WSRC-MS-99-00489

AN OPTIMIZATION SCHEME FOR HIGH LEVEL WASTE REMOVAL SEQUENCING USING ProdMod

RADIOACTIVE WASTE MANAGEMENT AND DISPOSAL

KEYWORDS: *high level waste, optimization, Savannah River Site*

P. K. PAUL and M. V. GREGORY
Westinghouse Savannah River Company
Aiken, South Carolina 29808

T. ALDEMIR
The Ohio State University
206 West 18th Avenue
Columbus, OH 43210

Abstract

A general purpose dynamic optimization scheme suitable for high level waste (HLW) complex operations has been developed. The optimizer is interfaced with the SPEEDUP™ based dynamic simulator ProdMod for flow of information between the optimizer and simulator, while the optimization is performed in the stand-alone FORTRAN based optimizer. The linear constructs and the mapping algorithm of the ProdMod have been used in the optimization scheme for the interface. The optimization scheme has been successfully implemented in generating waste blending batch sequences for one of salt processing options at the Savannah River Site (SRS) HLW complex. Parametric studies demonstrate that the devised optimization scheme is a realistic approach for guiding the operation of the HLW complexes.

I. Introduction

The United States Department of Energy (DOE) manages millions of cubic meters of nuclear waste that has been generated during the last 50 years of operations mostly related to the nation's defense activities. The total radioactivity of the DOE managed waste is about 1 billion curies, with 94% of the activity attributed to high level waste. The defense high level waste is located at the Hanford Site, the Idaho National Engineering and Environmental Laboratory (INEEL), and the Savannah River Site (SRS). The SRS is the first site in the United States where processing of wastes into glass form is taking place. At the SRS, more than 120,000 cubic meters of high level waste is stored with half a billion curies activity. The processing of high level waste into borosilicate glass for final disposal form began in 1996, and it is expected that it will take several decades to process all the wastes. Many waste processing operations are underway at different facilities at this site, and some new operations are in the process of coming on-line soon. Waste processing operations involve many highly interrelated facilities, each facility performing specific process functions. A state of the art computational tool can usefully guide the decades long process operations in an efficient manner. In support of the SRS waste processing operations in an efficient manner, a fast computational tool named Production Planning Model (ProdMod)[1,2,3] has been developed to simulate the dynamic response of all facilities at the SRS HLW complex in an integrated form for the entire life span.

Based on the input conditions provided by the user, the ProdMod simulator models SRS HLW complex for the next 30 years of operation, all the way to clean up of all waste storage tanks. To generate an optimal scenario, ProdMod needs optimal input parameters to be provided by an optimizer based on an objective function and the associated constraint equations. The HLW planners may consider one or several objectives during the entire range of operation. The objectives may be: 1) maximize the tank space in the early years of operations; 2) maximize the glass production rate or minimize the waste production rate; 3) minimize the time of operation; 4) minimize the cost of operation, etc. The operation of the entire HLW complex for many decades is a significantly complicated process. The operating, budgetary, and regulatory constraints and priorities dictate the nature of operation of the waste complex. These constraints and priorities are not known for all of the processing years in the future. The constraints and priorities are expected to change frequently. As a result, the waste processing operation will have to follow these changes from time to time. The optimized parameters for a given set of decision variables and objective function based on multi-year optimization results may not be applicable beyond a certain period of operation if the constraints and priorities are changed. The next phase of operation will always depend on the current operating, budgetary, and regulatory conditions. The planners have to generate optimal operating scenarios many times throughout the entire waste processing period. In general the objective function will depend on hundreds of parameters for the entire waste complex, besides the operating, regulatory, and budgetary constraints.

A general purpose dynamic optimization scheme has been devised in conjunction with the ProdMod simulator. The optimization scheme is suitable for the HLW complex operations and is able to handle different types of optimizations such as linear, nonlinear, dynamic, and stochastic. The optimization is performed in the stand-alone FORTRAN based optimizer, while the optimizer is interfaced with the SPEEDUP[4] based simulator for flow of information between the two. The scheme is suitable for generating optimal parameters for batch (discrete event) or continuous operation for a single or multiple optimization spans to cover the entire range of optimization. The ProdMod simulator has been adapted to simulate the waste processing operations using the optimal inputs provided by the optimizer for each event or time horizon, and then to generate information for the next stage of optimization, and so on.

A brief description of the SRS HLW waste complex and an overview of the ProdMod simulation of the SRS HLW complex are provided in Section II. Section III provides the algorithm for the general purpose dynamic optimization scheme. The waste removal sequencing for precipitate production in the salt processing operation is used to illustrate the implementation of the optimization algorithm. The operating rules and priorities of the waste removal sequencing of the example system and their implementations are described in Sections IV and V. The optimizer developed to perform the dynamic optimization is described briefly in Section VI. The problem is solved as a batch optimization problem as opposed to global optimization. Section VII justifies the use of the batch optimization. The results of parametric studies are summarized in Section VIII, followed by a conclusion in Section IX.

II. Background

The SRS has accumulated 127,000 m^3 (33.4 million gallons) of HLW in its 51 storage tanks from 40+ years' operation in support of the nation's nuclear defense programs. About 15.4 million gallons of the waste is in salt solution form called supernate, containing a variety of salts dominated by $NaNO_3$, NaOH, $NaNO_2$, and KNO_3. The waste in crystalline solid form called saltcake, created from supernate by evaporating excess water, accounts for 14.5 million gallons of the waste. Virtually all the radioactivity in the supernate and saltcake is Cs^{137}. The remaining 3.5 million gallons of waste is called sludge, in which suspended insoluble solid particles in a supernate layer are settled at the bottom of the tank. The sludge is mainly composed of oxides and hydroxides of Al, Fe, Ni, and Mn. Most of the radioactivity in the sludge is Sr^{90}.

The SRS HLW complex consists of several large facilities producing each of the final waste forms, plus other front-end and interface processes linking the facilities.[5,6] Figure 1[6] illustrates the operations with the processes indicated in boxes and the coupling connections between the processes as numbered principal streams. The incoming HLW (Stream 1), consisting of supernate and sludge, is stored in the HLW storage and evaporation facility (Tank

Farm). The Tank Farm safely stores these wastes until downstream facilities are available for further processing. Most of the supernate is evaporated to solid saltcake to reduce its volume and mobility. The decontaminated overheads from the evaporators combined with overheads from the separation processes and other low-level streams are sent to the Effluent Treatment Facility (ETF) (Stream 13). The sludges (Stream 2) are transferred to the Extended Sludge Processing (ESP) Facility via hydraulic slurrying techniques. In ESP, sludges with high aluminum content are processed to remove part of the insoluble aluminum compounds. All sludges are then washed with water to reduce their soluble salt contents. The spent wash water (Stream 3) is sent back to the Tank Farm, while the washed sludge (Stream 4) is sent to Vitrification. The sodium, potassium, and cesium rich crystallized saltcake is dissolved with water using hydraulic slurrying techniques. The supernate and/or dissolved salt (Stream 5) from different tanks are sent to the salt processing operation. Several salt processing options are under study at SRS. One of them is separation of the salt solution after precipitation and filtration steps into a very highly radioactive concentrated slurry (Stream 7) for vitrification and a very low radioactive filtrate (Stream 6) for solidification. This salt processing option is only discussed in this paper. In vitrification at the Defense Waste Processing Facility (DWPF), the precipitate is catalytically decomposed and separated into two streams: a mildly contaminated organic stream (Stream 11) which is sent to the Consolidated Incineration Facility (CIF) for organic waste destruction by burning; and an aqueous stream containing virtually all of the radionuclides which is combined with the washed sludge from ESP and sent to the glass melter. In DWPF, the precipitate is further washed to reduce the level of sodium nitrate and nitrite which had been added to inhibit tank corrosion to the acceptable levels of the glass melter. The spent wash water (Stream 8) from this process is sent back to the salt process for reuse. The washed sludge from ESP is chemically adjusted, stripping out a significant amount of mercury (Stream 12). The precipitate and sludge are mixed with glass frit and sent to the melter where the wastes are melted into a borosilicate glass matrix which is poured into a stainless steel canister (Stream 9). These canisters are temporarily stored in the Glass Waste Storage Building at SRS, awaiting for permanent disposal at the Federal Repository once it is built. The water vapor driven off from the melter along with other aqueous streams generated during vitrification are sent back to the Tank Farm for further processing (Stream 10). In ETF, the contaminants are separated and sent back to salt processing (Stream 15) where they are mixed with the decontaminated salt solution from salt processing. This mixture of LLW is then sent to the Saltstone Facility (Stream 6) where it is mixed with cement former and pumped as a wet grout to a Saltstone Vault (Stream 16). In the Saltstone Vault, the grout hydrates and cures, with the vault eventually closed as a landfill. The treated waste water (Stream 14) from the ETF flows to the Outfall (the site water environment).

