n}^{out} \quad \forall v = k \in U,\ n \in TIME \tag{4.13}$$

For each steady-state interceptor, a supply operator value, x_k^s, and target operator value, x_k^t, is given and L_k represents the optimizable flowrate of the agent through unit *k*.

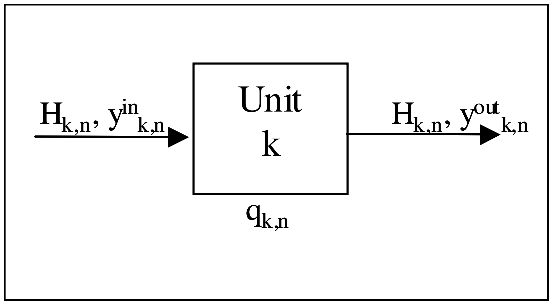

Figure 4.3. Property-Operator Balance for the k^th^ Unsteady-state Interceptor During the n^th^ Time Interval

Figure 4.3 is a diagram of an unsteady-state unit where $q_{k,n}$ represents the amount of accumulated property operator contained in the unit over time interval *n*. This load can be calculated by the following equation:

$$\int_{t_n} H_k (y_k^{in} - y_k^{out}) dt = \int_{t_n} q_k dt \tag{4.14}$$

where H_k is the instantaneous flowrate of the rich stream through unit *k*, $y^{in}{}_k$ and $y^{out}{}_k$ are the instantaneous property operators of the rich stream, and q_k is the instantaneous load of unit *k*. To simplify the formulation, we assume that the interceptor operates in a steady-state mode with its inlet and outlet property operators as well as flowrate taken to be constant over the time interval and designated by $y_{k,n}^{in}$, $y_{k,n}^{out}$, and $H_{k,n}$. These values change from one interval to another leading to a step-wise approximation of the nonlinear changes in flowrate, inlet and outlet property operators. Therefore, equation 4.14 becomes

$$H_{k,n} (y_{k,n}^{in} - y_{k,n}^{out}) \int_{t_n} dt = (q_{k,n+1} - q_{k,n}) \int_{t_n} dt \tag{4.15}$$

where $q_{k,n+1}$ is the load of unit k at the beginning of interval *n*+1. This simplifies to:

$$H_{k,n} (y_{k,n}^{in} - y_{k,n}^{out}) = \Delta q_{k,n} \tag{4.16}$$

where $\Delta q_{k,n}$ is the net load accumulation for unit *k* in time interval *n*.

The load of the interceptor may be tracked using the equation

$$q_{k,n+1} = q_{k,n} + \Delta q_{k,n} \qquad n \in TIME \tag{4.17}$$

where $q_{k,0} = 0$ when starting the time period with a clean unit.

Another equation that will be needed is an equation for the outlet property operator of the unsteady-state unit.

$$y_{k,n}^{out} = f_n(q_{k,n-1}, t, H_{k,n}, y_{k,n}^{in}, d_k, o_{k,n}, t_n) \tag{4.18}$$

where d_k is the vector of design variables for the k^{th} interceptor, $o_{k,\mathrm{n}}$ is the vector of operating variables of the k^{th} interceptor during the n^{th} time interval, and t_n is the duration of the n^{th} interval. Clearly, for the steady-state interceptors, there is no need to incorporate dependence on time intervals or duration.

The unsteady-state interceptors must be frequently taken offline and regenerated. A binary integer variable $Regen_{k,n}$ defined as follows:

$Regen_{k,n}$ = 1 if interceptor k is being regenerated during the n^{th} time interval

= 0 otherwise (4.19)

Therefore,

$$y_{k,n}^{out} = (1 - Regen_{k,n}) \cdot f_n(q_{k,n-1}, H_{k,n}, y_{k,n}^{in}, d_k, o_{k,n}, t_n) + Regen_{k,n} \cdot y_{k,n}^{in} \tag{4.20}$$

and the flowrate entering the k^{th} interceptor during the n^{th} time period is given by a revised form of equation 4.7:

$$H_{k,n} = (1 - Regen_{k,n}) \cdot (\sum_i g_{i,k,n} + \sum_v r_{v,k,n}) \tag{4.21}$$

Additionally, equation 4.17 can now be extended to include the effect of regeneration as follows:

$$q_{k,n} = (1 - Regen_{k,n}) \cdot (q_{k,n-1} + \Delta q_{k,n}) + Regen_{k,n} \cdot F_{k,n}(q_{k,n-1}, d_k, o_{k,n}, t_n) \tag{4.22}$$

where $F_{k,n}$ is the functional form of the regeneration period which provides the property load in unit k resulting after regeneration for a duration t_n.

The total annualized cost per episode (or per operating cycle) "*TAC*" of the interception network is given by:

$$TAC = \sum_n \sum_k [(1 - Regen_{k,n}) \cdot Cost\_Interception_{k,n} \cdot Regen_{k,n} \cdot Cost\_Regen_{k,n}] \tag{4.23}$$

The form and arguments of the cost expressions equation depend on the characteristics of the interceptor and its regeneration.

The foregoing formulation is a mixed-integer nonlinear program (MINLP) whose solution provides a synthesized interception and allocation network as well as the schedule to operate the system.

4.4 Case Study

To illustrate these concepts, let us consider an environmental application with one process source (a wastewater effluent) and one sink in which the objective is to reduce the color of a stream to be discharged into the environment. The effluent is discharged from dyeing and rinsing stages and is laden with color from dye. The color is measured by the American Dye Manufacturers Index, ADMI. The ADMI gives a reference of how much color is present compared to a colorless reference, clean water. The color mixing rule is given by Harell (2004) as:

$$\overline{ADMI}^{0.606} = \sum_i x_i ADMI_i^{0.606} \tag{4.24}$$

The effluent flowrate is 10 $^{kg}/_{min}$ and a property (ADMI) operator of 0.0050. The sink requires a flowrate of 10 $^{kg}/_{min}$ with a maximum property (ADMI) operator of 0.0002. One steady-state absorber and two unsteady-state fixed-bed adsorbers were used in the PIN. Experimental data for the performance of the fixed bed in removing color from the effluent as a function of time is given by:

$$y^{out} = 1.5\times10^{-4}\left[\tanh(0.08\times t-6)+1\right] \tag{4.25}$$

where t is measured in minutes and y^{out} is the adsorber outlet property (ADMI) operator. Regeneration (using steam) was found to approach a clean bed (i.e., $q = 0$ at end of regeneration).

An episode of 4 hours and 10 minutes was used and this was divided into 25 intervals, giving an interval size of 10 minutes. So, changes to the process were allowed every 10 minutes and the performance of the process cycled every 250 minutes.

The cost functions used in this case study consisted of both operating costs and capital costs. The operating cost was calculated by adding the cost of the steady-state MSA used in the absorber and the cost of regenerating the unsteady-state adsorbers. The steady-state MSA cost \$0.50 per kg MSA used and regeneration of the fixed beds cost \$0.50 for each regeneration. The annualized fixed capital cost for all three columns was calculated based on the height and diameter of the columns according to the following equation:

$$\text{Column Cost} = f\left(D^x + H^y\right) \tag{4.26}$$

where *f* is a cost factor for the type of column divided by the ratio of the optimization time horizon to the life expectancy of the equipment. D is the diameter; x is the diameter exponent (equal to 1 in this problem); H is the height; and y is the height exponent (equal to 0.9 in this problem). The objective function used in the case study was:

$$\text{MSACost} + \text{Regen}_1\text{Cost} + \text{Regen}_2\text{Cost} + 0.1\left(D_1 + H_1^{0.9}\right) + 0.1\left(D_2 + H_2^{0.9}\right) + 0.1\left(D_3 + H_3^{0.9}\right) \tag{4.27}$$

where MSACost is the cost of the total flowrate of the MSA stream in the steady state unit, $\text{Regen}_1\text{Cost}$ is the regeneration cost for the first unsteady-state adsorber, $\text{Regen}_2\text{Cost}$ is the regeneration cost for the second unsteady-state adsorber, and the final three terms are the capital cost for the three columns based on the diameters and height. The total cost was calculated as \$630,760/year, consisting of \$514,800 for the steady-state MSA stream, \$52,000 total regeneration cost, and \$21,320, \$24,960, and \$17,680 for the adjusted capital costs for the steady-state column, and the first and second unsteady-state columns respectively.

The diameters of the columns in this problem were calculated by dividing the maximum volumetric flowrate of the stream flowing through the column by an optimal superficial velocity of 1 $^{m}/_{min}$ to obtain an optimal cross-sectional area. The height of the steady-state absorber column was calculated using a Kremser-like equation based on the color operator, while the heights of the unsteady-state fixed-bed columns were calculated by multiplying a sizing factor by the maximum flow through the column.

The mixed-integer nonlinear program (MINLP) was run on Hyper LINGO version 8.0, from LINDO Systems Inc using branch-and-bound. The solution required five minutes using a desktop computer with a 2.52 GHz Pentium 4 processor and 512 MB of RAM.

Figures 4.4a, b, & c show the process conditions at various times as given by the optimization results. The units are numbered 1, 2, and 3 where unit 1 is the steady-state absorber and units 2 and 3 are the fixed-bed columns. These diagrams illustrate how the information obtained from this optimization solution may be used in process operation.

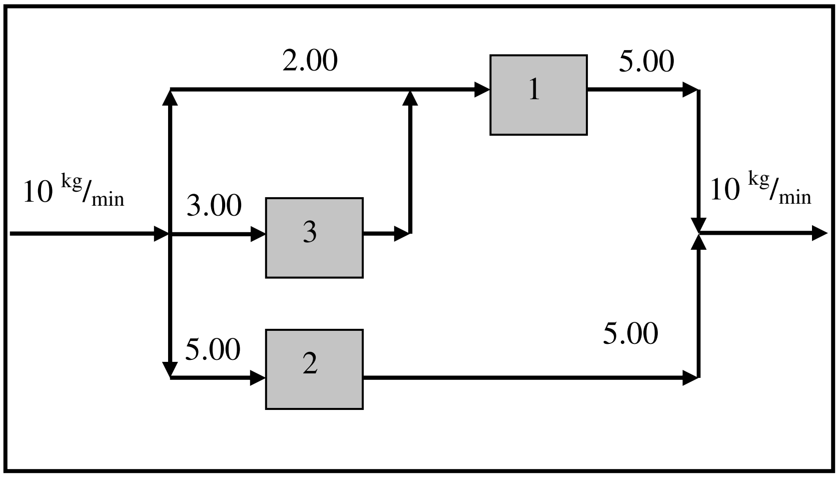

Figure 4.4a. Solution Structure for Sample Interval 7

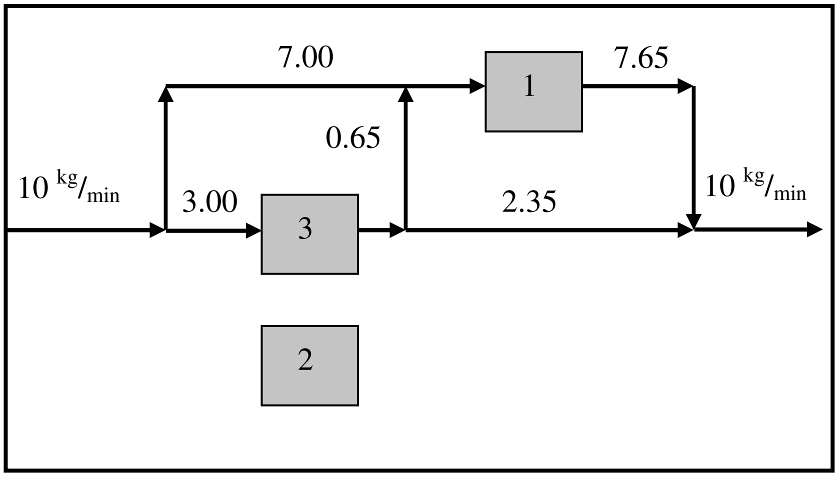

Figure 4.4b. Solution Structure for Sample Interval 8

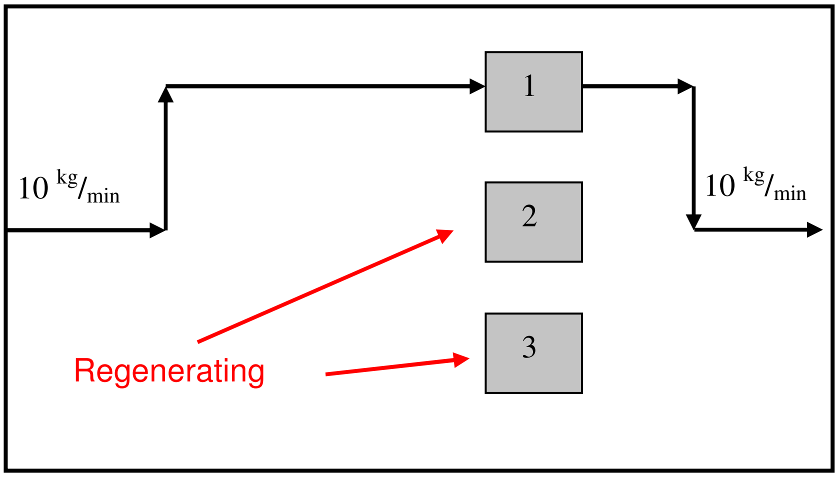

Figure 4.4c. Solution Structure for Sample Interval 25

4.5 Conclusions

This work demonstrates the utility of property integration tools in synthesizing and scheduling a mixed dynamic/steady-state network of sources, sinks and interceptors based on key properties for a process. A new procedure has been introduced for the synthesis and scheduling of property-interception and allocation networks. A basic feature of this problem was the incorporation of dynamic interception units whose performance was expressed in terms of input and output properties. The work has introduced a general framework and a mathematical program to address the synthesis and optimization of interception and allocation networks and the operational scheduling of the system based on properties. A case study was presented to demonstrate the applicability of the proposed formulation of the problem and the steps followed to address this problem in a real dyeing process with the color index (ADMI) as the key property. The case study illustrated that not only does this optimization formulation yield the optimum unit sizes and network configuration, but also the regeneration schedule for the fixed beds and the optimum distribution of flows to each unit to minimize both the operating and capital cost.

CHAPTER V
OPTIMAL DESIGN AND SCHEDULING OF SEMI-BATCH ADSORPTIVE/REGENERATIVE SYSTEMS FOR INDUSTRIAL WATER PURIFICATION

5.1 Introduction

Water purification has many uses and is accomplished by many different means. Water is the most widely used chemical in industrial processes, used in cooling, dilution, separation, and lubrication. Along with its use comes the need to remove contaminants, whether in the supply side (water for cooling towers and boilers) or on the discharge side (discharge wastewater treatment). Methods used to purify vary depending on the extent of purity required, on the amount of contaminants removed, and on the amount of water handled. Common technologies include carbon filtration, ion exchange, reverse osmosis, membrane filtration, and flocculation.

However these technologies are used, they all become saturated or used up and require some kind of regeneration or replacement. The regeneration operation should be conducted so that the usage of the purification technology is maximized while still meeting the requirements of the process. However, although much emphasis is placed on ensuring the purification operation functions optimally, little attention is paid to the regeneration operation. Optimizing this area may also result in better purification performance and reduced cost.

There has been much research into minimizing process water waste in chemical operations (Gabriel & El-Halwagi, 2005; Wang & Smith, 1994), but little work has been done on designing and configuring a complete system to treat that wastewater. Several papers exist on determining good process conditions (pH, chemical additives, etc.) in a water treatment system that has been otherwise already determined (Tlili et al., 2003; Clever et al., 2000) and work has been done on formulating detailed kinetic models of the reactions that occur in waste treatment (Li et al., 2002). However, nothing has been published to demonstrate a general, overall process optimization methodology to systematically design the best water treatment system (with its associated regeneration process).

