White Paper

OptiRamp® Optimized Scheduling

Complete Solution for Scheduling Problems

Vadim Shapiro
Dmitriy Khots, Ph.D.

Table of Contents

Introduction

The factors impacting production are numerous and complex. On a daily basis, the Production Scheduler must balance oil production with a wide range of constraints. Constructing a production balance model provides the Production Scheduler with several advantages, such as simulating and optimizing an entire month's production schedule during the month, which helps avoid pitfalls of manual or rule-based scheduling.

The *OptiRamp* Scheduling Submodule offered by Statistics & Control, Inc., (S&C) is a state-of-the-art solution to large-scale scheduling problems. The foundation of the *OptiRamp* Scheduling Submodule algorithms is a robust balance model, which is provided by the *OptiRamp* Model Construction Submodule. The *OptiRamp* Scheduling Submodule uses the balance model to build process operating forecasts in accordance with planned production.

The Scheduling Submodule forecast tools allow for various forecast horizons. The horizon can be chosen for the transition period (ranging from milliseconds to days, e.g., cyclic steam oil production) as well as budget periods (typically one month).

The *OptiRamp* Scheduling Submodule also uses sophisticated optimization techniques to determine the most efficient operating modes of a technological process using current conditions or customized "what if" scenarios. The pre-defined conditions can draw on the latest operational and market data as well as client input.

The submodule solves the optimization problem of maximizing process unit production while minimizing costs subject to constraints, such as limited fuel supply, steam availability, steam distribution system balance, and any relevant facility constraints. The production maximization problem is solved by decomposing the whole problem into the two sub-problems:

1. Production forecast
2. Selection of producers and consumers

The *OptiRamp* Scheduling Submodule uses predictive modeling and the Holt-Winters smoothing method to create accurate load forecasts. The submodule further uses genetic algorithms to search for a near-optimal combination of operating units as well as to find the optimal forecast parameters.

This submodule also provides a real-time alert system that allows for both system and custom alerts for specific events. The events can be based solely on operating data or could also incorporate current data points. The real-time functionality reduces decision-making response time, which saves the client valuable resources.

Balance Model

The *OptiRamp* Scheduling Submodule utilizes the material balance model constructed by the *OptiRamp* Model Construction Submodule. The material balance model can be described by the relationship $Input = Output + Accumulation$.

Given N producers, the behavior of each producer is characterized by three parameters: disabled period T_1; enabled period T_2; and the lag, τ, defining the producer's state when the cycle starts. Figure 1 depicts the step function for each producer.

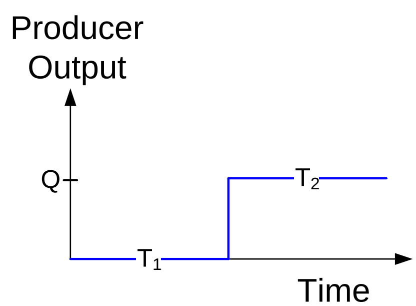

Figure 1. Step function for each producer

Output for each producer is given by function (1):

$$\overline{X}_i(t) = \begin{cases} 0, t \in [k(T_1 + T_2), k(T_1 + T_2) + T_1] \\ Q_i, t \in [k(T_1 + T_2), (k+1)(T_1 + T_2)] \end{cases}, \qquad (1)$$

where k is an integer. The lag parameter is used to define unit enable and disable times. The final producer output is given by $X_i(t) = \overline{X}_i(t - \tau_i)$, where τ_i denotes the lag for each producer and $i = 1,..., N$. The optimal lags are found using genetic algorithms. The material balance equation is then given by equation (2).

$$SystemOutput = \sum_{t \in [T_0, T_1], i=1,..,N} \overline{X}_i(t), \qquad (2)$$

The *OptiRamp* Scheduling Submodule is able to analyze balance models with thousands of producers/consumers because of its scalability and computational techniques.

Forecasting

The *OptiRamp* Scheduling Submodule uses the material balance models to forecast production. Depending on the forecast period, various forecasting methods can be used. Longer periods typically involve seasonality, while shorter forecast periods (for example, during process transition) do not have a seasonal component. The following forecasting methods are implemented in the *OptiRamp* Scheduling Submodule.