The SRS HLW complex represents a multi-decade mixed operation of continuous and batch processes. A computational tool called the Production Planning Model (ProdMod) models the entire HLW complex and simulates 29 years of operation (all the way to clean up of all the tanks). Ref. 1 describes ProdMod's modeling methodologies as applied to the SRS HLW complex. ProdMod has been designed as a pseudo-dynamic simulation code based on Aspen Technology's SPEEDUP™ software development package.[4] The dynamic nature of each plant process is cast in a linear form in which the time dependence is implicit. Another innovative approach implemented in ProdMod is mapping the event-space with the time-space and vice versa. That is, successive equations in the equation set can describe successive time intervals (i.e., "time-space") for continuous process, or successive events (i.e., "event-space") for process best described as discrete events. The mapping efficiently couples the continuous time-space simulation with the discrete event-space simulation. SPEEDUP treats all the interconnected processes as a single large process described by *n* variables, the dynamic behavior of which is represented by the following form:

$$\boldsymbol{F(x)} = \mathbf{0} \tag{1}$$
$$\boldsymbol{x} = [x_1, x_2, x_3, \ldots\ldots\ldots, x_n] \tag{2}$$

where, $\boldsymbol{x}$ is a real vector of dimension *n* and $\boldsymbol{F}$ is a real valued vector function of dimension *m* ($m \leq n$) which describes the process. The values of *m* and *n* depend on the physical model of the HLW complex. In order to solve Eq. (1) as a set of algebraic equations, a set of $n - m$ specifications of the form:

$$\boldsymbol{Mx} = \boldsymbol{w} \tag{3}$$

is imposed on Eq. (1) based on the physical nature of the complex being modeled. Here, $\boldsymbol{M}$ is a $(n - m)$ x n matrix such that its elements M_{ij} have the properties

$$M_{ij} = \{0,1\} \tag{4}$$
$$\sum_{j=1}^{n} M_{ij} = 1 \qquad \text{for} \quad i = 1, 2, 3, \ldots\ldots\ldots\ldots\ldots\ldots, n - m \tag{5}$$

and $\boldsymbol{w}$ is a real vector of dimension $n - m$.

In Eq. (1), the vector $\boldsymbol{x}$ comprises all variables over the entire time span of simulation, and the vector $\boldsymbol{F}$ comprises all the linear and nonlinear algebraic equations which describe the processes. The matrix $\boldsymbol{M}$ represents the $(n\text{-}m)$ known variables whose values are specified in single variable equation form, and $\boldsymbol{w}$ represents the column matrix of $n\text{-}m$ assigned values. SPEEDUP solves the process equations given by Eq. (1) in ProdMod in two steps. First, the equations are blocked into independently and sequentially solvable blocks by a block decomposition method. Then, the independently solvable blocks are solved by calling sparse linear and nonlinear solvers. ProdMod simulates the dynamic behavior of the SRS system for a specified set of input parameters. If the input parameters represent the optimal conditions for a process, ProdMod simulates the optimal behavior for that process.

III. Optimization Algorithm

The optimization scheme has been coupled to but performs its optimization outside the ProdMod simulator. A stand-alone optimizer (see Section VII) written in FORTRAN generates the optimized parameters for an optimization problem. The optimizer is interfaced with the simulator for the transfer of : a) necessary initial and simulated variables from the simulator to the optimizer, and, b) optimized parameters and other process control parameters from the optimizer to the simulator. The constraint equations and the objective function along with an appropriate optimization routine for a given optimization problem are incorporated in the optimizer. The optimizer then calculates the parameters of the constraint equations and the objective function, and generates optimized parameters by calling the optimization routine. The optimized parameters are passed along with other necessary variables to the simulator for the next stage of simulation. The flow of information between the simulator and the optimizer can be repetitive depending on the nature of the optimization problem.

The interface between the optimizer and the SPEEDUP based ProdMod simulator is schematically shown in Fig. 2. The exchange of information between the ProdMod simulator and the FORTRAN based optimizer is performed using SPEEDUP's EXTERNAL section and External Data Interface (EDI) feature. The ProdMod simulator generates a data file for the optimizer and reads the parameter file generated by the optimizer along with other necessary user input files. The EDI consists of a set of FORTRAN routines, executed in synchronization with the SPEEDUP simulation, which are used to read (using GET routine) or write (using PUT routine) data from or to a file plus open and close the files. The variables whose values are to be passed to the simulator from the optimizer via a data file are listed by name in the RECEIVE sub-section of the EXTERNAL section. Similarly, the TRANSMIT sub-section in the EXTERNAL lists the variable names whose values are to be passed from the simulator to the optimizer. The known variables are initialized by the user in the OPERATION section of ProdMod, and the unknown variables are calculated by solving equations specified in the ProdMod. The EDI feature supersedes an initial value specified in the OPERATION section by reading the value of the variable, listed in the RECEIVE sub-section, from a file. The OPERATION section's initial values are not superseded if the EDI's read feature remains inactive, although the RECEIVE sub-section is active. The ability to define the interface simply by listing the variable set is a convenient feature of SPEEDUP.

In order to simulate discrete events and continuous processes, ProdMod is constructed to advance simultaneously in both event-space and time-space. The optimization scheme is able to optimize parameters in both spaces. To optimize parameters in event-space (e.g., within batches and cycles in batch processing), the optimizer extracts the variables in event-space and the corresponding time-space necessary for generating optimized parameters over an event interval or over a number of event intervals. These optimized parameters are then used by the simulator corresponding to the particular interval or intervals in event-space. For example, to optimize waste blending for a particular salt processing batch, the optimizer needs to extract the information regarding the contents of waste in each tank prior to that batch, precipitate demand volume for that batch, along with other variables from the simulator. The waste contents in each tank are time-space variables, while batch precipitate demand volume and the optimized waste removal from a tank in a batch are event-space variables. The same principle is applicable for parameter optimization in time-space for continuous processes. In ProdMod, the initial input parameters are provided in the OPERATION section in hard coded format and these input parameters are EDI GET (see Fig. 2). So, the optimizer can start optimization process from any point in event- and time-space horizon, leaving the previous input condition as is. Using the same principle, the optimization process can be restarted for future extensions of the event- and time- horizon, leaving the previous optimized status intact provided the data interface files between the simulator and the optimizer are not disturbed.

The time- or event-span over which the optimization is performed depends on the constraints and objective function. To perform optimization over a desired span consisting of a sequence of several optimization spans, the optimization and the advancement of the simulation progress from one span to the next until reaching the final point. The ProdMod simulator has been developed based on linear constructs[1], and it may be possible to construct most optimization problems in a linear form under some approximations. But, for certain cases a fundamentally nonlinear form in the constraint equations or objective function may be unavoidable. In those situations, a user provided nonlinear optimization routine has to be incorporated in the optimizer to solve the problem.

Figure 3 shows the optimization algorithm in flowchart form. The first step is to create a file containing the invariant parameters used in the model. An initial file 'initial.dat' is created, through an EXTERNAL TRANSMIT and EDI PUT (see Fig. 2) by running ProdMod in steady state mode. The EDI PUT writes values of the variables listed in ProdMod's EXTERNAL TRANSMIT (see Fig. 2) to the file 'initial.dat'. The EXTERNAL TRANSMIT variables are a part of the constant variables of the waste complex at the initial starting point of optimization. The second step is to create an initial copy of the file containing the dynamic variables used in the model. Another file 'global_pre.dat' is created from a new ProdMod steady state run using a different EXTERNAL input section (with both RECEIVE and TRANSMIT options active) and with no active read in EDI GET. This step creates the file containing the values of the dynamic variables, listed in the TRANSMIT sub-section, which are subject to being updated by the optimizer. In this step, the RECEIVE (from the optimizer) and TRANSMIT (from the simulator) sub-sections of ProdMod's EXTERNAL section list the same variable names. But, the variables listed in the RECEIVE sub-section are not read from a file because of the inactive read in EDI GET, thereby preserving the initial values loaded into the ProdMod's OPERATION section. In the EDI feature, the READ option precedes the WRITE option if both options are active. This is the first time 'global_pre.dat' file is created and, this file is read by the optimizer along with other user supplied files necessary to compute the parameters for the optimization routine.

After successful optimization solution and acceptable optimization results, the optimizer creates an updated file 'global_post.dat'. The file 'global_post.dat' contains the optimized parameter set along with other updated and non-updated variables. The files 'global_pre.dat' and 'global_post.dat' have the same number of variables with identical variable names, but their values may not be identical. These variables represent a part of the dynamic variables, whose values are subject to change during an optimization stage, for the entire range of simulation. The ProdMod simulation run, with EXTERNAL RECEIVE and TRANSMIT sub-sections and having EDI read and write active in EDI GET and PUT routines, creates the file 'global_pre.dat' after reading 'global_post.dat'. If the simulation needs to be continued for the next stage of simulation, the optimizer reads the new file 'global_pre.dat' and performs the next stage of optimization provided the constraint equations and the objective function does not require any changes during this stage.

The files 'global_pre.dat' and 'global_post .dat' contain all the variables: those which have been updated up to *i*th and *i*+1th stage of optimization respectively, along with other un-updated dynamic variables which are subject to update during an optimization stage for the entire range of simulation. The above file manipulation makes it possible to use the same External section for all the sequential stages of optimization, from beginning to end. If the constraint equations and/or the objective function need to change at any stage, the optimizer is modified to create a new executable, and the process starts at the beginning of the flowchart as shown in Fig. 3.

IV. Example System Description

The sequencing of wastes from different waste tanks in the salt processing operation, to produce a specific amount of precipitate for DWPF, is used as an example system for the devised optimization scheme. Salt processing is a major operation in HLW treatment and was to take place in the In-Tank Precipitation (ITP) Facility which is currently being replaced due to excessive benzene generation. One of the proposed alternative salt processing operations is the Small Tank TPB to replace the ITP Facility, based on the same principle of operation but with major design changes to reduce benzene generation. In the salt processing operation, salt solutions from the HLW tanks are separated into two streams: a highly radioactive portion called the precipitate slurry and a minimally radioactive portion called the decontaminated filtrate. The soluble cesium is precipitated with sodium tetraphenylborate (NaTPB) and the strontium, uranium, and plutonium are adsorbed on monosodium titanate (MST) to form insoluble solids. The non-radioactive potassium and ammonium are also precipitated. The decontaminated filtrate is solidified into cement grout and disposed of at the Saltstone Vaults as treated low level waste. The highly radioactive processed precipitate, along with the processed sludge from another operation which is also highly radioactive, are vitrified into borosilicate glass poured into canisters at the DWPF for final disposal at the Federal Repository.