This work will describe a generic water purification operation and the difficulties involved in optimizing its design. A procedure to target the minimum liquid discharge, design the optimum system and operation schedule, and trade off additional discharge amounts with lower capital costs will be given. An industrial case study will be solved to illustrate the procedure. Although the case study will use a specific purification technology, the method used to optimize the system may be applied to any other technology.

5.2 Problem Description

Consider an industrial wastewater stream whose flowrate is F_{water} and contaminant concentration is c_{in}. It is desired to reduce the contaminant concentration to c_{out} using an adsorptive water purification system (e.g., ion exchange, adsorption). An adsorptive mass separating agent (MSA) is used to remove the contaminant and is circulated in the water purification unit. The used MSA with a contaminant concentration of x_{used} is cleaned with a regeneration operation and returned to the water-purification unit at a contaminant concentration of x_{clean}. Figure 5.1 is a schematic representation of the system. The time-averaged flowrate of the MSA, L as well as its inlet and outlet concentrations (x_{used} and x_{clean}) are unknown and to be determined through optimization. The MSA regeneration operation consists of multiple steps. First, the MSA is contacted with a regeneration agent, or regenerant. Next, the MSA is washed, or polished, with a different material, referred to as a wash. The nature of the regenerant and wash materials, as well as the specifics of the regenerant and wash flows, will depend on the MSA used in the process. As a result of the regeneration and the washing steps, a liquid waste is formed and discharged. The time-averaged flowrate of the discharged waste, D, is unknown.

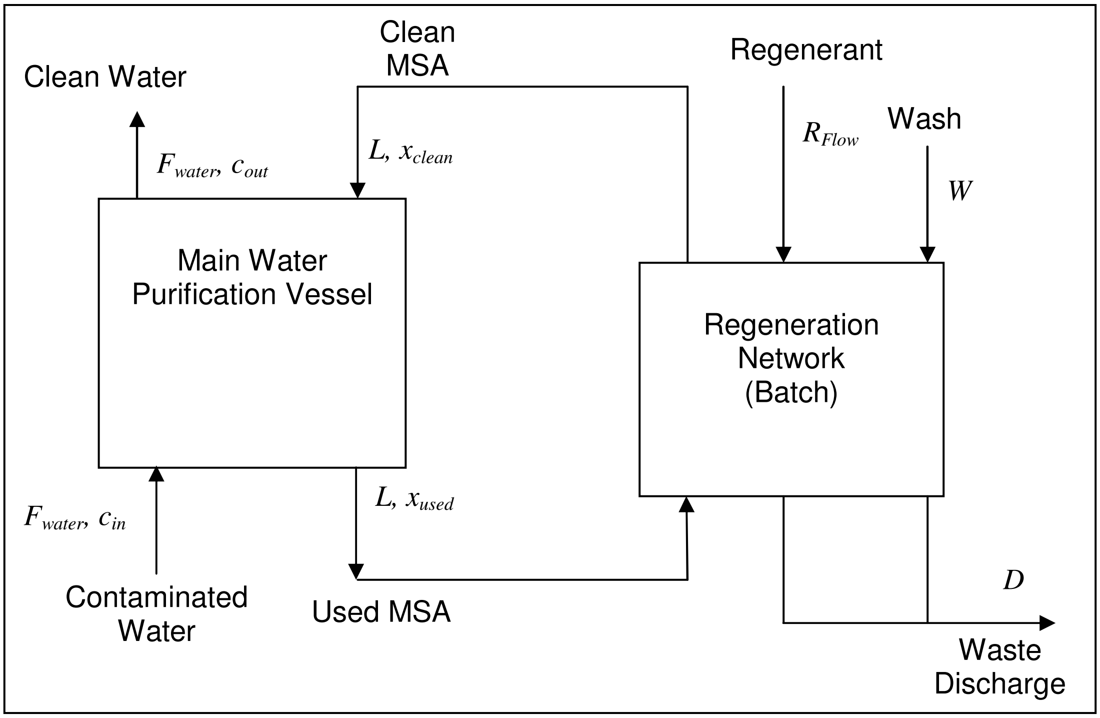

Figure 5.1. Water Treatment System

The main adsorption/ion-exchange unit receives a continuous flowrate of the wastewater. The regeneration and rinsing steps are batch operations and may involve multiple units for each step. The used MSA is continuously withdrawn from the main unit at a rate *L* while the clean MSA is periodically returned to the main unit in batches at an average rate *L* to make up for the MSA that is being taken to regeneration. Batches of clean MSA are periodically returned to the main vessel at a time-averaged rate *L* to make up for the MSA that is being taken to regeneration. Because the regeneration and wash steps are batch operations, the materials used for regeneration and washing are used in batches, but at average flows R_{Flow} and *W*. The discharged waste, *D*, is also a time-averaged calculation of a batch flow. The time-averaged flowrate of the returned MSA is described by the following equation:

$$L = \frac{\int_0^{\tau} l(t)dt}{\tau} \tag{5.1}$$

where *l(t)* is the instantaneous MSA flowrate at a time *t* and τ is the length of a system cycle. A system cycle is defined as the period between when a certain regeneration unit begins to fill with used MSA until the time the same unit begins the next batch. R_{Flow}, *W*, and *D* are averaged in the same way.

Table 5.1 summarizes the operating modes of streams in terms of continuous versus batch operations.

Table 5.1. Operating Modes of the Streams

Stream	**Mode**	**Symbol**	**Basis**
Incoming wastewater	Continuous	F_{water}	Actual Flowrate
MSA entering main purification vessel	Batch	*L*	Time-Averaged
MSA leaving main purification vessel	Continuous	*L*	Actual Flowrate
Regenerant entering all regeneration vessels	Batch	R_{Flow}	Time-Averaged
Wash (Water) entering all rinse vessels	Batch	*W*	Time-Averaged
Waste discharged from the system	Batch	*D*	Time-Averaged

Because MSA regeneration and washing are operated in a batch mode, multiple units may be used. The regeneration and washing network may be configured in one of two ways: single- or multi-purpose units. If the units are multi-purpose, the MSA is regenerated and washed in the same vessel. For single-purpose units, after the MSA is regenerated in one unit, it is transferred to another unit for washing. The configurations of the two alternatives are illustrated in Figures 5.2 and 5.3.

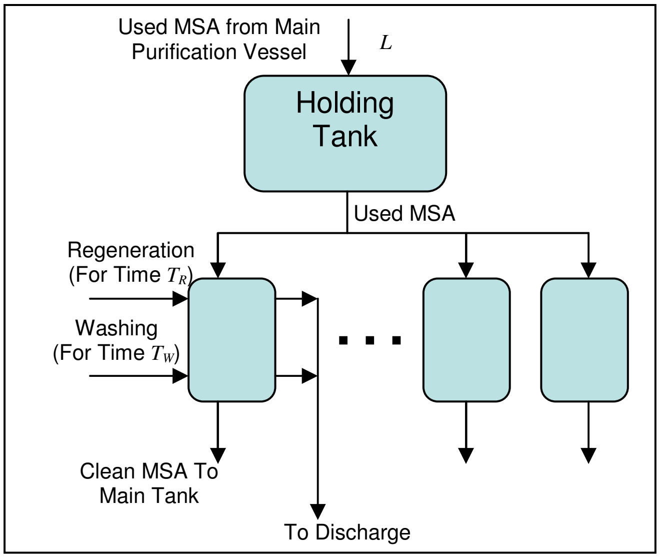

Figure 5.2
Multi-Use Unit Regeneration and Washing Network

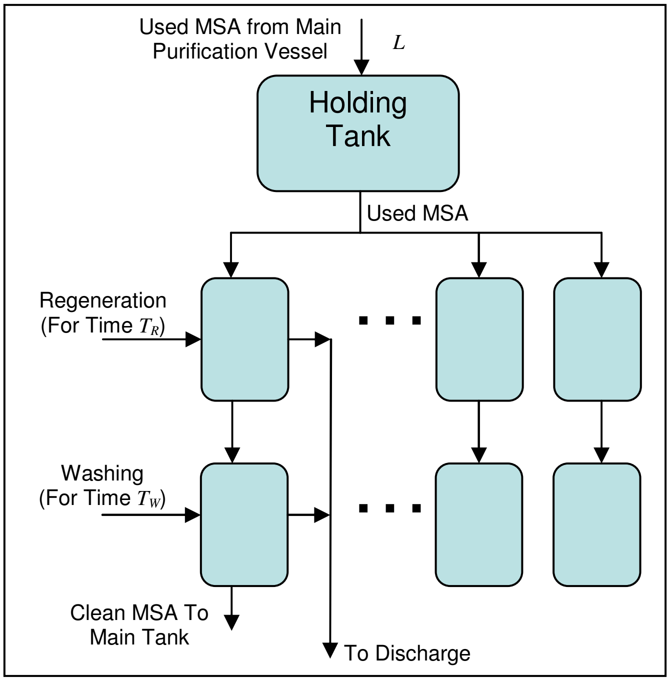

Figure 5.3
Single-Use Unit Regeneration and Washing Network

It is desired to develop a systematic and generally-applicable procedure that can achieve the following for a water purification system similar to that described above:

- Identify the optimal MSA technology
- Determine a target for minimum liquid discharge (MLD) of the waste leaving the system
- Optimize the size and operation of the main purification vessel
- Optimize the number, arrangement, design, and scheduling of the regeneration vessels
- Optimize the number, arrangement, design, and scheduling of the rinse vessels
- Identify the optimal operating schemes and flows of the regenerant and washing solution
- Establish an economic trade-off for discharging more than the MLD

5.3 Solution Approach

The aforementioned problem involves several complicating factors. Before formulating an optimization approach, it is beneficial to decompose the problem into several tractable stages and develop a solution approach for each portion. Later, these solution steps will be integrated to form the solution procedure. We propose the following three-pronged approach (illustrated by Figure 5.4):

1. **Targeting for MLD:** In this stage, a benchmark is established for the minimum target of waste discharge. This target is to be identified ahead of detailed design and without commitment to the final system configuration, operation, or schedule. The target should be determined based on incoming water data, desired purification objectives, as well as data and constraints for the MSA and the regenerating agent. The outcome of this benchmarking stage is to determine the minimum flow of the waste along with the associated flows of the MSA, the regenerant, and the rinse agent.

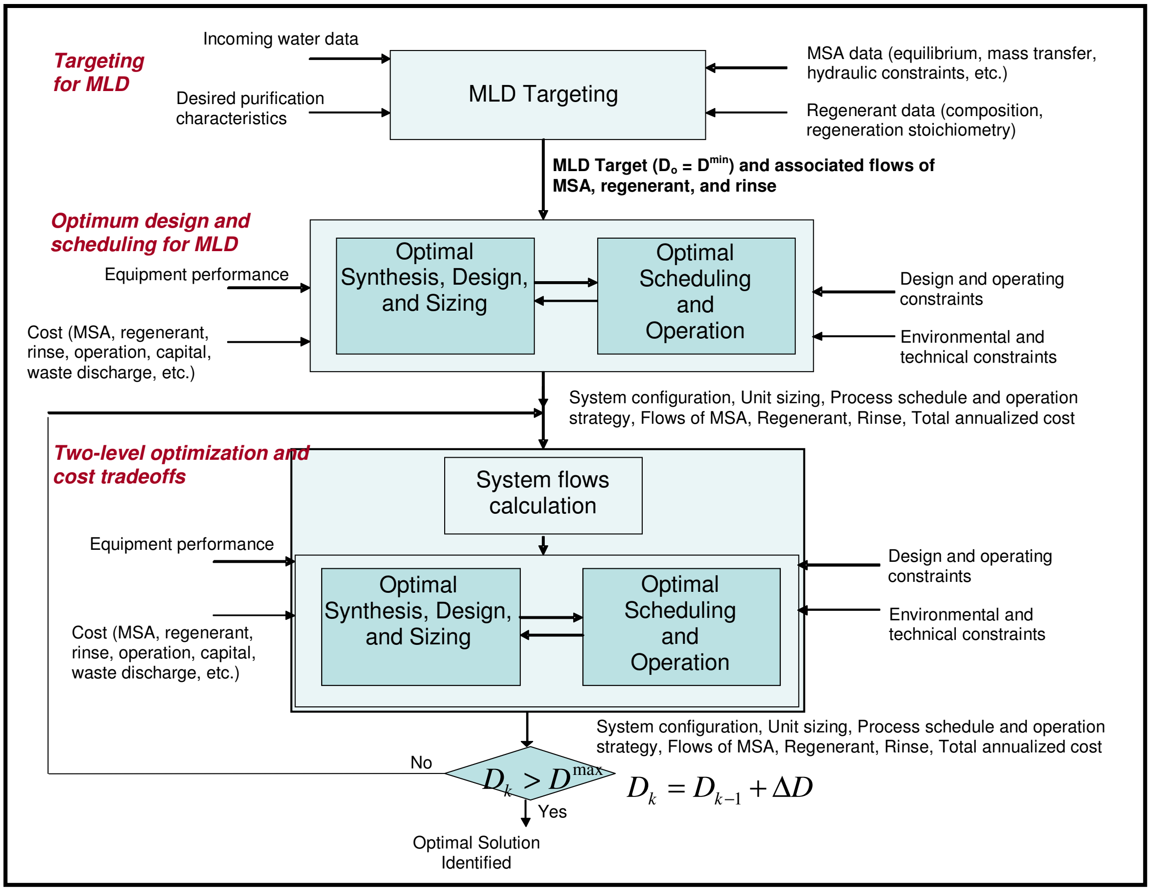
Targeting for MLD
Incoming water data
Desired purification characteristics
MLD Targeting
MSA data (equilibrium, mass transfer, hydraulic constraints, etc.)
Regenerant data (composition, regeneration stoichiometry)
MLD Target ($D_o = D^{min}$) and associated flows of MSA, regenerant, and rinse
Optimum design and scheduling for MLD
Equipment performance
Cost (MSA, regenerant, rinse, operation, capital, waste discharge, etc.)
Optimal Synthesis, Design, and Sizing
Optimal Scheduling and Operation
Design and operating constraints
Environmental and technical constraints
System configuration, Unit sizing, Process schedule and operation strategy, Flows of MSA, Regenerant, Rinse, Total annualized cost
Two-level optimization and cost tradeoffs
System flows calculation
Equipment performance
Cost (MSA, regenerant, rinse, operation, capital, waste discharge, etc.)
Optimal Synthesis, Design, and Sizing
Optimal Scheduling and Operation
Design and operating constraints
Environmental and technical constraints
System configuration, Unit sizing, Process schedule and operation strategy, Flows of MSA, Regenerant, Rinse, Total annualized cost
No
$D_k > D^{max}$
Yes
$D_k = D_{k-1} + \Delta D$
Optimal Solution Identified

Figure 5.4 Proposed Solution Approach

2. **Network Synthesis, Design, and Scheduling for MLD**: Once the target for MLD and associated flows are determined, one can proceed with the design and operation of the system to meet the process specifications at minimum cost. This involves the optimal identification of various many design and operating variables. For example: how many units for regeneration and washing should be used? How should they be configured? Should there be separate units for regeneration and washing or should they be combined in the same units? What are the optimum batch sizes for regeneration and washing? How often should the regeneration and washing steps occur? What are optimal sizes of the units used for purification, regeneration, and washing? The outcome of this stage should provide an optimum design and operating

scheme for the system and sufficient details about cost, configuration, sizing, and scheduling.

