Static Routing in Stochastic Scheduling: Performance Guarantees and Asymptotic Optimality
(*Online Appendix*)

Santiago R. Balseiro[1], David B. Brown[2], and Chen Chen[2]

[1]Graduate School of Business, Columbia University
[2]Fuqua School of Business, Duke University
srb2155@columbia.edu, dbbrown@duke.edu, cc459@duke.edu

First version: July 21, 2016
This version: May 19, 2018

B Additional Proofs and Derivations

B.1 Performance Bound on Static Routing With Dependent Jobs

Theorem B.1 (Extension of Theorem 4.1). *If $\frac{\mathbb{V}\mathrm{ar}[p_{jm}]}{\mathbb{E}[p_{jm}]^2} \leq \Delta$ holds for all jobs $j \in \mathcal{J}$ and machines $m \in \mathcal{M}$, then any static routing policy obtained from an optimal solution of (2) satisfies*

$$\begin{aligned} V^* \leq V^{\mathrm{R}} \leq V^* &+ \Big(\frac{M-1}{2M} + \frac{\Delta}{2}\Big) \sum_{j\in\mathcal{J}} w_j \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}] + \sum_{j\in\mathcal{J}} \alpha_j w_j \\ &+ \Big(2+\sqrt{\Delta M}\Big)\Big(\sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}]\Big)\Big(\sum_{j\in\mathcal{J}} \alpha_j \max_{m\in\mathcal{M}} r_{jm}\Big) + \frac{1}{2}\sum_{j\in\mathcal{J}} \beta_j \max_{m\in\mathcal{M}} r_{jm}. \end{aligned} \tag{1}$$

Proof. In the proof we use similar penalties as in Section 4.2 with the same parameters λ_{jm} and γ_{jm} as in Proposition 4.5. We first state and prove Lemma B.2, which bounds the parameters λ_{jm} in the routing penalty.

Lemma B.2. *Let $\lambda_{jm} = \nu_j^*/\mathbb{E}[p_{jm}] + \frac{1}{2}w_j$ with ν_j^* being an optimal Lagrange multiplier of the assignment constraint $\sum_{m\in\mathcal{M}} x_{jm} = 1$ in (2). We have the following inequalities*

$$0 \leq \lambda_{jm} \leq w_j + r_{jm} \sum_{i\in\mathcal{J}} \mathbb{E}[p_{im}]\,. \tag{2}$$

Proof. It suffices to show any optimal Lagrange multiplier ν_j^* of the assignment constraint $\sum_{m\in\mathcal{M}} x_{jm} = 1$ satisfies

$$0 \leq \nu_j^* \leq \frac{1}{2} w_j \mathbb{E}[p_{jm}] + r_{jm}\mathbb{E}[p_{jm}] \sum_{i\in\mathcal{J}} \mathbb{E}[p_{im}], \quad \forall m \in \mathcal{M}.$$

To show this, let $\mathbf{x}^* = (x_{jm}^*)_{j\in\mathcal{J}, m\in\mathcal{M}} \in \mathbb{R}^{J\times M}$ be an optimal solution and $\mu_{jm}^* \geq 0$ be an optimal Lagrange multiplier of the non-negativity constraint $x_{jm} \geq 0$ in (2). By the KKT conditions, for all jobs $j \in \mathcal{J}$ and machines $m \in \mathcal{M}$ we have

$$\frac{1}{2} w_j \mathbb{E}[p_{jm}] + \sum_{i\in\mathcal{J}} d_{ji}^m x_{im}^* = \nu_j^* + \mu_{jm}^*, \tag{3}$$

where $d_{ji}^m = \min(r_{im}, r_{jm})\mathbb{E}[p_{jm}]\mathbb{E}[p_{im}]$ is the (j,i)-th element in $\mathbf{D}_m$ as is defined in Lemma 3.1.

We first show $\nu_j^* \geq 0$. If not, from (3) we have $\mu_{jm}^* > 0$ for all $m \in \mathcal{M}$. Then according to the complementary slackness we have $x_{jm}^* = 0$ for all $m \in \mathcal{M}$. This contradicts the assignment constraint, which requires $\sum_{m\in\mathcal{M}} x_{jm}^* = 1$.

