

# Feasibility Analysis in Batch Process: A Machine Learning Approach

Piyush Kumar Kumawat, Nitin Dutt Chaturvedi\*

Process Systems Engineering Lab, Department of Chemical and Biochemical Engineering, Indian Institute of Technology Patna, Bihta, 147004, Patna, Bihar India  
[nitind@iitp.ac.in](mailto:nitind@iitp.ac.in)

Scheduling is a major issue in process operations, and it is essential for maximizing production output. Short-term scheduling is used in batch operations to allocate a set of restricted resources across time to manufacture one or more items according to a batch recipe. For such processes, a feasible design space can be described by the decision-maker as per the process constraints and objectives. A machine learning approach using support vector machines is implemented for the classification of the feasible region from a data set. An algebraic equation is obtained using support vectors (data points participating in creating classification boundary), which could be incorporated into the mathematical model. The equation will lead to attaining several objectives in the finite obtained region. In this paper, kernel-based support vector classification is tailored to a continuous-time formulation for the short-term scheduling of batch processes. First, using a classification SVC over a data set, it was observed that the mathematical model is infeasible for a set of production combination of two different products. Using the proposed methodology, a design space is presented graphically for feasible production targets. In the second case, the study is applied to perform optimization operations to target feasible region within a specified limit as a choice of decision-maker.

## 1. Introduction

Batch processing is widespread in the process industry, especially when specialized manufacturing is required. (e.g. pharmaceuticals, food, bio-chemicals and fine chemicals etc.). Objectives like minimising costs, making a profit or maximising profit can be achieved by optimum scheduling for a given time horizon. The mixed-integer linear programming (MILP) model for short-term production scheduling was proposed by Ierapetritou and Floudas (1998). The aim was to assess the optimum production schedule using the resources available over a given time horizon while meeting the production requirements at the end of the span. For short-term production scheduling, substantial progress has been made in the domain of short-term batch process scheduling, including the solution of industrial-sized problems. Mendez et al. (2006) presented a study of optimization strategies for batch process short-term scheduling. Kumawat and Chaturvedi (2021) proposed a formulation to handle uncertainties in the batch process for resource targeting.

In the past few years, data-driven techniques were implemented to the problems related to process systems engineering (PSE), focused on coupling data-based methodologies and classical optimization methods. These data-driven approaches are widely used for targeting uncertainties using stochastic and robust optimization methodologies (Gabrel et al, 2014). These techniques are also extended to define a design space and to analyse the feasibility of the process systems. Boukouvala et al. (2010) defined a design space of pharmaceutical processes using data-driven-based methods. Lisa and Ierapetritou (2019) presented a feasibility analysis for the integration of planning and scheduling problems.

Recently, Lee et. al (2018), discussed the recent progress in Machine Learning (ML) and predictive analytics by transforming that data into useful predictions and implications for the PSE. Grossmann and Harjunkoski (2019) discussed the future perspectives of ML applications. They also stated that ML and optimization are closely related to each other and would be valuable to PSE.

In this paper, the feasibility analysis of batch process network is discussed using support vector machines (SVM) classification, an ML technique. This work's ideology is based on projecting the STN feasible area onto the

space of production objectives. A set of algebraic equations for the classification of the data is developed and incorporated into the scheduling problem. Initially, the feasibility of the mathematical model of STN is obtained for different production demand. Further, it is extended to profitability analysis, targeting a feasible region for apposite profit and performing optimization in the generated region. Linear relationships will be used to approximate raw material and end product costs, while methodologies for determining the viability of production objectives will be thoroughly studied.

## 2. Problem Statement

Given the production recipe for each product, available units with their capacities, a time horizon of interest and job durations, for a batch process. A MILP formulation is modelled in order to obtain an optimal schedule w.r.t the objective.

The aim of this paper is to obtain a data-driven feasible region for different production targets using ML classification. The decision-maker can describe a feasible design space relying on the process constraints, production objectives etc. The proposed methodology is extended to perform optimization in the favourable design space.