3. **Cost Tradeoffs**: While MLD is desirable from an environmental perspective, it is important to identify the appropriate level of waste discharge that balances environmental and economic considerations. Here, we propose a two-level optimization approach. At the top level, the extent of waste discharge is varied. For each level of discharge, system synthesis, design, and scheduling can be used to evaluate the minimum cost solution at that level. In each iteration, the waste flow is increased according to:

$$D_k = D_{k-1} + \Delta D \tag{5.2}$$

Where k is the iteration index and ΔD is the increase in waste discharge in each iteration. The iterations are continued until the waste flow reaches a maximum allowable waste discharge (D^{max}). The results establish the Pareto tradeoff (or non-inferior curve) between the waste discharge (D) and the minimum cost. The entire procedure is repeated for each MSA technology under consideration to determine the optimal system to use.

5.3.1 Minimum Liquid Discharge Targeting Procedure:

The minimum liquid discharge problem may be formulated as a nonlinear optimization program. The program will minimize the objective function, which is the discharge of the system, made up of several components:

$$D = R_{XS} + W + B + Water \tag{5.3}$$

R_{XS} is the excess amount of regenerant above the stoichiometric amount used to regenerate the MSA. W is the flow of the material used for washing. B is the flow of the salt or other complex formed by the removed contaminant. The *Water* term describes water added to the regeneration system from several sources. One is water possibly used to dilute the regenerant, T. For some ion exchange and other technologies, acid is used to regenerate the MSA. Commercial acid is normally too concentrated to use, so water from some source must be used to dilute it to the proper concentration. The other source making up the *Water* term is water used to fluidize the MSA. Normally, the MSA used in a water purification application is a solid and when pumping it through the system, water is used to create a slurry so that pumps may be used. This fluidization water is denoted by a fluidized void fraction of the MSA, ε, times the flowrate of the MSA: εL. Obviously, to minimize the total liquid discharge, each of the components that make up the discharge must be minimized individually, with the exception of the removed contaminant flow, which will be fixed for given process requirements and MSA used.

The program will have several constraints to describe the material balances, purification, regeneration & washing performances, number of units, and unit cycle times. The material balance equations will describe the main ion exchange operation as well as the MSA regeneration and washing operations. The flows possible in the system will be limited by the sizes of the process units. The unit sizes also will affect and be partially determined by the process cycle time (operation schedule). The operation schedule and design of the process units may be optimized similar to the procedure described by Grooms et al. (2005). Unit performance equations are used to describe the behavior of the units as a function of their run times (time between regeneration) while operation costs balance superior unit performance due to frequent regeneration with increased operation and lost opportunity costs from taking units out of service. Mass balance and process requirement equations further constrain the optimization.

Using these constraints, the nonlinear optimization program may be solved. However, because the program is very nonlinear, the solution obtained will be a local optimum. That is, because the problem has multiple solutions, there is no way to guarantee that a particular solution is truly the global minimum. Therefore, an alternative solution method was created that used some simple insights about the process to simplify the problem and make it tractable without the loss of any generality.

The concept of targeting system performance is used extensively in process integration. The theory is to determine the design goal for whatever aspect of the process is of concern before the details of the process are designed. This way, important feasibility and design information are obtained before additional effort is spent in detailed work. Targeting concepts have been previously applied in many process integration areas, such as mass exchange networks (El-Halwagi & Manousiouthakis, 1990; Noureldin & El-Halwagi, 2000), material recycle/reuse networks (El-Halwagi et al., 2003), heat exchange networks (Manousiouthakis & Martin, 2004; Zhu et al., 1995), and property interception (Kazantzi & El-Halwagi, 2005; Eden et al., 2005).

By identifying the flows entering and exiting the regeneration system, we can quickly determine the flows that are discharged. From Figure 5.5, we can refine our equation for the discharge flow. We can see that in addition to the terms R_{XS}, W, and B from equation 5.2, we have the dilution water flow T, and the fluidization water term $(\varepsilon_1 - \varepsilon_2) \cdot L$. ε_1 is the fluidization void fraction of the MSA-water mixture entering the

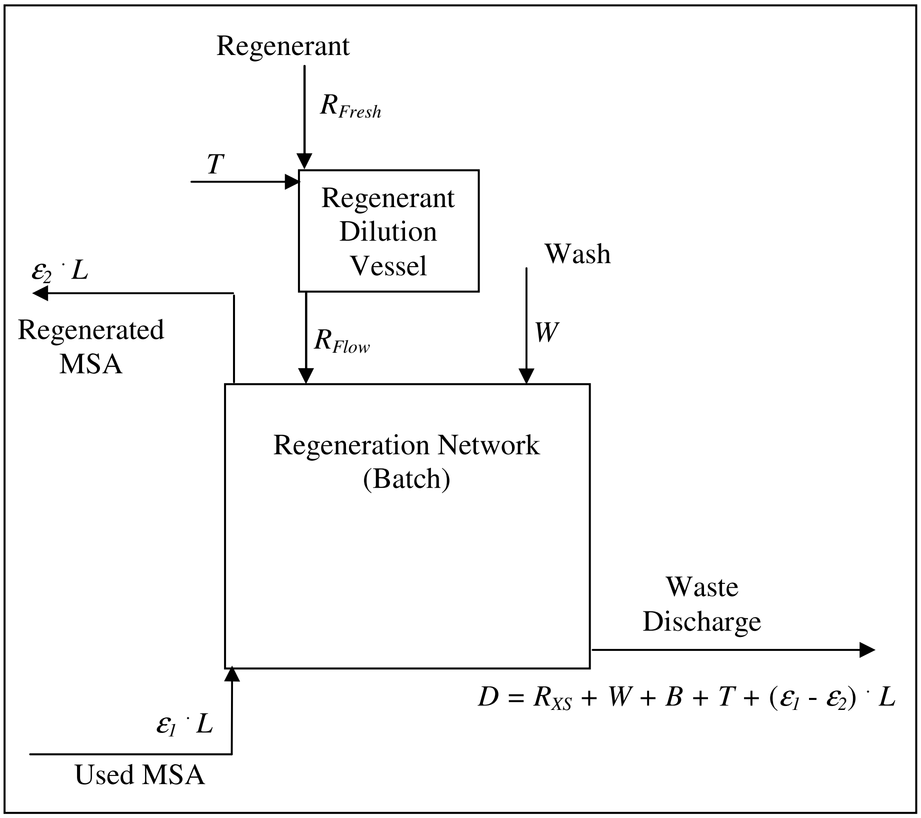

Figure 5.5 Discharge Targeting Diagram

regeneration system from the main purification vessel and ε_2 is the void fraction of the MSA-water mixture leaving the regeneration system and returning to the purification vessel. This term represents the net movement of fluidization water into or out of the regeneration system.

While several terms making up the liquid discharge will be determined based on the system parameters, we can draw some conclusions about the dilution, fluidization, and excess regenerant flows simply from our knowledge of the system. As can be seen from Figure 5.5, any regenerant that is used in excess of the stoichiometric amount required will be discharged unused. This is a significant cost because not only is more regenerant being purchased than is necessary, but it also results in an additional discharge cost. So a single mistake results in two additional costs. Thus, the amount of regenerant used should be carefully monitored to ensure there is no excess waste. In some situations, excess regenerant is necessary for mass transfer purposes but here also, the amount used should be precisely controlled at the necessary level to reduce both the cost and also the discharge amount. It will be assumed from here that proper control is achieved and excess regenerant is not used.

Regarding the fluidization water, it would seem that ε_2 should be maximized to minimize the total discharge. However, the fluid making up the void fraction leaving the regeneration system possibly contains regenerant material, which often harms the performance of the water purification process. Therefore, ε_2 should be minimized to ensure adequate purification and ε_1 should be minimized to reduce the liquid discharge. This can be accomplished through various operating strategies or methods. If they have similar values, the additional discharge from fluidization water will be negligible.

While water is normally required to dilute the regenerant, the source of that water can be used to minimize the discharge amount. Rather than using fresh water to dilute, the used wash material may normally be used. Depending on the relative values, this may reduce the discharge amount by as much as the entire amount required for dilution.

If *T* is greater than the wash flow, the entire used wash flow will be recycled and the fresh dilution flow may be reduced as shown:

$$T^{Fresh} = T - W \qquad \text{for } T > W \tag{5.4}$$

If *T* is less than or equal to the wash flow (the more common situation), the fresh dilution flow may be eliminated altogether and the recycled flow will be the difference between the two:

$$T^{\mathrm{Re}\,cy} = W - T \qquad \text{for } T \leq W \tag{5.5}$$

Since it is very common for the wash flow to be acceptable to be used to dilute the regenerant and the amount needed for dilution to be less than that required for washing, it will be assumed from this point that a portion of the used wash flow is used to dilute the regenerant and there will be no additional discharge from dilution needs.

The amount of contaminant discharged may be found from the extent of purification achieved. That is, the amount that is removed from the water being purified must be discharged after it is removed from the regenerated MSA. This may be calculated using the following equation:

$$I = F_{water} \cdot \left(c_{in} - c_{out}\right) \tag{5.6}$$

where *I* is the ion or other impurity that is removed from the contaminated water stream. It should be noted that *I* is independent of the MSA technology or the operating conditions. Different MSA technologies will have different methods for removing the contaminant. For some technologies, the contaminant will be bound to a salt or other complex. This combination will make up the *B* term in the discharge expression. For example: in ion exchange, the ion is bound to an oppositely charged ion from the regenerant to form a salt. For activated carbon regenerated with steam, the combination of the steam and the impurity will comprise the *B* term. So the specific calculation will vary with different MSA technologies. But the general equation is:

$$I + H_{Reg} = B \tag{5.7}$$

where H_{Reg} is the part of the regenerant that forms the complex with the removed impurity. So, *B* may be found knowing only the impurity removal requirements and the properties of the MSA technology being used.

The last component of the discharge flow is the used wash. The wash amount is a fixed ratio of the regenerated MSA flow:

$$W = W_{ratio} \cdot L \tag{5.8}$$

calculated using the minimum wash ratio W_{ratio}, specified by the MSA manufacturer. So the wash flow is solely dependent on the MSA flowrate. The MSA flowrate must be calculated using several pieces of data that may be obtained from the MSA manufacturer: the minimum regenerant ratio, the regeneration performance curve, and the water flow and purity specifications. The regeneration performance curve obtained from the manufacturer may be used to generate an equation relating the extent of regeneration (Δx) to the amount of regenerant used relative to the MSA flow ($^{R}/_{L}$).

$$\Delta x = f\left(\frac{R}{L}\right) \tag{5.9}$$

where f is some function obtained from the curve or data provided from the manufacturer. The minimum $^{R}/_{L}$ ratio may be used in the above function and the Δx for the minimum regenerant flow may then be calculated. Then, the water purification operation mass balance is used to determine the MSA flow:

$$L = \frac{F \cdot (c_{in} - c_{out})}{\Delta x} \tag{5.10}$$

The minimum regeneration flow is calculated using the equation:

$$R = \left(\frac{R}{L}\right)_{\min} \cdot L \tag{5.11}$$

We have thus targeted the minimum liquid discharge which is independent of the system or method of operation we choose from this point on using very basic process and material data.

Because in later steps of our design we may use a regeneration ratio larger than the minimum, we call the ratio of regenerant to MSA flow R_{ratio}:

$$R_{ratio} = \frac{R_{Flow}}{L} \tag{5.12}$$

5.3.2 Optimum Scheduling and Design for Minimum Discharge:

Once the minimum liquid discharge has been targeted, the unit sizes, operation schedule, and system configuration may be found.

As mentioned previously, two configurations of regeneration networks may be used – single-purpose or multi-purpose units. There is no advantage to using a mixture of the two systems, so a discrete choice of one of the two systems must be made. Total costs will be calculated for each system and the lower-cost configuration will be selected.

We can use some simple insights to easily determine the total unit volumes and cycle times of the two systems. If the time required for regeneration and washing (T_R & T_W) of the MSA is known, the volume of the necessary regeneration system may be calculated. For example, for the multi-use unit regeneration system, the time between regenerated MSA batches is $\frac{T_R + T_W}{N}$ where N is the number of regeneration units in the system. Since for a water flowrate, F_{water}, into the main vessel an average MSA flowrate, L, must be regenerated, the volume of a regeneration/wash unit must be the MSA flowrate plus the larger of the regenerant and the wash flow multiplied by the time,

$$\begin{aligned} V_R^{Multi-use} &= L \cdot \frac{T_R + T_W}{N} + L \cdot \arg\max\left[R_{ratio}, W_{ratio}\right] \cdot \frac{T_R + T_W}{N} \\ &= L \cdot \frac{T_R + T_W}{N} \cdot \left(1 + R_{ratio}\right) \qquad (5.13) \end{aligned}$$

if $R_{ratio} \geq W_{ratio}$ (the most common case). When multiplied by the number of units, we obtain the total volume of the multi-use regeneration system,

$$\begin{aligned} V_{Total}^{Multi-use} &= N \cdot L \cdot \frac{T_R + T_W}{N} \cdot \left(1 + R_{ratio}\right) \\ &= L \cdot \left(T_R + T_W\right) \cdot \left(1 + R_{ratio}\right), \quad \text{if } R_{ratio} \geq W_{ratio} \qquad (5.14) \end{aligned}$$

which is intuitively correct. The number of units will be the total volume divided by the volume of each unit:

$$N^{Multi-use} = \frac{V_{Total}^{Multi-use}}{V_R^{Multi-use}} = \frac{L \cdot (T_R + T_W) \cdot (1 + R_{ratio})}{V_R^{Multi-use}} \tag{5.15}$$

For the single-use unit regeneration system the process is similar. The time between regenerated MSA batches is $\frac{T_R}{N}$ and the volume of the MSA in each batch must be $L \cdot \frac{T_R}{N}$. So, the volume of a regeneration unit is

$$\begin{aligned} V_R^{Single-use} &= L \cdot \frac{T_R}{N} + L \cdot R_{ratio} \cdot \frac{T_R}{N} \\ &= L \cdot \frac{T_R}{N} \cdot (1 + R_{ratio}) \end{aligned} \tag{5.16}$$

The time between washed MSA batches is $\frac{T_W}{M}$ (where *M* is the number of washing units) and the volume of MSA in each batch is still $L \cdot \frac{T_R}{N}$. The volume of a wash unit may be calculated as

$$\begin{aligned} V_W^{Single-use} &= L \cdot \frac{T_R}{N} + L \cdot W_{ratio} \cdot \frac{T_R}{N} \\ &= L \cdot \frac{T_R}{N} \cdot (1 + W_{ratio}) \end{aligned} \tag{5.17}$$

Since a batch of $L \cdot \frac{T_R}{N}$ of MSA is returned to the main vessel at time intervals of $\frac{T_W}{M}$ and an average flowrate of *L* is desired,

$$\frac{L \cdot \frac{T_R}{N}}{\frac{T_W}{M}} = L \tag{5.18}$$

So,

$$\frac{T_R}{N} = \frac{T_W}{M} \tag{5.19}$$

Or rearranging,

$$M = N \cdot \frac{T_W}{T_R} \tag{5.20}$$

Using this, we can calculate the total volume of the single-use regeneration system:

$$V_{Total}^{Single-use} = N \cdot L \cdot \frac{T_R}{N} \cdot \left(1 + R_{ratio}\right) + M \cdot L \cdot \frac{T_R}{N} \cdot \left(1 + W_{ratio}\right)$$

$$= L \cdot \left[T_R \cdot \left(1 + R_{ratio}\right) + T_W \cdot \left(1 + W_{ratio}\right)\right] \tag{5.21}$$

As in the multi-use unit network, the number of regeneration units may be found by dividing the total regeneration volume by the single-unit volume:

$$N^{Single-use} = \frac{L \cdot T_R \cdot \left(1 + R_{ratio}\right)}{V_R^{Single-use}} \tag{5.22}$$

The number of wash units is found using a similar equation:

$$M^{Single-use} = \frac{L \cdot T_W \cdot \left(1 + W_{ratio}\right)}{V_W^{Single-use}} \tag{5.23}$$

By inspection, we see that the equations for total volume of the two systems (equations 5.14 & 5.21) are equivalent if R_{ratio}=W_{ratio}, which intuitively should be the case. So we can say that if the two parameters R_{ratio} and W_{ratio} are very similar, there is little savings in using separate units for regeneration and washing so the decision should be made based on other factors. However, if there is a large difference, separate single-use units would give a significantly smaller total volume, but a few more total units than the multi-use unit system. We will illustrate the two systems for a given application in the following section.

