Contents

List of Figures

List of Tables

1 Introduction

Scheduling is a decision-making process that plays an important role in most manufacturing and service industries. Scheduling problems arise in almost any type of industrial production facilities where given tasks need to be processes using specified resources. In a chemical process, production must be planned such that equipment, material and utilities are available at the manufacturing facility when they are needed to realize the production tasks. Production scheduling comprises the activity of planning in detail the production of a product or products in a given production facility. It boils down to the following main decisions (Harjunkoski et al., 2014):

- What production tasks to execute?
- Where to process the production tasks?
- In which sequence to produce?
- When to execute the production tasks?

For batch processes, short-term scheduling deals with the allocation of a set of limited resources over time to manufacture one or more products following a batch recipe (Méndez et al., 2006). Existing approaches rely on network-bsaed representation. Under the state-task network representation (Kondili et al., 1993), a task node is linked to a state node via an arc if the state is consumed or produced by the task. An example of a state task network is given in Fig. 5.1.

Alternatively, if units, states (materials) and utilities are viewed as resources consumed/produced by the task, then task-nodes are linked to resource-nodes, and the representation is referred to as a resource-task network (Fig. 1.1).

There has been significant development of optimization approaches to scheduling over the last two decades. Early attempts to establish general scheduling algorithms were based on the discretization of the time horizon into a number of intervals of equal and

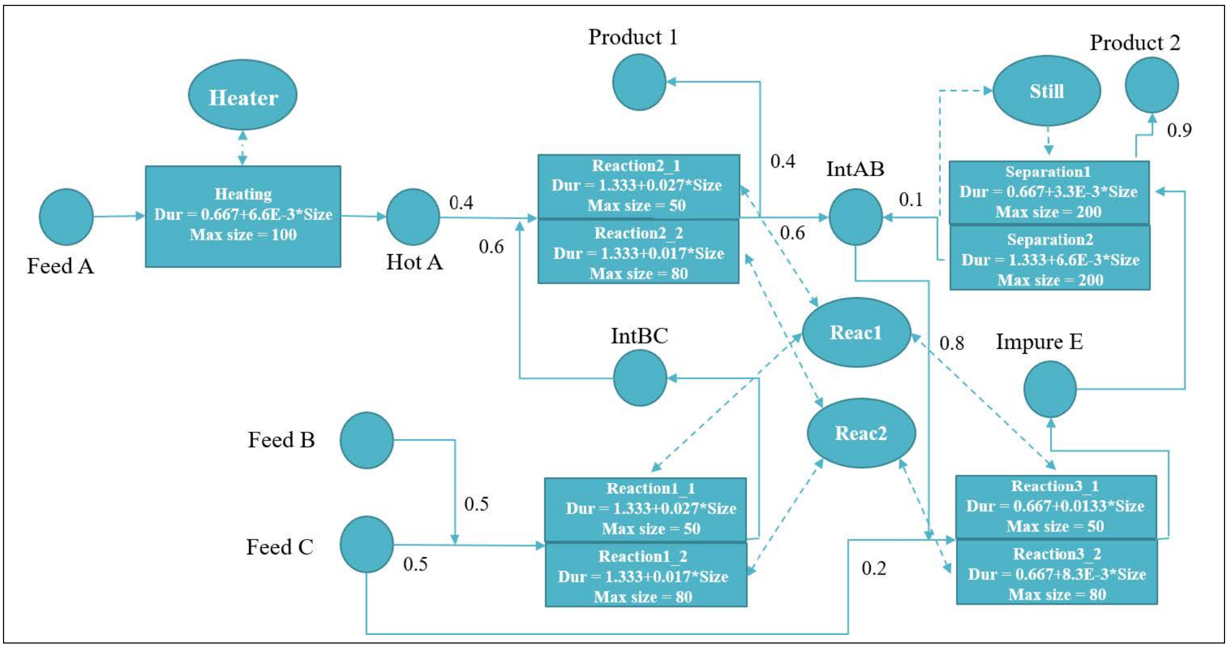


Fig. 1.1: Resource task network

fixed duration. In such a model, all system events are forced to coincide with one of the interval boundaries. Such a mathematical programming approach for the scheduling of multi-purpose, multi-product batch plants was proposed by Kondili et al. (1993). The main limitation of this approach resulted from the fact that the duration of all processing tasks must be a multiple of the discretization time, which implied a formulation with a high number of decision variables for reasonable accuracy for realistic applications. Shah et al. (1993) developed specific solution techniques for the proposed formulation so as to reduce the computational time. However, the discretization problem was still present. This encouraged work towards the development of efficient methods based on continuous-time representations, which was first introduced for scheduling problems by Sahinidis and Grossmann (1991).

Subsequently, several continuous time models for the scheduling of batch plants were developed over the next two decades. We describe five of them in Chapter 3:

1. Castro et al. (2004)

2. Maravelias and Grossmann (2003)

3. Giménez et al. (2009)

4. Ierapetritou and Floudas (1998)

5. Sundaramoorthy and Karimi (2005)

In realistic scenarios, many of the parameters associated with scheduling are not known exactly. Parameters such as processing time, yields, prices, etc. can vary with respect to time and are subject to unexpected deviations. Robust optimization is an approach that has been suggested to mitigate these uncertainties while designing a schedule. Robust optimization seeks to generate a solution that is immune to uncertainty by ensuring that it remains feasible for all possible realizations of the uncertain parameters from within a set chosen *a priori* by the modeler.

This work supports two frameworks to handle uncertainty of parameters:

- **Static Robust Optimization:** The first application of robust optimization in process scheduling was by Lin et al. (2004). This work, which utilized box uncertainty sets, was later extended by Janak et al. (2007) to consider uncertainty sets derived from probabilistic information. Li and Ierapetritou (2008) considered box, ellipsoidal and budget uncertainty sets. All these single-stage approaches are collectively referred to as Static Robust Optimization (SRO).

- **Adjustable Robust Optimization:** SRO approaches are generally conservative, as they assume that all of the decisions have to be made "here-and-now", before the schedule begins to be implemented. In reality, many of the decisions can be "wait-and-see", meaning that they can be delayed until a later point when the a subset of the uncertain parameters have revealed their values. To handle such multi-stage decision making strategies, Adjustable Robust Optimization (ARO) is used, where an optimal policy is derived instead of a single, static solution. The optimal policy constitutes a family of solutions that are parameterized in the uncertain parameter realizations (Lappas and Gounaris, 2016).

2 Problem Statement

Indices

i = tasks

j = units

s = states

u = utilities

Sets

$\mathcal{J}$ = Set of available processing tasks

$\mathcal{S}$ = Set of states (materials)

$\mathcal{S}$ = Set of utilities

$\mathcal{I}$ = Set of processing tasks

$\mathcal{I}_j$ = Set of processing tasks that can be performed in unit j

$\mathcal{I}_s^p$ = Set of tasks that produce state s

$\mathcal{I}_s^c$ = Set of tasks that consume state s

$\mathcal{I}_u$ = Set of tasks that consume utility u

$\mathcal{I}^{zw}$ = Set of tasks that produce a zero-wait state

Parameters

α_i = Fixed processing time of task i

β_i = Processing time of task i per unit batch

γ_{iu} = Fixed consumption of utility u by task i

δ_{iu} = Consumption of utility u by task i per unit batch

ρ_{is} = proportion of state s in the total production/consumption by task i

$B_i^{\min}$ = minimum capacity for task i in unit j

$B_i^{\max}$ = maximum capacity for task i in unit j

$U_u^{\max}$ = maximum availability of utility u

S_{s0} = Initial amount of state s

$S_s^{\max}$ = maximum storage capacity for state s

D_s = Demand of state s at the end of horizon

P_s = Price of state s

H = Horizon

Given the above data for units, states, utilities, tasks, horizon and demands, a feasible schedule has to be obtained in order to either maximize profit within horizon H or minimize the time taken to produce demands D_s for states s.

3 | Scheduling models used

This chapter briefly describes the scheduling models supported by the online optimization tool. All the models are continuous time, mixed-integer linear models and can be readily solved with standard MILP optimization solvers. All models can account for two objectives: maximization of profit and minimization of makespan.

3.1 CBMN 2004

This is a global event-based continuous time process scheduling model that was first introduced by Castro et al. (2004). It uses a resource task network based uniform-time-grid continuous time representation. One of the key aspects of this model is the set of tightening constraints (Equation 3.1.10) which are not necessary for completeness but are typically included so as to improve its linear programming relaxation and overall solution time. Below we present a formulation identical to the one published by Castro et al. (2004), but with slight modeling changes to make it amenable to a State Task Network (STN) representation (Lappas and Gounaris, 2016).

$$\min_{\substack{W,T,B,\\U,S,G,z}} \; z \tag{3.1.1}$$

$$\text{s.t. } z \geq \sum_{s \in \mathcal{S}} P_s (S_{s0} - S_{sN}) \tag{3.1.2}$$

$$z \geq (T_N - T_1) \tag{3.1.3}$$

$$T_{n'} - T_n \geq \sum_{i \in \mathcal{I}_j} (\alpha_i W_{inn'} + \beta_i B_{inn'}) \qquad \forall j \in \mathcal{J}, \forall n \in \mathcal{N}, \forall n' \in \mathcal{N}_n^+ \tag{3.1.4}$$

$$W_{inn'} B_i^{\min} \leq B_{inn'} \leq W_{inn'} B_i^{max} \qquad \forall j \in \mathcal{J}, \forall n \in \mathcal{N}, \forall n' \in \mathcal{N}_n^+ \tag{3.1.5}$$

$$G_{jn} = G_{j(n-1)} + \sum_{i \in \mathcal{I}_j} \left[\sum_{n' \in \mathcal{N}_n^+} W_{inn'} - \sum_{n' \in \mathcal{N}_n^-} W_{in'n} \right] \qquad \forall j \in \mathcal{J}, \forall n \in \mathcal{N} : \{n > 1\} \tag{3.1.6}$$

$$S_{sn} = S_{s(n-1)} + \sum_{i \in \mathcal{I}_s^p} \rho_{is} \sum_{n' \in \mathcal{N}_n^-} B_{in'n} - \sum_{i \in \mathcal{I}_s^c} \rho_{is} \sum_{n' \in \mathcal{N}_n^+} B_{inn'} \qquad \forall s \in \mathcal{S}, \forall n \in \mathcal{N} \tag{3.1.7}$$