## 3. Methodology

The methodology presented to solve the defined problem is elucidated in three sections. In section 3.1, the mathematical model for STN scheduling is presented. Section 3.2 explains the basic mathematics linked with the SVM algorithm, which required a classify the data. Section 3.3 presents the implementation of SVM to the batch process.

### 3.1 Batch Scheduling

The mathematical model for targeting maximum profit to be generated by processing raw materials to finished products if formulated.

$$\sum_{s_{in} \in S} y(s_{in}, j, n) \leq 1 \quad \forall j \in J, n \in N \quad (1)$$

$$W_{min} y(s_{in}, j, n) \leq \sum_{s_{in} \in S} m(s_{in}, j, n) \leq W_{max} y(s_{in}, j, n) \quad \forall j \in J, n \in N \quad (2)$$

$$\sum_{s_{in}} m(s_{in}, j, n - 1) = \sum_{s_{out}} m(s_{out}, j, n - 1) \quad (3)$$

$$q(s, j, n) = q(s, j, n - 1) + \sum_{s_{in}} m(s_{in}, j, n) - \sum_{s_{out}} m(s_{out}, j, n) \quad \forall s_{in} \in S_{in}, s_{out} \in S_{out}, j \in J, n \in N \quad (4)$$

$$T_p(s_{out}, j, n) = T_p(s_{out}, j, n) + \alpha(s_{in}, j) y(s_{in}, j, n) + \beta(s_{in}, j) m(s_{in}, j, n - 1) \quad (5)$$

$$\forall s_{in} \in S_{in}, s_{out} \in S_{out}, j \in J, n \in N$$

$$T_p(s_{in}, j, n) \geq T_p(s'_{out}, j, n) \quad \forall s \in S, j \in J, n \in N \quad (6)$$

$$T_p(s, j, n) \leq H \quad \forall s \in S, j \in J, n \in N \quad (7)$$

$$q(s, j, n) \leq Q_u(s, j) \quad \forall s \in S, j \in J, n \in N \quad (8)$$

Objective:

$$\max \sum_p \sum_{s \in S} \sum_{j \in J} \sum_{n \in N} CP(p) m(s_p, j, n) - \sum_r \sum_{s \in S} \sum_{j \in J} \sum_{n \in N} CR(r) m(s_r, j, n) \quad (9)$$

To maximize the overall profit is objective in the above formulation. Here, the allocation restrictions are given by Eq(1), which states that at any event point 'n', only one of the tasks can be completed in each unit. The capacity limitations of production units and storage and are expressed by the constraints given in Eq(2). Eq(3) states the material balance around a particular unit 'j'. Eq(4) implies a general mass balance for the amount of state 's' stored at a time point 'n'. Similarly, the constraint is given in Eq(5) are express the duration constraint. Similarly, Eq(6) denotes a sequence constraint: a state 's' can only be utilized in a single unit, and only after all preceding states have been processed (if 's<sub>out</sub>' is a previous state of 's<sub>in</sub>'). Eq(7) constrains the time horizon of interest that the appearance of all states should be within it. Eq(8) describes the limit of the maximum allowed amount of state 's' in unit 'j'. Eq(9) is the objective of the MILP model, the first term denotes the profit generated from the finished products

A and B. Second terms represent the total cost of the three raw materials. Here  $CP_p$  and  $CR_r$  are the cost/ton of the finished products and raw materials respectively.

### 3.2 Classification using Support Vector Machines

SVM is a machine learning model that can perform linear and nonlinear classification, regression, and outlier identification in high-dimensional space, SVM constructs a hyperplane or sequence of hyperplanes with the maximum distance from each class's nearest training data points. This results in a strong separation and the larger the margin, the smaller the classifier's generalization error.