Now that the total regeneration and washing system volumes have been found, the individual unit sizes will be determined. The maximum allowable diameter of a unit may be calculated in many different methods, one of which will be demonstrated in the case study in the next section. The maximum diameter should be used because based on economy of scale, it is more cost-effective to use a small number of large units than many smaller units. However, normally the diameter of the unit will be limited by mass transfer hydrodynamic effects. It must be ensured that there is sufficient mass transfer so that the MSA is sufficiently regenerated and washed. After finding the diameter, the volume of the individual units may be calculated. Next, the number of units required is found by dividing the total regeneration (or wash) volume required (as calculated previously) by the individual unit volumes as in equations 5.15, 5.22, and 5.23. The number and sizes of units in both configurations of the regeneration network are found because the cost of both systems will be calculated and compared to determine which is the optimal configuration. The diameter and volume of the main water purification vessel may also be calculated based on the water and MSA flows required (*F* & *L*) to ensure sufficient mass transfer is present. The number and sizes of all of the main process vessels has now been found.

To calculate the cost of the system, the final component of the operation cost must be found. Although the MSA is not consumed in the purification and regeneration operations, it will gradually become deactivated or unusable. Thus, a small amount of make-up MSA will be part of the operation cost of the system. The rest of the operation cost will consist of discharge disposal, regenerant and wash material cost, and pumping and utility costs. The cost of the multi-use unit regeneration system will be found according to the following equation:

$$Cost^{Multi-use} = \frac{\{(\text{Purification Unit Cost}) + (\text{Regeneration Unit Cost})\}}{(\text{Equipment Lifetime})}$$
$$+ (\text{Regenerant Cost}) + (\text{Wash Cost}) + (\text{Make - up MSA Cost})$$
$$+ (\text{Pumping Cost}) + (\text{Utility Cost}) \qquad (5.24)$$

The cost for the single-use unit regeneration system will be calculated in the same way, but with additional consideration of the washing units:

$$Cost^{Multi-use} = \frac{\{(\text{Purification Unit Cost}) + (\text{Regeneration Unit Cost}) + (\text{Wash Unit Cost})\}}{(\text{Equipment Lifetime})}$$
$$+ (\text{Regenerant Cost}) + (\text{Wash Cost}) + (\text{Make - up MSA Cost})$$
$$+ (\text{Pumping Cost}) + (\text{Utility Cost}) \qquad (5.25)$$

5.3.3 Two-level Optimization Procedure:

The two-level procedure to optimize the system is shown graphically in Figure 5.6. If the total discharge amount is the only system parameter of concern because of environmental, disposal, or other factors, then the minimum liquid discharge targeting will give the feasibility of the system and the process described above to design and schedule the system for minimum discharge will give the optimal design and operation schedule for the system. However, if possible, a two-level optimization procedure will be

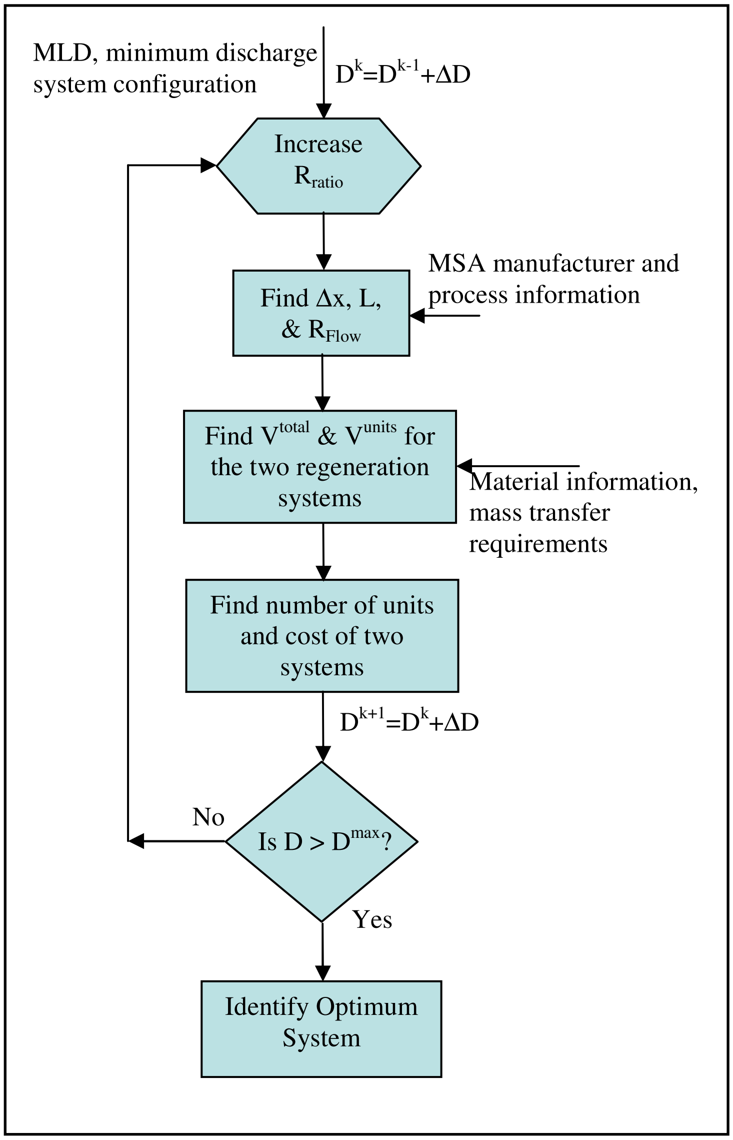
MLD, minimum discharge system configuration
$D^k=D^{k-1}+\Delta D$
Increase R_{ratio}
MSA manufacturer and process information
Find Δx, L, & R_{Flow}
Find V^{total} & V^{units} for the two regeneration systems
Material information, mass transfer requirements
Find number of units and cost of two systems
$D^{k+1}=D^k+\Delta D$
No
Is $D > D^{max}$?
Yes
Identify Optimum System

Figure 5.6
Two-level Water Purification System Optimization Procedure

followed to balance operating and capital costs. Any other available MSA technologies will also be evaluated in this way to compare costs and discharge amounts from a range of separation alternatives.

If the discharge amount is flexible and it is possible to discharge a quantity greater than the minimum, then it may be possible to further optimize the purification system. After the minimum liquid discharge is targeted and the optimal system design and schedule is formulated, the design process is repeated using a slightly larger discharge amount. The quantity discharged is incrementally increased until reaching some maximum value and the total system cost is calculated for each increment. Using a plot of the data, a trade-off between operational costs and capital costs should be seen and the minimum cost for each configuration (multi-use regeneration units versus single-use units) is determined. The configuration and process variables resulting in the minimum cost are recorded. The entire sequence may then be repeated for any additional MSA technologies that are available and the minimum cost for each technology may be compared to choose the best technology.

5.4 Case Study

The industrial case that will be studied is purification of 600 $^{gal}/_{min}$ of water from an underground well for use in cooling. The water is high in sodium which must be reduced from a concentration of 50,000 ppm to 10,000 ppm. The MSA technology under consideration is an ion exchange resin. Some process and MSA parameters are given in Table 5.2.

Table 5.2. Case Study Parameters

F_{water}	600 gpm
c^{in}	0.05
c^{out}	0.01
R_{ratio}^{Min}	0.6
W_{ratio}	0.1
T_R	20 minutes
T_W	10 minutes
C_V	2.7
C_R	0.6
C_W	0.1

Table 5.2 Continued	
C_B	0.15
C_L	1.5
R_{Flux}^{Min}	$3\ ^{\mathrm{L}}/_{\mathrm{ft2}}$
R_{Flux}^{Max}	$5\ ^{\mathrm{L}}/_{\mathrm{ft2}}$

We will examine regeneration systems with both single- and multi-purpose units. Using the purification requirements, the amount of sodium removed is calculated as shown in equation 5.6:

$$I = F_{water} \cdot (c_{in} - c_{out}) = \left(37.85 \,{}^{\mathrm{kg}}\!/_{\mathrm{s}}\right) \cdot (0.05 - 0.01) = 1.51\ {}^{\mathrm{kg}}\!/_{\mathrm{s}} \qquad (5.26)$$

Δx may be determined using an expression for the regeneration performance of the ion exchange resin:

$$\Delta x = 0.2 \cdot \ln\left(30 \cdot \left({}^{R}\!/_{L}\right)_{\min}\right) = 0.2 \cdot \ln[(30) \cdot (0.6)] = 0.578 \qquad (5.27)$$

Next, the MSA flowrate may be calculated from equation 10:

$$L = \frac{F_{water} \cdot (c_{in} - c_{out})}{\Delta x} = \frac{I}{\Delta x} = \frac{1.51 \,{}^{\mathrm{kg}}\!/_{\mathrm{s}}}{0.578} = 2.62\ {}^{\mathrm{kg}}\!/_{\mathrm{s}} \qquad (5.28)$$

Now, the minimum regenerant flowrate may be found using the MSA flowrate and the minimum regenerant ratio as in equation 5.11:

$$R_{Flow} = \left(\frac{R}{L}\right)_{\min} \cdot L = (0.6) \cdot \left(2.62 \,{}^{\mathrm{kg}}\!/_{\mathrm{s}}\right) = 1.57\ {}^{\mathrm{kg}}\!/_{\mathrm{s}} \qquad (5.29)$$

Note that if dilution of the regenerant were necessary, the mass of the dilution water would be added to the above calculation. The wash flow is similarly found using equation 8.

$$W = W_{ratio} \cdot L = (0.1) \cdot \left(2.62 \,{}^{\mathrm{kg}}\!/_{\mathrm{s}}\right) = 0.26 \,{}^{\mathrm{kg}}\!/_{\mathrm{s}} \qquad (5.30)$$

The regeneration process using HCl to remove the impurity, sodium, from the ion exchange resin and replaces it with a hydrogen ion. In this way, NaCl salt is created and discharged. The regeneration reaction is shown in the following equation:

$$\text{Na - Resin} + \text{HCl} \rightarrow \text{NaCl} + \text{H - Resin} \tag{5.31}$$

The amount of brine, *B*, discharged is calculated using a 1:1 mole ratio between HCl and NaCl from the above equation:

$$B = \frac{I}{M_{Na}} \cdot M_{NaCl} = \frac{1.51 \,{}^{\text{kg}}\!/\!_{\text{s}}}{23} \cdot (58.45) = 3.84 \,{}^{\text{kg}}\!/\!_{\text{s}} \tag{5.32}$$

For our application, we must use 10% excess regenerant to ensure sufficient regeneration, so 10% of the regenerant will be discharged. Our MSA does not require dilution of the acid, so we don't have to recycle used wash flow to dilute the acid. Also, the difference between the water fluidization amounts (ε_1 and ε_2) is negligible. So, the minimum discharge amount is found as by adding the brine and wash volumetric flows:

$$\begin{aligned} D = W + R_{XS} + B &= \left(0.26 \,{}^{\text{kg}}\!/\!_{\text{s}}\right) \cdot \left(\frac{1\,\text{L}}{1\,\text{kg}}\right) + (0.1) \cdot \left(1.57 \,{}^{\text{kg}}\!/\!_{\text{s}}\right) \cdot \left(\frac{1\,\text{L}}{1.18\,\text{kg}}\right) \\ &\quad + \left(3.84 \,{}^{\text{kg}}\!/\!_{\text{s}}\right) \cdot \left(\frac{1\,\text{L}}{2.165\,\text{kg}}\right) \\ &= 2.167 \,{}^{\text{L}}\!/\!_{\text{s}} \end{aligned} \tag{5.33}$$

We have targeted for the minimum liquid discharge and this is feasible for our application so we continue to designing the optimum system and operation for our process. First we will design an optimal process with the multi-use regeneration system configuration, then using the single-use configuration. The total regeneration volume required for the multi-use system is calculated using equation 5.14:

$$\begin{aligned} V_{Total}^{Multi-use} &= L \cdot (T_R + T_W) \cdot (1 + R_{ratio}) \\ &= \left[\left(2.62 \,{}^{\text{kg}}\!/\!_{\text{s}}\right) \cdot \left(\frac{1\,\text{L}}{1.5\,\text{kg}}\right) \cdot \left(60 \,{}^{\text{sec}}\!/\!_{\text{min}}\right)\right] \cdot (20\,\text{min} + 10\,\text{min}) \cdot (1 + 0.6) \\ &= 5030\,\text{L} = 1329\,\text{gal} \end{aligned} \tag{5.34}$$

To use equation 5.15 to calculate the number of regeneration units, we must find the volume of a single unit. In this case, we are given liquid flux limits from the MSA manufacturer to ensure sufficient mass transfer. We are given limits of 3-5 $^{\mathrm{L}}/_{\mathrm{s\cdot ft^2}}$ for the regeneration flux. We wish to have the maximum size unit possible, so we use the lower limit to calculate the diameter. The equation to calculate the diameter of a unit using a given flux is:

$$D_R = \sqrt{\frac{R_{Flow}/N}{R_{Flux}} \cdot \left(\frac{4}{\pi}\right)} \tag{5.35}$$

The volume of a unit with this diameter would be calculated and used in equation 5.15 to find the number of units. This would give:

$$N^{Multi-use} = \frac{V_R^{Total}}{V_R} = \frac{V_{Total}^{Multi-use}}{\sqrt{\frac{4}{\pi} \cdot \left(\frac{H}{D}\right)_{Spec}} \cdot \left(\frac{R_{Flow}}{N \cdot R_{Flux}}\right)^{3/2}} \tag{5.36}$$

which is an implicit equation in *N*. The equation may be explicitly solved for *N* to give the following equation:

$$N^{Multi-use} = \left(\frac{R_{Flux}}{R_{Flow}}\right)\left[\frac{V_{Total}^{Multi-use}}{\sqrt{\frac{4}{\pi} \cdot \left(\frac{H}{D}\right)_{Spec}}}\right]^{2/3} \tag{5.37}$$

To use this equation we will need to calculate the liquid flow of regenerant using a density of 1.18 kg/L for the acid. As mentioned above, if dilution of the regenerating acid were necessary, the total diluted mass would be included in this calculation.

$$R_{Flow} = \left(1.57\,^{\mathrm{kg}}/_{\mathrm{s}}\right) \cdot \left(\frac{1\,\mathrm{L}}{1.18\,\mathrm{kg}}\right) = 1.33\,^{\mathrm{L}}/_{\mathrm{s}} \tag{5.38}$$