Next, since $\mu_{jm}^* \geq 0$ for all $m \in \mathcal{M}$, from (3) we have for all $m \in \mathcal{M}$,

$$\nu_j^* \leq \frac{1}{2} w_j \mathbb{E}[p_{jm}] + \sum_{i\in\mathcal{J}} d_{ji}^m x_{im}^* \leq \frac{1}{2} w_j \mathbb{E}[p_{jm}] + r_{jm}\mathbb{E}[p_{jm}] \sum_{i\in\mathcal{J}} \mathbb{E}[p_{im}]\,,$$

because $x_{im}^* \leq 1$ and $d_{ji}^m \leq r_{jm}\mathbb{E}[p_{jm}]\mathbb{E}[p_{im}]$. The result follows. □

We now prove Theorem B.1. We will restrict attention to non-anticipative policies where idling all machines at the same time is not allowed. This idling constraint is trivially satisfied by the optimal non-anticipative policy, and it allows us to bound the starting times of each job in the perfect information problem.

First, note that for the upper bound on V^{R}, Proposition 4.3 still holds. To derive a lower bound on V^* using a penalized perfect information relaxation, we define the sequencing penalty Y_{S} to be

$$Y_{\mathrm{S}} \triangleq \sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}} r_{jm} S^{\pi}_{jm}\big(p_{jm} - \mathbb{E}[p_{jm}|\mathcal{F}^{\pi}_{S^{\pi}_{jm}}]\big), \tag{4}$$

and the routing penalty Y_{R} to be

$$Y_{\mathrm{R}} \triangleq \sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}} x^{\pi}_{jm}\Big[\lambda_{jm}\big(\mathbb{E}[p_{jm}|\mathcal{F}^{\pi}_{S^{\pi}_{jm}}] - p_{jm}\big) + \gamma_{jm}\big(\mathbb{E}[p^2_{jm}|\mathcal{F}^{\pi}_{S^{\pi}_{jm}}] - p^2_{jm}\big)\Big]. \tag{5}$$

We set the parameters $(\lambda_{jm})_{j\in\mathcal{J},m\in\mathcal{M}}$ and $(\gamma_{jm})_{j\in\mathcal{J},m\in\mathcal{M}}$ as in the statement of Proposition 4.5. Since S^{π}_{jm} is measurable with respect to $\mathcal{F}^{\pi}_{S^{\pi}_{jm}}$ for all non-anticipative policies, we obtain by the law of total expectation that the sequencing penalty Y_{S} is dual feasible. An analogous argument shows the routing penalty Y_{R} is also dual feasible, as x^{π}_{jm} is measurable with respect to $\mathcal{F}^{\pi}_{S^{\pi}_{jm}}$ for all non-anticipative policies. Let

$$V^{\mathrm{H}}(\mathbf{p}) = \min_{\mathbf{x}\in\mathcal{X}}\Big\{\sum_{j\in\mathcal{J}} w_j C_j + Y_{\mathrm{S}} + Y_{\mathrm{R}}\Big\},$$

be the penalized perfect information relaxation problem for sample path $\mathbf{p}\in\mathbb{R}^{J\times M}$. We have

$$V^{\mathrm{H}} \triangleq \mathbb{E}[V^{\mathrm{H}}(\mathbf{p})] \le V^*, \tag{6}$$

is a lower bound on V^*, because the penalty $Y_{\mathrm{S}} + Y_{\mathrm{R}}$ is dual feasible.

Next, define the penalties as in Section 4:

$$\tilde{Y}_{\mathrm{S}} \triangleq \sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}} r_{jm} S^{\pi}_{jm}\big(p_{jm} - \mathbb{E}[p_{jm}]\big),$$

$$\tilde{Y}_{\mathrm{R}} \triangleq \sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}} x^{\pi}_{jm}\Big[\lambda_{jm}\big(\mathbb{E}[p_{jm}] - p_{jm}\big) + \gamma_{jm}\big(\mathbb{E}[p^2_{jm}] - p^2_{jm}\big)\Big],$$

and let $\tilde{V}^{\mathrm{H}}(\mathbf{p})$ be the perfect information problem with these penalties:

$$\tilde{V}^{\mathrm{H}}(\mathbf{p}) = \min_{\mathbf{x}\in\mathcal{X}}\Big\{\sum_{j\in\mathcal{J}} w_j C_j + \tilde{Y}_{\mathrm{S}} + \tilde{Y}_{\mathrm{R}}\Big\}.$$