Given a set of training vectors  $x_i \in \mathbb{R}^p$ ,  $i = 1, 2, \dots, k$  in two classes, and an output vector  $O \in \{-1, 1\}^k$ , the goal is to find  $w \in \mathbb{R}$  and  $b \in \mathbb{R}$  as a result of which the prediction made by  $\text{sign}(w^T \phi(x) + b)$  is correct for the vast majority of samples. Support Vector Classification (SVC) solves the following primal problem (9-11):

$$\min_{w, b, \varepsilon} \frac{1}{2} w^T w + C \sum_{i=1}^k \varepsilon_i ; s. t. \quad (10)$$

$$O_i (w^T \phi(x) + b) \geq 1 - \varepsilon_i \quad (11)$$

$$\varepsilon_i \geq 0, \forall i \quad (12)$$

Intuitively, the goal is to maximize the margin (by minimizing  $\|w\|^2 = w^T w$ ). Also, when a sample is misclassified or falls within the margin border, a penalty is incurred. Ideally, the equation  $O_i (w^T \phi(x) + b) \geq 1$  would be true for all data, which indicates a perfect prediction. Because the issues aren't always fully separable with a hyperplane, some samples are permitted to be separated from their proper margin boundary ' $\varepsilon_i$ '. the penalty term C works as an inverse regularization parameter, controlling the strength of the penalty. The dual of the primal problem (12-14) is:

$$\min_{\delta} \frac{1}{2} \delta^T Q \delta - e^T \delta; s. t. \quad (13)$$

$$Q^T \delta = 0 \quad (14)$$

$$0 \leq \delta_i \leq C, \quad \forall i \quad (15)$$

where ' $e$ ' is an all-ones vector, and ' $Q$ ' is a  $(k \times k)$  positive semidefinite matrix  $Q_{uv} = z_u z_v K(x_u, x_v)$ , where  $K(x_u, x_v) = \phi(x_u)^T \phi(x_v)$  is the kernel. The terms  $\delta_i$  are called the dual coefficients, and they are upper-bounded by ' $C$ '. This dual representation highlights that the training vectors are implicitly transferred into a higher dimensional space by the function is highlighted by this dual representation ' $\phi$ '. Commonly used kernel functions are the polynomial kernel, radial basis function (RBF) kernel with bandwidth and sigmoid kernel. (Amrani et. al, 1999)

The output of the decision function for a given sample ' $x$ ' becomes as follows Eq(16) and the predicted class correspond to its sign.

$$\sum_{u \in SV} z_u \delta_u K(x_u, x) + b \quad (16)$$

It should be noted that the sum of the support vectors will consist of the samples that lie within the margin because the dual coefficients are zero for the other samples.

### 3.3 Implementation of SVC in the batch process

To obtain the algebraic equation of classification to describe the feasibility of production targets, it is assumed that data in the form of historical information of the scheduling problem is available. In this case, the scheduling model (1-8) is used to generate data of different production targets and their corresponding feasibility information. The steps of the methodology are presented as a flow chart in Figure 1 and explained below:

**Step 1:** The data set with different production targets  $\{P_A, P_B\}$  as input and classifying each instance as feasible  $[-1]$  or infeasible  $[1]$ .

**Step 2:** The collected data is now divided into two different sets, one to train the data using SVM classifier, and the second as a test set to check the fitting of the model.

**Step 3:** Train the data to create an ML model with the appropriate kernel function and parameters.

**Step 4:** Model fitting can be judged based on the classification report of the test data. Better the classification report better the fitting of the trained data. For the improper fitting of input parameters, step 3 should be implemented again with modified parameters.

**Step 5:** A graph for the feasible region generated from the trained data can be obtained. It will assist the decision-maker to choose feasible production targets.

**Step 6:** Using the model coefficients i.e. dual coefficients, support vectors and intercept, an algebraic equation for the classifier can be obtained.

**Step 7:** The algebraic equation can now be used and incorporated into the mathematical model.

The resultant model with Eq(16) as an additional constraint will now only be viable for the feasible region of the production space. Various objectives and optimization problems can be achieved and will be feasible for the defined area by SVC

