Online Appendix for "An Approximate Dynamic Programming Approach for Production-Delivery Scheduling under Non-stationary Demand"

April 20, 2021

Problem Formulation

Given the full information of orders over planning horizon, the IPDS problem being studied becomes a deterministic but challenging optimization problem. We provide a mathematical formulation for the deterministic problem. The following parameters and variables are defined and employed.

We briefly explain the model as follows. The average waiting time is defined in (1). Equations (2)-(3) imply that there is at most one order which is the first to be processed on machine m. Equations (4)-(5) imply that each order either precedes another order or is the last to be processed on a machine. Equation (6) indicates that each order is either the first to be processed on machine or succeeds another order. Equation (7) indicates that the starting processing time of an order is not earlier than its release time. Equations (8)-(10) define the completion time of orders. Equation (11) implies that the last vehicle departure time is not less than the maximum order completion time. Equation (12) implies that each order cannot be delivered before its completion time. Equations (13)-(15) indicate that each order must be allocated to some vehicle and choose a vehicle departure time. Equation (16) indicates the vehicle capacity cannot be violated.

Table 1: Parameters and decision variables

Parameters:	
$\mathcal{J}$	Set of customer orders, $\mathcal{J}$=$\{1, 2, ..., J\}$
$\mathcal{M}$	Set of identical parallel machines, $\mathcal{M}$= $\{1, 2, ..., M\}$
i, j	Indices of customer orders
m	Index of identical parallel machines
τ	Transportation time
p_j	Processing time of order j
a_j	Arrival time of order j
r_j	Release time of order j
d_ℓ	Vehicle departure time of vehicle ℓ, $\ell = 1, \ldots, L + 1$
q_ℓ	Vehicle capacity of vehicle ℓ
Λ	A sufficiently large number
Decision variables:	
S_j	Starting processing time of order j
C_j	Completion time of order j
D_j	Departure time of order j from the plant
y_{mi}	Binary variable which takes value 1 if order i is the first order processed on machine m ; otherwise it takes 0.
x_{ij}	Binary variable which takes value 1 if order j is processed immediately after order i and no other order is processed in between on that machine; otherwise it takes 0.
x_{jJ+1}	Binary variable which takes value 1 if order j is the last order processed on a machine, where order $J + 1$ is a dummy order.
$z_{j\ell}$	Binary variable which takes value 1 if order j is delivered by vehicle ℓ; otherwise it takes 0.

$$\min \quad \sum_{j=1}^{J}(D_j + \tau - a_j)/J \tag{1}$$

$$s.t. \quad 1 \geq \sum_{j=1}^{J} y_{jm}, \quad \forall m \in \mathcal{M} \tag{2}$$

$$y_{jm} \in \{0,1\}, \quad \forall j \in \mathcal{J}, \ m \in \mathcal{M} \tag{3}$$

$$1 = \sum_{j=1, i \neq j}^{J+1} x_{ij}, \quad \forall i \in \mathcal{J} \tag{4}$$

$$x_{ij} \in \{0,1\}, \quad \forall i,j \in \mathcal{J}, \ i \neq j \tag{5}$$

$$1 = \sum_{m=1}^{M} y_{mj} + \sum_{i=1, i \neq j}^{J} x_{ij}, \quad \forall j \in \mathcal{J} \tag{6}$$

$$S_j \geq r_j, \quad \forall j \in \mathcal{J} \tag{7}$$

$$C_j \geq y_{mj} p_j, \quad \forall j \in \mathcal{J}, \ \forall m \in \mathcal{M} \tag{8}$$

$$C_j \geq C_i + p_j - \Lambda(1 - x_{ij}), \quad \forall i,j \in \mathcal{J}, \ i \neq j \tag{9}$$

$$C_j \geq S_j + p_j, \quad \forall j \in \mathcal{J} \tag{10}$$

$$d_{L+1} \geq C_j, \quad \forall j \in \mathcal{J} \tag{11}$$

$$D_j \geq C_j, \quad \forall j \in \mathcal{J} \tag{12}$$

$$1 = \sum_{\ell=1}^{L+1} z_{j\ell}, \quad \forall j \in \mathcal{J} \tag{13}$$

$$D_j = \sum_{\ell=1}^{L+1} z_{j\ell} d_\ell, \quad \forall j \in \mathcal{J} \tag{14}$$

$$z_{j\ell} \in \{0,1\}, \quad \forall j \in \mathcal{J}, \ \forall \ell = 1, \ldots, L+1 \tag{15}$$

$$q_\ell \geq \sum_{j=1}^{J} z_{j\ell}, \ \forall \ell = 1, \ldots, L+1 \tag{16}$$

Code

The code is written only for academic purpose. Although the correctness has been carefully checked, the quality such as standardability, clarity, generality, and efficiency has not been well considered.

The code was written in MATLAB R2020a. Before running the code, please make sure that you have installed YALMIP and Gurobi, and connectted them to MATLAB.

The instance is randomly generated under stationary demand environment with $\lambda = 420$.

When loading the test instance, please modify the path accordingly.