We have that

$$V^{\mathrm{H}}(\mathbf{p}) \ge \tilde{V}^{\mathrm{H}}(\mathbf{p}) + \min_{\mathbf{x}\in\mathcal{X}}\Big\{Y_{\mathrm{S}} + Y_{\mathrm{R}} - \tilde{Y}_{\mathrm{S}} - \tilde{Y}_{\mathrm{R}}\Big\}. \tag{7}$$

Using the assumption of limited dependence (10) and (11), we can bound the difference in penalties as follows:

$$Y_{\mathrm{S}} + Y_{\mathrm{R}} - \tilde{Y}_{\mathrm{S}} - \tilde{Y}_{\mathrm{R}} \ge -\sum_{j\in\mathcal{J}}\alpha_j\sum_{m\in\mathcal{M}} r_{jm}S_{jm} - \sum_{j\in\mathcal{J}}\alpha_j\sum_{m\in\mathcal{M}} x_{jm}|\lambda_{jm}| - \sum_{j\in\mathcal{J}}\beta_j\sum_{m\in\mathcal{M}} x_{jm}|\gamma_{jm}|.$$

We next bound each of these terms. For the first term, we use the fact that the start time of every job $j\in\mathcal{J}$ must satisfy $S_j \le C_j \le \sum_{i\in\mathcal{J}}\max_{m\in\mathcal{M}} p_{im}$ because idling all machines at the same time is not allowed. Moreover, because each job is processed by one machine we have that

$\sum_{m\in\mathcal{M}} r_{jm} S_{jm} \le \big(\max_{m\in\mathcal{M}} r_{jm}\big) S_j$. Therefore,

$$\sum_{j\in\mathcal{J}} \alpha_j \sum_{m\in\mathcal{M}} r_{jm} S_{jm} \le \Big(\sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} p_{jm}\Big)\Big(\sum_{j\in\mathcal{J}} \alpha_j \max_{m\in\mathcal{M}} r_{jm}\Big).$$

For the second term, the assignment constraint and Lemma B.2 imply that

$$\sum_{j\in\mathcal{J}} \alpha_j \sum_{m\in\mathcal{M}} x_{jm}|\lambda_{jm}| \le \sum_{j\in\mathcal{J}} \alpha_j w_j + \Big(\sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}]\Big)\Big(\sum_{j\in\mathcal{J}} \alpha_j \max_{m\in\mathcal{M}} r_{jm}\Big).$$

For the third term, we use $\gamma_{jm} = -\frac{1}{2} r_{jm}$ and the assignment constraint to obtain that

$$\sum_{j\in\mathcal{J}} \beta_j \sum_{m\in\mathcal{M}} x_{jm}|\gamma_{jm}| \le \frac{1}{2} \sum_{j\in\mathcal{J}} \beta_j \max_{m\in\mathcal{M}} r_{jm}.$$

Taking expectations on both sides of (7) gives

$$\begin{aligned} V^{\text{H}} \ge \mathbb{E}\left[\tilde{V}^{\text{H}}(\mathbf{p})\right] - \sum_{j\in\mathcal{J}} \alpha_j w_j - \left(2 + \sqrt{\Delta M}\right)\Big(\sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}]\Big)\Big(\sum_{j\in\mathcal{J}} \alpha_j \max_{m\in\mathcal{M}} r_{jm}\Big) \\ - \frac{1}{2} \sum_{j\in\mathcal{J}} \beta_j \max_{m\in\mathcal{M}} r_{jm}, \end{aligned} \tag{8}$$

because

$$\mathbb{E}\Big[\max_{m\in\mathcal{M}} p_{jm}\Big] \le \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}] + M^{1/2} \max_{m\in\mathcal{M}} \left(\mathbb{V}\text{ar}[p_{jm}]\right)^{1/2} \le \left(1 + \sqrt{\Delta M}\right) \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}],$$

due to Devroye (1979) and $\mathbb{V}\text{ar}[p_{jm}] \le \Delta \mathbb{E}[p_{jm}]^2$ for all jobs $j \in \mathcal{J}$ and machines $m \in \mathcal{M}$. The result follows from using Proposition 4.3 to bound V^{R} from above, Proposition 4.5 to bound $\tilde{V}^{\text{H}}(\mathbf{p})$ from below, (6), and (8). □