Exponential Smoothing

Exponential smoothing is a common forecasting technique that is similar to moving average time series forecasting. The goal is to forecast either a discrete or a continuous variable based on its historical values so that the older values get exponentially smaller weights.

Single exponential smoothing is a single parameter forecast given by equation (3):

$$S_t = \alpha \cdot X_t + (1-\alpha) \cdot S_{t-1}, \tag{3}$$

where S_t is the forecast value at time *t*, X_t is the series value at time *t*, and $0 < \alpha < 1$ is the smoothing parameter.

Due to the recursive nature of this algorithm, it is crucial to select the appropriate initial value. The *OptiRamp* Scheduling Submodule allows the following selection: first time series element and average of *N* initial series elements.

Double Exponential Smoothing

Double exponential smoothing is a useful technique when a trend component is suspected in the time series. Thus, two time series components (level and trend) need to be smoothed. The recursive algorithm for this approach is given by equation (4):

$$S_t = \alpha \cdot y_t + (1-\alpha) \cdot (S_{t-1} + b_{t-1}), \tag{4}$$

where $b_t = \beta \cdot (S_t - S_{t-1}) + (1-\beta) \cdot b_{t-1}$ and $0 < \alpha, \beta < 1$ are the smoothing parameters. The *OptiRamp* Scheduling Submodule allows the following initial value selection: $S_1 = y_1$ and $b_1 = \dfrac{y_n - y_1}{n-1}$ for a positive integer, *n*.

Holt-Winters Smoothing

Holt-Winters smoothing, also known as triple exponential smoothing, is the optimal forecasting method for a time series that has both a trend and seasonality. The *OptiRamp* Scheduling Submodule supports smoothing for both additive and multiplicative seasonality.

For multiplicative seasonality, the recursive algorithm is defined in the set of equations represented as equation (5):

$$\overline{R}_t = \alpha \cdot \frac{y_t}{\overline{S}_{t-L}} + (1-\alpha)\cdot\left(\overline{R}_{t-1} + \overline{G}_{t-1}\right) \text{ (overall smoothing)}$$

$$\overline{G}_t = \beta \cdot \left(\overline{S}_t - \overline{S}_{t-1}\right) + (1-\beta)\cdot \overline{G}_{t-1} \text{ (trend smoothing)} \tag{5}$$

$$\overline{S}_t = \gamma \cdot \frac{y_t}{\overline{S}_t} + (1-\gamma)\cdot \overline{S}_{t-L} \text{ (seasonal smoothing),}$$

where $0 < \alpha, \beta, \gamma < 1$ are the smoothing parameters and L is the season length. The forecast for the next time frame is then given by equation (6).

$$y_t = \left(\overline{R}_{t-1} + \overline{G}_{t-1}\right)\cdot \overline{S}_{t-L} \tag{6}$$

This method also allows for predictive horizon forecasting *T* time increments ahead. This forecast is given by equation (7):

$$y_{t+T} = \left(\overline{R}_{t-1} + T \cdot \overline{G}_{t-1}\right)\cdot \overline{S}_{t+T-L} \tag{7}$$

The initial conditions are given by $\overline{G}_0 = \dfrac{\overline{y}_m - \overline{y}_1}{(m-1)\cdot L}$, $\overline{R}_0 = \overline{x}_1 - \dfrac{L}{2}\overline{G}_0$ and

$\overline{S}_t = \dfrac{\overline{x}_t}{\overline{x}_i - [(L+1)/2 - j]\cdot \overline{G}_0}$, where $\overline{x}_i$ is the average for the season corresponding to the time increment *t* and *j* is the index of time increment *t* within the season.