$$\begin{aligned} U_{un} &= U_{u(n-1)} \\ &+ \sum_{u \in \mathcal{I}_u} \left[\sum_{n' \in \mathcal{N}_n^+} (\gamma_{iu} W_{inn'} + \delta_{iu} B_{inn'}) - \sum_{n' \in \mathcal{N}_n^-} (\gamma_{iu} W_{in'n} + \delta_{iu} B_{in'n}) \right] \\ &\qquad \forall u \in \mathcal{U}, \forall n \in \mathcal{N} : \{n > 1\} \end{aligned} \tag{3.1.8}$$

$$\begin{aligned} T_{n'} - T_n \leq \bar{M} \left[1 - \sum_{i \ \ in \mathcal{I}_j \cap \mathcal{I}^{\mathrm{zw}}} W_{inn'} \right] \quad + \sum_{i \in \mathcal{I}_j} (\alpha_i W_{inn'} + \beta_{inn'}) \\ \forall j \in \mathcal{J}, \forall n \in \mathcal{N}, \forall n' \in \mathcal{N}_n^+ \end{aligned} \tag{3.1.9}$$

$$\begin{aligned} \sum_{i \in \mathcal{I}_j} \sum_{\substack{n' \in \mathcal{N}: \\ \{n' \geq n\}}} \sum_{n'' \in \mathcal{N}_{n'}^+} (\alpha_i W_{in'n''} + \beta_i B_{in'n''}) \quad \leq T_N - T_n \\ \forall j \in \mathcal{J}, \forall n \in \mathcal{N} \end{aligned} \tag{3.1.10}$$

$$T_N = H \tag{3.1.11}$$

$$S_{sN} \geq D_s \qquad \forall s \in \mathcal{S} \tag{3.1.12}$$

$$0 \leq G_{jn} \leq 1 \qquad \forall j \in \mathcal{J}, \forall n \in \mathcal{N} \tag{3.1.13}$$

$$0 \leq S_{sn} \leq S_s^{\max} \qquad \forall s \in \mathcal{S}, n \in \mathcal{N} \tag{3.1.14}$$

$$0 \leq U_{un} \leq U_u^{\max} \qquad \forall u \in \mathcal{U}, n \in \mathcal{N} \tag{3.1.15}$$

$$W_{inn'} = 0 \qquad \forall i \in \mathcal{I}, \forall n \in \mathcal{N}, \forall n' \in \mathcal{N} \backslash \mathcal{N}_n^+ \tag{3.1.16}$$

$$B_{inn'} = 0 \qquad \forall i \in \mathcal{I}, \forall n \in \mathcal{N}, \forall n' \in \mathcal{N} \backslash \mathcal{N}_n^+ \tag{3.1.17}$$

$$T_1 = 0 \tag{3.1.18}$$

$$G_{jN} = 0 \qquad \forall j \in \mathcal{J} \tag{3.1.19}$$

$$W_{inn'} \in \{0, 1\} \qquad \forall i \in \mathcal{I}, \forall n \in \mathcal{N}, \forall n' \in \mathcal{N} \tag{3.1.20}$$

3.2 M&G 2003

This model, described by Maravelias and Grossmann (2003) is a Global Event-Based Model using State Task Network process representation, which accounts for variable batch sizes and processing times. It uses a continuous time representation that is common for all units. The assignment constraints are binary variables defined for tasks instead of units with different binary variables to denote whether a task starts, finishes or continues over event points. The start time of tasks are eliminated so time matching constraints are used for the finish times of tasks only. It also accounts for various storage policies.

$$\min\ z \tag{3.2.1}$$

$$\text{s.t. } z \geq \sum_{s\in\mathcal{S}} P_s(S_{s0} - S_{sN}) \tag{3.2.2}$$

$$z \geq (T_N - T_1) \tag{3.2.3}$$

$$T_1 = 0 \tag{3.2.4}$$

$$T_N = H \tag{3.2.5}$$

$$T_{n+1} \geq T_n \qquad \forall n \tag{3.2.6}$$

$$\sum_{i\in\mathcal{I}_j} Ws_{\text{in}} \leq 1 \qquad \forall j, \forall n \tag{3.2.7}$$

$$\sum_{i\in\mathcal{I}_j} Wf_{\text{in}} \leq 1 \qquad \forall j, \forall n \tag{3.2.8}$$

$$\sum_{n} Ws_{\text{in}} = \sum_{n} Wf_{\text{in}} \qquad \forall i \tag{3.2.9}$$

$$S_{sn} = S_{sn-1} + \sum_{i\in O(s)} B^{O}_{isn} - \sum_{i\in I(s)} B^{I}_{isn} \qquad \forall s, \forall n > 1 \tag{3.2.10}$$

$$S_{sn} \leq S^{\max}_{s} \qquad \forall s, \forall n \tag{3.2.11}$$

$$U_{un} = U_{un-1} - \sum_{i} U^{O}_{iun-1} + \sum_{i} U^{I}_{iun} \qquad \forall u, \forall n \tag{3.2.12}$$

$$U_{un} \leq U^{\max}_{u} \qquad \forall u, \forall n \tag{3.2.13}$$

$$\sum_{i\in\mathcal{I}_j} \sum_{n'\leq n} (Ws_{in'} - Wf_{in'}) \leq 1 \qquad \forall j, \forall n \tag{3.2.14}$$

$$Wf_{i0} = 0 \qquad \forall i \tag{3.2.15}$$

$$Ws_{i0} = 0 \qquad \forall i, n = |N| \tag{3.2.16}$$

$$D_{in} = \alpha_i Ws_{in} + \beta Bs_{in} \qquad \forall i, \forall n \tag{3.2.17}$$

$$Tf_{in} \leq Ts_{in} + D_{in} + H(1 - Ws_{in}) \qquad \forall i, \forall n \tag{3.2.18}$$

$$Tf_{in} \geq Ts_{in} + D_{in} + H(1 - Ws_{in}) \qquad \forall i, \forall n \tag{3.2.19}$$

$$Tf_{in} - Tf_{in-1} \leq HWs_{in} \qquad \forall i, \forall n \tag{3.2.20}$$

$$Tf_{in} - Tf_{in-1} \geq D_{in} \qquad \forall i, \forall n \tag{3.2.21}$$

$$Ts_{in} = T_n \qquad \forall i, \forall n \tag{3.2.22}$$

$$Tf_{in} \leq T_n + H(1 - Wf_{in}) \qquad \forall i, \forall n \tag{3.2.23}$$

$$Tf_{in} \geq T_n - H(1 - Wf_{in}) \qquad \forall i \in I^{\mathrm{zw}}, \forall n \tag{3.2.24}$$

$$B_i^{min} Ws_{in} \leq Bs_{in} \leq B_i^{max} Ws_{in} \qquad \forall i, \forall n \tag{3.2.25}$$

$$B_i^{min} Wf_{in} \leq Bf_{in} \leq B_i^{max} Wf_{in} \qquad \forall i, \forall n \tag{3.2.26}$$

$$\begin{aligned} &B_i^{min}(\sum_{n'<n} Ws_{in'} - \sum_{n' \leq n} Wf_{in'}) \\ &\qquad \leq Bp_{in} \leq B_i^{max}(\sum_{n'<n} Ws_{in'} - \sum_{n' \leq n} Wf_{in'}) \end{aligned} \qquad \forall i, \forall n \tag{3.2.27}$$

$$Bs_{in-1} + Bp_{in-1} = Bp_{in} + Bf_{in} \qquad \forall i, \forall n \tag{3.2.28}$$

$$B_{isn}^I = \rho_{is} Bs_{in} \qquad \forall i \in \mathcal{I}_s^c, \forall n, \forall s \tag{3.2.29}$$

$$B_{isn}^I \leq B_i^{max} \rho_{is} Ws_{in} \qquad \forall i \in \mathcal{I}_s^c, \forall n, \forall s \tag{3.2.30}$$

$$B_{isn}^O = \rho_{is} Bf_{in} \qquad \forall i \in \mathcal{I}_s^p, \forall n, \forall s \tag{3.2.31}$$

$$B_{isn}^O \leq B_i^{max} \rho_{is} Wf_{in} \qquad \forall i \in \mathcal{I}_s^p, \forall n, \forall s \tag{3.2.32}$$

$$U_{iun}^I = \gamma_{iu} Ws_{in} + \delta_{iu} Bs_{in} \qquad \forall i, \forall u, \forall n \tag{3.2.33}$$

$$U_{iun}^O = \gamma_{iu} Wf_{in} + \delta_{iu} Bf_{in} \qquad \forall i, \forall u, \forall n \tag{3.2.34}$$

3.3 GHM 2009

This model, proposed by Giménez et al. (2009) is a based on a continuous-time representation that does not require tasks to start or end exactly at a time point, thus reducing the number of time points needed to represent a solution. Processing units are modeled as being as being in different activity states to allow storage of input/output materials. Time variables for "idle" and "storage" periods of a unit are introduced to enable the matching between tasks and time points without big-M constraints. Material transfer variables explicitly account for unit connectivity. Inventory variables for storage in processing units are incorporated to model non-simultaneous and partial material transfers.