The entire inventory of supernate and saltcake will be processed, in many cycles with a number of batches in each cycle, in the salt processing operation. In each batch, the supernate and dissolved salt solution from different waste tanks are blended in Tank 48, dilution water (if needed) is added to reduce the Na concentration of the blended salt solution to a specified molarity (~5.0 molar) for optimum precipitation, and then NaTPB is added to produce precipitate slurry. The addition of MST is not included in the example system implementation, because the precipitate production contribution due to MST is negligible. At the end of a batch, the precipitate slurry is concentrated to 10 wt% solids by dewatering through crossflow filtration. At the end of a cycle which consists of several batches, the 10 wt% precipitate slurry is washed to reduce the Na molarity. The washed precipitate slurry would be sent to DWPF where the precipitate slurry is converted to borosilicate glass after mixing with sludge and frit through different chemical operations. The blending of waste in the salt processing operation is subject to a number of operating rules and priorities that arise due to operational limitations, regulatory requirements, and budgetary constraints.

V. Implementation

The optimization scheme described in Section III has been implemented for the example system outlined in Section IV in order to sequence wastes for each batch sequentially in an optimal manner while satisfying the operating rules and priorities. Performing optimization for each batch independently is identified as batch optimization. The rules and priorities modeled in the form of constraint equations and an objective function are given in Table I, the middle column presents the operating rules and priorities and the right column is the corresponding implementation form. Some of the priorities and constraints are implemented by initializing the parameters at the beginning and then updating those for each batch interactively as required, by the user manually or by the optimizer automatically. The remaining constraints and priorities are imposed explicitly by the constraint equations and the objective function which are expressed in linear form. The batch optimization involves 103 non-negative decision variables which are the supernate volume (x_i) from 51 tanks, the dissolved salt solution (y_i) taken from 51 tanks, and the needed dilution water volume (w_d). The subscript i runs from 1 to 51 for 51 tanks. For numerical simplicity, all 51 tanks are used in the optimization although two tanks (Tanks 17 and 20) have already been closed. The supernate and the dissolved salt solution volumes for those two tanks have been set to zero. An optimizer has been developed in FORTRAN to perform the batch optimization for the example system. The optimizer interfaces with the ProdMod simulator to read the variables required to calculate the parameters of the constraint equations and the objective function, and then calculates the optimized parameters using a linear programming SIMPLEX algorithm. The optimized parameters are then passed to the simulator, and after simulation the optimizer extracts the required variables from the simulator for optimization of the next batch. The batch optimization continues until wastes from all the tanks are removed or the user wants to stop. A brief description of the optimizer is provided in Section VII.

The operating rules and priorities listed in Rows 1 through 3 in Table I are achieved though the objective function (the equation in Row 1) and assigning appropriate values for c_i, and thereby maximize the available space and also removing wastes from certain tanks. The higher the priority-rank (c_i), the higher is the probability that waste from the high priority tank will be removed earlier. The user has the option of changing the priority of a tank at any batch. The objective function maximizes the amount of waste to be processed (maximizing tank space) from the high priority tanks after satisfying all the constraints that are imposed. The third term ($q.w_d$) in the objective function is used to make the objective function a general one. The coefficient q may be zero, positive or negative. Section VIII.B discusses the significance of this term. The rules listed in Row 4 are implemented by setting 0 or 1 for the parameter $tank\_avail_i$. If the parameter is set to zero for a tank, the tank is not available for transfer of waste. The user has the option to set this parameter for any tank at any batch. The optimizer automatically checks for the presence of supernate in a tank at each batch and sets the dissolved salt solution volume parameter (f_i) to zero if supernate is present; otherwise it generates dissolved salt solution volume if salt is available. The uniformity of Na, K, Cs concentrations in the salt solution (supernate + dissolved salt) and the upper limit of Cs concentration in 10 wt% precipitate slurry in a batch as identified in Rows 5 and 6 are controlled by the constraint equations in those two rows. The upper and lower limits of Na, K, and Cs concentrations of the salt solution are specified by the user. The operating rules listed in Rows 7, 8, and 9 are explicitly represented by the constraint equations in the respective rows.

VI. Justification for Batch Optimization

The example optimization problem implemented has used batch-by-batch optimization instead of global optimization. The justifications in using this batch optimization are as follows:

1. Mimic the SRS waste complex operation: Due to constant operational, regulatory, and budgetary changes, the Savannah River Site publishes a '10 Year Plan' for the next 10

years of operation and the 'System Plan'[7] for the entire range of operation (to clean up all waste tanks) twice a year on an average. These operational plans are developed based on the best available information at that time. This suggests that optimized batch sequences generated by global optimization, if possible, will not be applicable after six months to a year. In salt processing operation, every batch will be sequenced based on the conditions present at that time with an eye to the foreseeable future. This again suggests that globally optimized parameters, if feasible, are valid only for a single batch. It will be necessary to generate optimized parameters many times, each set of which will be invalid after one batch of operation in any case. Thus the batch optimization reflects the operational strategy which is in place.

2. Strict operating window: A regulatory commitment of closing several tanks before a specific time is in place. In batch optimization it is easy to fulfill this type of tank closure commitment by elevating the waste removal priority-ranks. Waste removal instrumentation must be in place before waste can be removed from a tank. At any time only a few tanks will have the waste removal instrumentation in place. It is desirable to move the waste removal instrumentation from tank to tank as infrequently as possible. As a result, it is desirable that waste removal from a tank is continued until all wastes are removed. The batch optimization can easily manage this kind of restriction by allowing or disallowing waste removal from a tank and increasing or decreasing the waste removal priority-rank. Due to specific closure dates for some tanks and acute limitations for available space in the drop tanks, it will be necessary to transfer wastes from certain tanks to other tanks for temporary storage. The batch optimization can accommodate this type of inter-tank waste transfer efficiently.

3. Dynamic nature of the problem: The selection of a drop tank (where the concentrate is sent) in an evaporator system, the space available in the drop tank, and the drop tank fill rate are all dynamic in nature. This information is based on the operation of the entire waste complex, and is only known from the simulator. In case of batch optimization, this information is extracted from the simulator at each batch. For global optimization, a realistic assumption must be made to determine these dynamic parameters in advance for all future batches over which optimization is to be performed. This is a challenging task. Moreover, the number of batches required to process all wastes is not known. Hence, the range of global optimization will be determined by trial and error after running global optimizations several times, which is very time consuming.

4. Infeasible solution: The example system operates with many restricted constraints. It is very likely that an attempt to achieve a global optimization will end up having infeasible solutions in most cases. It will not be easy to find a feasible solution from an infeasible one in the case of global optimization. On the other hand, for batch optimization it would be easy to find a feasible solution from an infeasible one by relaxing some of the constraints.

5. Special nature of the objective function: The objective discussed in this problem is to maximize the tank space, i.e. to process the maximum amount of wastes in a batch. Space in the Tank Farm is at a premium for its normal operation. Hence, achieving this

objective is critical for the initial stage of the operation, particularly in the first few years of clean up operations. The batch optimization is perfectly suited to meeting this objective, and at the same time the constraint equations guide smooth operation throughout its life span.

VII. Description of the Optimizer

An optimizer has been developed to implement the optimization scheme for the example system described in Sections III, IV and V. This section provides a brief description of the optimizer and the implementation steps of the example system using a linear programming algorithm. The optimizer has been developed in FORTRAN. Its objectives are to:

1. Read the input files generated by the ProdMod simulator and created by the user.
2. Read interactive user inputs.
3. Calculate the parameters of the constraint equations and the objective function.
4. Incorporate the appropriate optimization routine.
5. Calculate optimized parameters using the optimization routine.
6. Control process parameters for advancement of the simulation.
7. Write optimized and process control parameters to a file for the ProdMod simulator to be used for the next stage of simulation.
8. Write necessary variables to a file to allow restart of the optimization in the future.

The ProdMod simulator performs simulations of the entire high level waste complex at SRS for a specified number of years. The current structure is set to simulate 29 years of operation, with the first four years represented on a monthly basis and the rest as annual operations. This structure can be changed easily by changing a few parameter values in the simulator. The salt and sludge processing operations are batch processes; thus these two processes are modeled in discrete event-space. The current ProdMod structure has first 20 salt processing cycles with multiple batches in each cycle, followed by another 25 cycles with a single batch per year. In the current application, out of the first 20 cycles, only 6 cycles are set as active sets containing several batches in each cycle. The number of batches in each of the other 14 cycles is set to zero. Table II shows the supernate decant schedule from a typical waste tank for precipitate production during salt processing operation in event-space (in cycles and batches). An evaporator active/inactive status is shown in Table III for all time intervals (active = 1.0; inactive = 0.0) in time-space (in months and years) since the evaporators run as continuous processes. In the ProdMod simulator, the input variables are set in the OPERATION section for all years of operation and for all batches and cycles. If a variable is not applicable for any event or time period, the variable is set to zero for completeness. If simulation beyond a certain point in the event- or time-space is not desired, dummy numbers or zeros must be supplied for that range in the OPERATION section for completeness. The values in Tables II and III are the user supplied values provided here as an example. The values of interest for the optimization range are generated by the optimizer; thus the user supplied values for this range are superseded. The optimization scheme devised in Section III is a general purpose one. It can start optimization at any point in the simulation event or time interval. It can optimize parameters for any number of event or time intervals for linear and nonlinear problems. For nonlinear optimization problems, the optimizer needs to have a nonlinear optimization routine. The optimization can be repeated any number of times for the subsequent event or time intervals until it reaches the end of the simulation. The optimization can be stopped at the end of a particular event or time interval and can be restarted from that point for the remaining event or time interval(s) at a later time.

