Multiway Interval Partial Least Squares for Batch Process Performance Monitoring

Shallon Stubbs,† Jie Zhang, and Julian Morris*  

Centre for Process Analytics and Control Technology, School of Chemical Engineering and Advanced Materials, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.

Supporting Information

ABSTRACT: The method of interval Partial Least Squares (iPLS) is combined with multiway partial least-squares (MPLS) to allow the building of enhanced statistical process performance monitoring models. A novel algorithm is proposed for segmenting batch duration, or spectral data in applications employing spectroscopy data into several subintervals for which independent PLS models can be constructed. The approach deviates from the method of using subintervals of equal length and the practice of choosing only a subset of these subintervals for prediction and/or monitoring. The proposed approach provides dramatic reduction in the number of subintervals required and subsequently the number of PLS models required to give improved prediction and monitoring performance. The proposed method, MiPLS, is applied to the well-known benchmark fed-batch penicillin production simulator, Pensim, for quality variable prediction and fault detection.

1. INTRODUCTION

Multivariate statistical process control, or process performance monitoring, methods based on multiway principal component analysis (MPCA) and projection to latent structures (PLS) have proven their significant value in batch analysis and monitoring.6−8 These methods have been widely studied and extensively applied across many industries.7−14 They are essentially the only powerful approaches for the analysis and monitoring of batch processes when mechanistic models are not available. This work focuses on the MPCA approach proposed by Nomikos and MacGregor2 as a basis for the extension proposed using interval partial least-squares.

Interval Partial Least Squares (iPLS) is a local modeling methodology quite widely used in the analysis and modeling of spectroscopic data and variable selection to optimize the predictive power of PLS regression models as well as to aid in data interpretation. iPLS has been applied successfully to the spectroscopic monitoring and prediction of quality variables, in particular using mid-infrared and near-infrared spectra. The technique has been reported in applications ranging from pharmaceutical medicinal explorations - measuring the content of flavone an active ingredient in a rare medicinal plant called snow lotus using near-infrared spectroscopy,15 determination of total polyphenols content in green tea using FT-NIR spectroscopy,16 for simultaneous active substance detection in pharmaceutical formulations,17 to the detection of quality parameters in biodiesel/diesel blends,18 and for the detection of contaminants in lubricating oil.19 The principle of the method involves splitting a spectrum into several equidistance subintervals or regions and then developing a PLS model for each subinterval using a selected number of latent variables.

To the authors’ knowledge there have been no studies into how iPLS might contribute to the analysis and interpretation of process malfunctions and faults within a multivariate statistical process performance monitoring scheme. In this paper the iPLS implementation differs from previously published papers in process analytics in two key ways. The first is that the subintervals are not equally spaced, and the interval slicing is optimally selected based upon an algorithm to minimize the overall root mean squared error (RMSE) of prediction. Second, all subintervals are employed as the aim of the model is to support the detection of fault conditions developing at any stage in the batch run which may have adverse impact on the target product quality. The proposed iPLS model is developed and applied to batch-wise unfolded data to provide a Multiway iPLS (MiPLS) monitoring model. The model’s fault detection performance and quality variable prediction capabilities are demonstrated using data generated from the well-known benchmark fed-batch penicillin fermentation simulator20 based on a detailed mechanistic model.21 It is shown that optimal splitting of the intervals can dramatically minimize the number of subintervals required, and subsequently the number of PLS models required, to represent the full duration of the batch process.

The paper is structured as follows. Section 2 reviews several data unfolding methods that have been proposed in the literature and introduces the concept of batch duration interval splicing. The proposed method of combining the two concepts is explained in Section 2.2. Section 3 provides a descriptive overview of the fed-batch penicillin process simulator and describes the development of the MiPLS process monitoring and predictive models employed in the study. Section 4 presents the results and analysis obtained using the proposed process monitoring and prediction models. Section 5 presents some conclusions.
2. COMBINING DATA UNFOLDING AND INTERVAL SPLICING TECHNIQUES

2.1. Three-Dimensional Data Unfolding. MPLS is an extension of PLS to deal with data in three dimension arrays. By three dimensions we refer to variables sampled over a fixed time duration collected across several batches (repeat runs of the same process) as shown in Figures 1 and 2. The MPLS is essentially the equivalent of PLS applied to the unfolded three dimension data set $X (I \times J \times K)$ where $I$ is the number of batches, $J$ is the number of variables, and $K$ represents the number of measurement samples for the duration of the batch run. As shown in Figures 1 and 2 the data can be unfolded in one of three different modes: (i) and (ii) time, (iii) batch, and (iv) variable.