$$\min\ z \tag{3.3.1}$$

$$\text{s.t. } z \geq \sum_{s \in \mathcal{S}} P_s (S_{s0} - S_{sN}) \tag{3.3.2}$$

$$z \geq (T_N - T_1) \tag{3.3.3}$$

$$E_{jn} + W_{jn} + S^I_{jn} + S^O_{jn} = 1 \qquad \forall j, n < N \tag{3.3.4}$$

$$E_{jn} = Z_{jn} + \sum_{i \in \mathcal{I}_j} X_{ijn} \qquad \forall j, n < 1 \tag{3.3.5}$$

$$Z_{jn} = Z_{j(n-1)} + \sum_{i \in \mathcal{I}_j} X_{ij(n-1)} - \sum_{i \in \mathcal{I}_j} Y_{ijn} \qquad \forall j, n > 1 \tag{3.3.6}$$

$$\bar{T}^{\mathrm{LB}}_{jn} \leq H \sum_{i \in (\mathcal{I}_j \setminus \mathcal{I}^{\mathrm{CZW}})} X_{ijn} \qquad \forall j, n < N \tag{3.3.7}$$

$$\bar{T}^{\mathrm{EE}}_{jn} \leq H \sum_{i \in (\mathcal{I}_j \setminus \mathcal{I}^{\mathrm{PZW}})} Y_{ijn} \qquad \forall j, n > 1 \tag{3.3.8}$$

$$\bar{T}^S_{jn} \leq H(S^I_{jn} + S^O_{jn}) \qquad \forall j, n < N \tag{3.3.9}$$

$$T^W_{jn} \leq H(W_{jn}) \qquad \forall j, n < N \tag{3.3.10}$$

$$\begin{aligned} T_{n+1} - T_n - H(1 - S^I_{jn} - S^O_{jn} - W_{jn}) \leq \\ \bar{T}^S_{jn} + \bar{T}^W_{jn} \leq T_{n+1} - T_n \end{aligned} \qquad \forall j, n < N \tag{3.3.11}$$

$$\begin{aligned} T_n \geq & \sum_{1 < n' \leq n} \sum_{i \in \mathcal{I}_j} (\alpha_{ij} Y_{ijn'} + \beta_{ij} B^E_{ijn'}) \\ & + \sum_{1 < n' \leq n} \bar{T}^{\mathrm{EE}}_{jn'} + \sum_{n' < n} (\bar{T}^{\mathrm{LB}}_{jn'} + \bar{T}^{\mathrm{S}}_{jn'} + \bar{T}^{\mathrm{W}}_{jn'}) \end{aligned} \qquad \forall j, n > 1 \tag{3.3.12}$$

$$\begin{aligned} H - T_n \geq & \sum_{1 < n' \leq n} \sum_{i \in \mathcal{I}_j} (\alpha_{ij} X_{ijn'} + \beta_{ij} B^S_{ijn'}) \\ & + \sum_{n' \geq n} \bar{T}^{\mathrm{EE}}_{jn'} + \sum_{n \leq n' < N} (\bar{T}^{\mathrm{LB}}_{jn'} + \bar{T}^{\mathrm{S}}_{jn'} + \bar{T}^{\mathrm{W}}_{jn'}) \end{aligned} \qquad \forall j, n > 1 \tag{3.3.13}$$

$$\begin{aligned}\sum_{n>1}\bar{T}^{\mathrm{EE}}_{jn} + \sum_{n<N}(\bar{T}^{\mathrm{LB}}_{jn} + \bar{T}^{\mathrm{S}}_{jn} + \bar{T}^{\mathrm{W}}_{jn}) \\ + \sum_{n>1}\sum_{i\in\mathcal{I}_j}(\alpha_{ij}Y_{ijn} + \beta_{ij}B^{\mathrm{E}}_{ijn}) = H\end{aligned} \qquad \forall j \tag{3.3.14}$$

$$\beta^{\mathrm{MIN}}_{ij}Y_{ijn} \leq B^{E}_{ijn} \leq \beta^{\mathrm{MAX}}_{ij}Y_{ijn} \qquad \forall i, j\in\mathcal{J}_i, n>1 \tag{3.3.15}$$

$$B^{\mathrm{S}}_{ijn} \leq B^{\mathrm{MAX}}_{ij}X_{ijn} \qquad \forall i, j\in\mathcal{J}_i, n<N \tag{3.3.16}$$

$$\sum_{i\in\mathcal{I}_j}B^{\mathrm{P}}_{ijn} \leq \max_{i\in\mathcal{I}_j}\{B^{\mathrm{MAX}}_{ij}\}Z_{jn} \qquad \forall j, n \tag{3.3.17}$$

$$B^{\mathrm{S}}_{ijn} + B^{\mathrm{P}}_{ijn} = B^{\mathrm{P}}_{ij(n+1)}B^{\mathrm{E}}_{ij(n+1)} \qquad \forall i, j\in\mathcal{J}_i, n<N \tag{3.3.18}$$

$$\begin{aligned}I^{S}_{mkn} = I^{S}_{mk(n-1)} - \sum_{j\in\mathcal{J}_k}F^{\mathrm{VU}}_{mkjn} - \sum_{k'\in\mathcal{K}_k}F^{\mathrm{VV}}_{mkk'n} \\ + \sum_{j\in\mathcal{J}_k}F^{\mathrm{UV}}_{mjkn} + \sum_{k'\in\mathcal{K}_k}F^{\mathrm{VV}}_{mk'kn}\end{aligned} \qquad \forall m\notin(\mathcal{M}^{\mathrm{NIS}}\cup\mathcal{M}^{\mathrm{ZW}}), k\in\mathcal{K}_m, n \tag{3.3.19}$$

$$I^{\mathrm{S}}_{mkn} \leq S^{\mathrm{MAX}}_{mk} \qquad \forall m, k\in\mathcal{K}^{\mathrm{D}}, n>1 \tag{3.3.20}$$

$$\sum_{m\in\mathcal{M}_k}S^{S}_{mkn} \leq 1 \qquad \forall k\in\mathcal{K}^{S}, n>1 \tag{3.3.21}$$

$$\begin{aligned}I^{\mathrm{I}}_{mjn} =& I^{\mathrm{I}}_{mj(n-1)} + \sum_{k\in(\mathcal{K}_j\cap\mathcal{K}_m)}F^{\mathrm{VU}}_{mkjn} \\ &+ \sum_{j'\in\mathcal{J}_j}F^{\mathrm{UU}}_{mj'jn} + \sum_{i\in(\mathcal{I}_j\cap\mathcal{I}^{C}_m)}\sum\rho_{im}B^{\mathrm{S}}_{ijn}\end{aligned} \qquad \forall m, j, n \tag{3.3.22}$$

$$\sum_{m\in\mathcal{M}}I^{\mathrm{I}}_{mjn} \leq \max_{i}\{B^{\mathrm{MAX}}_{i,j}S^{\mathrm{I}}_{jn}\} \qquad \forall j, n<N \tag{3.3.23}$$

$$\begin{aligned}I^{\mathrm{O}}_{mjn} = I^{\mathrm{O}}_{mj(n-1)} + \sum_{i\in(\mathcal{I}_j\cap\mathcal{I}^{p}_m)}\rho_{im}B^{\mathrm{E}}_{ijn} \\ - \sum_{k\in(\mathcal{K}_j\cap\mathcal{K}_m)}F^{UV}_{mjkn} - \sum_{j'\in\mathcal{J}_j}F^{UU}_{mjj'n}\end{aligned} \qquad \forall m, j, n>1 \tag{3.3.24}$$

$$\sum_{m\ in\mathcal{M}}I^{\mathrm{O}}_{mjn} \leq \max_{i}\{B^{\mathrm{MAX}}_{ij}\}S^{\mathrm{O}}_{jn} \qquad \forall j, n<N \tag{3.3.25}$$

$$\begin{aligned}Q_{rn} = Q_{r(n-1)} + \\ \sum_{i\in\mathcal{I}_r}\sum_{j\in\mathcal{J}_i}[\gamma_{ijr}(X_{ijn} - Y_{ijn}) + \delta_{ijr}(B^{S}_{ijn} - B^{E}_{ijn})]\end{aligned} \qquad \forall r, n \tag{3.3.26}$$

$$Q_{rn} \leq \rho^{\mathrm{MAX}}_{r} \qquad \forall r, n \tag{3.3.27}$$

$$\sum_{j \in \mathcal{K}_m} I^{\mathrm{S}}_{mkn} \geq d_m \qquad \forall n \in \mathcal{M}^S, n = N \tag{3.3.28}$$

3.4 I&F 1998

This model (Ierapetritou and Floudas, 1998) was the first introduction of the concept of event points which correspond to a sequence of time instances located along the time axis of each unit. The location of event points is different for each unit, allowing different tasks to start at different times in each unit for the same event point. The timings of tasks are accounted through special sequencing constraints involving big-M constraints. The resulting model requires less event points compared to corresponding global or slot-based methods.

$$\min \; z \tag{3.4.1}$$

$$\text{s.t. } z \geq \sum_{s \in \mathcal{S}} P_s (S_{s0} - S_{sN} - \sum_{i \in \mathcal{I}^p_s} \sum_{j \in \mathcal{J}_i} B_{iN}) \tag{3.4.2}$$

$$z \geq T_{ijn} + \alpha_i W_{iN} + \beta_i B_{iN} - T^s_{ij0} \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_i \tag{3.4.3}$$

$$\sum_{i \in \mathcal{I}_j} W_{in} = G_{jn} \qquad \forall j \in \mathcal{J}, \forall n \in \mathcal{N} \tag{3.4.4}$$

$$B^{\min}_i \leq B_{in} \leq B^{\max}_i W_{in} \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_i, \forall n \in \mathcal{N} \tag{3.4.5}$$

$$S_{sn} \leq S^{\max}_s \qquad \forall s \in \mathcal{S}, \forall n \in \mathcal{N} \tag{3.4.6}$$

$$S_{sn} = S_{s(n-1)} + \sum_{i \in \mathcal{I}^p_s} \rho_{is} \sum_{j \in \mathcal{J}_i} B_{i(n-1)} - \sum_{i \in \mathcal{I}^c_s} \rho_{is} \sum_{j \in \mathcal{J}_i} B_{in} \qquad \forall s \in \mathcal{S}, \forall n \in \mathcal{N} \tag{3.4.7}$$

$$S_{sN} \geq D_s \qquad \forall s \in \mathcal{S} \tag{3.4.8}$$

$$T^f_{ijn} = T^s_{ijn} + \alpha_i W_{in} + \beta_i B_{in} \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_i, \forall n \in \mathcal{N} \tag{3.4.9}$$

$$T^s_{ij(n+1)} \geq T^f_{ijn} - H(2 - W_{in} - G_{jn}) \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_i, \forall n \in \mathcal{N}, n \neq N \tag{3.4.10}$$

$$T^s_{ij(n+1)} \geq T^s_{ijn} \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_i, \forall n \in \mathcal{N}, n \neq N \tag{3.4.11}$$

$$T^f_{ij(n+1)} \geq T^f_{ijn} \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_i, \forall n \in \mathcal{N}, n \neq N \tag{3.4.12}$$

$$T^s_{ij(n+1)} \geq T^f_{i'jn} - H(2 - W_{i'n} - G_{jn}) \qquad \forall j \in \mathcal{J}, \forall i \in \mathcal{I}_j, \forall i' \in \mathcal{I}_j, i \neq i', \forall n \in \mathcal{N}, n \neq N \tag{3.4.13}$$

$$T^s_{ij(n+1)} \geq \sum_{n' \in \mathcal{N}, n' \leq \mathcal{N}} \sum_{i' \in \mathcal{I}_j} (T^f_{i'jn'} - T^s_{i'jn'}) \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_i, \forall n \in \mathcal{N}, n \neq \mathcal{N} \tag{3.4.14}$$

$$T^f_{ijn} \leq H \qquad i \in \mathcal{I}, \forall j \in \mathcal{J}_i, n \in \mathcal{N} \tag{3.4.15}$$

$$T^s_{ijn} \leq H \qquad i \in \mathcal{I}, \forall j \in \mathcal{J}_i, n \in \mathcal{N} \tag{3.4.16}$$