The example system used in Section IV is a linear programming optimization problem in event-space for a single interval. To begin an optimization for a specific number of event or time intervals starting from a particular event or time interval, the optimizer reads the initial conditions prior to the optimization starting point from a file generated by the ProdMod simulator. The simulator generates this file using SPEEDUP's EDI and EXTERNAL features. The optimizer reads user supplied values from files and also interactively during optimization. The optimizer calculates the optimized parameters for the desired event or time intervals after calculating the parameters of the constraint equations and the objective function of the optimization problem. The optimizer also sets some variables for future time/event intervals starting from the optimization starting point. The variables prior to the optimization starting point are not changed. The optimized parameters and other necessary variables are written to a file. The ProdMod simulator reads that file using its EDI and EXTERNAL features, supersedes the hard coded input values in its OPERATION section, and then advances the simulation. The simulation results are valid up to the last event-/time-space interval and their corresponding time/event interval over which the optimization is performed. The results beyond that point are dummy representations.

The objective function and the constraint equations of waste removal sequencing (the example system) given in Table I can be written in matrix form as follows:

$$\text{Maximize} \quad Z = \boldsymbol{cx} \tag{6}$$

subject to

$$\boldsymbol{Ax} \begin{bmatrix} \leq \\ \\ \geq \\ \\ = \end{bmatrix} \boldsymbol{b} \tag{7}$$

and

$$\boldsymbol{x} \geq \mathbf{0} \tag{8}$$

where, A is a matrix of dimension mxn for the parameters of the constrain equations
m is the number of constraint equations
n is the number of decision variables
Z is the value of the objective function
$\boldsymbol{c}$ is the row vector with n columns for the parameters of the objective function

$\boldsymbol{c} = [c_1, c_2, c_3, \ldots\ldots\ldots\ldots\ldots\ldots, c_n]$ (9)
x is the decision variable vector
b is the vector for the right hand side constants of the constraint equations
0 is the null vector
***x*, *b*,** and **0** are the column vectors with *n, m,* and *n* rows, respectively

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ . \\ x_{51} \\ y_1 \\ . \\ y_{51} \\ w_d \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ b_{m-1} \\ b_m \end{bmatrix}, \quad \boldsymbol{0} = \begin{bmatrix} 0 \\ . \\ 0 \\ 0 \\ . \\ 0 \\ 0 \end{bmatrix} \qquad (10)$$

The example system is formulated as a linear optimization problem with the number of decision variables and the number of constraint equations are almost equal. Hence, a SIMPLEX (Primal) algorithm based optimization routine[8] has been incorporated into the optimizer.

The sequence of operations and the functions of the routines in the optimizer are given in Fig. 4, which is a system specific and expanded version of Fig. 3. The variable names used in Fig. 4 correspond to those used in Table I. The important functions of the optimizer are described in this paragraph. After initialization and setting the default parameters, the optimizer asks the user whether it is a new optimization run or an extension from the previous optimization run. For a new optimization run, the optimizer reads the file 'initial.dat' generated by the ProdMod simulator for the fixed variables whose values remain constant during simulation. The optimizer also reads the values for the variables c_i (waste removal priority-ranks) and $tank\_avail_i$ (waste removal allowability) from the user supplied input files. For a new optimization run, the optimizer reads the optimization starting point in the event-space or time-space interactively from the user. The user chooses the optimization starting point, say Batch 13 for the example system optimization problem. From Table II, Batch 13 is the second batch in Cycle 3. The optimizer then synchronizes the optimization starting point with the variables in the other space. The optimizer then reads the file 'global_pre.dat' generated by the ProdMod simulator for the dynamic variables whose values may get updated during the next stage(s) of optimization and simulation. The optimizer will not change the values of any variable up to Batch 1 in Cycle 3 (intervals before the simulation starting point) in the OPERATION section of the ProdMod simulator (see Fig. 2). If the optimization never started before, the optimizer initializes the values of the dynamic variables for the future time-/event-space intervals. The optimizer then displays current information of the waste complex (e.g., supernate, salt, dissolved salt volumes and their composition for each tank, evaporator and its associated drop tank status) as a user option. The user can change these status variables interactively. The user can change the waste removal priority-ranks and waste removal allowability for any tank at each stage of optimization upon reviewing the current status. The optimizer displays with change option the evaporator and its associated drop (salt receiving) tank status. The optimizer also displays the components of the current parameters of the constraint equations, and the user has the choice to change them interactively during optimization.

At this point, the optimizer proceeds into the optimization calculation. After calculating a_{ij} (the matrix elements of A) and b_i, the optimizer uses the optimization routine to generate optimized parameters. The user has a choice to generate another set of optimized parameters (say, the current set is not acceptable due to operational choices), after reviewing the current set, by changing the parameters of the constraint equations and the objective function interactively. If no feasible solution exists for a given condition, the user can change the optimization parameters interactively to find a feasible solution. In the ProdMod structure, the annual regime consists of a single batch (one batch per cycle and one cycle per year). But in real life operation, a cycle will have several batches. So, in the optimizer the user has the choice to specify the number of sub-batches in the annual batch regime. The optimized parameters over the sub-batches are combined to generate annual batch optimized parameters. After an acceptable and successful optimization, the optimizer writes the optimized parameter and other necessary variable values to a file ('global_post.dat') for the ProdMod simulator to use for the next stage of simulation. The other necessary variables are, for example, evaporator status, drop tank selection, salt dissolution start time, etc. At this stage the optimizer automatically also creates a restart file ('restart.dat') in order to provide the option for the user to continue optimization at a later time. After successful simulation using 'global_post.dat', the ProdMod simulator writes a file 'global_pre.dat' for the optimizer for the next stage of optimization. The user can continue this process until the simulation is complete or the user wants to quit and/or restart it at some other time.

VIII. Parametric Studies

Two different waste removal scenarios have been studied in performing sensitivity analyses of the parameters in the constraint equations and the objective function in Table I. The scenarios have direct impact on blending wastes from different tanks at SRS.

To determine the input data for the scenarios, the parameters are calculated in the following manner: Waste removal from a tank is permitted or not by setting a value of 1 or 0 for the $tank\_avail_i$ parameters. The waste removal priority-ranks c_i for different tanks have been set by assigning a number; the bigger the rank number, the higher the waste removal priority. The supernate (e_I), saltcake, and dissolved salt (f_i) volumes along with their associated Na (Na_{x_i} and Na_{y_i}), K (K_{x_i} and K_{y_i}), and Cs (Cs_{x_i} and Cs_{y_i}) concentrations for all tanks are extracted one time from the ProdMod initial simulation run. The Na, K, and Cs concentrations of supernate and dissolved salt are assumed to remain the same for all batches. The saltcake is dissolved once the supernate is gone; the dissolved salt volume is 2.7 times the saltcake volume after adjustment for trapped sludge volume which is 2%. The parameters $STPB_{x_i}$ and $STPB_{y_i}$ are calculated based on the concentrations of K and Cs in supernate and dissolved salt solution in a tank. The parameters PF_{x_i} and PF_{y_i} are ratios of the solid masses to the 10 wt% precipitate slurry mass. The solids in the precipitate slurry consist of solids due to addition of NaTPB. It has been assumed that no solids leave the slurry during the filtration process to concentrate the slurry to 10 wt% solids. The number of sub-batches used in each annual batch is four. In addition, the optimizer sets default values or the user can over-write the default values for the parameters: Na_{u_lim}, Na_{l_lim}, K_{u_lim}, K_{l_lim}, *batch_vol, Cs_lim,* Cs_{l_lm}, and *batch_size*.

VIII.A. Scenario A

A sequence has been generated with the object of removing wastes from Tanks 1 through 24 as rapidly as possible, starting in Cycle 6/Batch 3 of the salt process. In this scenario, the waste blending for all batches prior to Cycle 6/Batch 3 remains unchanged as defined in the Operation section of ProdMod. Tanks 1 through 24 are given high priority-ranks (c_i parameters) starting at 100 for Tank 1 and decreasing by 1 for the subsequent tanks to achieve the objective. Waste removals from Tanks 27, 29 and 41 are also given high priorities (priority-rank 100) because each evaporator system must have enough space for concentrated solution. The remaining tanks are set to a 0 priority-rank. The optimizer calculates the parameter w_W based on the *batch_vol* parameter of the previous batch. The other parameters along with their defined values are given below.

batch_vol = 40 kgal
(for annual batch 4 x 40 = 160 kgal)
batch_size = 600 kgal
Cs_lim = 39 Ci/gal
Cs_{l_lim} = 0.5 Ci/gal
Na_{u_lim} = 10 molar
Na_{l_lim} = 4 molar
K_{u_lim} = 0.2 molar
K_{l_lim} = 0.001 molar
dl_lim = 5.0 molar

Table IV presents the results for the first several batches of the optimized batch sequences. The key results are, of course, the order and volumes of the waste transfers determined by the optimizer. In the table, the column under “Batch Size” includes salt solution (supernate + dissolved salt), NaTPB which is approximately 25 kgal in each batch, and 125 kgal of spent wash water from Late Wash with very low Na concentration. In all these sequenced batches, no dilution water has been needed because of the addition of wash water with low Na concentration. At the end of each batch, the optimized batch sequence is passed to ProdMod for time advancement. The simulated results are then extracted in the optimizer, and the optimized waste removal sequence for the next batch is generated. This process is repeated to generate the batch sequences. The batch size in Cycle 6/Batch 3 is below the 600 kgal limit. In Cycle 6/Batch 3, the tanks have only supernate or saltcake. Thus, the batch could not hit the maximum limits in satisfying the constraints. The optimizer dissolves saltcake if there is no supernate. As a result dissolved salts are available in some tanks for the next batch, and the batch sizes hit their limits.

The last two sub-batches (Sub-Batches 3 and 4 in Batch 9, which are not shown in Table IV) also have batch sizes below their limits. At this point of the simulation, the tanks have been cleaned and there are not enough wastes to be taken out from the tanks with positive priority-rank. In each sub-batch, wastes have been blended from tanks ranging from two to four tanks with the exception of two sub-batches only.