Each of the three general unfolding categories (time, batch, and variable) has two alternative implementations making the total number of two-dimensional structure representations equal to six. However, due to the equivalence of some of the unfolded structures, the number of unique 2-dimensional data set structures reduces to three. For instance, (ii) and (iv) are of the same dimension $(KJ \times J)$ but with different ordering of the rows.

2.2. Combining Data Unfolding and Interval Slicing. The process involves first splitting the batches along identical time slice intervals. Each subinterval is then unfolded according to the unfolding approach (iv) in Figure 2. Data blocks extracted from identical intervals across each batch are combined into $r$ subinterval grouping, and a PLS model is constructed for each subinterval as shown in Figure 3.

Critical to the success of iPLS is the process of arriving at the ideal interval subdivision across all batches or the selection of a subset of the intervals which will minimize the overall RMSE. Chen et al.\textsuperscript{15} used a genetic algorithm to select a subset of the intervals that most efficiently contributes to the prediction of the quality parameter.

In this subsection we describe a novel procedure which has been effectively applied to derive the efficient interval subdivision across all batches. The approach renders the selection of subintervals unnecessary as it will be demonstrated that if the batch, or spectrum, duration is optimally spliced then the number of subintervals required is significantly less than choosing an arbitrary fixed-length subinterval. The algorithm was inspired by an algorithm proposed by Kegl et al.\textsuperscript{22} for constructing principal curves using polygonal line segments. The principal curve algorithm was initialized by using the first linear principal component of the data set. The algorithm then iteratively identifies new segment vertices by adding one vertex to polygonal curve in each iteration step. The new position of each vertex was determined by minimizing an average squared distance criterion penalized by a measure of the local curvature. The stopping criterion is based on a heuristic complexity measure, determined by the number of segments $k$, the dimension of data matrix, and the average squared distance of the polygonal line from the data points.

Figure 4 shows a flowchart of the proposed algorithm, which is initialized by defining first a minimum subinterval length or step size $S_{\text{min}}$. All subsequently selected interval lengths are integer multiples of this minimum interval spacing. The process is best described as an iterative binary subdivision process that terminates when either a predefined maximum number of subdivisions is reached or the algorithm deems that further subdivision of the intervals would not achieve any significant reduction in the RMSE of prediction. To evaluate the degree of success or failure of a given subdivision stage, we define two

Figure 1. Time-mode unfolding methods (i) and (ii).

Figure 2. Batch-mode unfolding (iii) and variable-mode unfolding (iv).

Figure 3. Batch-wise interval segmented unfolding of data set.
measures - the total and segment RMSE, given by eq 1 and eq 2, respectively

\[
RMSET = \sum_{i=1}^{r} \frac{n_i}{N} \text{RMSE}_i
\]

\[
\text{RMSE} = \sqrt{\frac{\sum (\hat{y} - y)^2}{n_i}}
\]

where \(n_i\) is the length (number of samples) of the \(i^{th}\) subinterval, \(N\) is the full length of the batch duration (total number of samples), \(r\) is the total number of segments the batch is currently divided into, and \(\hat{y}\) is the estimated/predicted value of the quality variable measurement \(y\) based upon the PLS model of the \(i^{th}\) subinterval.

It is noted that the total RMSE is computed as a weighted sum of the interval RMSE such that the largest subinterval will have the most influence on the overall value. Identifying the best binary split of any given segment is a sequential process of evaluating all split scenarios as illustrated in the flowchart (Figure 4) and the example shown in Figure 5. The partitioning is adjustment in discrete step size defined by the minimum subinterval. The binary division algorithm compares the segment RMSE value (\(\text{RMSE}_i\)) before a split with the weighted sum of the split version of the interval to determine whether splitting the segment into two parts achieves any improvement over keeping the segment as a single block. The impact of a given subinterval binary splitting on the total RMSE (\(RMSET\)) is also factored-in using a combined weighted criterion computed as the weighted sum of two ratios

\[
\gamma_i = \alpha \frac{\text{RMSE}^{+}_T}{\text{RMSE}_T^+} + (1 - \alpha) \frac{\text{RMSE}^{+}_i}{\text{RMSE}_i^+}
\]

where \(\text{RMSE}^{+}_T\) is the evaluated total RMSE after a given subinterval is split, and \(\text{RMSE}^{+}_i\) is the total RMSE before splitting. Likewise, \(\text{RMSE}^{+}_i\) is the segment RMSE of the selected