3.5 S&K 2005

This model (Sundaramoorthy and Karimi, 2005) considers a synchronous slot-based time representation where the time horizon is divided into multiple time slots of varying lengths. It uses generalized recipe diagrams for process representation, wherein a storage task is used to model the mixing and splitting of the same material streams. Tasks are allowed to continue processing over multiple time slots. Tasks are allowed to finish before the end of the time slot, making the model inherently similar to the global event-based models, except for the differences in accounting the various balances. No resources other than materials and equipment are considered, hence it cannot support instances with utilities.

$$\min \; z \tag{3.5.1}$$

$$\text{s.t. } z \geq \sum_{s \in \mathcal{S}} P_s(S_{s0} - S_{sN}) \tag{3.5.2}$$

$$z \geq \sum_{n \in \mathcal{N}} SL_n \tag{3.5.3}$$

$$\sum_{n=1}^{N} SL_n \leq H \tag{3.5.4}$$

$$G_{jn} = \sum_{i\in\mathcal{I}_j} Ws_{in} \qquad 0 \leq n < N \tag{3.5.5}$$

$$B_i^{min} Ws_{in} \leq Bp_{in} \leq B_i^{max} Ws_{in} \qquad i > 0 \tag{3.5.6}$$

$$Wp_{in} = Wp_{i(n-1)} + Ws_{i(n-1)} - Wf_{in} \qquad 0 < n < N \tag{3.5.7}$$

$$G_{jn} = \sum_{i\in\mathcal{I}_j} Wf_{in} \qquad 0 < n < N \tag{3.5.8}$$

$$\sum_{i\in\mathcal{I}_j} Wp_{in} + G_{jn} = 1 \qquad 0 < n < N \tag{3.5.9}$$

$$Wp_{in} + Ws_{in} \leq 1 \qquad 0 < n < N \tag{3.5.10}$$

$$Wp_{in} + Wf_{in} \leq 1 \qquad 0 < n < N \tag{3.5.11}$$

$$t_{j(n+1)} \geq t_{jn} + \sum_{i\ in\mathcal{I}_j} (\alpha_i Ws_{in} + \beta_i Bp_{in}) - SL_{n+1} \qquad n < N \tag{3.5.12}$$

$$Bs_{in} = Bs_{i(n-1)} + Bp_{i(n-1)} - Bf_{in} \qquad i > 0, n > 0 \tag{3.5.13}$$

$$B_i^{min} Wp_{in} \leq Bs_{in} \leq B_i^{max} Wp_{in} \qquad i > 0, 0 < n < N \tag{3.5.14}$$

$$B_i^{min} Wf_{in} \leq Bf_{in} \leq B_i^{max} Wf_{in} \qquad i > 0, 0 < n < N \tag{3.5.15}$$

$$t_{jn} \leq \sum_{i\ in\mathcal{I}_j} (\alpha_i Wp_{in} + \beta_i Bs_{in}) \qquad 0 < n < N \tag{3.5.16}$$

$$S_{sn} = S_{s(n-1)} + \sum_{\substack{i\in\mathcal{I}_s^p\\ i\neq 0}} \sum_{j\in\mathcal{J}_i} \rho_{is} Bf_{in} - \sum_{\substack{i\in\mathcal{I}_s^c\\ i\neq 0}} \sum_{j\in\mathcal{J}_i} \rho_{is} Bp_{in} \qquad \forall s \in \mathcal{S}, \forall n \tag{3.5.17}$$

$$SL_n \leq \max_j \left[\max_{i\in\mathcal{I}_j} (\alpha_i + \beta_i B_i^{max}) \right] \qquad \forall n \tag{3.5.18}$$

$$t_{jn} \leq \max_{j\in\mathcal{J}_i} (\alpha_i + \beta_i B_i^{max}) \qquad \forall j \in \mathcal{J}, \forall n \tag{3.5.19}$$

$$S_{sn} \leq S_s^{max} \qquad \forall s \in \mathcal{S}, \forall n \tag{3.5.20}$$

$$Bp_{in}, Bs_{in}, Bf_{in} \leq B_i^{max} \qquad \forall i \in \mathcal{I}, \forall n \tag{3.5.21}$$

4 | GUI Design

This chapter describes the basic structure of the instance builder web page and the various assertions and conditions required for a well defined scheduling instance.

4.1 Instance structure

To facilitate compatibility with backend C++ code, the JavaScript instance object is converted to a JSON string. JSON (JavaScript Object Notation) is a text format that is completely language independent but uses conventions that are familiar to many different programming languages, including C, C++, Java, JavaScript, Python and others. These properties make JSON an ideal data-interchange format.

JSON is built on two universal data structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

Virtually all programming languages support these structures in one form or other. In JSON, they take on these forms:

An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

4.2 Instance object

The structure of the Instance JSON object is as shown in Figure 4.1. The empty arrays `[]` for `"Units"`, `"States"`, `"Tasks"`, `"Utilities"` and `"Orders"` in the structure contain the JSON objects as specific to the instance. `"Name"` is a string denoting the given name for the instance. `"Horizon"` is an integer or float. `"isCompleteInstance"` is a boolean `true` or `false` value denoting if an instance is completely specified.

```
{
    "Name": "Scheduling_Instance",
    "Horizon": 8,
    "Units": [],
    "States": [],
    "Orders": [],
    "Utilities": [],
    "Tasks": [],
    "isCompleteInstance": false
}
```

Fig. 4.1: Instance JSON

```
{
    "Name": "Unit1",
    "MaximumCapacity": 100
}
```

Fig. 4.2: Unit JSON object

```
{
    "StateName": "State1",
    "StateInitialLevel":
       100,
    "StateMaxLevel": 100,
    "IsZeroWait": true,
    "IsUIS": false,
    "Price": 10
}
```

Fig. 4.3: State JSON object

```
{
    "Name": "Utility1",
    "MaximumAvailability":
       250
}
```

Fig. 4.4: Utility JSON object

```
{
    "StateName": "Product1",
    "Amount": 100
}
```

Fig. 4.5: Order (demand) JSON object

Figures 4.2 - 4.6 show the structures of the unit, state, utility, order and task JSON objects respectively.

An instance is deemed ‘complete’ if:

1. At least one valid unit

    - Positive maximum capacity

2. At least two states

    - Non-negative maximum capacity

  - Initial level is less than or equal to maximum capacity
  - At least one state with positive initial level

3. At least one valid task
   - At least one compatible unit with non-zero processing time (at least one of α or β is non-zero)
   - At least one consumed state
   - At least one produced state

4. Positive horizon and at least one state to be sold with positive price OR At least one state in demand with positive demand amount

```
{
    "TaskName": "Task1",
    "CompatibleUnits": [
        {
        "UnitName": "Unit1",
        "alpha": 5,
        "beta": 1
        }
    ],
    "ConsumedStates": [
        {
        "ConStateName": "State1",
        "consRatio": 1
        }
    ],
    "ProducedStates": [
        {
        "ProdStateName": "State2",
        "prodRatio": 1
        }
    ],
    "ConsumedUtilities": [
        {
        "ConsUtilName": "Utility1",
        "CompUnit": "Unit1",
        "gamma": 2,
        "delta": 0.2
        }
    ]
}
```

Fig. 4.6: Task JSON object

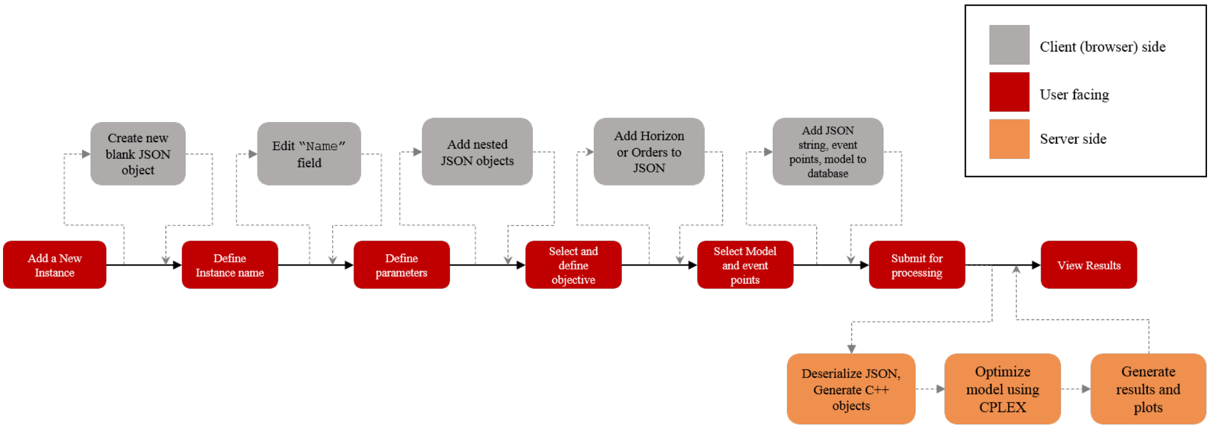

Client (browser) side
User facing
Server side
Create new blank JSON object
Edit "Name" field
Add nested JSON objects
Add Horizon or Orders to JSON
Add JSON string, event points, model to database
Add a New Instance
Define Instance name
Define parameters
Select and define objective
Select Model and event points
Submit for processing
View Results
Deserialize JSON, Generate C++ objects
Optimize model using CPLEX
Generate results and plots

Fig. 4.7: Overall process flow

4.3 Overall scheduling process flow

Figure 4.7 shows the underlying process behind creating and submitting a scheduling instance. At each stage of the instance building process, the JSON object is converted to a string by means of the `JSON.stringify()` method in JavaScript and saved in a backend mySQL database. A more detailed process flow for parameter and objective input is shown in Fig. 4.8. Once the the instance is complete and amenable to optimization, in the deterministic version, the user has the option of selecting the number of event points and scheduling model to be used. In the uncertain version, the user has additional choices to select uncertain parameters and adjustable variables.