Using this, the number of regeneration units in the multi-use unit system is calculated:

$$N^{Multi-use} = \left(\frac{R_{Flux}}{R_{Flow}}\right)\left[\frac{V_{Total}^{Multi-use}}{\sqrt{\frac{4}{\pi} \cdot \left(\frac{H}{D}\right)_{Spec}}}\right]^{2/3}$$

$$= \left(\frac{3\,\mathrm{L}/_{\mathrm{s}\cdot\mathrm{ft}^2}}{1.33\,\mathrm{L}/_{\mathrm{s}}} \right) \left\{ \frac{(5030\,\mathrm{L}) \cdot \left(\dfrac{1\,\mathrm{ft}^3}{28.32\,\mathrm{L}} \right)}{\sqrt{\dfrac{4}{\pi}} \cdot (3)} \right\}^{2/3}$$

$$= 31.61\,\text{units} = 32\,\text{units} \qquad (5.39)$$

We may now find the sizes of the individual units using equation 5.35:

$$D_R^{Multi-use} = \sqrt{\frac{R_{Flow}/N}{R_{Flux}} \cdot \left(\frac{4}{\pi} \right)} = \sqrt{\frac{1.33\,\mathrm{L}/_{\mathrm{s}}}{(32) \cdot \left(3\,\mathrm{L}/_{\mathrm{s}\cdot\mathrm{ft}^2} \right)} \cdot \left(\frac{4}{\pi} \right)} = 0.133\,\mathrm{ft}$$

(5.40)

This procedure is repeated to find the total system volume, number of units, and unit sizes for the single-use unit regeneration system. The procedure will be identical to that followed above, but the equations will be slightly different. First, we find the total volumes of the regeneration and washing systems using equation 5.21:

$$V_{R\,Total}^{\text{Single-use}} = L \cdot T_R \cdot (1 + R_{ratio}) = \left(2.62\,\mathrm{kg}/_{\mathrm{s}} \right) \cdot \left(\frac{1\,\mathrm{L}}{1.5\,\mathrm{kg}} \right) \cdot (20\,\mathrm{min}) \left(\frac{60\,\mathrm{s}}{1\,\mathrm{min}} \right) \cdot (1 + 0.6)$$

$$= 3360\,\mathrm{L} = 887.6\,\mathrm{gal} \qquad (5.41)$$

$$V_{W\,Total}^{\text{Single-use}} = L \cdot T_W \cdot (1 + W_{ratio}) = \left(2.62\,\mathrm{kg}/_{\mathrm{s}} \right) \cdot \left(\frac{1\,\mathrm{L}}{1.5\,\mathrm{kg}} \right) \cdot (10\,\mathrm{min}) \left(\frac{60\,\mathrm{s}}{1\,\mathrm{min}} \right) \cdot (1 + 0.1)$$

$$= 1155\,\mathrm{L} = 305.1\,\mathrm{gal} \qquad (5.42)$$

The number of regeneration and washing units is now found using an equation similar to equation 5.37. The calculations are performed using the total regeneration and washing system volumes, the regenerant and wash liquid flows, and the lower liquid flux limit of 3 $\mathrm{L}/_{\mathrm{s}\cdot\mathrm{ft}2}$.

$$N^{Single-use} = \left(\frac{R_{Flux}}{R_{Flow}} \right) \left[\frac{V_{R\,Total}^{Single-use}}{\sqrt{\dfrac{4}{\pi}} \cdot \left(\dfrac{H}{D} \right)_{Spec}} \right]^{2/3}$$

$$= \left(\frac{3\,\mathrm{L}\big/_{\mathrm{s}\cdot\mathrm{ft}^2}}{1.33\,\mathrm{L}\big/_{\mathrm{s}}} \right) \left\{ \frac{(3360\,\mathrm{L})\cdot\left(\frac{1\,\mathrm{ft}^3}{28.32\,\mathrm{L}}\right)}{\sqrt{\frac{4}{\pi}}\cdot(3)} \right\}^{2/3}$$

$$= 24.16\,\text{units} = 25\,\text{units} \tag{5.43}$$

$$M^{Single-use} = \left(\frac{R_{Flux}}{W}\right)\left[\frac{V_{W\,Total}^{Single-use}}{\sqrt{\frac{4}{\pi}}\cdot\left(\frac{H}{D}\right)_{Spec}}\right]^{2/3}$$

$$= \left(\frac{3\,\mathrm{L}\big/_{\mathrm{s}\cdot\mathrm{ft}^2}}{0.26\,\mathrm{L}\big/_{\mathrm{s}}} \right) \left\{ \frac{(1155\,\mathrm{L})\cdot\left(\frac{1\,\mathrm{ft}^3}{28.32\,\mathrm{L}}\right)}{\sqrt{\frac{4}{\pi}}\cdot(3)} \right\}^{2/3}$$

$$= 60.64\,\text{units} = 61\,\text{units} \tag{5.44}$$

Finally, we find the sizes of the individual regeneration and washing units in the single-use unit system configuration using equation 5.35:

$$D_R^{Single-use} = \sqrt{\frac{R_{Flow}\big/N}{R_{Flux}}\cdot\left(\frac{4}{\pi}\right)} = \sqrt{\frac{1.33\,\mathrm{L}\big/_{\mathrm{s}}}{(25)\cdot 3\,\mathrm{L}\big/_{\mathrm{s}\cdot\mathrm{ft}^2}}\cdot\left(\frac{4}{\pi}\right)} = 0.15\,\mathrm{ft} \tag{5.45}$$

$$D_W^{Single-use} = \sqrt{\frac{W\big/N}{R_{Flux}}\cdot\left(\frac{4}{\pi}\right)} = \sqrt{\frac{0.26\,\mathrm{L}\big/_{\mathrm{s}}}{(61)\cdot 3\,\mathrm{L}\big/_{\mathrm{s}\cdot\mathrm{ft}^2}}\cdot\left(\frac{4}{\pi}\right)} = 0.043\,\mathrm{ft} \tag{5.46}$$

The size of the main water purification unit may also be calculated by using a lower flux limit governing mass transfer in the purification stage:

$$D_{Pur} = \sqrt{\frac{L+F_{water}}{L_{Flux}}\cdot\left(\frac{4}{\pi}\right)} = \sqrt{\frac{1.75\,\mathrm{L}\big/_{\mathrm{s}} + 34.41\,\mathrm{L}\big/_{\mathrm{s}}}{2\,\mathrm{L}\big/_{\mathrm{s}\cdot\mathrm{ft}^2}}\cdot\left(\frac{4}{\pi}\right)} = 4.8\,\mathrm{ft} \tag{5.47}$$

The total process cost of the two systems may now be calculated using equations 5.24 and 5.25. For our application, the pumping and utility costs are negligible compared to the other costs for the process. The make-up MSA flow is calculated using a deactivation factor of 1% of the circulating flow:

$$L_{Make-up} = y_{MSA} \cdot L = (0.01) \cdot \left(2.62 \, {}^{\mathrm{kg}}\!/\!{}_{\mathrm{s}} \right) = 0.026 \, {}^{\mathrm{kg}}\!/\!{}_{\mathrm{s}} \tag{5.54}$$

The total process cost of the multi-use regeneration unit system will be:

$$Cost^{Multi-use} = C_V \cdot \left\{ N \cdot \left[D_R^{0.7} + \left(D_R \cdot \left(\frac{H}{D} \right)_{Spec} \right)^{0.7} \right] + \left(D_{Pur} \right)^{0.7} + \left(D_{Pur} \cdot \left(\frac{H}{D} \right)_{Spec} \right)^{0.7} \right\} \cdot \frac{1}{ELT}$$

$$+ C_L \cdot L_{Make-up} + C_D \cdot D + C_R \cdot R_{Flow} + C_W \cdot W$$

$$= (2.7) \cdot \left\{ (32) \cdot \left[(0.13\,\mathrm{ft})^{0.7} + \left[(0.13\,\mathrm{ft}) \cdot (3) \right]^{0.7} \right] + \left[(4.8\,\mathrm{ft})^{0.7} + \left[(4.8\,\mathrm{ft}) \cdot (3) \right]^{0.7} \right] \right\} \cdot \left(\frac{1}{5} \right)$$

$$+ (1.5) \cdot \left(0.026 \, {}^{\mathrm{kg}}\!/\!{}_{\mathrm{s}} \right) + (0.15) \cdot \left(2.034 \, {}^{\mathrm{L}}\!/\!{}_{\mathrm{s}} \right) + (0.6) \cdot \left(1.56 \, {}^{\mathrm{kg}}\!/\!{}_{\mathrm{s}} \right) + (0.1) \cdot \left(0.26 \, {}^{\mathrm{kg}}\!/\!{}_{\mathrm{s}} \right)$$

$$= {}^{\$19.5 \times 10^3}\!/\!{}_{\mathrm{period}} \tag{5.55}$$

where C_V is the volumetric cost of a vessel, *ELT* is the expected lifetime of the capital equipment, C_L is the cost of the MSA, C_D is the volumetric cost of any material discharged, C_R is the unit cost of regenerant, and C_W is the unit cost of wash material. The cost of the entire process with the single-use regeneration unit system will be:

$$Cost^{Single-use} = C_V \left\{ N \cdot \left[D_R^{0.8} + \left(D_R \cdot \left(\frac{H}{D} \right)_{Spec} \right)^{0.7} \right] + M \cdot \left[D_W^{0.8} + \left(D_W \cdot \left(\frac{H}{D} \right)_{Spec} \right)^{0.7} \right] \right\} \cdot \frac{1}{ELT}$$

$$+ C_V \cdot \left(D_{Pur} \right)^{0.7} \left(D_{Pur} \cdot \left(\frac{H}{D} \right)_{Spec} \right)^{0.7} \cdot \frac{1}{ELT} + C_L \cdot L_{Make-up} + C_D \cdot D + C_R \cdot R_{Flow} + C_W \cdot W$$

$$= 2.7 \cdot \left\{ 25 \cdot \left[(0.15)^{0.7} + ((0.15) \cdot (3))^{0.7} \right] + 61 \cdot \left[(0.04)^{0.7} + ((0.04) \cdot (3))^{0.7} \right] + (4.8\,\mathrm{ft})^{0.7} + \left[(4.8\,\mathrm{ft}) \cdot (3) \right] \right\} \cdot \left(\frac{1}{5} \right)$$

$$+ (1.5) \cdot \left(0.026 \, {}^{\mathrm{kg}}\!/\!{}_{\mathrm{s}} \right) + (0.15) \cdot \left(2.034 \, {}^{\mathrm{L}}\!/\!{}_{\mathrm{s}} \right) + (0.6) \cdot \left(1.56 \, {}^{\mathrm{kg}}\!/\!{}_{\mathrm{s}} \right) + (0.1) \cdot \left(0.26 \, {}^{\mathrm{kg}}\!/\!{}_{\mathrm{s}} \right)$$

$$= {}^{\$28.6 \times 10^3}\!/\!{}_{\mathrm{period}} \tag{5.56}$$

So the total cost is $19,500 per period for the single-use system versus $28,600 for the multi-use system using the minimum stoichiometric regenerant flow.

Now that two optimal systems have been designed, along with their operational parameters, the two-level optimization procedure is followed. For our application, the maximum allowable discharge amount is 2.3 $^{L}/_{s}$ due to environmental regulations. So, the regenerant flow is incrementally increased until the maximum discharge amount is reached. The above calculations are repeated for each case and the costs of the two regeneration system configurations are calculated. Figures 5.7 and 5.8 show the results of those calculations for the current case study.

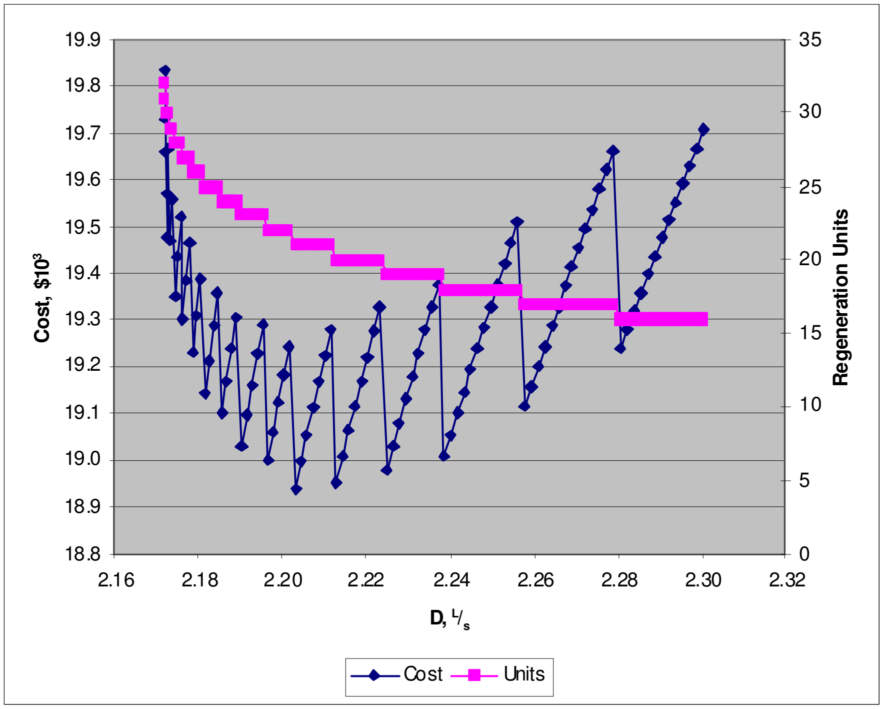

Figure 5.7. Multi-use Unit Regeneration System Cost Profile

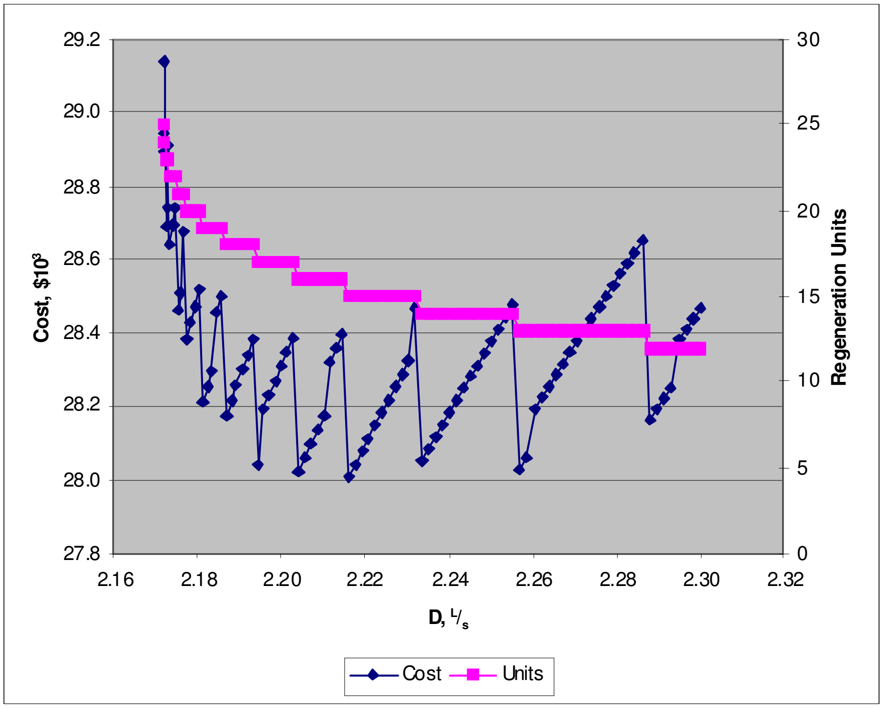

Figure 5.8. Single-use Unit Regeneration System Cost Profile

The total cost of each system is calculated and plotted for increasing regenerant flow. The number of units is also plotted against the regenerant flow. From these calculations, we can see that the minimum cost for the multi-use unit regeneration system is $18,940 per period at a discharge flow of 2.203 $^{L}/_{s}$ while the single-use system minimum cost is $28,000 per period with a flow of 2.216 $^{L}/_{s}$. So, we would choose to use the multi-use unit regeneration system with 21 units, which yields the minimum cost.