B.2 Additional Details on Example 6.1

Because weights are one, we can express V^{A} as

$$V^{\text{A}} = \mathbb{E}\Bigg[\underbrace{\sum_{i=1}^{J/2} C_{2i-1}}_{(a)} + \underbrace{\sum_{i=1}^{J/2} C_{2i}\mathbf{1}_{\{p_{2i-1}=0\}}}_{(b)} + \underbrace{\sum_{i=1}^{J/2} C_{2i}\mathbf{1}_{\{p_{2i-1}=1\}}}_{(c)}\Bigg],$$

where (a) represents the completion time of odd-numbered jobs, (b) represents the completion time of even-numbered jobs processed immediately, and (c) represents the completion time of even-numbered jobs delayed to the end of sequence. The odd-numbered job $2i-1$ has to wait for all jobs with $j < 2i-1$ to either complete processing on the machine or transfer towards the end of the sequence. Let t_i be the total processing time of job pair (p_{2i-1}, p_{2i}) excluding the time to process job $2i$ if delayed. We have

$$\mathbb{E}[t_i] = \mathbb{E}[p_{2i-1}] + \mathbb{E}[p_{2i}\mathbf{1}\{p_{2i-1} = 0\}] = \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot(1 - 2q_i) = 1 - q_i.$$

Thus,

$$\mathbb{E}[(a)] = \sum_{i=1}^{J/2} \Big(\mathbb{E}[p_{2i-1}] + \sum_{j<i} \mathbb{E}[t_j]\Big) = \sum_{i=1}^{J/2} \Big(\frac{1}{2} + \sum_{j<i}(1-q_j)\Big) = \frac{1}{8}J^2 - \sum_{i=1}^{\frac{J}{2}} \sum_{j<i} q_j.$$

Similar, we have

$$\mathbb{E}[(b)] = \sum_{i=1}^{J/2} \mathbb{P}(p_{2i-1}=0)\Big(\mathbb{E}[p_{2i}|p_{2i-1}=0] + \sum_{j<i}\mathbb{E}[t_j]\Big) = \sum_{i=1}^{J/2} \frac{1}{2}\Big((1-2q_i) + \sum_{j<i}(1-q_j)\Big)$$
$$= \frac{1}{16}J^2 + \frac{1}{8}J - \frac{1}{2}\sum_{i=1}^{\frac{J}{2}}\sum_{j<i} q_j - \sum_{i=1}^{\frac{J}{2}} q_i.$$

To calculate (c), we first calculate the expectation conditioning on the realized processing times of all odd-numbered jobs $\{p_{2i-1}\}_{i=1}^{\frac{J}{2}}$, then we take expectation over these processing times. We let $N = \sum_{i=1}^{\frac{J}{2}} p_{2i-1}$ be the number of odd-numbered jobs with realized processing times of one. Altogether N even-numbered jobs will be moved towards the end of the sequence, because job $2i$ is delayed if $p_{2i-1} = 1$. Because the machine begins processing these jobs at time $N + \sum_{i=1}^{\frac{J}{2}}(1 - p_{2i-1})\mathbb{E}[p_{2i}|p_{2i-1} = 0]$, we have that

$$\mathbb{E}\left[(c) \mid \{p_{2i-1}\}_{i=1}^{\frac{J}{2}}\right] = N\Big(N + \sum_{i=1}^{\frac{J}{2}}(1-p_{2i-1})\mathbb{E}[p_{2i}|p_{2i-1}=0]\Big) + \sum_{i=1}^{\frac{J}{2}}\sum_{j\le i} p_{2i-1}p_{2j-1}\mathbb{E}[p_{2j}|p_{2j-1}=1]$$
$$= N\Big(N + \sum_{i=1}^{\frac{J}{2}}(1-p_{2i-1})(1-2q_i)\Big) + \sum_{i=1}^{\frac{J}{2}}\sum_{j\le i} p_{2i-1}p_{2j-1}2q_j.$$