For additive seasonality, the recursive algorithm is defined as the set of equations represented as equation (8):

$$\overline{R}_t = \alpha \cdot \left(y_t - \overline{S}_{t-L}\right) + (1-\alpha)\cdot\left(\overline{R}_{t-1} + \overline{G}_{t-1}\right) \text{ (overall smoothing)}$$

$$\overline{G}_t = \beta \cdot \left(\overline{S}_t - \overline{S}_{t-1}\right) + (1-\beta)\cdot \overline{G}_{t-1} \text{ (trend smoothing)} \tag{8}$$

$$\overline{S}_t = \gamma \cdot \left(y_t - \overline{S}_t\right) + (1-\gamma)\cdot \overline{S}_{t-L} \text{ (seasonal smoothing)}$$

The forecast for the next time frame is then given by $y_t = \overline{R}_{t-1} + \overline{G}_{t-1} + \overline{S}_{t-L}$.

Note that the smoothing parameters $0 < \alpha, \beta, \gamma < 1$ are best estimated by genetic algorithms that minimize the fitness function given by the forecast mean absolute percentage error (MAPE). Genetic algorithms employed by the *OptiRamp* Scheduling Submodule are described in the next section.

Predictive Modeling

The *OptiRamp* Scheduling Submodule uses some of the predictive modeling algorithms employed in the *OptiRamp* Model Construction Submodule to create regression forecasts based on predictors other than time series variables. Given the forecast target variable, ordinary least squares and maximum likelihood estimations are used to estimate the autoregressive moving average (ARMA) and the autoregressive moving average with exogenous inputs (ARMAX) coefficients. In particular, the linear version of the *ARMAX(p, q, b)* model can be defined as shown in equation (9):

$$X_t = \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \sum_{i=1}^{q} \eta_i d_{t-i} \,, \tag{9}$$

where the first sum is the pth order autoregressive component, the second sum is the qth order moving average, and the third sum is the exogenous variable time series. The autoregressive component measures the dependence of a series value on its past values, while the moving average component uses errors between series and forecast values to estimate the series' next value.

This portfolio of tried-and-true forecasting methods allows the *OptiRamp* Scheduling Submodule to create robust and accurate production forecasts.

Genetic Algorithms

The Scheduling Submodule employs genetic algorithms to find optimal solutions to efficient operating mode problems as well as forecast parameter search problems.

Genetic algorithms mimic nature's selection process in order to find the "fittest" solution. The algorithm is based on two critical components:

- Solution space coding
- Fitness function identification

In case of an optimal operating mode search, the solution space can be coded as a binary string, with ith position = 0 meaning "ith unit off" and ith position = 1 meaning "ith unit on." The solution space power shows factorial growth with the number of units available for optimization. The fitness function is defined as the optimization objective—either production maximization or cost minimization.

Given N operating units, the algorithm begins by choosing a random sample of size M from the solution space. Each element of the sample is an N-dimensional binary vector $\vec{S}_i$, $i = 1,..., N$. The fitness function can then be evaluated as $F_i = F\left(\vec{S}_i, X_i\right)$, where X_i is a vector of exogenous variables that take effect depending on the selected solution $\vec{S}_i$. An iterative process follows, choosing certain solutions for crossover or mutation to create children solutions until a stopping criterion is met.

The *OptiRamp* Scheduling Submodule accommodates the following selection methods of parent solutions. The fitness proportionate selection method defines probability of selection much like a roulette chance: $p_i = \frac{F_i}{\sum_{j=1}^{M} F_j}$. The tournament selection method creates random subsets of the initial solutions, and the "winner" of each tournament is defined by $\max F_i$ (or $\min F_i$), with r^th^ tournament place probability given by $p \cdot (1-p)^{r-1}$ and p being the probability of first place.

Once the fittest subset of the initial population is selected for breeding, a genetic algorithm proceeds to execute crossover and mutation operations on the parent solutions. Crossover is achieved by taking two (or more) parents and recombining their binary strings, which results in two (or more) children solutions. Mutation can be defined by randomly switching off a single operating unit $\vec{S}_i$ (single point mutation) or rearranging operating unit modes (swapping, inversion, or scramble). Visually, these operators can be described in Figure 2.