---

**Figure 4.** Flowchart of binary interval splitting algorithm.
interval before it is subdivided, and $RMSE_i^{+}$ is the average segment RMSE after the segment is split into two subparts. The weighting parameter $\alpha$ takes on values over the range from 0 to 1, setting the parameter closer to 1 allows more emphasis to be placed on improving the total RMSE of the fit. By design, therefore, $\gamma$ will take on values less than 1 if a splitting of the data set along a given time line reduces the overall RMSE and will be equal to or greater than 1 if an experimental splitting of the batch fails to improve the prediction fit.

Based upon the criterion of eq 3, the algorithm is biased to favor splitting of the largest subinterval in any given stage since a registered improvement in the segment RMSE resulting from binary splitting exercise will impact on the RMSET value more if that subinterval represents a large proportion of the total batch duration. Therefore, as indicated in the flowchart (Figure 4), the algorithm will first target the largest subinterval and may either revert to the previous state if the split is rejected or proceed, in either case targeting the next largest subinterval for splitting. After the optimal subdivision of the batch duration is computed, the final stage of the process involves developing a PLS model to fit each interval

$$E_{k+1} = X - \sum_{i=1}^{k} T_i p_i$$
$$F_{k+1} = Y - \sum_{i=1}^{k} U_i q_i$$

(4)

where $k$ is the number of latent variables used, $X$ is the process measurements, and $Y$ is the process quality variables. The process measurement and quality variable are therefore simultaneously decomposed into a set of scores $T$ and $U$ using a matrix of loading vectors $p$ and $q$, respectively. A linear inner relation between the scores ensures that the scores of $X$ are generated to be most correlated while being highly predictive of $Y$.

The binary division interval splicing algorithm was first evaluated using simulated data defined by eq 5 where the coefficients $a_1$ and $a_2$ were designed to take on different parameter values over the different time intervals indicated.

$$y(t) = a_1 x_1(t) + a_2 x_2(t) + n(t)$$

$$\begin{align*}
  a_1 &= 0.75; a_2 = 0.15; \quad 0 < t \leq 100 \\
  a_1 &= 0.10; a_2 = 0.75; \quad 100 < t \leq 160 \\
  a_1 &= 0.35; a_2 = 0.35; \quad 160 < t \leq 225 \\
  a_1 &= 0.50; a_2 = 1.25; \quad 225 < t \leq 300
\end{align*}$$

(5)

Subsequently, the linear relationship between the response variable $y$ and predictor variables $x_1$ and $x_2$ is dependent upon the current time or sample instance. The data are corrupted with random noise $n(t)$ with zero mean and variance equal to 10% of the true measurement variances. Three different scenarios were considered and are summarized in tabular form which is available as Supporting Information (simulation results evaluating the capability and accuracy of proposed interval division algorithm) where $S_{\text{min}}$ is the minimum subinterval length. Ten batches of data were generated each spanning a batch duration of 300 samples. A value for $S_{\text{min}} = 15$ was selected. Multiway unfolding and interval splitting algorithm applied. Figure 6 shows the prediction fitting evolution for the response variable $y$ as the algorithm discovers the optimal splicing of batch data set range. The RMSE reduction with discovery of the intervals progresses as follows: (a) Total RMSE = 0.28403, (b) Total RMSE = 0.13022, (c) Total RMSE = 0.050297, and (d) Total RMSE = 0.028955. The vertical blue lines in each of the plots from (a) to (d) indicate the intervals in the order they are discovered (identified) by the proposed binary division algorithm.

Figure 5. An illustrative example of binary division algorithm for interval splitting.
The algorithm gave the best results when the parameters $\alpha$ and $\gamma_{\text{max}}$ were assigned values in the ranges $0.6 < \alpha < 1.0$ and $0.8 < \gamma_{\text{max}} < 0.9$, respectively. If $\gamma_{\text{max}}$ is chosen too close to the value of 1.0 and/or $\alpha$ chosen to be 0.5 or less, the binary division algorithm can be too sensitive to provide reduction in $\text{RMSE}_i$ and had the tendency of overshooting the actual number of segment intervals required in some cases.