It is apparent from Table IV that in most cases once tanks have been selected for supernate or dissolved salt removal in a sub-batch, the same tanks remain selected in subsequent batches until all their supernate or dissolved salt are removed. This mimics the actual operational strategy in removing wastes. Because of the logistics and cost involved in transferring waste from a tank, blending of wastes from more than four tanks in a sub-batch, or switching tanks from batch to batch, is not a viable option. By changing the priority-rank of a tank or forcing a tank to be unavailable for waste removal or by adjusting some other parameters within the operating range, it is easily possible to select or deselect a particular tank in a batch sequence.

At the end of the simulation, i.e. after 9 annual batches, there has been only one tank (Tank 19) out of first 24 targeted tanks left with some waste. The objective function (first Row in Table I) is defined in such a way that the value of the objective function should decrease as the sub-batches progress. But, Table IV shows that the value of the objective function of Sub-Batch 1 in Batch 2 is higher than that of Sub-Batch 4 in Batch 1. The priority-rank of Tank 29 is higher than that of Tank 3, and the dissolved salt of Tank 29 is not available until Sub-Batch 1 in Batch 2. The availability of dissolved salt in Tank 29 contributes higher value to the objective function in Sub-Batch 1 of Batch 2. Again, the dissolved salt from Tank 1 is not available until Sub-Batch 4 in Batch 2. Each of the sub-batches in Batch 2 uses dissolved salt from Tank 29 or Tank 1. As a result, the value of the objective function in each of the sub-batches in Batch 2 is higher than the sub-batches objective function values in Batch 1. It may be noted that supernate from Tank 45 has been taken in several sub-batches by the optimization algorithm although its addition does not increase the value of the objective function because of its zero priority-rank. The addition of supernate from Tank 45 was necessary only to satisfy the constraint equations. Another important observation from Tables IV and V is that the optimizer has not selected waste from a tank in a sub-batch whenever that selected amount would have been so small as to be not economically and technically feasible to transfer (except in sub-batch 5 for Case 2 in Table V for which the volume is 5 kgal).

A modified waste removal sequence case has also been generated with the objective being to clean up all the tanks instead of just 1 though 24 tanks. In this case, the waste removal priority of the tanks has been set arbitrarily. The batch size limits and the number of sub-batches in an annual batch have been varied. In the early part of sequencing, the number of sub-batches in an annual batch has been set to four. But in the later part, the number of sub-batches has been increased to seven. The values of the other parameters are the same as those mentioned above. It has been observed that the optimizer was unable to find a feasible solution only for three occasions. The existing conditions at those points were such that the precipitate demand could not be met after satisfying all the constraint equations. The precipitate demand volume was changed and then the batch optimization was able to progress. The optimizer has successfully generated batches all the way to clean up of all the tanks. For the last sub-batch, the precipitate demand volume had to decrease to get a feasible solution because the available salt solution was insufficient to make enough precipitate to meet the demand. There was only 76 kgal of salt solution left unprocessed at the end of last batch.

VIII.B. Scenario B

The second case study has been performed to demonstrate the effect of varying the coefficient *q* of the dilution water decision variable in the objective function (Row 1 in Table I). Three different cases have been studied: 1) null coefficient ($q = 0$), 2) penalty coefficient ($q = -50$), and 3) credit coefficient ($q = 100$). Using Scenario A's parameter values, three different sequences are generated for these choices and the results are presented in Table V. Unlike Scenario A, no wash water has been used to dilute Na concentration of the salt solution in this scenario. For the null coefficient case (Case 1), addition of dilution water will not affect the value of the objective function. For Case 2, the coefficient *q* has a value - 50 which reduces the value of the objective function. Hence, the optimizer will be reluctant to add dilution water to the salt solution in a batch. On the other hand, there is an incentive in Case 3 where the value of *q* is 100. In this case, the optimizer will add dilution water which is simulated as waste from a tank of priority-rank 100. From Table V, it is apparent that no dilution water has been added in the first three sub-batches for Case 1. In Case 2, no dilution water has been added until sub_batch number 7. In Case 3, dilution water has been added in all sub-batches but the first sub-batch. This sensitivity study shows that the coefficient *q* in the objective function can play a very important role in sequencing high level waste at SRS over different time horizons. The available space in the Tank Farm at SRS is a crucial issue for the next several years of operation. Case 2 is a suitable scenario for the next several years which will try to maximize the available space without adding dilution water. The optimizer will try to extract waste from low Na concentration tanks. Once the near term issue of Tank Farm available space is passed, Case 3 may be a suitable scenario for some time. During this time, waste from high Na concentration tanks will get some preference. If Case 3 is in operation for some time, the waste with elevated Na concentration is expected to be depleted to some extent. At that point, Case 1 may be the preferred choice.

IX. Conclusion

A general purpose dynamic optimization methodology has been developed and coupled to the ProdMod simulator, taking advantage of its pseudo-dynamic constructs and time- and event-space mapping. The devised optimization scheme is suitable for a large and complex dynamic system consisting of tens of thousands of variables. The optimization scheme is open to all types of optimization: linear, nonlinear, dynamic, and stochastic. A stand alone optimizer to perform optimization has been developed which interfaces with the ProdMod simulator to extract data from the simulator and pass the optimized input to the simulator for the next stage of optimization. This repeated process can continue until the simulation comes to its final stage. Depending on the nature of the optimization problem, the optimizer may be customized to incorporate suitable optimization routines and to calculate the needed parameters for the optimization routine.

The optimization scheme has been fully coupled with the ProdMod simulator and has demonstrated success in generating optimized waste blending batch sequences using a linear programming algorithm. Parametric studies show that batch optimization in blending waste from different tanks for precipitate production is a realistic approach for this type of complex dynamic system where some of the parameters are not known in advance and the system is constrained by many operating rules and priorities, some of which might also change with time. This scheme can be easily implemented to optimize other operating parameters, to determine strategies for running the High Level Waste complex in an economically efficient and environmentally sound manner, by using appropriate constraint equations and objective functions.

Nomenclature

batch_size = maximum allowable volume combined in a batch
batch_vol = 10 wt% precipitate slurry volume produced in a batch
c_i = waste removal priority-rank of tank *i*
Cs_{x_i} , Cs_{y_i} = supernate, dissolved salt Cs concentration in tank *i*
Cs_lim = Cs concentration limit in 10 wt% precipitate slurry
dl_lim = Na concentration limit of salt solution for dilution water addition
e_i , f_i = supernate, dissolved salt volume available in tank *i*
Na_{x_i}, Na_{y_i} = supernate, dissolved salt Na concentration in tank *i*
$Na_{w_{LW}}$ = Na concentration of LateWash addition water
Na_{w_d} = Na concentration of dilution water
PF_{x_i}, PF_{y_i} = 10 wt% precipitate slurry per unit supernate, dissolved salt volume
q = coefficient of dilution water
$STPB_{x_i}, STPB_{y_i}$ = STPB needed per unit vol. of supernate, dissolved salt
$tank\_avail_i$ = supernate/dissolved salt availability in tank *i*
w_d , w_W = dilution, wash water volume addition
X_{u_lim} , X_{l_lim} = upper limit, lower limit of *X* (Na/K/Cs) concentration of blended salt solution
x_i , y_i = supernate, dissolved salt taken from tank *i*
X_{x_i} , X_{y_i} = supernate, dissolved salt *X* (Na/K/Cs) concentration in tank *i*

Acknowledgments

This work was performed under the auspices of the U. S. Department of Energy under Contract DE-AC09-96SR18500. Accordingly, the U.S. government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. government purposes.

References

1. M. V. Gregory and P. K. Paul, “Efficient Mathematical Modeling of an Integrated High Level Waste Processing Complex,” (Submitted to Nuclear Technology).

2. P. K. Paul, M. V. Gregory, and M. N. Wells, “Computer Based Re-Engineering of the Planning Process at the Savannah River Site,” *Trans. Am. Nucl. Soc.*, **77**, 484-486, November 1997.

3. M. V. Gregory, P. K. Paul et al., “Planning Tools Supporting High Level Waste Processing at the Savannah River Site,” DOE Pollution Prevention Conference XII Proceedings, Chicago, Illinois, July 9-11, 1996.

4. “User’s Manual, SPEEDUP,” Aspen Technology Inc., Cambridge, Massachusetts, 1993. SPEEDUP is a Trade Mark of Aspen Technology.

5. A. B. Scott, Jr. and N. R. Davis, “Processing High-Level Waste at the Savannah River Site,"” RADWASTE, Vol 3, No 3, May 1996

6. P. D. d’Entremont et al., “High-Level Waste System Process Interface Description,” WSRC-TR-94-442, Westinghouse Savannah River Company, September 1994.

7. N. R. Davis and M. N. Wells, “High Level Waste System Plan, Revision 9,” HLW-OVP-97-0068, Westinghouse Savannah River Company, April 1998.