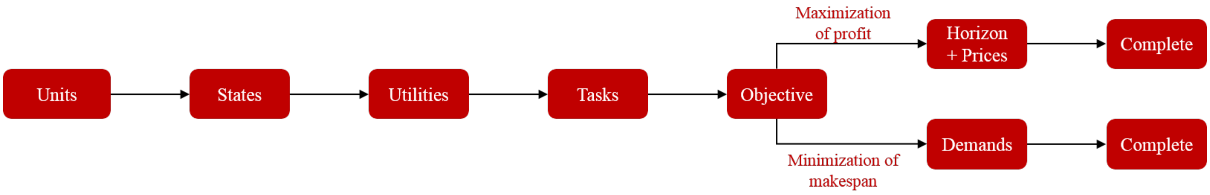


Fig. 4.8: Parameter and objective input

On submitting, a C++ JSON deserializer converts the JSON string to C++ objects. These are then used to build a mixed-integer model in CPLEX according to the model specified by the user.

On successful completion, the following statistics are displayed on the results page:

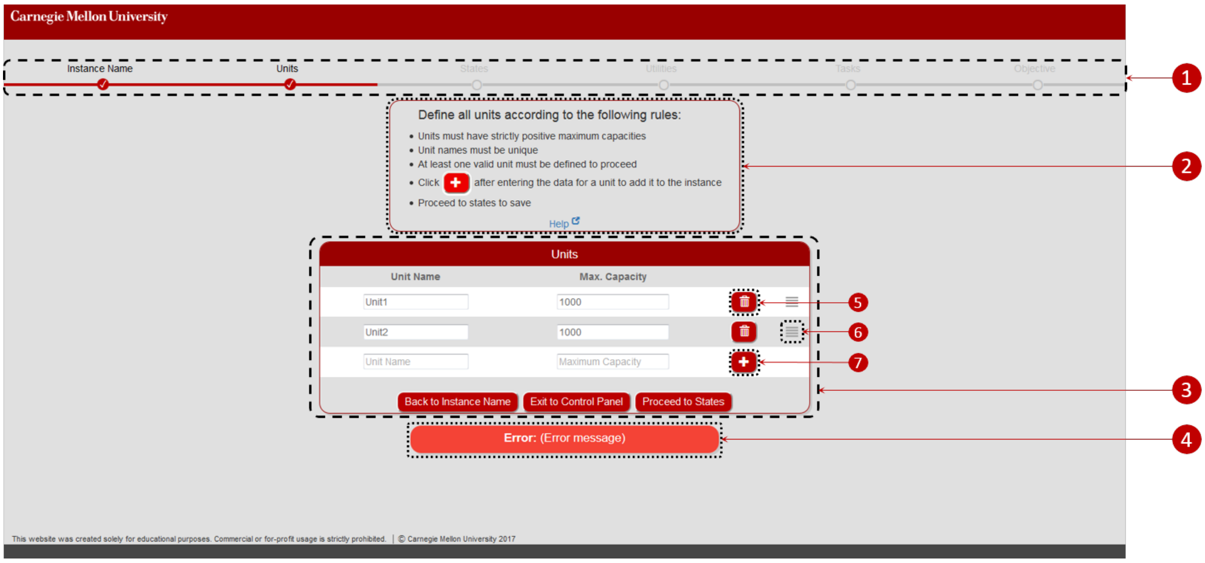
- Run time
- Forumlation

- Number of event points
- Objective type
- Solver status
- Objective value
- Number of constraints
- Number of binary variables
- Number of continuous variables
- Number of continuous variables
- Nodes
- Root node relaxation
- Relative gap (%)

If the user has elected to incorporate uncertainty into the instance, the following additional information is displayed:

- Uncertain Parameter(s)
- Level of uncertainty
- Adjustable Variable(s)

Additionally, a Gantt chart for the optimal schedule and material inventory charts are displayed. If utilities are specified by the user, then a utility consumption plot is also displayed.



Define all units according to the following rules:

- Units must have strictly positive maximum capacities
- Unit names must be unique
- At least one valid unit must be defined to proceed
- Click ✚ after entering the data for a unit to add it to the instance
- Proceed to states to save

Units

Unit Name	Max. Capacity
Unit1	1000
Unit2	1000
Unit Name	Maximum Capacity

Error: (Error message)

This website was created solely for educational purposes. Commercial or for-profit usage is strictly prohibited. | © Carnegie Mellon University 2017

Fig. 4.9: Instance builder elements: 1. Progress bar 2. Instructions 3. Input table 4. Error message(s) 5. Delete item 6. Drag and drop to rearrange 7. Add new item

4.4 Instance builder

Figure 4.9 shows the elements in the instance builder GUI. The screenshot in the figure shows the elements in the Units input table. The structure of the States, Utility and Tasks input table follows the same structure. A progress tracker is provided to track the status of the instance being edited.

Error messages are displayed to ensure that the user input is valid and the instance completion criteria listed in section 4.2 are satisfied. Once a particular set of inputs is completed, clicking on the proceed button will display the next input table.

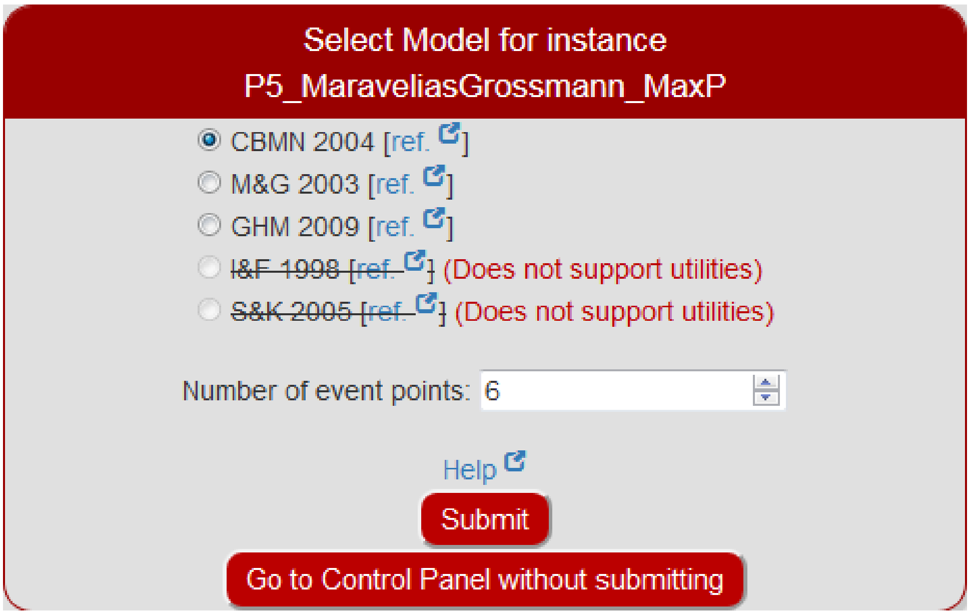
4.5 Submission page

On successful completion, an instance becomes available for submission. A screenshot of the model selection table is shown in Fig. 5.7. The CBMN 2004 model is selected by default for both the deterministic and uncertain versions. The I&F 1998 and S&K 2005 models do not support instances with utilities involved.

4.5.1 Uncertainty frameworks

If the user elects to incorporate uncertainty in the instance, four types of uncertain variables are available:

1. Fixed processing time (α)



Select Model for instance P5_MaraveliasGrossmann_MaxP

● CBMN 2004 [ref.]

○ M&G 2003 [ref.]

○ GHM 2009 [ref.]

○ ~~I&F 1998 [ref.]~~ (Does not support utilities)

○ ~~S&K 2005 [ref.]~~ (Does not support utilities)

Number of event points: 6

Fig. 4.10: Model selection table for an instance involving utilities

2. Fixed and variable processing time ($\alpha + \beta$)

3. Task yield (ρ_p)

4. Fixed processing time and task yield ($\alpha + \rho_p$)

The availability of SRO and ARO frameworks for each of the sets of uncertain variables is shown in Table 4.1. If ARO is selected, the uncertain parameters selected dictate the availability of adjustable variables.

Table 4.1: Uncertainty frameworks and adjustable variables

	Frameworks		Adjustable variables		
Uncertain parameters	SRO	ARO	T	S	B + S
α	✓	✓	✓	✗	✓
$\alpha + \beta$	✓	✓	✓	✗	✗
ρ_p	✗	✓	✗	✓	✗
$\alpha + \rho_p$	✗	✓	✓	✓	✗

5 | Case Study

5.1 Instance description

In this chapter, we present a case study of a well known scheduling instance from Kondili et al. (1993), optimized for the deterministic case using the described online tool. The production of two products 1 and 2 from three feed stocks A, B and C takes place according to the STN representation given in Fig. 5.1. This instance has four intermediate states and five tasks.