3. FED-BATCH PENICILLIN SIMULATOR PREDICTION AND FAULT MONITORING

3.1. Fed-Batch Penicillin Production Process Simulator Overview. Commercial quantities of secondary metabolites are produced using filamentous micro-organisms. Formation of the target product (here penicillin) occurs after the cells have grown, and it is common practice to first grow the micro-organisms in a batch culture and then to promote the synthesis of the antibiotic by means of a fed-batch operation. Process data were generated using the PenSim simulator. The input variables are substrate feed rate and substrate feed temperature; the manipulated variables are acid/base and heating/cooling water flow rates; the internal state variables are culture volume, generated heat, carbon dioxide, dissolved oxygen, biomass, penicillin, and substrate feed concentrations; and the controlled variables used are pH and bioreactor temperature with glucose addition being carried out in open-loop. The typical offline measurements of biomass and penicillin concentration were assigned as the process quality variable for which the predictive model was developed.

The relationship between these variables is characterized by nonlinear dynamic equations, and the process is multistage in nature. In the initial preculture stage most of the necessary cell mass is usually generated after which penicillin production commences at the exponential growth phase. Penicillin production then continues until cell growth reaches the stationary phase. To ensure high penicillin productivity a minimum cell growth rate is maintained by feeding glucose continuously into the system during cultivation instead of applying it all at once. The duration of each batch is 400 h, comprising a preculture stage (about 45 h) followed by a final fed-batch stage. All batches are of the same duration, and measurement samples are collected with a sampling interval of 0.5 h making each batch essentially comprising of 800 samples (400 h). The set points for the controlled variables and initial conditions of the input variables are tabulated and available in the Supporting Information. The proposed MiPLS model was then used to detect faults in a simulated fed-batch penicillin cultivation process. Twenty batches were used for each of model training and validation and five ‘unseen’ batches for testing and evaluation of each of the various simulated fault conditions.

3.2. Prediction and Process Monitoring. The input/output structures of the predictive and process monitoring models are shown in Figure 7. The PLS2 predictive model estimates the quality variables – biomass and penicillin concentrations using eleven measured process variables. The same set of input variables was employed to develop the process monitoring model. The process monitoring model is built using bootstrap aggregation regression of PLS1 models (see later discussion) with each PLS1 model providing an output that is representative of the estimation of one of the six selected online measured variables (substrate concentration, dissolved oxygen concentration, CO$_2$ concentration, pH, reactor temperature, and generated heat) from the eleven predictor variables. Estimation or reconstruction of a given
output variable is based on the remaining variables or subset thereof consisting of input variables, control, and manipulated variables. Along with employing an interval PLS approach, the model development also incorporates variable selection to identify the minimum subset of the ten remaining process variables that is most reflective of the particular output variable being estimated.

Table 1 shows details of each of the iPLS1 process monitoring models, indicating the interval segmentation evolved by the proposed binary splitting interval algorithm. The selected variables used by each PLS1 model as inputs are indicated by their index number in column 2 of the table - ‘predictor variables’. The process variables, for which ‘not used’ appears in the column, are the five external input variables to the process, and as such no PLS models were constructed to reconstruct these variables.

The method of variable selection incorporated with the interval PLS model identifies the minimum subset of predictor variables to apply in the specific model without significantly degrading the prediction RMSE for the particular output variable being estimated. Figure 8 illustrates the design of the variable selection process. The technique uses a search tree algorithm inspired by Bratley’s Algorithm employed in a periodic task scheduling applications to identify a feasible schedule from a set of non-pre-emptive tasks. The process starts out by using all n predictor variables and then branches into n nodes. Each branch node represents a case where one of the predictor variables is removed. The node with the minimum RMSE value is then selected, the other branches are terminated, and the process is repeated again for the selected node. The tree terminates when there is no node among the set of new branches provides any significant improvement in the RMSE value over that of the previous level.

In the development of the PLS models, both with and without variable selection, a bootstrapped PLS regression algorithm was employed. The algorithm incorporation within an iterative variable selection procedure is shown in Table 2.