From the graphs, we see that for a constant number of units, the cost increases as the discharge flow increases due to the increasing sizes of the units and the increased usage of regenerant. However, as the discharge flow increases, the units become large enough to reduce the number used. Using one fewer unit dramatically reduces the cost of the system and results in the sharp drop seen in the graphs. However, as the discharge flow increases even more, the unit diameters required become so large that even reducing the number used does not make up for the increasing size of the units. So, from this type of plot, it is easy to identify the number of units that will yield the minimum cost.

When planning the operation of the system, the discharge flow should be chosen that is slightly above the minimum-cost point so that system disturbances don't harm the operation of the system. For example, if the system were operated at the absolute minimum-cost point (D = 2.203 $^{L}/_{s}$ for the multi-use system here) and the regenerant flow dropped temporarily for some reason (control disturbance, blockage in a pipe, etc.), then there would be insufficient flow in the units for adequate mass exchange. This would result in incomplete regeneration and ultimately, poor treated water quality. Thus, the discharge flow should be slightly higher than minimum for the desired number of units (perhaps 2.21 $^{L}/_{s}$ in our example) which would be at a point slightly above the bottom on the 21-unit line in our graph. The additional cost for this safety margin may be easily seen and balanced against the need for quality assurance. Thus, the graphs not only illustrate the optimum system design and operating conditions, they also yield valuable insights into the system's performance at many different operating conditions.

In our example, the values for R_{ratio} and W_{ratio} were very close, so there was little savings that resulted in using separate units for washing. Using separate units increased the number of units needed and this more than outweighed the savings in the size of the units. This caused the multi-use unit regeneration system to be less expensive than the single-use unit configuration. For another type of ion exchange resin that uses values for R_{stoic} and W_{stoic} that have a larger difference, the outcome will be different.

5.5 Conclusions

This paper illustrates the usefulness of applying process integration and optimization principles to real unsteady-state processes. Utilizing process integration targeting techniques, the minimum liquid discharge possible was targeted to determine the system's feasibility. After feasibility was established, a method was shown to synthesize an optimal regeneration network with minimum liquid discharge that included the number and configuration of regeneration and wash units, the flowrates of the regenerant and wash, and the regeneration and wash cycle times. For systems with flexibility in the allowable discharge amount, a two-level optimization procedure was illustrated that balanced the environmental and operational costs with capital costs. The complete process will be repeated for any additional water purification technologies available and their minimum costs compared. Thus, optimal technology and system will be selected. Although much emphasis is placed on selecting a technology and process conditions that will accomplish the desired process operation such as water purification, this work demonstrates that that related operations (regeneration, in this case) are also important factors to consider.

CHAPTER VI
CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

This dissertation has focused on applying the tools and methodologies of process integration to systems exhibiting unsteady-state behavior. While there has been much work done in the past on varying types of unsteady-state systems, approaching the problem using the perspective of process integration helps to formulate more general solution methodologies than those previously introduced. To this end, three hybrid steady- and unsteady-state design and operation synthesis problems were solved: a general mass exchange network, a general property-interception network, and a water purification system based on an actual industrial process.

The mass exchange network problem illustrated a general method to optimize the design and operation of a separation network with both steady-state and dynamic mass exchange units. A network superstructure that included all possible network configurations was used to enable the optimization procedure to choose the optimal structure for each time period. A time horizon model was used to model the behavior and regeneration of the unsteady-state units. This enabled the optimization process to accomplish the complex trade-off of balancing better unit performance and unit availability with the cost of regeneration and lost opportunity cost of a unit being unavailable. Even for a very simple system such as the 3-unit case study solved, the trade-offs are much too complex to be solved by inspection and thus the tool of mathematical optimization is essential. Not only was the system structure optimized, but the design of the unit sizes was also included. While detailed specifics such as internal structures and differential fluid equations should be done at a later, more detailed design stage, finding the optimal unit sizes not only helps give a more informed preliminary process design, but it also includes the trade-off between capital and operational costs in the initial process design. While similar design methods have been proposed, none has optimized a process as generally as the procedure demonstrated here. Because it was created as a general methodology it is focused on designing new process, but it is even more straightforward for use in optimizing operation of existing processes. This would be a special case of the more general problem because certain parameters such as unit sizes and technologies available would be specified from the start. Also, although the system designed in the case study operated within an otherwise steady-state process (the process sources and sinks were steady-state), this is also a special case. It is equally possible for a source to be an unsteady-state feed coming from a batch reactor or a periodic supplier. Although introducing a global optimization algorithm was not a primary goal of this research, because the formulation resulted in a mixed-integer nonlinear optimization program, a global optimization method was created to identify the global solution. A two-level optimization procedure using a linearized, relaxed program and the original nonlinear program was able to identify the global solution of the case study shown.

The property interception network problem was the first effort at using the new tool of property integration for unsteady-state processes. Property integration is a useful approach to problems that are either intractable or very tedious using conventional component-based design. Early work in the area focused on identifying direct recycle techniques and eventually synthesis of interception networks with the condition that their behavior was steady-state. This work removes that constraint so that now a network with any type of behavior and any type of process unit may be optimized. Using the property integration tools, separating agents needed for the network may be quickly identified based on their property characteristics without a trial-and-error search process. This design method is consistent with research now going on whose goal is to synthesis or modify materials to accomplish a given goal knowing the properties required.

The water purification process optimization work illustrated an example of a specific industrial process which exhibited both steady-state and dynamic characteristics being optimized. The main water purification process was a nearly steady-state operation while the regeneration operation is inherently a batch process. Utilizing targeting techniques used in other areas of process integration, the minimum liquid discharge possible was targeted. This targeting can determine if a system is feasible without even doing preliminary designs. After determining a particular system's feasibility, a method was given to design the system and find the number of regeneration and wash units, their configuration, the optimal scheme and flows of the regeneration and wash materials, the sizes of the units, and the regeneration and wash cycle times. Finally, after the system was designed for minimum liquid discharge, a method was given to balance the environmental and capital costs of the system with the operation costs. So that if there is flexibility in the discharge quantity, the discharge flow may be increased if this would result in a lower total system cost. This process is repeated for each water purification technology that is available for use. The minimum costs for each technology are compared and the end result is that the optimal technology and system is used. The case study illustrated the many insights that may be gathered about a process's behavior under many different operating conditions and its sensitivity to different system parameters. It also illustrated how the information from the optimization process could be used to balance operational robustness with the additional cost of sub-optimal operation.

5.2 Recommendations for Future Work

This work highlights several opportunities for further study that would greatly benefit the process integration field. First, and possibly most importantly, process optimization is often limited by the tools available. For process problems that involve stream mixing and splitting equations, nonlinear process models, and interactions between process intervals, the optimization programs generated are normally nonlinear, often utilizing integer or binary variables. While this ensures the process model is realistic, it creates many difficulties in the solution process since mixed integer nonlinear (MINLP) models are among the most difficult optimization problems to solve. The fact that there is no general method that may be used to solve these problems discourages the formulation of optimization procedures for these types of processes. As work continues in this area and better global optimization algorithms are developed, more general and detailed process optimization methodologies may be developed utilizing these advanced mathematical solution tools.

The process optimization methodologies introduced in this dissertation focused on various interception processes, but the ideas and concepts used are not specific to that type of operation. While the methods used attempted to optimize a process very generally and with few restrictions, they may be extended to even more general optimization methods. It is very conceivable to include reactors and the other components of chemical processes in the solution formulation. This is simply an extension of the process integration perspective. Single units should not be optimized because what is optimal for one unit very likely isn't optimal for the network as a whole. With the same logic, it would be better to optimize the entire chemical process: supply, reaction, refining, and waste treatment instead of one of these single parts of the process. Thus, it would be ideal to include every type of operation when optimizing a process. The approach presented here may be easily expanded to include many different types of units in the superstructures illustrated in the previous chapters. Such a formulation would necessarily be very complex, with many process variables such as flowrates, pressure, temperature, reaction extent, numerous different components, and varying technologies for each process operation. But this is the goal for process integration: to produce methods to optimize processes as generally as possible.

REFERENCES

Al-Khayyal, F.A. and Falk, J.E. (1983). Jointly constrained biconvex programming, *Mathematics of Operations Research*, **8**(2): 273-286.

Alle, A., Papageorgiou, L.G., and Pinto, J.M. (2004). A mathematical programming approach for cyclic production and cleaning scheduling of multistage continuous plants, *Computers and Chemical Engineering*, **28:** 3-15.

Cervantes, A.M., Wachter, A., Tutuncu, R.H., and Biegler, L.T. (2000). A reduced space interior point strategy for optimization of differential algebraic systems, *Computers and Chemical Engineering*, **24:** 39-51.

Clever, M., Jordt, F., Knauf, R., Rabiger, N., Rudebusch, M., and Hilker-Scheibel, R. (2000). Process water production from river water by ultrafiltration and reverse osmosis, *Desalination*, **131:** 325-336.

Dunn, R. F. and El-Halwagi, M.M. (2003). Process integration technology review: background and applications in the chemical process industry, *Journal of Chemical Technology and Biotechnology*, **78**: 1011-1021.

Eden, M.R., Jørgensen, S.B., Gani, R., and El-Halwagi, M.M. (2002). Property integration – a new approach for simultaneous solution of process and molecular design problems, *Computer Aided Chemical Eng*ineering, **10**: 79-84.

Eden, M.R., Jørgensen, S.B., Gani, R., and El-Halwagi, M.M. (2004). A novel framework for simultaneous separation process and product design, *Chemical Engineering and Processing*, **43**(5): 595-608.

Eden, M. R., Eljack, F.T., Abdelhady, A.F., Gabriel, F., Qin, X., and El-Halwagi, M.M. (2005). Targeting optimum resource allocation using reverse problem formulations and property clustering techniques, *Computers and Chemical Engineering*, **29**: 2304-2317.

El-Halwagi, M.M. and Manousiouthakis, V. (1990). Automatic synthesis of mass exchange networks with single-component targets, *Chemical Engineering Science*, **45**(9): 2813-2831.

El-Halwagi, M.M. (1997). *Pollution Prevention through Process Integration*. San Diego: Academic Press.

El-Halwagi, M.M., Gabriel, F., and Harell, D. (2003). Rigorous graphical targeting for resource conservation via material recycle/reuse networks, *Industrial & Engineering Chemistry Research*, **42**: 4319-4328.

El-Halwagi, M. M., I. M. Glasgow, M. R. Eden, and X. Qin. (2004). Property integration: componentless design techniques and visualization tools, *AIChE Journal*, **50**(8): 1854-1869.

Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., and Kallrath, J. (2005). Global optimization in the 21st century: advances and challenges, *Computers and Chemical Engineering*, **29**: 1185-1202.

Gabriel, F. B. and El-Halwagi, M.M. (2005). Simultaneous synthesis of waste interception and material reuse networks: problem reformulation for global optimization, *Environmental Progress*, **24**(2): 171-180.

Grooms, D., Kazantzi, V., and El-Halwagi, M.M. (2005). Optimal synthesis and scheduling of hybrid dynamic/steady-state property integration networks, *Computers and Chemical Engineering*, **29**: 2318-2325.

Grossmann, I.E. and Biegler, L.T. (2004). Part II. Future perspective on optimization, *Computers and Chemical Engineering*, **28:** 1193-1218.

Harell, D.A. (2004). Resource Conservation and Allocation via Process Integration. Ph.D. diss., Texas A&M University.

Ierapetritou, M. G. and Floudas, C.A. (1998a). Effective continuous-time formulation for short-term scheduling. Part 1. Multipurpose batch processes, *Industrial and Engineering Chemistry Research*, **37**: 4341-4359.

Ierapetritou, M. G. and Floudas, C.A. (1998b). Effective continuous-time formulation for short-term scheduling. Part 2. Continuous and semi-continuous processes, *Industrial and Engineering Chemistry Research*, **37**: 4360-4374.

Jain, V. and Grossmann, I. E. (1998). Cyclic scheduling of continuous parallel-process units with decaying performance, *AIChE Journal*, **44**(7): 1623-1636.

Jockenhovel, T., Biegler, L.T., and Wachter, A. (2003). Dynamic optimization of the Tennessee Eastman process using the OptControlCentre, *Computers and Chemical Engineering*, **27**(11): 1513-1531.

Kazantzi, V., Harell, D., Gabriel, F., Qin, X., and El-Halwagi, M.M. (2004a). Property-based integration for sustainable development, *Proceedings of European Symposium on Computer-Aided Process Engineering 14 (ESCAPE 14)*, Eds. A. Barbosa-Povoa and H. Matos, 1069-1074, Elsevier.

Kazantzi, V., Qin, X., Gabriel, F., Harell, D., and El-Halwagi, M.M. (2004b). Process modification through visualization and inclusion techniques for property based integration, *Proceedings of the Sixth Foundations of Computer Aided Design (FOCAPD)*, Eds. C. A. Floudas and R. Agrawal, 279-282, CACHE Corp.

Kazantzi, V. and El-Halwagi, M.M. (2005). Targeting material reuse via property integration, *Chemical Engineering Progress*, **101**(8): 28-37.

Kondili, E., Pantelides, C.C., and Sargent, R.W.H. (1993). A general algorithm for short-term scheduling of batch operations. I. MILP formulation, *Computers and Chemical Engineering*, ***17***: 211–227.

Lee, K.H., Park, H.I., and Lee, I.B. (2001). A novel nonuniform discrete time formulation for short-term scheduling of batch and continuous processes, *Industrial and Engineering Chemistry Research,* **40**: 4902-4911.

Li, Hebi, Crittenden, J.C., Mihelcic, J.R., and Hautakangas, H. (2002). Optimization of biofiltration for odor control: model development and parameter sensitivity, *Water Environment Research*, **74**(1): 5-16.

Manousiouthakis, V. and Martin, L. (2004). A minimum area targeting scheme for single component MEN and HEN synthesis, *Computers and Chemical Engineering*, **28**(8): 1237-1247.

Maravelias, C.T. and Grossman, I.E. (2003). Minimization of the makespan with a discrete-time state-task network formulation, *Industrial and Engineering Chemistry Research*, **42**: 6252-6257.

Maravelias, C.T. and Grossman, I.E. (2004). A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants, *Computers and Chemical Engineering*, **28**: 1921-1949.

McCormick, G.P. (1976). Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems, *Mathematical Programming,* **10**: 147-175.

Noureldin, M.B. and El-Halwagi, M.M. (2000). Pollution-prevention targets through integrated design and operation, *Computers and Chemical Engineering*, **24**: 1445-1453.

Pinto, J.M. and Grossmann, I.E. (1994). Optimal cyclic scheduling of multistage continuous multiproduct plants, *Computers and Chemical Engineering*, **18**(9): 797-816.

Qin, X., Gabriel, F., Harell, D., and El-Halwagi, M.M. (2004). Algebraic techniques for property integration via componentless design, *Industrial and Engineering Chemistry Research*, **43**: 3792-3798.

Quesada, I. and Grossmann, I.E. (1995). Global optimization of bilinear process networks with multicomponent flows, *Computers and Chemical Engineering*, **19**(12), 1219-1242.