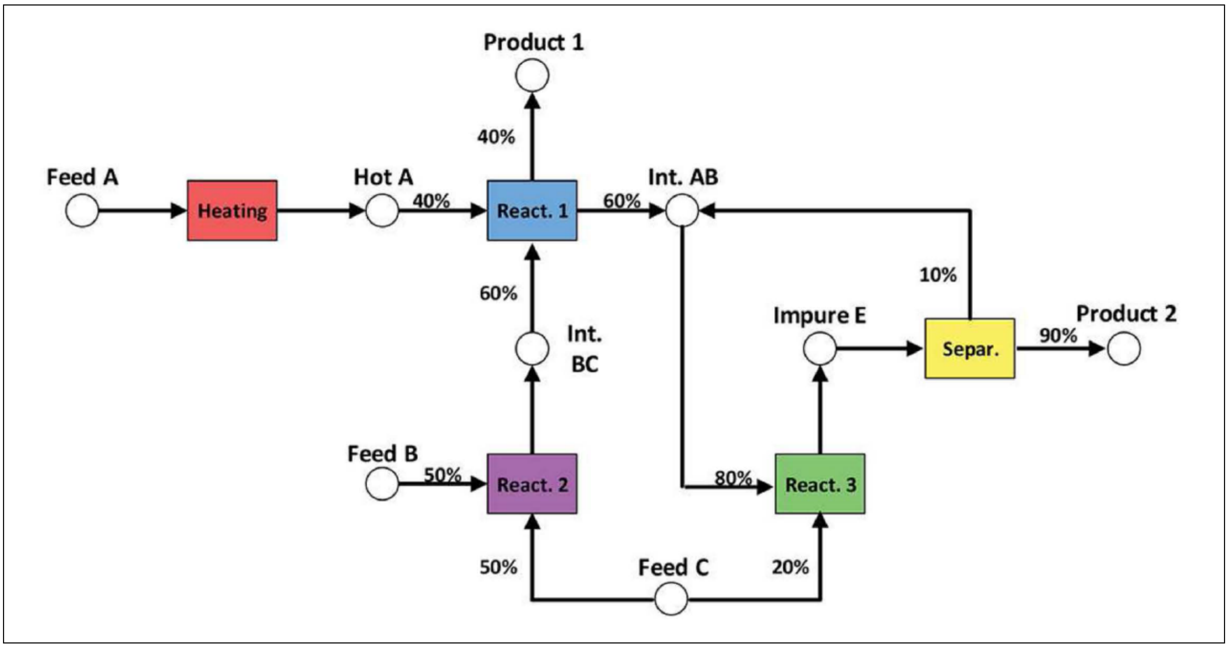


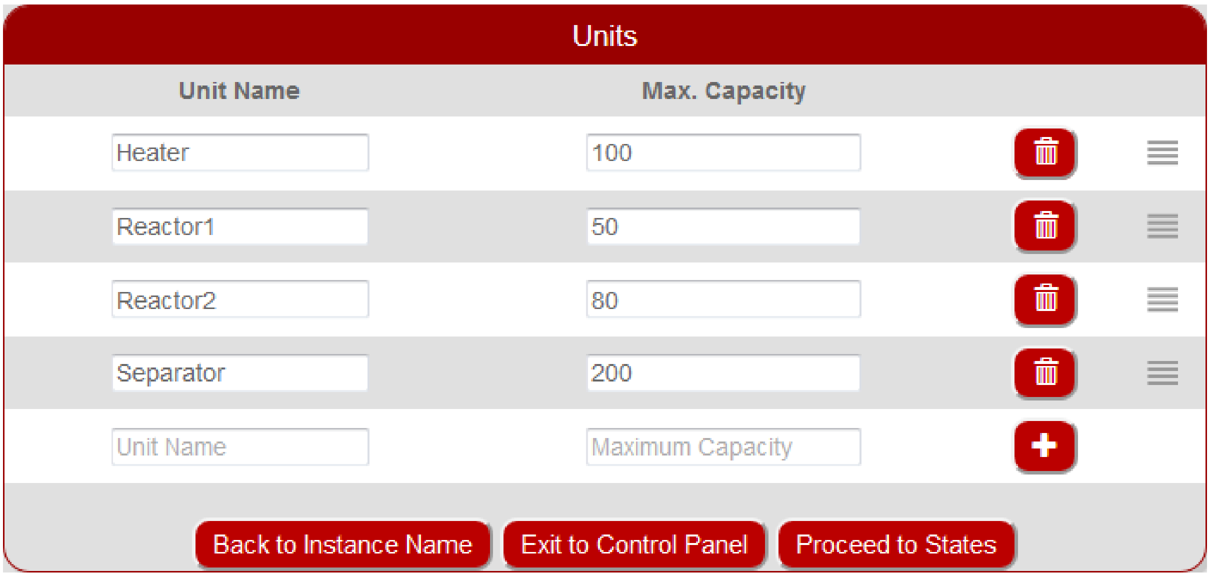
Fig. 5.1: State task network for the example instance

Table 5.1 shows the state maximum capacity and initial load data for the problem. The available unit, task compatibility and processing time data is given in Table 5.2.

5.2 Defining instance parameters

5.2.1 Defining units

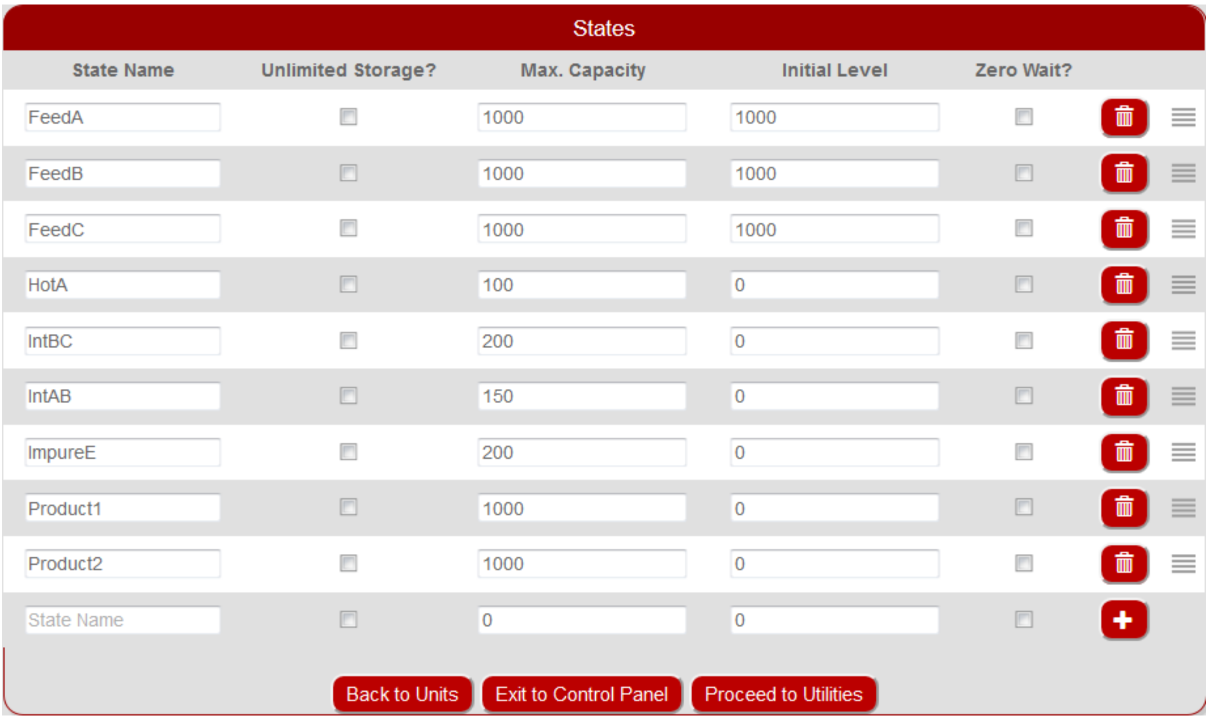
Unit names and maximum capacities from Table 5.2 are input into the units table of the web tool as shown in Fig. 5.2.



Units

Unit Name	Max. Capacity
Heater	100
Reactor1	50
Reactor2	80
Separator	200
Unit Name	Maximum Capacity

Fig. 5.2: Units input into webtool



States

State Name	Unlimited Storage?	Max. Capacity	Initial Level	Zero Wait?
FeedA	□	1000	1000	□
FeedB	□	1000	1000	□
FeedC	□	1000	1000	□
HotA	□	100	0	□
IntBC	□	200	0	□
IntAB	□	150	0	□
ImpureE	□	200	0	□
Product1	□	1000	0	□
Product2	□	1000	0	□
State Name	□	0	0	□

Fig. 5.3: States input into webtool

Table 5.1: Problem data (states)

State	**Capacity**	**Initial load**	**Price (per unit)**
Feed A	1000	1000	-
Feed B	1000	1000	-
Feed C	1000	1000	-
Hot A	100	-	-
Int. BC	200	-	-
Int. AB	150	-	-
Impure E	200	-	-
Product 1	1000	-	10
Product 2	1000	-	10

Table 5.2: Problem data (units & tasks)

Unit	Heater		Reactor 1		Reactor 2		Separator	
Maximum Load	100		50		80		200	
Tasks	α	β	α	β	α	β	α	β
Heating	0.667	0.007						
Reaction 1			1.334	0.027	1.334	0.017		
Reaction 2			1.334	0.027	1.334	0.017		
Reaction 3			0.667	0.013	0.667	0.008		
Separation							1.334	0.007