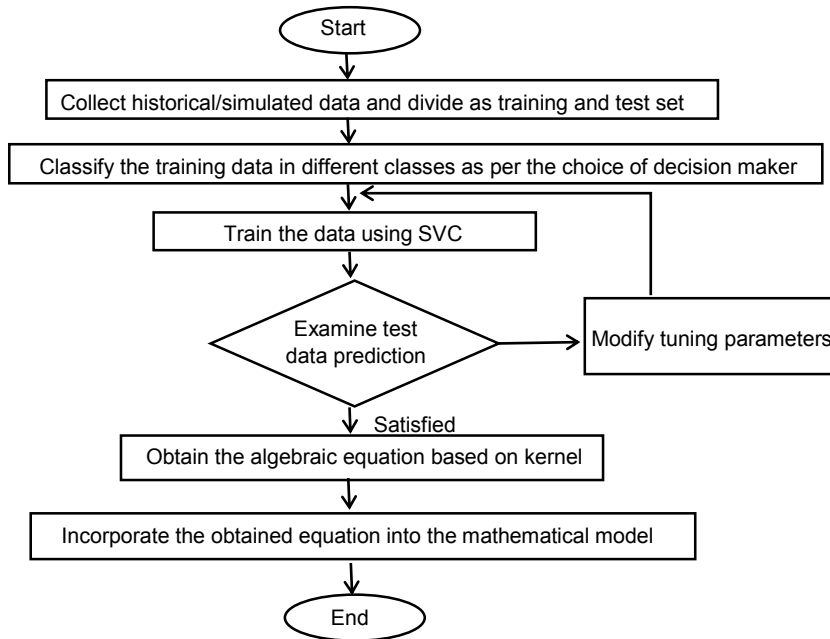


Figure 1. Flow chart for the proposed methodology

**4. Illustrative Example**

To demonstrate the methodology, an example is adapted from Ierapetritou and Floudas (1998), it involves the production of two products using three raw materials (Figure 2a) with six event points. The maximum production that can be achieved is calculated to be 139.75 kg comprised of 87.75 kg of product A and 52 kg of product B in 8 hours; as the time horizon of interest (H). In this paper, the data set for 120 inputs are generated using a MILP model (1) - (9) for model feasibility. The mixed-integer linear programming (MILP) model is solved in GAMS 28.4.2 using CPLEX solver. The feasibility analysis is performed and analysed for two different cases.

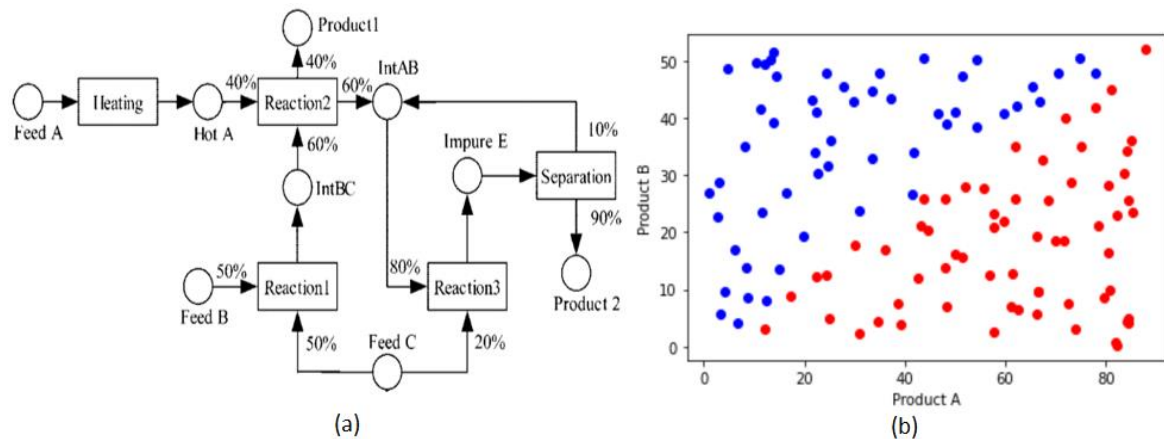


Figure 2. (a) State Task Network (STN) representation of the batch process (b) simulated data points

Case 1: Feasible region for STN

A set of historical/simulated data is required as different production targets for feasibility analysis. Figure (2b) shows the data of various product target (A and B) and their classification in the form of feasibility; red marks for the infeasible and blue for the feasible points. The input data set is trained using SVC algorithm to classify

data on the production space by a hyperplane. It can be observed that the data set can be linearly separable, here, a linear kernel is used for the training model. The classification model is trained in Python 3.0 using the library scikit-learn (Pedregosa, 2011) with default parameters.