Rich, S.H. and Prokopakis, G.J. (1987). Multiple routings and reaction paths in project scheduling, *Industrial & Engineering Chemistry Research*, **26**: 1940-1943.

Shah, N., Pantelides, C.C., & Sargent, R.W.H. (1993). A general algorithm for short-term scheduling of batch operations. II. Computational issues, *Computers and Chemical Engineering*, **17**: 229–244.

Sharratt, P.N. (1993). Waste minimisation for batchwise extraction processes, Institution of Chemical Engineers Symposium Series, **132** (*Effluent Treatment and Waste Minimisation*): 81-94.

Shelley, M.D. and El-Halwagi, M.M. (2000). Componentless design of recovery and allocation systems: a functionality-based clustering approach, *Computers and Chemical Engineering*, **24**: 2081-2091.

Soileau, C. S. (1998). Synthesis of Dynamic Mass Exchange Networks. M.S. Thesis, Auburn University.

Szitkai, Z., Farkas, T., Lelkes, Z., Rev, E., Fonyo, Z., and Kravanja, Z. (2006). Fairly linear mixed integer nonlinear programming model for the synthesis of mass exchange networks, *Industrial and Engineering Chemistry Research*, **45**: 236-244.

Tlili, M. M., Manzola, A.S., and Ben Amor, M. (2003). Optimization of the preliminary treatment in a desalination plant by reverse osmosis, *Desalination*, **156**: 69-78.

Wang, Y.P. and Smith, R. (1995). Time pinch analysis, *Chemical Engineering Research and Design*, **73**(8): 905-914.

Zhu, X.X., O'Neill, B.K., Roach, J.R., and Wood, R.M. (1995). Area-targeting methods for the direct synthesis of heat exchanger networks with unequal film coefficients, *Computers and Chemical Engineering*, **19**(2): 223-239.

APPENDIX A

MASS EXCHANGE NETWORK OPTIMIZATION LINGO CODE

A.1 Nonlinear Program

```
min=LCost+R1Cost+R2Cost+D1+D2+D3+1E11*devsum;

Data:
dt=10;
a1=3000;
b1=1E53;
m1=4E-6;
a2=3000;
b2=1E53;
m2=4E-6;
tf=4;
Cr=0.5;

xs=0;
m=8.4E-8;
eps=0.06;
Cj=10;

N=25;
V1=1;
V2=1;
V3=1;
EndData

Sets: Interval /1..N/: G1, G2, G3, G4, F11, F12, F13, F14, F21, F22,
F23, F24, F31, F32, F33, F34, F41, F42, F43, F44, z1in, z2in, z3in,
z4in, z1out, z2out, z3out, z4out, L, t1, t2, I1, I2, Int1, Int2, pos1,
neg1, pos2, neg2, C1out, C2out, C3out, C4out, C1in, C2in, C3in, C4in,
C11, C12, C13, C14, C21, C22, C23, C24, C31, C32, C33, C34, C41, C42,
C43, C44, IT1, IT2;
Endsets

@For( Interval(i): I1(i)=Int1(i)+pos1(i)-neg1(i) );
@For( Interval(i): I2(i)=Int2(i)+pos2(i)-neg2(i) );
devsum=@Sum( Interval(i): pos1+neg1+pos2+neg2 );

@For( Interval(i): @BIN(Int1(i)) );
@For( Interval(i): @BIN(Int2(i)) );
@For( Interval(i): z1in(i)<=1 );
@For( Interval(i): z2in(i)<=1 );

xt=0.19;
@For( Interval(i): G4(i)=10);
@For( Interval(i): z4out(i)=0.005 );
@For( Interval(i): z4in(i)=0.0002 );

@For( Interval(i): F11(i)=0 );
@For( Interval(i): F22(i)=0 );
@For( Interval(i): F33(i)=0 );
```

```
@For( Interval(i): G1(i)=F11(i)+F12(i)+F13(i)+F14(i) );
@For( Interval(i): G2(i)=F21(i)+F22(i)+F23(i)+F24(i) );
@For( Interval(i): G3(i)=F31(i)+F32(i)+F33(i)+F34(i) );
@For( Interval(i): G4(i)=F41(i)+F42(i)+F43(i)+F44(i) );

@For( Interval(i): C1out(i)=C11(i)+C12(i)+C13(i)+C14(i) );
@For( Interval(i): C2out(i)=C21(i)+C22(i)+C23(i)+C24(i) );
@For( Interval(i): C3out(i)=C31(i)+C32(i)+C33(i)+C34(i) );
@For( Interval(i): C4out(i)=C41(i)+C42(i)+C43(i)+C44(i) );

@For( Interval(i): G1(i)=F11(i)+F21(i)+F31(i)+F41(i) );
@For( Interval(i): G2(i)=F12(i)+F22(i)+F32(i)+F42(i) );
@For( Interval(i): G3(i)=F13(i)+F23(i)+F33(i)+F43(i) );
@For( Interval(i): G4(i)=F14(i)+F24(i)+F34(i)+F44(i) );

@For( Interval(i): C1in(i)=C11(i)+C21(i)+C31(i)+C41(i) );
@For( Interval(i): C2in(i)=C12(i)+C22(i)+C32(i)+C42(i) );
@For( Interval(i): C3in(i)=C13(i)+C23(i)+C33(i)+C43(i) );
@For( Interval(i): C4in(i)=C14(i)+C24(i)+C34(i)+C44(i) );

@For( Interval(i): C1out(i)=z1out(i)*G1(i) );
@For( Interval(i): C2out(i)=z2out(i)*G2(i) );
@For( Interval(i): C3out(i)=z3out(i)*G3(i) );
@For( Interval(i): C4out(i)=z4out(i)*G4(i) );

@For( Interval(i): C1in(i)=z1in(i)*G1(i) );
@For( Interval(i): C2in(i)=z2in(i)*G2(i) );
@For( Interval(i): C3in(i)=z3in(i)*G3(i) );
@For( Interval(i): C4in(i)=z4in(i)*G4(i) );

@For( Interval(i): C11(i)=z1out(i)*F11(i) );
@For( Interval(i): C12(i)=z1out(i)*F12(i) );
@For( Interval(i): C13(i)=z1out(i)*F13(i) );
@For( Interval(i): C14(i)=z1out(i)*F14(i) );
@For( Interval(i): C21(i)=z2out(i)*F21(i) );
@For( Interval(i): C22(i)=z2out(i)*F22(i) );
@For( Interval(i): C23(i)=z2out(i)*F23(i) );
@For( Interval(i): C24(i)=z2out(i)*F24(i) );
@For( Interval(i): C31(i)=z3out(i)*F31(i) );
@For( Interval(i): C32(i)=z3out(i)*F32(i) );
@For( Interval(i): C33(i)=z3out(i)*F33(i) );
@For( Interval(i): C34(i)=z3out(i)*F34(i) );
@For( Interval(i): C41(i)=z4out(i)*F41(i) );
@For( Interval(i): C42(i)=z4out(i)*F42(i) );
@For( Interval(i): C43(i)=z4out(i)*F43(i) );
@For( Interval(i): C44(i)=z4out(i)*F44(i) );

@For( Interval(i): C1in(i)-C1out(i)=L(i)*xt );

@For( Interval(i): z2out(i)=m1*t1(i) );
t1(1)=10;
I1(1)=1;
t1(N)=0;

@For( Interval(i) | i #GE# 2: t1(i)=IT1(i)+dt*I1(i) );
```

```
@For( Interval(i) | i #GE# 2: IT1(i)=t1(i-1)*I1(i) );

@For( Interval(i): z3out(i)=m2*t2(i) );
t2(1)=10;
I2(1)=1;
t2(N)=0;

@For( Interval(i) | i #GE# 2: t2(i)=IT2(i)+dt*I2(i) );

@For( Interval(i) | i #GE# 2: IT2(i)=t2(i-1)*I2(i) );

@For( Interval(i): G2(i)<=10*I1(i) );
@For( Interval(i): G3(i)<=10*I2(i) );

Rsum1=@SUM( Interval(i): (1-I1(i)) );
R1Cost=Rsum1*Cr;
Rsum2=@SUM( Interval(i): (1-I2(i)) );
R2Cost=Rsum2*Cr;
Lsum=@SUM( Interval(i): L(i) );
LCost=Lavg*Cj;
N*Lavg=Lsum;

@For( Interval(i): D1>=3*G1(i) );
@For( Interval(i): D2>=3*G2(i) );
@For( Interval(i): D3>=3*G3(i) );
```