5.2.2 Defining states

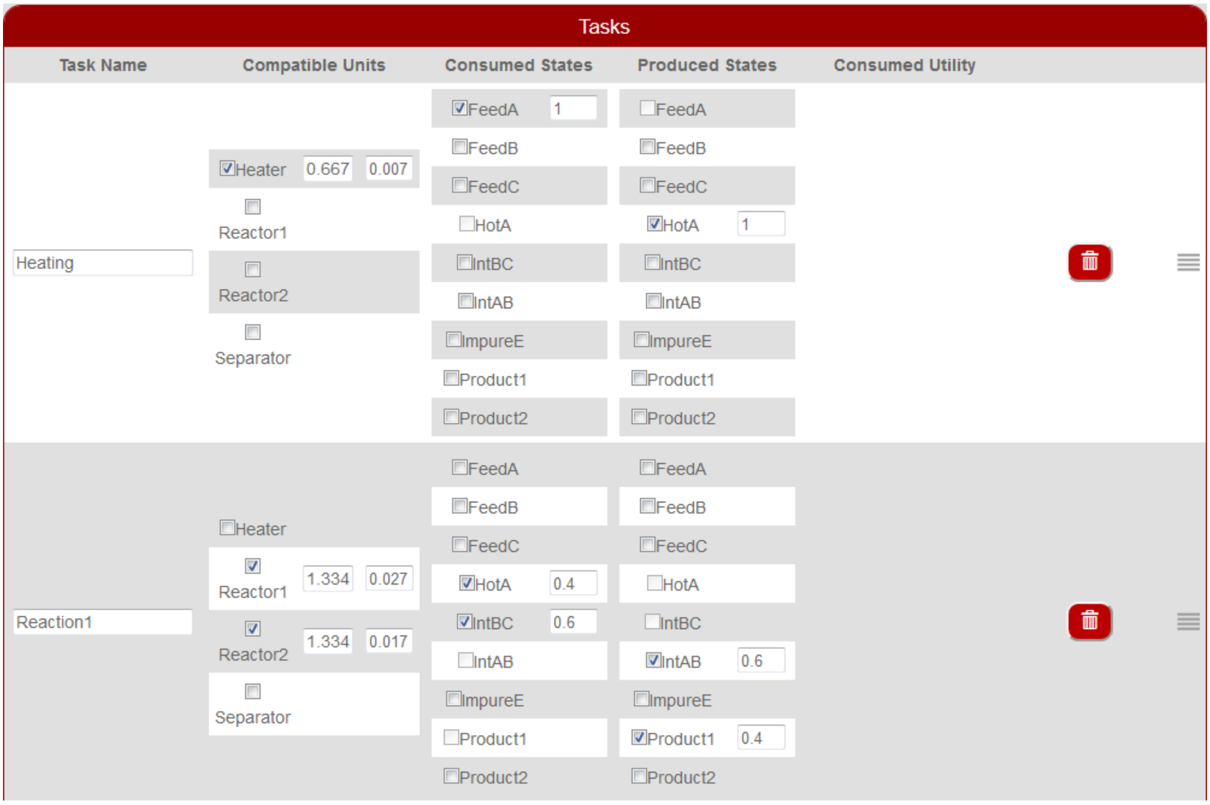
State names, maximum storage capacities and initial levels from Table 5.1 are input into the units table of the web tool as shown in Fig. 5.3.

5.2.3 Defining tasks

This instance does not involve utilities. Hence we can skip utilities input. For each task in Table 5.2, the values of α and β are input after selecting the appropriate compatible unit(s). Fig. 5.4 shows the tasks table after successful input of tasks Heating and Reaction 1.

5.2.4 Objective

For this case, we will consider the case of maximization of profit. After successful completion of the tasks input, the objective selection options are available as shown in Fig. 5.5. After clicking on “Maximize profit”, the horizon and price input screen appears as shown in Fig. 5.6. The instance definition is complete after input of horizon and price



Tasks

Task Name	Compatible Units	Consumed States	Produced States	Consumed Utility
Heating	☑Heater 0.667 0.007 ☐ Reactor1 ☐ Reactor2 ☐ Separator	☑FeedA 1 ☐FeedB ☐FeedC ☐HotA ☐IntBC ☐IntAB ☐ImpureE ☐Product1 ☐Product2	☐FeedA ☐FeedB ☐FeedC ☑HotA 1 ☐IntBC ☐IntAB ☐ImpureE ☐Product1 ☐Product2	
Reaction1	☐Heater ☑ Reactor1 1.334 0.027 ☑ Reactor2 1.334 0.017 ☐ Separator	☐FeedA ☐FeedB ☐FeedC ☑HotA 0.4 ☑IntBC 0.6 ☐IntAB ☐ImpureE ☐Product1 ☐Product2	☐FeedA ☐FeedB ☐FeedC ☐HotA ☐IntBC ☑IntAB 0.6 ☐ImpureE ☑Product1 0.4 ☐Product2	

Fig. 5.4: Tasks input into webtool

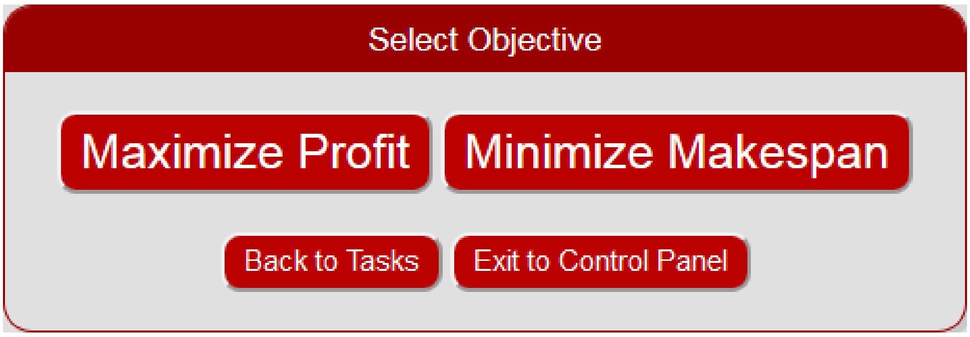


Fig. 5.5: Objective selection

data for Product 1 and Product 2 from Table 5.1. We will consider an 8 hour horizon for this case study.

5.3 Submission

Once the instance is complete, the submit button in the control panel will be activated. Clicking the “Submit” button will display the model selection table as shown in Fig. 5.7. For the deterministic version, the five models described in Chapter 3 are available. The CBMN 2004 model is selected by default.

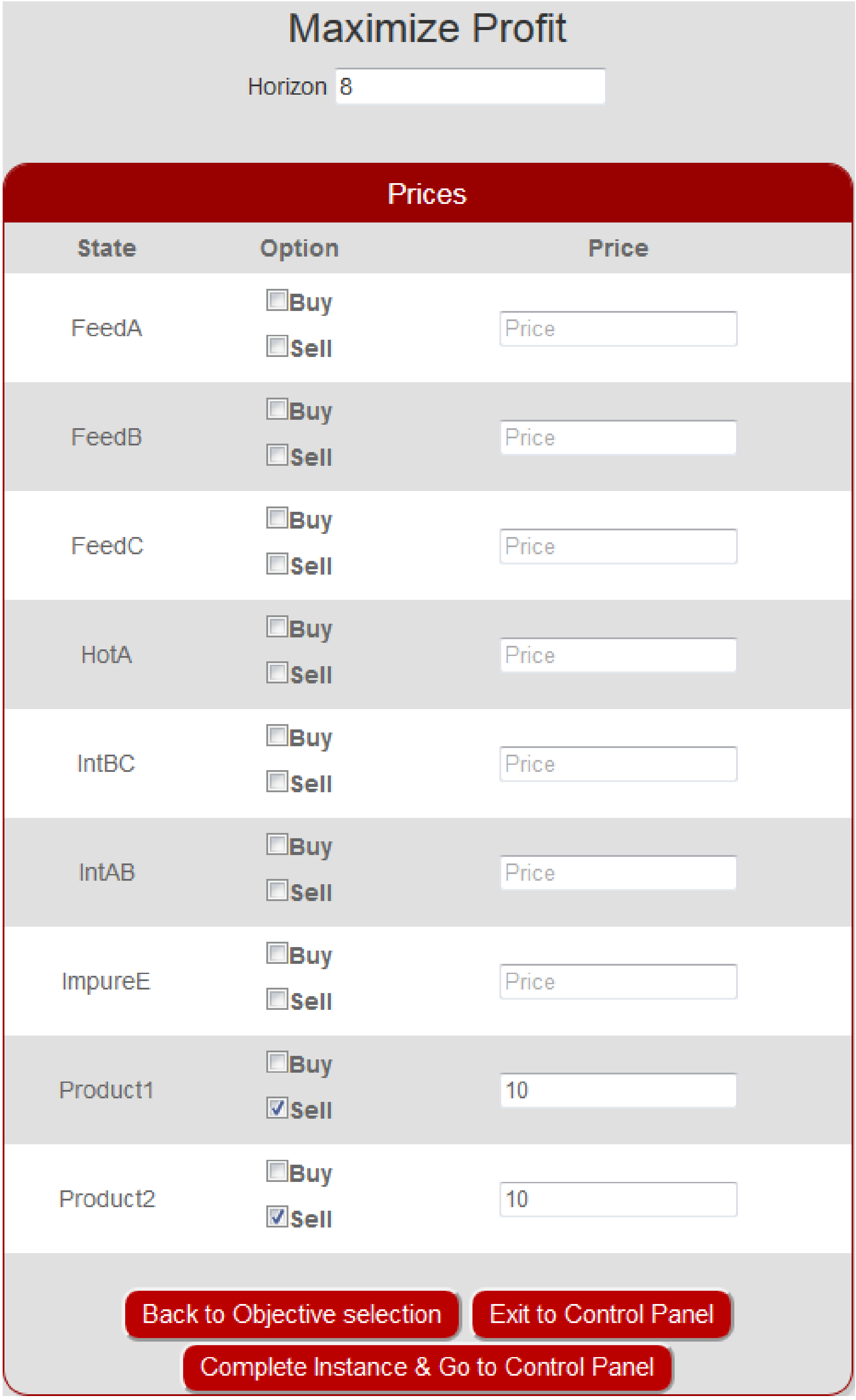
5.3.1 Number of event points

Continuous-time formulations for short-term scheduling are based on a set of time points (unit specific or global), which are non-uniformly distributed along the scheduling horizon. A limitation of these approaches is that the number of time points required to represent the optimal schedule is unknown a priori, so multiple MILP models need to be solved to reach the optimal solution. A suitable strategy is to start by adopting a small number of event points and gradually increase this number and repeatedly solve the model until the objective values does not show an improvement. For this particular instance, 5 event points result in the optimal schedule. Clicking the “Submit” on this page will add the instance to the processing queue.

5.4 Results

Once the processing of the instance has completed, the “Submission Status” field on the control panel for the instance will change to “Results Available”. Clicking on the “Results” button will show model statistics and the optimized Gantt Chart as shown in Fig. 5.8.

A proven optimal solution is found fairly quickly by CPLEX. The objective value reported by the tool is the same as that reported in previous literature for this instance. Inventory level charts for Product 1 and Product 2 show that around 70 units of Product 1 and 80 units of Product 2 will be manufactured.