8. NSWC Library of Mathematical Subroutines, NSWCDD/TR-92/425 (1993), Naval Surface Warfare Center, Dahlgren, Virginia.

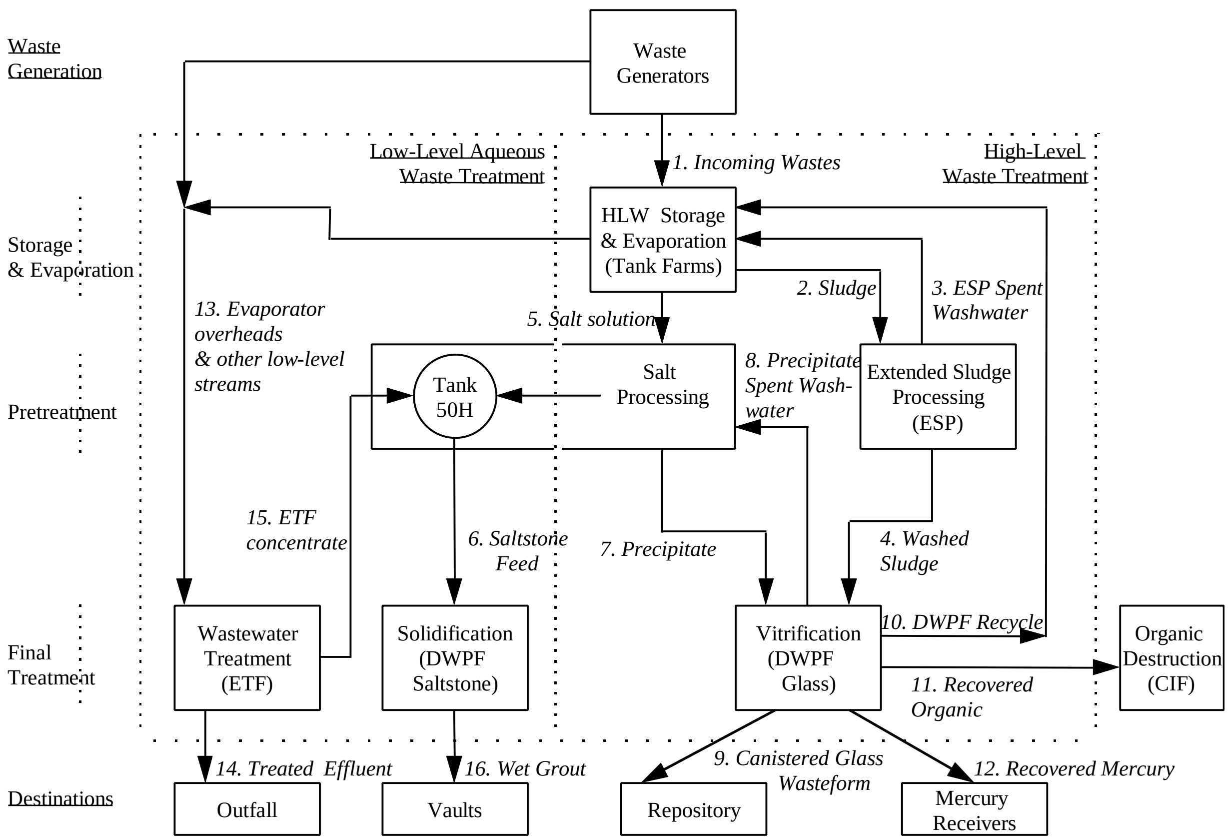
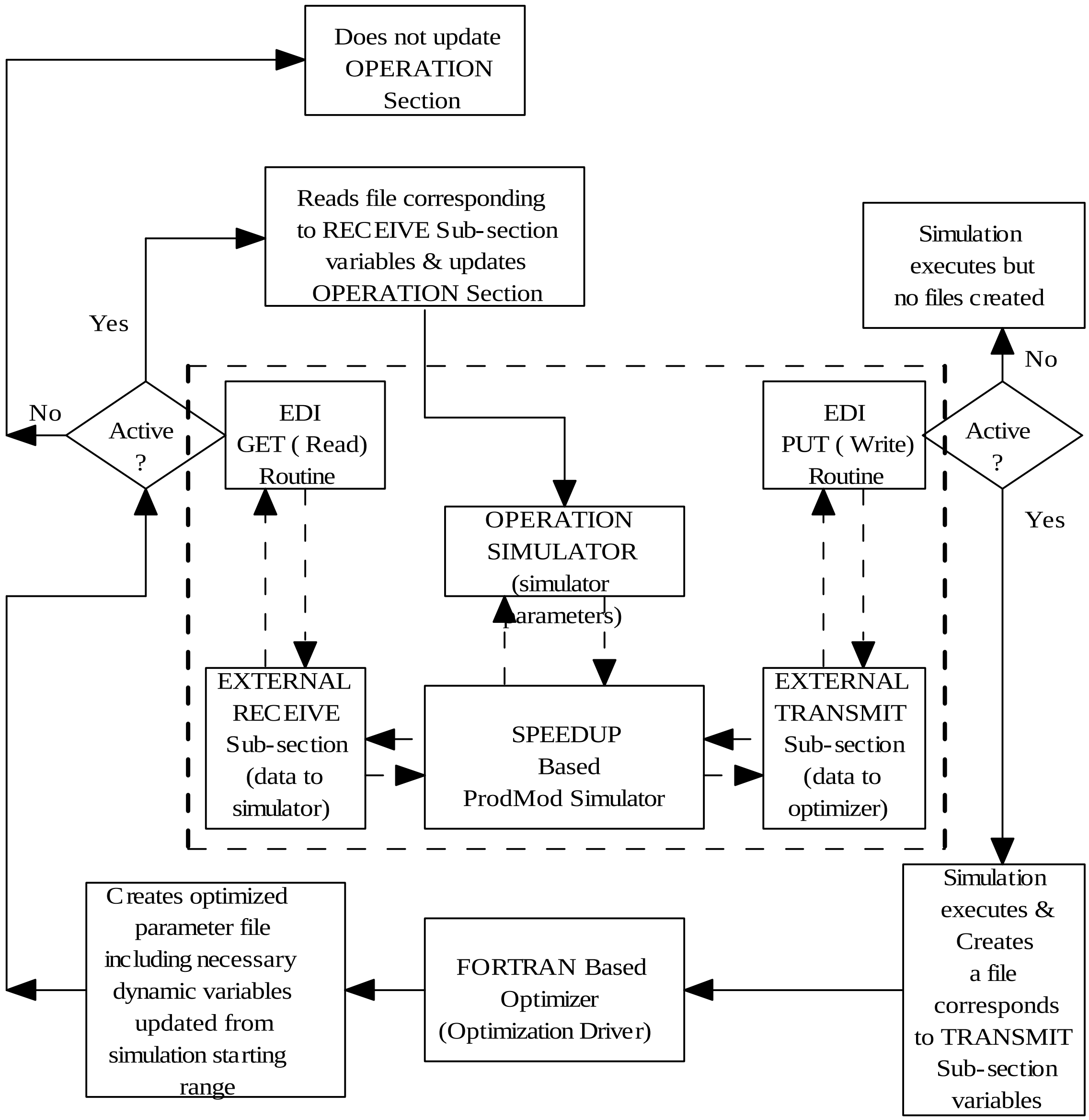


Fig. 1. SRS HLW Process Flow Streams Modeled in ProdMod



Note: Blocks & the broken arrows inside the dotted rectangle show the SPEEDUP structured data interfacing mechanism within the ProdMod simulator