A.2 Linearized Program

```
min=LCost+R1Cost+R2Cost+D1+D2+D3+1E11*devsum;

Data:
dt=10;
a1=3000;
b1=1E53;
m1=4E-6;
a2=3000;
b2=1E53;
m2=4E-6;
tf=4;
Cr=0.5;

xs=0;
m=8.4E-8;
eps=0.06;
Cj=10;

N=25;
V1=1;
V2=1;
V3=1;
EndData

Sets: Interval /1..N/: G1, G2, G3, G4, F11, F12, F13, F14, F21, F22,
F23, F24, F31, F32, F33, F34, F41, F42, F43, F44, z1in, z2in, z3in,
z4in, z1out, z2out, z3out, z4out, L, t1, t2, I1, I2, Int1, Int2, pos1,
neg1, pos2, neg2, C1out, C2out, C3out, C4out, C1in, C2in, C3in, C4in,
C11, C12, C13, C14, C21, C22, C23, C24, C31, C32, C33, C34, C41, C42,
C43, C44, IT1, IT2;
Endsets

@For( Interval(i): I1(i)=Int1(i)+pos1(i)-neg1(i) );
@For( Interval(i): I2(i)=Int2(i)+pos2(i)-neg2(i) );
devsum=@Sum( Interval(i): pos1+neg1+pos2+neg2 );

@For( Interval(i): @BIN(Int1(i)) );
@For( Interval(i): @BIN(Int2(i)) );
@For( Interval(i): z1in(i)<=1 );
@For( Interval(i): z2in(i)<=1 );

xt=0.19;
@For( Interval(i): G4(i)=10);
@For( Interval(i): z4out(i)=0.005 );
@For( Interval(i): z4in(i)=0.0002 );

z4outU = 0.005;
z4outL = 0.005;
z4inU = 0.002;
z4inL = 0.002;
G4U = 10;
G4L = 10;
zoutU = 0.5;
zoutL = 0;
zinU = 0.005;
zinL = 0.0002;
GU = 10;
GL = 0;
FU = 10;
FL = 0;
IL=0;
IU=1;
tL=0;
tU=240;

@For( Interval(i): F11(i)=0 );
@For( Interval(i): F22(i)=0 );
@For( Interval(i): F33(i)=0 );

@For( Interval(i): G1(i)=F11(i)+F12(i)+F13(i)+F14(i) );
@For( Interval(i): G2(i)=F21(i)+F22(i)+F23(i)+F24(i) );
@For( Interval(i): G3(i)=F31(i)+F32(i)+F33(i)+F34(i) );
@For( Interval(i): G4(i)=F41(i)+F42(i)+F43(i)+F44(i) );

@For( Interval(i): C1out(i)=C11(i)+C12(i)+C13(i)+C14(i) );
@For( Interval(i): C2out(i)=C21(i)+C22(i)+C23(i)+C24(i) );
@For( Interval(i): C3out(i)=C31(i)+C32(i)+C33(i)+C34(i) );
@For( Interval(i): C4out(i)=C41(i)+C42(i)+C43(i)+C44(i) );

@For( Interval(i): G1(i)=F11(i)+F21(i)+F31(i)+F41(i) );
@For( Interval(i): G2(i)=F12(i)+F22(i)+F32(i)+F42(i) );
@For( Interval(i): G3(i)=F13(i)+F23(i)+F33(i)+F43(i) );
@For( Interval(i): G4(i)=F14(i)+F24(i)+F34(i)+F44(i) );
```

```
@For( Interval(i): C1in(i)=C11(i)+C21(i)+C31(i)+C41(i) );
@For( Interval(i): C2in(i)=C12(i)+C22(i)+C32(i)+C42(i) );
@For( Interval(i): C3in(i)=C13(i)+C23(i)+C33(i)+C43(i) );
@For( Interval(i): C4in(i)=C14(i)+C24(i)+C34(i)+C44(i) );

@For( Interval(i): C11(i)>=z1out(i)*FL+zoutL*F11(i)-zoutL*FL );
@For( Interval(i): C11(i)>=z1out(i)*FU+zoutU*F11(i)-zoutU*FU );
@For( Interval(i): C11(i)<=z1out(i)*FU+zoutL*F11(i)-zoutL*FU );
@For( Interval(i): C11(i)<=z1out(i)*FL+zoutU*F11(i)-zoutU*FL );
@For( Interval(i): C12(i)>=z1out(i)*FL+zoutL*F12(i)-zoutL*FL );
@For( Interval(i): C12(i)>=z1out(i)*FU+zoutU*F12(i)-zoutU*FU );
@For( Interval(i): C12(i)<=z1out(i)*FU+zoutL*F12(i)-zoutL*FU );
@For( Interval(i): C12(i)<=z1out(i)*FL+zoutU*F12(i)-zoutU*FL );
@For( Interval(i): C13(i)>=z1out(i)*FL+zoutL*F13(i)-zoutL*FL );
@For( Interval(i): C13(i)>=z1out(i)*FU+zoutU*F13(i)-zoutU*FU );
@For( Interval(i): C13(i)<=z1out(i)*FU+zoutL*F13(i)-zoutL*FU );
@For( Interval(i): C13(i)<=z1out(i)*FL+zoutU*F13(i)-zoutU*FL );
@For( Interval(i): C14(i)>=z1out(i)*FL+zoutL*F14(i)-zoutL*FL );
@For( Interval(i): C14(i)>=z1out(i)*FU+zoutU*F14(i)-zoutU*FU );
@For( Interval(i): C14(i)<=z1out(i)*FU+zoutL*F14(i)-zoutL*FU );
@For( Interval(i): C14(i)<=z1out(i)*FL+zoutU*F14(i)-zoutU*FL );
@For( Interval(i): C21(i)>=z2out(i)*FL+zoutL*F21(i)-zoutL*FL );
@For( Interval(i): C21(i)>=z2out(i)*FU+zoutU*F21(i)-zoutU*FU );
@For( Interval(i): C21(i)<=z2out(i)*FU+zoutL*F21(i)-zoutL*FU );
@For( Interval(i): C21(i)<=z2out(i)*FL+zoutU*F21(i)-zoutU*FL );
@For( Interval(i): C22(i)>=z2out(i)*FL+zoutL*F22(i)-zoutL*FL );
@For( Interval(i): C22(i)>=z2out(i)*FU+zoutU*F22(i)-zoutU*FU );
@For( Interval(i): C22(i)<=z2out(i)*FU+zoutL*F22(i)-zoutL*FU );
@For( Interval(i): C22(i)<=z2out(i)*FL+zoutU*F22(i)-zoutU*FL );
@For( Interval(i): C23(i)>=z2out(i)*FL+zoutL*F23(i)-zoutL*FL );
@For( Interval(i): C23(i)>=z2out(i)*FU+zoutU*F23(i)-zoutU*FU );
@For( Interval(i): C23(i)<=z2out(i)*FU+zoutL*F23(i)-zoutL*FU );
@For( Interval(i): C23(i)<=z2out(i)*FL+zoutU*F23(i)-zoutU*FL );
@For( Interval(i): C24(i)>=z2out(i)*FL+zoutL*F24(i)-zoutL*FL );
@For( Interval(i): C24(i)>=z2out(i)*FU+zoutU*F24(i)-zoutU*FU );
@For( Interval(i): C24(i)<=z2out(i)*FU+zoutL*F24(i)-zoutL*FU );
@For( Interval(i): C24(i)<=z2out(i)*FL+zoutU*F24(i)-zoutU*FL );
@For( Interval(i): C31(i)>=z3out(i)*FL+zoutL*F31(i)-zoutL*FL );
@For( Interval(i): C31(i)>=z3out(i)*FU+zoutU*F31(i)-zoutU*FU );
@For( Interval(i): C31(i)<=z3out(i)*FU+zoutL*F31(i)-zoutL*FU );
@For( Interval(i): C32(i)<=z3out(i)*FL+zoutU*F31(i)-zoutU*FL );
@For( Interval(i): C32(i)>=z3out(i)*FL+zoutL*F32(i)-zoutL*FL );
@For( Interval(i): C32(i)>=z3out(i)*FU+zoutU*F32(i)-zoutU*FU );
@For( Interval(i): C32(i)<=z3out(i)*FU+zoutL*F32(i)-zoutL*FU );
@For( Interval(i): C32(i)<=z3out(i)*FL+zoutU*F32(i)-zoutU*FL );
@For( Interval(i): C33(i)>=z3out(i)*FL+zoutL*F33(i)-zoutL*FL );
@For( Interval(i): C33(i)>=z3out(i)*FU+zoutU*F33(i)-zoutU*FU );
@For( Interval(i): C33(i)<=z3out(i)*FU+zoutL*F33(i)-zoutL*FU );
@For( Interval(i): C33(i)<=z3out(i)*FL+zoutU*F33(i)-zoutU*FL );
@For( Interval(i): C34(i)>=z3out(i)*FL+zoutL*F34(i)-zoutL*FL );
@For( Interval(i): C34(i)>=z3out(i)*FU+zoutU*F34(i)-zoutU*FU );
@For( Interval(i): C34(i)<=z3out(i)*FU+zoutL*F34(i)-zoutL*FU );
@For( Interval(i): C34(i)<=z3out(i)*FL+zoutU*F34(i)-zoutU*FL );
@For( Interval(i): C41(i)>=z4out(i)*FL+z4outL*F41(i)-z4outL*FL );
@For( Interval(i): C41(i)>=z4out(i)*FU+z4outU*F41(i)-z4outU*FU );
@For( Interval(i): C41(i)<=z4out(i)*FU+z4outL*F41(i)-z4outL*FU );
```

```
@For( Interval(i): C41(i)<=z4out(i)*FL+z4outU*F41(i)-z4outU*FL );
@For( Interval(i): C42(i)>=z4out(i)*FL+z4outL*F42(i)-z4outL*FL );
@For( Interval(i): C42(i)>=z4out(i)*FU+z4outU*F42(i)-z4outU*FU );
@For( Interval(i): C42(i)<=z4out(i)*FU+z4outL*F42(i)-z4outL*FU );
@For( Interval(i): C42(i)<=z4out(i)*FL+z4outU*F42(i)-z4outU*FL );
@For( Interval(i): C43(i)>=z4out(i)*FL+z4outL*F43(i)-z4outL*FL );
@For( Interval(i): C43(i)>=z4out(i)*FU+z4outU*F43(i)-z4outU*FU );
@For( Interval(i): C43(i)<=z4out(i)*FU+z4outL*F43(i)-z4outL*FU );
@For( Interval(i): C43(i)<=z4out(i)*FL+z4outU*F43(i)-z4outU*FL );
@For( Interval(i): C44(i)>=z4out(i)*FL+z4outL*F44(i)-z4outL*FL );
@For( Interval(i): C44(i)>=z4out(i)*FU+z4outU*F44(i)-z4outU*FU );
@For( Interval(i): C44(i)<=z4out(i)*FU+z4outL*F44(i)-z4outL*FU );
@For( Interval(i): C44(i)<=z4out(i)*FL+z4outU*F44(i)-z4outU*FL );

@For( Interval(i): C1out(i)>=z1out(i)*GL+zoutL*G1(i)-zoutL*GL );
@For( Interval(i): C2out(i)>=z2out(i)*GL+zoutL*G2(i)-zoutL*GL );
@For( Interval(i): C3out(i)>=z3out(i)*GL+zoutL*G3(i)-zoutL*GL );
@For( Interval(i): C4out(i)>=z4out(i)*G4L+z4outL*G4(i)-z4outL*G4L );
@For( Interval(i): C1out(i)>=z1out(i)*GU+zoutU*G1(i)-zoutU*GU );
@For( Interval(i): C2out(i)>=z2out(i)*GU+zoutU*G2(i)-zoutU*GU );
@For( Interval(i): C3out(i)>=z3out(i)*GU+zoutU*G3(i)-zoutU*GU );
@For( Interval(i): C4out(i)>=z4out(i)*G4U+z4outU*G4(i)-z4outU*G4U );
@For( Interval(i): C1out(i)<=z1out(i)*GU+zoutL*G1(i)-zoutL*GU );
@For( Interval(i): C2out(i)<=z2out(i)*GU+zoutL*G2(i)-zoutL*GU );
@For( Interval(i): C3out(i)<=z3out(i)*GU+zoutL*G3(i)-zoutL*GU );
@For( Interval(i): C4out(i)<=z4out(i)*G4U+z4outL*G4(i)-z4outL*G4U );
@For( Interval(i): C1out(i)<=z1out(i)*GL+zoutU*G1(i)-zoutU*GL );
@For( Interval(i): C2out(i)<=z2out(i)*GL+zoutU*G2(i)-zoutU*GL );
@For( Interval(i): C3out(i)<=z3out(i)*GL+zoutU*G3(i)-zoutU*GL );
@For( Interval(i): C4out(i)<=z4out(i)*G4L+z4outU*G4(i)-z4outU*G4L );

@For( Interval(i): C1in(i)>=z1in(i)*GL+zinL*G1(i)-zinL*GL );
@For( Interval(i): C2in(i)>=z2in(i)*GL+zinL*G2(i)-zinL*GL );
@For( Interval(i): C3in(i)>=z3in(i)*GL+zinL*G3(i)-zinL*GL );
@For( Interval(i): C4in(i)>=z4in(i)*G4L+z4inL*G4(i)-z4inL*G4L );
@For( Interval(i): C1in(i)>=z1in(i)*GU+zinU*G1(i)-zinU*GU );
@For( Interval(i): C2in(i)>=z2in(i)*GU+zinU*G2(i)-zinU*GU );
@For( Interval(i): C3in(i)>=z3in(i)*GU+zinU*G3(i)-zinU*GU );
@For( Interval(i): C4in(i)>=z4in(i)*G4U+z4inU*G4(i)-z4inU*G4U );
@For( Interval(i): C1in(i)<=z1in(i)*GU+zinL*G1(i)-zinL*GU );
@For( Interval(i): C2in(i)<=z2in(i)*GU+zinL*G2(i)-zinL*GU );
@For( Interval(i): C3in(i)<=z3in(i)*GU+zinL*G3(i)-zinL*GU );
@For( Interval(i): C4in(i)<=z4in(i)*G4U+z4inL*G4(i)-z4inL*G4U );
@For( Interval(i): C1in(i)<=z1in(i)*GL+zinU*G1(i)-zinU*GL );
@For( Interval(i): C2in(i)<=z2in(i)*GL+zinU*G2(i)-zinU*GL );
@For( Interval(i): C3in(i)<=z3in(i)*GL+zinU*G3(i)-zinU*GL );
@For( Interval(i): C4in(i)<=z4in(i)*G4L+z4inU*G4(i)-z4inU*G4L );

@For( Interval(i): C1in(i)-C1out(i)=L(i)*xt );

@For( Interval(i): z2out(i)=m1*t1(i) );
t1(1)=10;
I1(1)=1;
t1(N)=0;

@For( Interval(i) | i #GE# 2: t1(i)=IT1(i)+dt*I1(i) );
```

```
@For( Interval(i) | i #GE# 2: IT1(i)>=t1(i-1)*IL+tL*I1(i)-tL*IL );
@For( Interval(i) | i #GE# 2: IT1(i)>=t1(i-1)*IU+tU*I1(i)-tU*IU );
@For( Interval(i) | i #GE# 2: IT1(i)<=t1(i-1)*IU+tL*I1(i)-tL*IU );
@For( Interval(i) | i #GE# 2: IT1(i)<=t1(i-1)*IL+tU*I1(i)-tU*IL );

@For( Interval(i): z3out(i)=m2*t2(i) );
t2(1)=10;
I2(1)=1;
t2(N)=0;

@For( Interval(i) | i #GE# 2: t2(i)=IT2(i)+dt*I2(i) );

@For( Interval(i) | i #GE# 2: IT2(i)>=t2(i-1)*IL+tL*I2(i)-tL*IL );
@For( Interval(i) | i #GE# 2: IT2(i)>=t2(i-1)*IU+tU*I2(i)-tU*IU );
@For( Interval(i) | i #GE# 2: IT2(i)<=t2(i-1)*IU+tL*I2(i)-tL*IU );
@For( Interval(i) | i #GE# 2: IT2(i)<=t2(i-1)*IL+tU*I2(i)-tU*IL );

@For( Interval(i): G2(i)<=10*I1(i) );
@For( Interval(i): G3(i)<=10*I2(i) );

Rsum1=@SUM( Interval(i): (1-I1(i)) );
R1Cost=Rsum1*Cr;
Rsum2=@SUM( Interval(i): (1-I2(i)) );
R2Cost=Rsum2*Cr;
Lsum=@SUM( Interval(i): L(i) );
LCost=Lavg*Cj;
N*Lavg=Lsum;

@For( Interval(i): D1>=3*G1(i) );
@For( Interval(i): D2>=3*G2(i) );
@For( Interval(i): D3>=3*G3(i) );
```

APPENDIX B

PROPERTY INTERCEPTION NETWORK OPTIMIZATION LINGO CODE

```
min=LCost+R1Cost+R2Cost+0.1*(D1+H1^0.9)+0.1*(D2+H2^0.9)+0.1*(D3+H3^0.9)
;

Data:
dt=10;
a2=1.5E-4;
b2=6;
c2=0.08;
a3=1.5E-4;
b3=6;
c3=0.08;
tf=4;
Cr=0.5;

xs=0;
m=8.4E-8;
eps=0.06;
Cj=100;

N=25;
V1=1;
V2=1;
V3=1;
U1=10;
L1=5;
U2=5;
L2=0;
U3=3;
L3=0;
EndData

Sets: Interval /1..N/: G1, G2, G3, G4, F12, F13, F14, F21, F23, F24,
F31, F32, F34, F41, F42, F43, z1in, z2in, z3in, z4in, z1out, z2out,
z3out, z4out, L, xt, t2, t3, yLM1, Regen2, Regen3, Run2, Run3;
Endsets

@For( Interval(i): @BIN(Regen2(i)) );
@For( Interval(i): @BIN(Regen3(i)) );
@For( Interval(i): @BIN(Run2(i)) );
@For( Interval(i): @BIN(Run3(i)) );
@For( Interval(i): z1in(i)<=1 );
@For( Interval(i): z2in(i)<=1 );

@For( Interval(i): xt(i)<=0.19 );
@For( Interval(i): G4(i)=10);
@For( Interval(i): z4out(i)=0.005 );
@For( Interval(i): z4in(i)=0.0002 );

@For( Interval(i): G1(i)=F12(i)+F13(i)+F14(i) );
```

```
@For( Interval(i): G2(i)=F21(i)+F23(i)+F24(i) );
@For( Interval(i): G3(i)=F31(i)+F32(i)+F34(i) );
@For( Interval(i): G4(i)=F41(i)+F42(i)+F43(i) );

@For( Interval(i): G1(i)=F21(i)+F31(i)+F41(i) );
@For( Interval(i): G2(i)=F12(i)+F32(i)+F42(i) );
@For( Interval(i): G3(i)=F13(i)+F23(i)+F43(i) );
@For( Interval(i): G4(i)=F14(i)+F24(i)+F34(i) );

@For( Interval(i):
z1in(i)*G1(i)=z2out(i)*F21(i)+z3out(i)*F31(i)+z4out(i)*F41(i) );
@For( Interval(i):
z2in(i)*G2(i)=z1out(i)*F12(i)+z3out(i)*F32(i)+z4out(i)*F42(i) );
@For( Interval(i):
z3in(i)*G3(i)=z1out(i)*F13(i)+z2out(i)*F23(i)+z4out(i)*F43(i) );
@For( Interval(i):
z4in(i)*G4(i)=z1out(i)*F14(i)+z2out(i)*F24(i)+z3out(i)*F34(i) );

@For( Interval(i): 0=G1(i)*(z1in(i)-z1out(i))-L(i)*xt(i) );

!@For( Interval(i): G2(i)<=(1/4*3.14*D2^2)*V2 );
!@For( Interval(i): G1(i)>=L1 );
@For( Interval(i): G2(i)=U2*(1-Regen2(i))*Run2(i) );
!@For( Interval(i): F24(i)>=L2*(1-Regen2(i))*Run2(i) );
@For( Interval(i): G3(i)=U3*(1-Regen3(i))*Run3(i) );
!@For( Interval(i): F34(i)>=L3*(1-Regen3(i))*Run3(i) );

@For( Interval(i): z2out(i)=a2*(2-2/(@EXP(2*(c2*t2(i)-b2))+1))*(1-
Regen2(i))+z2in(i)*Regen2(i) );
t2(1)=10;
Regen2(1)=0;
@For( Interval(i) | i #NE# 1: t2(i)=(t2(i-1)+dt*Run2(i))*(1-Regen2(i))
);
t2(N)=0;
@For( Interval(i): Run2(i)<=1-Regen2(i) );

@For( Interval(i): z3out(i)=a3*(2-2/(@EXP(2*(c3*t3(i)-b3))+1))*(1-
Regen3(i))+z3in(i)*Regen3(i) );
t3(1)=10;
Regen3(1)=0;
@For( Interval(i) | i #NE# 1: t3(i)=(t3(i-1)+dt*Run3(i))*(1-Regen3(i))
);
t3(N)=0;
@For( Interval(i): 0=1-Run3(i)-Regen3(i) );

Rsum1=@SUM( Interval(i): Regen2(i) );
R1Cost=Rsum1*Cr;
Rsum2=@SUM( Interval(i): Regen3(i) );
R2Cost=Rsum2*Cr;
Lsum=@SUM( Interval(i): L(i) );
LCost=Lavg*Cj;
N*Lavg=Lsum;

@For( Interval(i): H1*yLM1(i)+z1out(i)=z1in(i) );
@For( Interval(i): yLM1(i)^3=1/2*(z1in(i)-
m*xt(i))*z1out(i)^2+1/2*(z1in(i)-m*xt(i))^2*z1out(i) );
@For( Interval(i): H2=0.5*@max(Interval: G2) );
```

```
@For( Interval(i): H3=0.5*@max(Interval: G3) );

D1=(4/3.14*@max(Interval: G1)/V1)^(1/2);
D2=(4/3.14*@max(Interval: G2)/V2)^(1/2);
D3=(4/3.14*@max(Interval: G3)/V3)^(1/2);
```

VITA

Daniel Douglas Grooms received his Bachelor of Science degree in chemical engineering from Lamar University in May, 2002. He entered the Ph.D. program at Texas A&M University in August of the same year. He published a paper, “Optimal synthesis and scheduling of hybrid dynamic/steady-state property integration networks,” in *Computers and Chemical Engineering* in 2005.

Mr. Grooms may be reached at:

Daniel Grooms
Invista
P.O. Box 1003
Orange, TX 77631-1003