Thus, we obtain by taking expectations over $\{p_{2i-1}\}_{i=1}^{\frac{J}{2}}$

$$\mathbb{E}[(c)] = \frac{1}{8}J^2 - \big(\frac{J}{2} - 2\big)\sum_{i=1}^{\frac{J}{2}} q_i + \sum_{i=1}^{\frac{J}{2}}\sum_{j<i} q_j + \frac{1}{2}\sum_{i=1}^{\frac{J}{2}}\sum_{j>i} q_j$$
$$\le \frac{1}{8}J^2 - \big(\frac{J}{2} - 2\big)\sum_{i=1}^{\frac{J}{2}} q_i + \frac{3}{2}\sum_{i=1}^{\frac{J}{2}}\sum_{j<i} q_j,$$

where the inequality is because q_i are sequenced in decreasing order.

Combining (a), (b), and (c) yields

$$V^{\mathrm{A}} \le \frac{5}{16}J^2 + \frac{1}{8}J + \big(1 - \frac{J}{2}\big)\sum_{i=1}^{\frac{J}{2}} q_i.$$

C Dynamic Programming Formulation

The optimal policy of the scheduling problem described in Section 2 can be obtained by solving Bellman's Equation, using backward induction. For sake of simplicity of presentation, we assume all random variables p_{jm} take only positive integral values, that is, $\mathbb{P}\{p_{jm} \in \mathbb{N}\} = 1$. Then we can further assume without loss of generality that jobs can only be started at integral points in time $t \in \mathbb{N}$.

We define the state to be (W, P, ℓ), where $W \subseteq \mathcal{J}$ is the subset of jobs waiting for service, and $P \subseteq \mathcal{J} \times \mathcal{M}$ is the subset of job-machine pairs under process. $\ell = (\ell_j)_{j \in \mathcal{J}} \in \mathbb{R}^J$ is defined as follows: if job j is currently under process on some machine, then ℓ_j is the elapsed processing time of job j; if job j is waiting for service, then $\ell_j = 0$; otherwise if job j is completed, ℓ_j can take an arbitrary value.

To describe Bellman's Equation, we introduce some helpful notation. Suppose $P \subseteq \mathcal{J} \times \mathcal{M}$ is a subset of job-machine pairs. We define two operations $\mathcal{J}(\cdot)$ and $\mathcal{M}(\cdot)$ on P such that $\mathcal{J}(P)$ and $\mathcal{M}(P)$ give the jobs and machines under process in the job-machine pairs in P, respectively. Also, for each set $A \subseteq \mathcal{J}$, we define the vector $\mathbf{I}_A = (\mathrm{I}^j_A)_{j \in \mathcal{J}} \in \{0, 1\}^J$ to be $\mathrm{I}^j_A = 1$ if $j \in A$, and $\mathrm{I}^j_A = 0$ otherwise.

Let $V(W, P, \ell)$ be the cost-to-go function when the state is (W, P, ℓ). If $W = \varnothing$, set $V(W, P, \ell) = \sum_{(j,m) \in P} w_j \mathbb{E}\left[p_{jm} - l_j | p_{jm} > l_j\right]$. If $W \neq \varnothing$, then the action set is

$$\mathcal{A}(W, P) = \{\hat{P} \subseteq W \times \mathcal{M} \setminus \mathcal{M}(P) : \text{if } (j, m), (j', m') \in \hat{P}, \text{then } j \neq j' \text{and } m \neq m'\}.$$

That is, $\mathcal{A}(W, P)$ is the set of all possible job-machine pairs that can be assigned when jobs in W are waiting and job-machine pairs in P are under process. Thus, if $\hat{P} \in \mathcal{A}(W, P)$ is chosen, the decision maker will assign job j to currently idle machine m for each job-machine pair $(j, m) \in \hat{P}$.