Fig. 2. Interface Between ProdMod Simulator and Optimizer

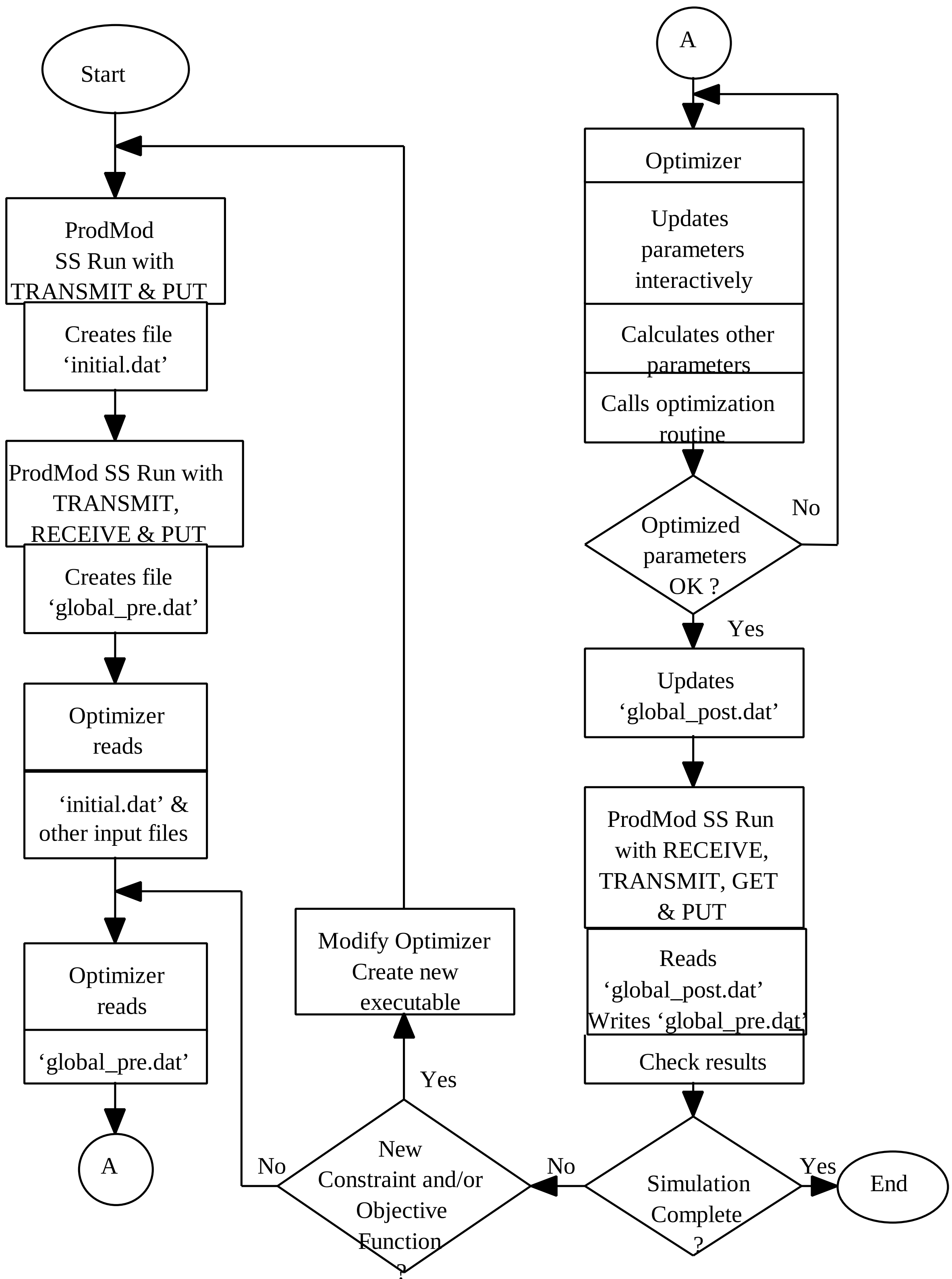


Fig. 3. Optimization Scheme Algorithm Flowchart

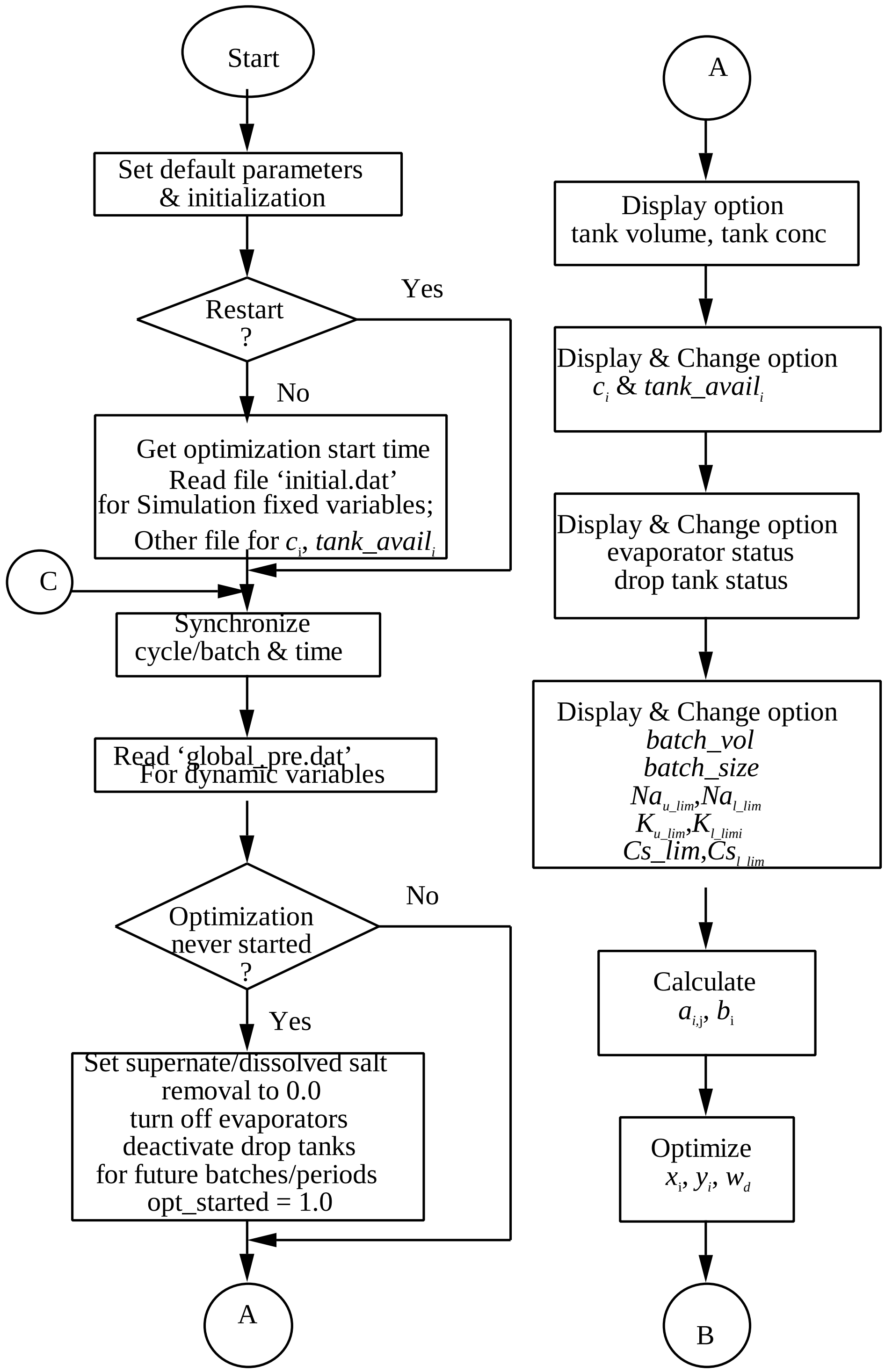


Fig. 4. Optimizer Functional Flowchart

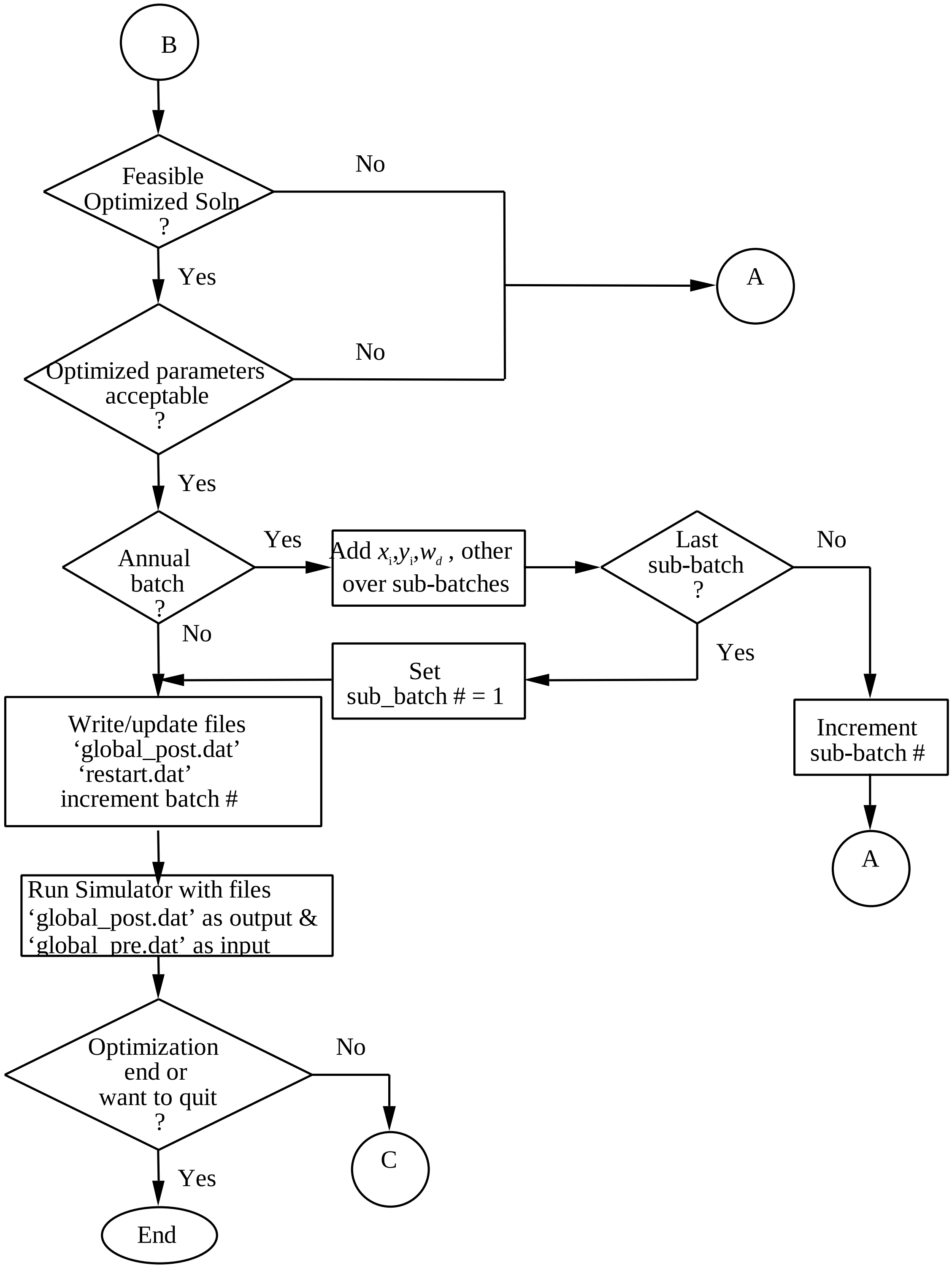


Fig. 4. Optimizer Functional Flowchart (Continued)