A decision boundary is generated based on the input data for the classification of the dataset (Figure 3a). The equation of the hyperplane can be obtained from the support vectors and incorporated into the mathematical model with the objective as the choice of decision-maker. The nature of the resultant model will be MILP with respect to the linear objective. The model has 98 % accuracy for the test set and considered to be reliable for the classification.

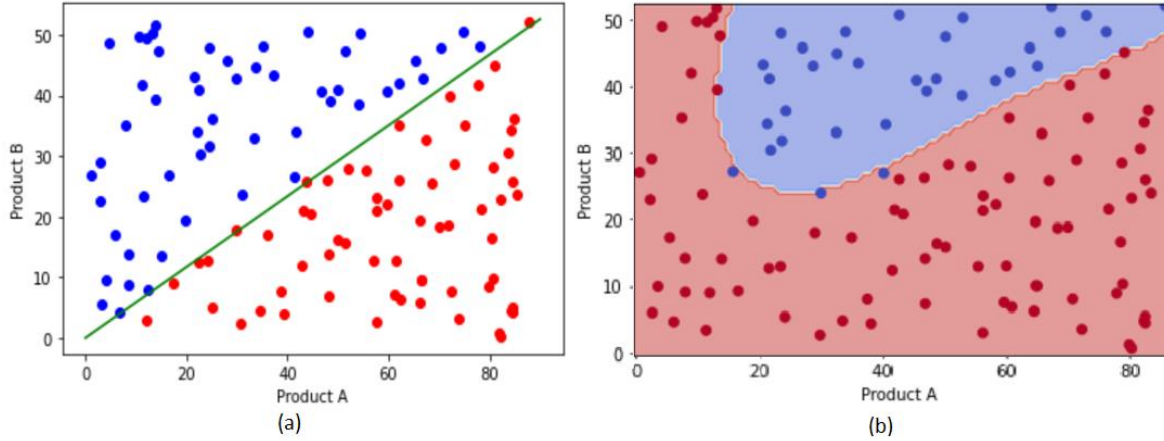


Figure 1. Decision Boundary (a) feasible operations (b) minimum profit

#### Case 2: Targeting minimum profitability

Profit is a major concern for planners and sometimes it is important to target minimum profit to be generated for operating a process. This can be attained using a mathematical expression for profit capping. However, the optimization problem eventually results in a single product combination as a target for the minimization problem. The classification problem will additionally result in a feasible area for the feasible production target considering minimum profit.

RBF kernel is used for the SVC model to classify the nonlinear dataset. The blue region in Figure 3(b) represents the approximate feasible area of production, in order to generate a minimum profit of \$ 200. The algebraic equation can be generated using the sum of the support vectors of the samples that lie within the margin. With an objective to minimize resource requirement cost (Eq(17)), optimization is performed within the feasible region. Eq(18) classifies the feasible region and should be incorporated in the optimization model. Eq(19) limits the minimum and maximum production ( $p^{min}$ ,  $p^{max}$ ) to be produced in the fixed time horizon (H). The resultant MINLP model with constraints (1)-(8), Eqs(17)-(19) and objective (Eq(17)) will only be viable for feasible options as per the choice of the planner for the targeted minimum profit.

$$\min \sum_r \sum_{s \in S} \sum_{j \in J} \sum_{n \in N} CR(r)m(s_r, j, n) \quad (17)$$

$$\sum_{u \in SV} z_u \delta_i \exp(-\gamma \|x_u - P\|) + b \leq 0 \quad (18)$$

$$p^{min} \leq p \leq p^{max} \quad (19)$$

The resultant MINLP model is solved using GAMS/CONOPT solver. The model generates a minimum resource cost of \$ 710.67 within the feasible region (assuring minimum targeted profit).

## 5. Conclusions

In this paper, a data-driven approach is proposed to provide a feasible region of production targets. It comprises a classification method, SVC, to describe the feasible area projected from data-driven classifiers on the product space. The classifiers are then transformed into the planning problem and can also be modified as the choice of the decision-maker. The resultant problem with the integrated classifier would only result in feasible solutions

on the basis of training data. As per the kernel function used for the classification, the nature of the final model may be MILP or NLP. The model is solved for the batch process to produce two products from three raw materials for a fixed time horizon of interest. Classification is performed on simulated data set to obtain a feasible production space. From the case study design space is created to obtain a minimum profit from the batch operation. In future work, the formulation could be extended with demand-based production planning and multi-facility planning problems to satisfy the demand.