In addition, define $\mathcal{F} \subseteq P$ to be the random set of job-machine pairs that are currently under process and will be completed by the beginning of next stage, i.e., for each set $F \subseteq P$,

$$\mathbb{P}\{\mathcal{F} = F | P, \ell\} = \prod_{(j,m) \in P \setminus F} \mathbb{P}\{p_{jm} > \ell_j + 1 | p_{jm} > \ell_j\} \prod_{(j,m) \in F} \mathbb{P}\{p_{jm} = \ell_j + 1 | p_{jm} > \ell_j\}.$$

Then the Bellman Equation can be written as:

$$V(W, P, \ell) = \min_{\hat{P} \in \mathcal{A}(W,P)} \left\{ \sum\nolimits_{j \in \mathcal{J}(P) \cup W} w_j + \mathbb{E}_{\mathcal{F} | P \cup \hat{P}, \ell} \left[V\left(W \setminus \mathcal{J}(\hat{P}), \{P \cup \hat{P}\} \setminus \mathcal{F}, \ell + \mathbf{I}_{\mathcal{J}\left(\{P \cup \hat{P}\} \setminus \mathcal{F}\right)} \right) \right] \right\}.$$

The optimal performance is given by $V^* = V(\mathcal{J}, \varnothing, \mathbf{0})$.

D Binary Linear Programming Formulation for Optimal Static Routing

In this section, we provide an approach to calculating an exact solution to optimization problem (1) using 0-1 integer programming. The approach may be viable when the number of jobs is not large.

We can write problem (1) in matrix form as

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{m \in \mathcal{M}} \mathbf{c}'_m \mathbf{x}_m + \frac{1}{2} \mathbf{x}'_m \tilde{\mathbf{D}}_m \mathbf{x}_m, \tag{9}$$

where $\mathbf{c}_m = (w_j \mathbb{E}[p_{jm}])_{j \in \mathcal{J}} \in \mathbb{R}^J$, and $\tilde{\mathbf{D}}_m = (\tilde{d}_{ij}^m) \in \mathbb{R}^{J \times J}$ is the matrix such that

$$\tilde{d}_{ij}^m = \begin{cases} 0 & i = j, \\ w_i \mathbb{E}[p_{jm}] & j \prec_m i, \\ w_j \mathbb{E}[p_{im}] & i \prec_m j. \end{cases}$$

We claim that the optimal value of problem (9) does not change if we replace the non-negative constraint $x_{jm} \geq 0$ with the binary constraint $x_{jm} \in \{0, 1\}$ for each job-machine pair. To see this, suppose $\mathbf{x}$ is an optimal solution for (9), then using the method of conditional probabilities we can derandomize $\mathbf{x}$ to a feasible binary solution whose performance cannot be any worse. On the other hand, every feasible binary solution can not be any better than $\mathbf{x}$, because $\mathbf{x}$ is an optimal solution to (9).

Define new binary variables $y_{ij}^m \triangleq x_{im} x_{jm}$. Then the objective of problem (9) is equivalent to

$$\sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} c_{jm} x_{jm} + \sum_{m \in \mathcal{M}} \sum_{i < j \in \mathcal{J}} y_{ij}^m \tilde{d}_{ij}^m.$$

We can guarantee that $y_{ij}^m = x_{im} x_{jm}$ using the following constraints:

$$\begin{aligned} & y_{ij}^m \leq x_{jm}, \\ & y_{ij}^m \leq x_{im}, \\ & y_{ij}^m \geq x_{jm} + x_{im} - 1, \\ & y_{ij}^m \in \{0, 1\}. \end{aligned}$$

As a result, problem (9) is equivalent to

$$\begin{aligned} \underset{x_{jm}, y_{ij}^m}{\text{minimize}} \quad & \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} c_{jm} x_{jm} + \sum_{m \in \mathcal{M}} \sum_{i < j \in \mathcal{J}} y_{ij}^m \tilde{d}_{ij}^m \\ \text{subject to} \quad & \sum_{m \in \mathcal{M}} \mathbf{x}_m = \mathbf{1}, \\ & y_{ij}^m \leq x_{jm}, \\ & y_{ij}^m \leq x_{im}, \\ & y_{ij}^m \geq x_{jm} + x_{im} - 1, \\ & x_{jm} \in \{0, 1\}, \\ & y_{ij}^m \in \{0, 1\}, \end{aligned}$$

which is a 0-1 integer programming problem.

References

Devroye, L. P. (1979), 'Inequalities for the completion times of stochastic pert networks', *Mathematics of Operations Research* **4**(4), 441–447.