![Figure 8. Variable selection search tree algorithm.](image-url)
yield further reduction in RMSE the algorithm terminates. The overall algorithm is a nested three loop structure.

It was found that in this particular application study no more than four segments were needed to create the best MiPLS1 model. The PLS1 fault detection model residuals, computed as the difference between the estimated and the actual measured outputs, were then combined to form the monitoring metric. Both the squared prediction error (SPE) statistics and Hotelling’s $T^2$ were experimented with as possible monitoring metrics

$$SPE_k = \sum_{i=1}^{m} e^2_{ik}$$

$$T^2_k = \sum_{i=1}^{m} \theta_{ik}^2 / \sigma_k$$

where $k$ is the sample instance, $m$ is the number of variables for which the residuals are computed, and $\sigma_k$ is a time varying covariance computed and applied as proposed in ref 25. The same monitoring statistics have been used by other process monitoring studies using the Pensim simulator.20,25 It is noted that in this study it was necessary to apply a time dependent variance to compute the $T^2$ statistics due to a difference in variance characterizing the different intervals of the iPLS model.

4. PREDICTION AND MONITORING RESULTS

4.1. Predictive Model Performance. For the PLS2 predictive model alternatives considered for predicting biomass and penicillin concentration, the variable selection algorithm employed predictor variables 3 to 7 and 9 to 11. The evolution and reduction in the overall RMSE through the stages of the binary division algorithm used to identify the boundaries of the intervals employed by MiPLS model is shown in Figure 9. It can be observed that the initial RMSE corresponding to a non-iPLS model, that is, a standard MPLS approach, is dramatically improved when the final MiPLS model is evolved. In all cases studied the interval splicing achieved reduction of the initial RMSE to less than 30% of its initial value.

The plots in Figure 10 show the reconstruction/prediction of biomass concentration and penicillin concentration when the final MiPLS2 model is employed (stage 5 of Figure 9), in comparison to a conventional PLS model employing no interval segmentation (stage 1 of Figure 9). The vertical blue lines highlight the intervals identified and employed by the MiPLS algorithm (note that the identified ‘sample’ intervals are converted into batch times). The results were obtained using five ‘unseen validation batches’ not used in the initial model building. It is interesting to observe that the intervals discovery has located intervals that reflect the phase nature of the fermentation, namely the preculture phase of around 45 h but which can vary from batch to batch, followed by about 355 h of the fed-batch stage.

4.2. Process Performance Monitoring. In order to investigate the potential of MiPLS to the detection of both abrupt and incipient (subtle) faults, the three process malfunctions were simulated, and the detection capabilities of the SPE and Hotelling’s $T^2$ statistics monitoring statistics were evaluated. For each type of fault and occurrence (abrupt and incipient), four batches were generated with different fault commencement times (50, 100, 150, and 200 h). Fault 1 simulated a sudden pH controller failure, and fault 2 simulated both a sudden decrease in substrate flow rate of 15% and a gradual fault with a rate of decrease of 0.002% of the nominal value of the variable per sample (0.5 h). Fault 3 simulated both sudden and incipient faults using the same magnitudes as for fault 2 for the case of decrease in agitator power. The abrupt faults were all readily detectable performing equally as well as conventional multivariate statistical process control technologies studies by others using the Pensim simulator, and consequently only the results of the incipient subtle faults are discussed here.

Figure 11 compares the fault detection performance of incipient faults 2 and 3 initiated at time $t = 150$ h. The left
vertical bar indicates the time of fault initiation, the right vertical bar indicates the time of fault detection, and the horizontal line represents the control limit. It can be observed that the detection of incipient fault 3 is achieved more readily than incipient fault 2. The horizontal dashed line represents the 99% control limit.

Figure 12 illustrates the detection of incipient fault 2 for three different fault initiation times 50 h, 100 h, and 200 h. The point of detection of the incipient faults and the detection delay time are indicated by the vertical dotted-lines. For the case of this particular incipient fault the results indicated a trend of reduced detection delay time for fault initiated later in the process. For the other incipient fault simulated the detection delay time was relatively constant. In comparison to fault detection performance of MPCA models presented by Birol et al.20 and Lee et al.,25 the proposed model results in much more effective capturing and representation of the process dynamics and nonlinearities. Instead of just one general PLS model, several models are implemented with the freedom to select the best number of latent variables for each specific model to most accurately capture the dominant process features characterizing the specific interval within the batch duration.