### Nomenclature

#### Sets

$$J = \{j | j = \text{unit}\}$$

$$N = \{n | n = \text{event point}\}$$

$$S = \{s | s = \text{any state}\}$$

$$R = \{r | r = \text{raw material}\}$$

$$P = \{p | p = \text{product}\}$$

$$S_{in} = \{s_{in} | s_{in} = \text{input state}\} \in S$$

$$S_{out} = \{s_{out} | s_{out} = \text{output state}\} \in S$$

#### Continuous Variables (Batch Process)

$$m(s, j, n) \quad \text{Amount of state 's' exits or enter to/from unit 'j' at a time point 'n'}$$

$$T_p(s, j, n) \quad \text{Time at which state 's' appears in unit 'j' at event point 'n'}$$

$$Q_u(s, j) \quad \text{Maximum allowed amount of state 's' in unit 'j'}$$

$$q(s, j, n) \quad \text{Amount of state 's' in unit 'j' at time point 'n'}$$

#### Binary Variable (Batch Process)

$$y(s, j, n) \quad \text{Usage of state 's' in unit 'j' at time point 'n'}$$

#### SVM Notations

$$i \quad \text{Training vector indices}$$

$$u \quad \text{Set of support vectors } \in i$$

$$w \quad \text{Weight}$$

$$b \quad \text{Bias}$$

#### Batch Process Parameters

$$\alpha \quad \text{Constant coefficient of processing time}$$

$$\beta \quad \text{Variable coefficient of processing time}$$

$$G \quad \text{Fixed production demand}$$

$$H \quad \text{Time horizon of interest}$$

$$CR(r) \quad \text{Cost of raw material } r$$

$$CP(P) \quad \text{Cost of product } P, \text{ where } P \in P_A, P_B$$

### Acknowledgements

The authors would like to thank Science and Engineering Research Board (SERB), India and Indian Institute of Technology, Patna for providing the research funding under grant no. ECR/2018/000197.

### References

- Amrani S., Wu S., 1999, Improving support vector machine classifiers by modifying kernel functions, *Neural Networks*, 12, 783-789.
- Boukouvala F., Muzzio F.J., Ierapetritou M.G., 2010, Design Space of Pharmaceutical Processes Using Data-Driven-Based Methods, *Journal of Pharmaceutical Innovation*, 5, 119-137.
- Gabrel V., Murat C., Thiele A., 2014, Recent advances in robust optimization: An overview, *European Journal of Operational Research*, 235, 471-483.
- Grossmann I.E., Harjunkoski I., 2019, Process Systems Engineering: Academic and industrial perspectives, *Computers and Chemical Engineering*, 126, 474-484.
- Ierapetritou M.G., Floudas C.A., 1998, Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes, *Industrial & Engineering Chemistry Research*, 37, 4341-4359.
- Kumawat P.K., Chaturvedi N.D., 2021, Robust resource targeting in continuous and batch process, *Clean Techn Environ Policy*, 1-16.
- Lee J.H., Shin J., Realf M.J., 2018, Machine learning: an overview of the recent progresses and implications for the process systems engineering field, *Computers and Chemical Engineering*, 114, 111-121.
- Lisa S.D., Ierapetritou M.G., Floudas C.A., 2019, Data-driven feasibility analysis for the integration of planning and scheduling problems, *Optimization and Engineering*, 20, 1029-1066.
- Mendez C.A., Cerda J., Grossmann I.E., Harjunkoski I., Fahl M., 2006, State-of-the-art review of optimization methods for short-term scheduling of batch processes, *Computers and Chemical Engineering*, 30, 913-946.
- Pedregosa F., Varoquaux G., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E., 2011, Scikit-learn: machine learning in python, *Journal of Machine Learning Research*, 12, 2825-2830.