Table I. Implementation of Operating Rules and Priorities

Row	Operating Rule and Priorities	Implementation
1	Maintain emergency tank space (to allow accumulation of new waste) as required by the Tank Farm Safety Analysis Report (SAR).	Objective Function : Maximize waste removal in a batch from the high prioritized tanks : Maximize $\sum_{i=1}^{51} c_i.x_i + \sum_{i=1}^{51} c_i.y_i + q.w_d$ Decision variables: x_i, y_i, wd (i = 1,2,3, …,51)
2	Enable continued operation of the evaporators by providing enough space for the evaporator drop tanks.	Assigning higher number for c_i (in the objective function) for the evaporator drop tanks
3	Have priority to remove waste from tanks with a history of leakage and tanks with no secondary containment and leak detection equipment.	Assigning higher number for c_i (in the objective function) for the those tanks
4	Waste from certain tanks can not be removed for certain period of time due to operational procedure. Waste removal instrumentation must be in place before waste removal can take place from a tank. Dissolved salt is not available until all supernate is decanted form a tank.	Assigning 0 or 1 to $tank\_avail_i$ of the constraint equations: $x_i \le e_i \,.\, tank\_avail_I$ $y_i \le f_i \,.\, tank\_avail_i$ with i = 1, 2, 3, ……., 51
5	Na, K, Cs concentrations in the blended salt solution should be uniformly distributed over the batches, if possible.	Upper limit and lower limit: (X for Na, K, Cs) $\sum_{i=1}^{51}(X_{x_i} - X_{u\_lim}).x_i + \sum_{i=1}^{51}(X_{y_i} - X_{u\_lim}).y_i \le 0$ $\sum_{i=1}^{51}(X_{x_i} - X_{l\_lim}).x_i + \sum_{i=1}^{51}(X_{y_i} - X_{l\_lim}).y_i \ge 0$
6	Cs concentration in the 10 wt% precipitate slurry should not exceed the SAR dictated limit based on facility shielding design criterion.	$\sum_{i=1}^{51} Cs_{x_i}.x_i + \sum_{i=1}^{51} Cs_{y_i}.y_i \le batch\_vol.Cs\_lim$
7	Dilution water needs to be added if Na concentration of the blended salt solution (supernate + dissolved salt + spent wash water, if any) exceeds a specified limit (~ 5 molar).	$\sum_{i=1}^{51}(Na_{x_i} - dl\_lim).x_i + \sum_{i=1}^{51}(Na_{y_i} - dl\_lim).y_i +$ $(Na_{w_d} - dl\_lim).w_d \le (dl\_lim - Na_{w_{LW}}).w_W$
8	The size of each batch consisting of salt solution, spent wash water, dilution water, and NaTPB can not exceed ITP tank capacity or any other operating set limit.	$\sum_{i=1}^{51}(1+STPB_{x_i}).x_i + \sum_{i=1}^{51}(1+STPB_{y_i}).y_i + w_d \le$ $(batch\_size - w_W)$
9	The amount of precipitate produced in a batch must meet DWPF demand.	$\sum_{i=1}^{51} PF_{x_i}.x_i + \sum_{i=1}^{51} PF_{y_i}.y_i = batch\_vol$

Table II. Sample Supernate Decant Schedule (User Input)
from a Waste Tank in Event-Space

```
bdect_XX =    <

#     batch -- >   0     1     2     3     4          - cycle 1
                   0.0   0.0   0.0   200.  0.0
#     batch -- >   0     1     2     3     4          - cycle 2
                   0.0   350.  0.0   0.0   0.0
#     batch -- >   0     1     2     3     4          - cycle 3
                   0.0   0.0   0.0   100.  0.0
#     batch -- >   0     1     2     3                - cycle 4
                   0.0   175.  200.  0.0
#     batch -- >   0     1     2     3                - cycle 5
                   0.0   0.0   0.0   0.0
#     batch -- >   0     1     2     3     4          - cycle 6
                   0.0   0.0   0.0   0.0   0.0  >
```

---

bdect_XX - supernate removal from Tank XX, kgal
Batch 0 (dummy batch) in each cycle is designed to accommodate cycle
(for computational purpose, batch 0 in cycle 1 is considered as batch 1 and then subsequent batches including dummy batches are incremented)
Records starting with ‘#’ are comment cards

Table III. Sample Evaporator Status (User Input) in Time-Space

```
xx_evap_state   =  <
#      month  -->   1      2      3      4      5      6      7      8      9
                    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
#      month  -->   10     11     12     13     14     15     16     17     18
                    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
#      month  -->   19     20     21     22     23     24     25     26     27
                    1.0    0.0    0.0    0.0    0.0    1.0    1.0    1.0    1.0
#      month  -->   28     29     30     31     32     33     34     35     36
                    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
#      month  -->   37     38     39     40     41     42     43     44     45
                    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
#      year   -->   FY01          FY02          FY03          FY04          FY05
                    1.0           1.0           1.0           1.0           1.0
#      year   -->   FY06          FY07          FY08          FY09          FY10
                    1.0           0.0           0.0           1.0           1.0
#      year   -->   FY11          FY12          FY13          FY14          FY15
                    1.0           1.0           1.0           1.0           1.0
#      year   -->   FY16          FY17          FY18          FY19          FY20
                    1.0           1.0           1.0           1.0           0.0
#      year   -->   FY21          FY22          FY23          FY24          FY25
                    1.0           1.0           1.0           1.0           0.0
```

---

xx_evap_state - xx evaporator status (on-line : 1.0 ; off-line : 0.0)
mon 1 is the starting month, i.e. simulation start time
Records starting with ‘#’ are comment cards

Table IV. A Sample Optimized Batch Sequence

Cycle & Batch Number	Tank Number	Volume (kgal)	Supernate / Dissolved Salt	Batch Size (kgal)	Objective Function Value
Cycle 6/Batch 3	11	218	supernate	368	19620
Cycle 6/Batch 4	29	71	supernate	600	44522
	2	378	dissolved salt		
Annual Regime Starts					
Batch 1					
Sub-Batch 1	29	71	supernate	600	44522
	2	378	dissolved salt		
Sub-Batch 2	29	71	supernate	600	44522
	2	378	dissolved salt		
Sub-Batch 3	27	19	supernate	600	44507
	29	43	supernate		
	2	283	dissolved salt		
	3	105	dissolved salt		
Sub-Batch 4	27	46	supernate	600	44192
	3	404	dissolved salt		
Batch 2					
Sub-Batch 1	27	80	supernate	600	45000
	29	370	dissolved salt		
Sub-Batch 2	1	19	supernate	600	45000
	27	46	supernate		
	29	385	dissolved salt		
Sub-Batch 3	4	68	supernate	600	44696
	41	34	supernate		
	29	347	dissolved salt		
Sub-Batch 4	4	49	supernate	600	44853
	1	401	dissolved salt		

Table IV. A Sample Optimized Batch Sequence (Continued)

Cycle & Batch Number	Tank Number	Volume (kgal)	Supernate / Dissolved Salt	Batch Size (kgal)	Objective Function Value
Batch 3					
Sub-Batch 1	4	49	supernate	600	44853
	1	401	dissolved salt		
Sub-Batch 2	4	49	supernate	600	44853
	1	401	dissolved salt		
Sub-Batch 3	4	78	supernate	600	44766
	1	68	dissolved salt		
	29	304	dissolved salt		
Sub-Batch 4	4	53	supernate	600	44655
	7	31	supernate		
	29	366	dissolved salt		
Batch 4					
Sub-Batch 1	7	84	supernate	600	44396
	29	365	dissolved salt		
Sub-Batch 2	7	42	supernate	600	43949
	27	33	supernate		
	45	7	supernate		
	29	367	dissolved salt		
Sub-Batch 3	27	71	supernate	600	43180
	45	14	supernate		
	29	141	dissolved salt		
	41	224	dissolved salt		
Sub-Batch 4	27	73	supernate	600	43162
	45	14	supernate		
	41	363	dissolved salt		
Batch 5					
Sub-Batch 1	27	73	supernate	600	43162
	45	14	supernate		
	41	363	dissolved salt		
Sub-Batch 2	27	73	supernate	600	43162
	45	14	supernate		
	41	363	dissolved salt		
Sub-Batch 3	27	73	supernate	600	43162
	45	14	supernate		
	41	363	dissolved salt		

Table V. Sensitivity of Dilution Water Coefficient

	Case 1: Null Coefficient		Case 2: Penalty Coefficient		Case 3: Credit Coefficient	
Sub-Batch Number	Tank Number	Volume (kgal)	Tank Number	Volume (kgal)	Tank Number	Volume (kgal)
1	1 11 23 29	19 s 93 s 331 s 231 s 0 dw	1 11 23 29	19 s 93 s 331 s 231 s 0 dw	1 11 23 29 0	19 s 93 s 331 s 231 s 0 dw
2	13 20 23 24 28 29 3	98 s 18 s 142 s 111 s 23 s 28 s 255 ds 0 dw	13 20 23 24 28 29 3	98 s 18 s 142 s 111 s 23 s 28 s 255 ds 0 dw	13 28 3	107 s 35 s 281 ds 251 dw
3	13 17 24 28 3	105 s 123 s 161 s 42 s 242 ds 0 dw	13 17 24 28 3	106 s 123 s 161 s 42 s 242 ds 0 dw	13 28 3	107 s 35 s 281 ds 251 dw
4	13 17 21 28 2 19	106 s 156 s 119 s 47 s 198 ds 34 ds 14 dw	13 17 21 28 2 19	25 s 156 s 119 s 62 s 277 s 34 ds 0 dw	13 28 3	107 s 35 s 281 ds 251 dw

Table V. Sensitivity of Dilution Water Coefficient (Continued)

	Case 1: Null Coefficient		Case 2: Penalty Coefficient		Case 3: Credit Coefficient	
Sub-Batch Number	Tank Number	Volume (kgal)	Tank Number	Volume (kgal)	Tank Number	Volume (kgal)
5	13 28 2	107 s 35 s 281 s 251 dw	13 19 28 43 2	5 s 247 s 13 s 339 s 70 ds 0 dw	13 28 3	107 s 35 s 281 ds 251 dw
6	13 28 2	107 s 35 s 281 ds 251 dw	13 33 43 2	14 s 256 s 110 s 295 ds 0 dw	13 28 3	107 s 35 s 281 ds 251 dw
7	13 43 2	105 s 116 s 264 ds 187 dw	13 33 43 2	10 s 109 s 151 s 326 ds 78 dw	13 28 43 2 3	106 s 21 s 49 s 250 ds 24 ds 224 dw

s - supernate
ds - dissolved salt
dw - dilution water