Maximize Profit

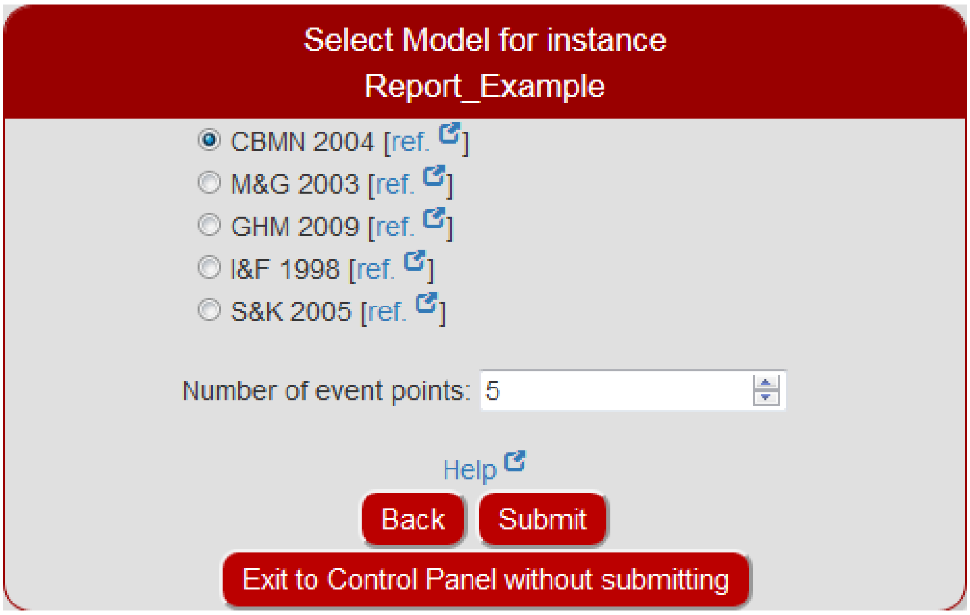
Horizon 8

Prices

State	Option	Price
FeedA	☐Buy ☐Sell	Price
FeedB	☐Buy ☐Sell	Price
FeedC	☐Buy ☐Sell	Price
HotA	☐Buy ☐Sell	Price
IntBC	☐Buy ☐Sell	Price
IntAB	☐Buy ☐Sell	Price
ImpureE	☐Buy ☐Sell	Price
Product1	☐Buy ☑Sell	10
Product2	☐Buy ☑Sell	10

Back to Objective selection
Exit to Control Panel
Complete Instance & Go to Control Panel

Fig. 5.6: Horizon and price input



Select Model for instance Report_Example

● CBMN 2004 [ref.]
○ M&G 2003 [ref.]
○ GHM 2009 [ref.]
○ I&F 1998 [ref.]
○ S&K 2005 [ref.]

Number of event points: 5

Fig. 5.7: Number of event points and model selection

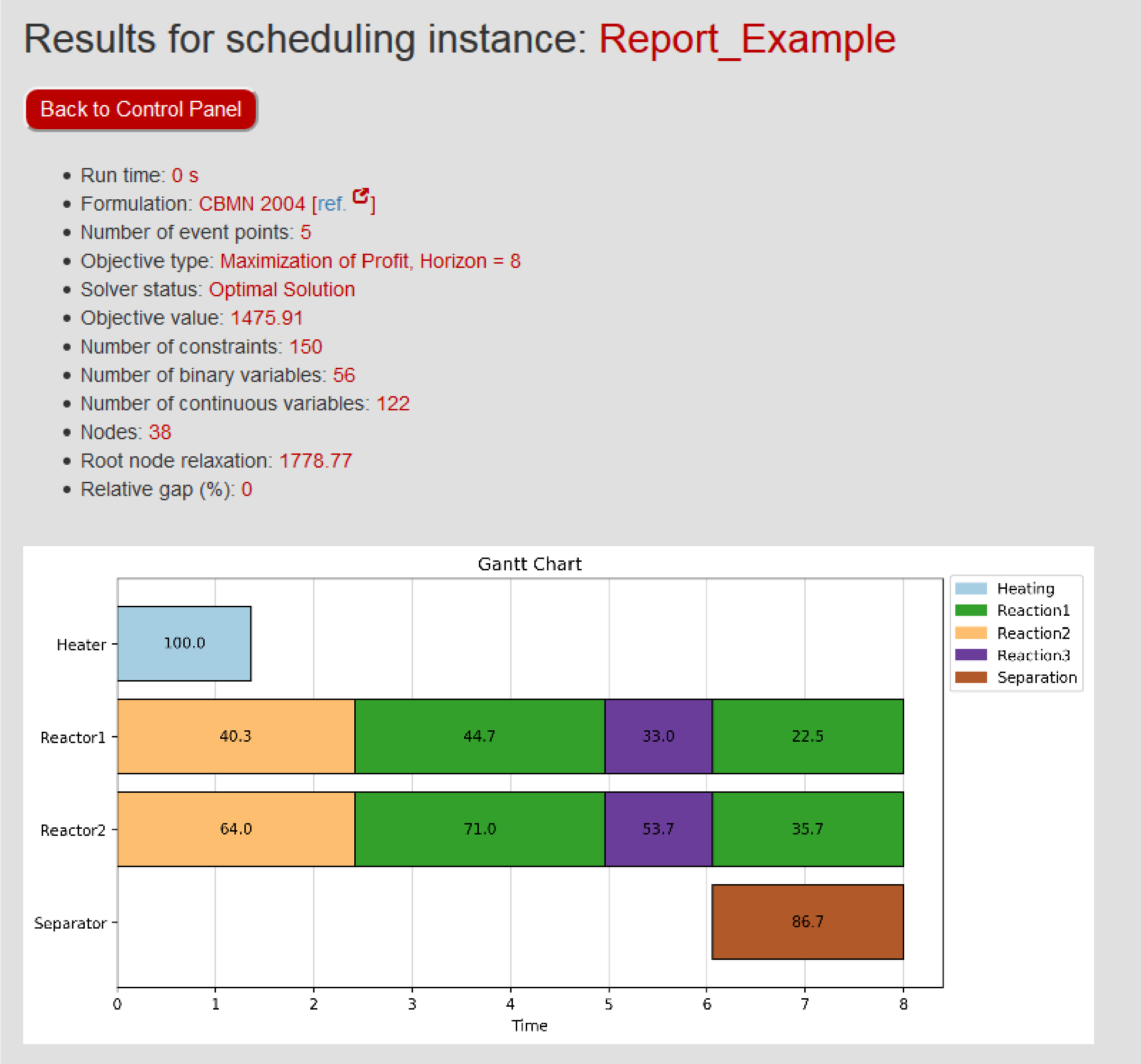
5.5 Handling uncertain parameters with robust optimization

In this section, we will consider uncertainty in fixed processing time (α). The parameters are allowed to vary in a decision-dependent polyhedron. The original α_i values specified by the user are regarded as nominal values α_i^0. The maximum allowable magnitude of each parameter's deviation from its nominal value is a percentage of the latter (e.g. ± 10% of its nominal value).

In addition, we also consider one general affine correlation to postulate an upper limit on the cumulative processing time deviation among all tasks to be performed. The quantities $\xi \in [0,1]$ and $\phi \in [0,1]$ are used to parameterize the uncertainty set. For low, medium and high uncertainty levels, the values of ξ and ϕ are set as follows:

1. **Low:** $\xi = 0.1$, $\phi = 0.0$
2. **Medium:** $\xi = 0.2$, $\phi = 0.5$
3. **High:** $\xi = 0.3$, $\phi = 1.0$

5.5.1 Static robust optimization



Results for scheduling instance: Report_Example

Back to Control Panel

- Run time: 0 s
- Formulation: CBMN 2004 [ref.]
- Number of event points: 5
- Objective type: Maximization of Profit, Horizon = 8
- Solver status: Optimal Solution
- Objective value: 1475.91
- Number of constraints: 150
- Number of binary variables: 56
- Number of continuous variables: 122
- Nodes: 38
- Root node relaxation: 1778.77
- Relative gap (%): 0

Fig. 5.8: Model statistics and Gantt chart

References

Castro, P. M., Barbosa-Pvoa, A. P., Matos, H. A., and Novais, A. Q. Simple Continuous-Time Formulation for Short-Term Scheduling of Batch and Continuous Processes. *Industrial & Engineering Chemistry Research*, 43(1):105–118 (2004).

Giménez, D. M., Henning, G. P., and Maravelias, C. T. A novel network-based continuous-time representation for process scheduling: Part I. Main concepts and mathematical formulation. *Computers & Chemical Engineering*, 33(9):1511 – 1528 (2009).

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Mndez, C., Sand, G., and Wassick, J. Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, 62(Supplement C):161 – 193 (2014).

Ierapetritou, M. G. and Floudas, C. A. Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. *Industrial & Engineering Chemistry Research*, 37(11):4341–4359 (1998).

Janak, S. L., Lin, X., and Floudas, C. A. A new robust optimization approach for scheduling under uncertainty: II. Uncertainty with known probability distribution. *Computers & Chemical Engineering*, 31(3):171 – 195 (2007).

Kondili, E., Pantelides, C., and Sargent, R. A general algorithm for short-term scheduling of batch operations – I. MILP formulation. *Computers & Chemical Engineering*, 17(2):211 – 227 (1993).

Lappas, N. H. and Gounaris, C. E. Multi-stage adjustable robust optimization for process scheduling under uncertainty. *AIChE Journal*, 62(5):1646–1667 (2016).

Li, Z. and Ierapetritou, M. G. Robust Optimization for Process Scheduling Under Uncertainty. *Industrial & Engineering Chemistry Research*, 47(12):4148–4157 (2008).

Lin, X., Janak, S. L., and Floudas, C. A. A new robust optimization approach for scheduling under uncertainty:: I. Bounded uncertainty. *Computers & Chemical Engineering*, 28(6):1069 – 1085 (2004), FOCAPO 2003 Special issue.

Maravelias, C. T. and Grossmann, I. E. New General Continuous-Time StateTask Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants. *Industrial & Engineering Chemistry Research*, 42(13):3056–3074 (2003).

Méndez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I., and Fahl, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, 30(6):913 – 946 (2006).

Sahinidis, N. and Grossmann, I. MINLP model for cyclic multiproduct scheduling on continuous parallel lines. *Computers & Chemical Engineering*, 15(2):85 – 103 (1991).

Shah, N., Pantelides, C., and Sargent, R. A general algorithm for short-term scheduling of batch operationsII. Computational issues. *Computers & Chemical Engineering*, 17(2):229 – 244 (1993), An International Journal of Computer Applications in Chemical Enginnering.

Sundaramoorthy, A. and Karimi, I. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chemical Engineering Science*, 60(10):2679 – 2702 (2005).