5. CONCLUSIONS

The iPLS implementation presented differs from previously published papers in process analytics in two key ways. The first is that the subintervals are not equally spaced and the interval slicing is optimally selected based upon an algorithm to minimize the overall root mean squared error (RMSE) of prediction. Second, all subintervals are employed as the aim of the model is to support the detection of fault conditions developing at any stage in the batch run and which may have adverse impact on the target product quality. The proposed multiway iPLS (MiPLS) modeling approach combined with the proposed algorithm for identifying the optimal interval subdivision is shown to produce significant reduction in the RMSE over the stand MPLS approach. The usually adopted fixed-interval length iPLS approach typically involves the use of a genetic algorithm to select a subset of the intervals that are best predictive of the quality variable. Employing the binary algorithm has facilitated the use of a much smaller set of define segments/interals to implement the iPLS model. It was demonstrated that optimal selection can significantly reduce the number of segments/interals needed. The proposed binary division algorithm offers an efficient alternative to employing a fix size interval selection approach to the implementation of an iPLS model. It is noted that in the study presented the fault detection capabilities of the proposed MiPLS process monitoring scheme did not provide as dramatic an improvement in comparison to other more conventional approaches and those where wavelet decompositions have been used26–29 but does demonstrate that the MiPLS approach is arguably just as effective and deserving further study. Overall, both the Hotelling’s $T^2$ statistics performed more reliably than the SPE statistics. It is conjectured that this is because it accounts for the difference in the residual variance across the different intervals of the PLS model fit. To prevent false alarm conditions for the SPE statistics a variable control limit over different interval time spans of the batch duration, similar to the control limits employed by Lee et al.,25 could be used.

The studies presented demonstrate the potential of the proposed Multiway interval PLS multiple model approach for a new process performance monitoring method to that using a single PLS model through the increased flexibility and degrees of freedom inherent to the iPLS method. Employing subinterval dividing and the development of independent PLS models for each interval segment provides for more effective capturing and representation of the process dynamics and nonlinearities. Instead of just one general PLS model, several models are implemented with the freedom to select the best number of latent variables for each specific model to most accurately capture the dominant process features characterizing the specific interval within the batch duration.

Figure 11. Detection of incipient faults 2 and 3 using Hotelling’s $T^2$ statistics.

Figure 12. Detection of incipient Fault 2 using $T^2$ statistics: a) fault initiated at $t = 50$ h; b) fault initiated at $t = 100$ h; c) fault initiated at $t = 200$ h.
and other batch equalization methods such as ‘indicator variable approach’ have been widely explored. Only a few articles address the online real time issues although a recent article discusses the challenges and potential solutions. Such solutions could provide the batch data structures required for advanced process monitoring strategies including MiPLS. For batch processes with natural occurring intervals as determined by the batch recipe, one approach would be to build a separate MiPLS model for each recipe-based phase. An alternative approach is where the batch trajectories are subdivided into operating regions, which could be recipe-driven, with a MiPLS model being fitted within each region. These individual models are spliced together to provide an overall nonlinear global model.

■ ASSOCIATED CONTENT

Supporting Information
Tables related to simulation results evaluating the capability and accuracy of proposed interval division algorithm and initial conditions and set points used in the simulation. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author
E-mail: julian.morris@ncl.ac.uk.

Present Address
CAE Group, R&D Centre, OLED Business, Samsung Display Co., Ltd., South Korea. E-mail: shallon76@yahoo.com.

Author Contributions
The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

Notes
The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The work was supported by Dorothy Hodgkin Postgraduate Award and BP International Ltd. The authors acknowledge the School of Chemical Engineering and Advanced Materials and the Centre for Process Analytics and Control Technology (CPACT).

■ REFERENCES

(18) Ferrao, M. F.; Viera, M. D.; Pazos, R. E. P.; Fachini, D.; Gerbase, A. E.; Marder, L. Simultaneous determination of quality parameters of biodiesel/diesel blends using HATR-FTIR spectra and PLS, iPLS or sPLS regressions. Pflu 2011, 90, 701.
(23) Bratley, P.; Florian, M.; Robillard, P. Scheduling with earliest start and due date constraints. Naval Res. Q. 1971, 18, (4).