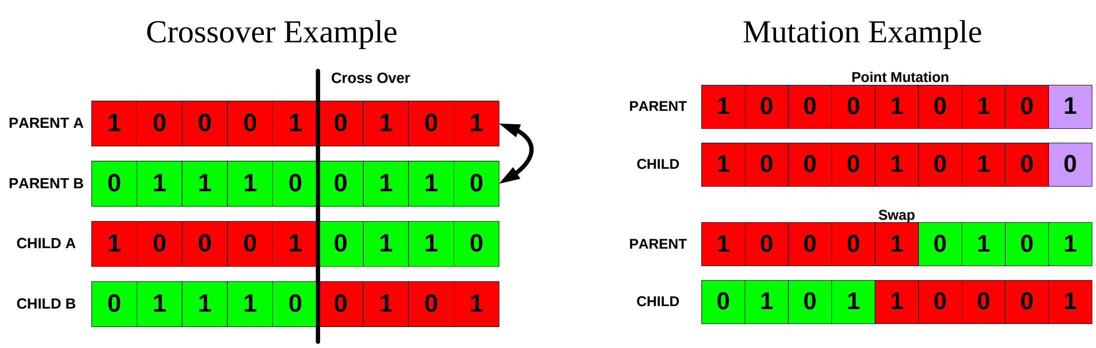

Figure 2. Crossover and mutation operations

To achieve both operations, the *OptiRamp* Scheduling Submodule uses a robust random number generator.

The resulting population is evaluated for fitness, and the process continues until one of the following criteria is reached:

- A solution is found that satisfies the objective function
- New generation does not produce a better fit (within parameter $\varepsilon > 0$)
- Max number of population generation iterations is exceeded

- Allocated resources are exceeded

The *OptiRamp* Scheduling Submodule genetic algorithm implementation is designed to recalculate the optimal solution in real time based on changes in the current operating and market conditions. The submodule uses the alert system to continuously monitor current conditions and to make automated retraining decisions based on the variance in attributes impacting the fitness function.

Alert System

The *OptiRamp* Scheduling Submodule alert system provides automated alerts based on system- and user-defined events. The system events are defined as those that impact any of the Scheduling Submodule's components. All field variables involved in exogenous forecasting and efficient operating mode search are monitored using statistical process control (SPC) methods. Continuous variable distributions are built and alerts are generated if the variable value is outside of $n\sigma$ control limits, where *n* is a positive integer. Alerts are further color coded to denote the severity of a detected change. Outlier detection is employed to distinguish between rarely occurring extreme values and an underlying process change.

Discrete variables are monitored through a series of transformations, where binary indicators are created for each statistically significant category and are individually tracked. Furthermore, a clustering data mining technique is applied to collapse categories into highly discretized field variables for ease of tracking and visualization. Binary indicator means are measured in parametrically defined time intervals to determine whether the process is in control.

System alerts are also used to automatically recalibrate forecasting and efficient operating mode search algorithms so that forecasts are kept as accurate as possible according to the MAPE function (discussed in the Forecasting section) and so that the optimization reflects the most current process changes.

The *OptiRamp* Scheduling Submodule gives the user/operator flexibility to define events based on any attribute used within the system. In addition to standard mathematical operators, such as LT, LE, GT, GE, and EQ, the system provides logic operators, such as AND, OR, XOR, and nested IF-THEN statements. The user/operator can learn about the occurrence of a user-defined event via HMI.

About Statistics & Control, Inc.

S&C—an engineering consulting and technology company headquartered in West Des Moines, IA—solves complex challenges for customers through its unique technology and its highly seasoned team of professionals. The company has a global portfolio spanning the energy, oil and gas, utility, and digital oil field industry sectors. S&C provides clients with turbomachinery control solutions that easily integrate with the existing system as well as *OptiRamp*® solutions, which focus on process and power analytics to optimize processes and, in turn, reduce costs and increase reliability. S&C also provides consulting, dynamic system studies, modeling, automation, training and OTS, and support services.

Statistics & Control, Inc.
4401 Westown Pkwy, Suite 124
West Des Moines, IA 50266 USA
Phone: 1.515.267.8700
Fax: 1.515.267.8701