Online Supplement of the Paper "Chance-Constrained Surgery Planning Under Conditions of Limited and Ambiguous Data"

Yan Deng, Siqian Shen, Brian Denton

E-mails: yandeng@umich.edu, siqian@umich.edu, btdenton@umich.edu

Department of Industrial and Operations Engineering

University of Michigan, Ann Arbor, MI, USA

Appendix A: Ambiguity Set Configuration

A.1. Support and empirical distribution of $\boldsymbol{\xi}$

There are a variety of ways of deriving the support Ξ and empirical distribution function f_0 of $\boldsymbol{\xi}$ to construct the ambiguity set $\mathcal{F}_\phi(d)$ in (11). We start with building the marginal empirical distribution of the one-dimensional ξ_i of each surgery $i \in \mathcal{S}$, to estimate the joint empirical distribution f_0. This follows how historical data points are collected in the context of surgery planning – practitioners can easily collect samples of the duration time of each surgery but not their joint realizations. We will also partition the support of ξ_i into finite intervals, and count the frequency of a realization appearing in each interval to estimate the marginal empirical distribution function. We describe the details of the method as follows.

Suppose that we have N_o historical data points, $\{\hat{\xi}_i^n\}_{n=1}^{N_\text{o}}$, of the random duration time ξ_i for each surgery $i \in \mathcal{S}$. (The value N_o could be different for different surgeries in $\mathcal{S}$.) Interval $[\underline{\xi}_i, \overline{\xi}_i]$ denotes the range of the random duration ξ_i, where $\underline{\xi}_i$ and $\overline{\xi}_i$ can be set based on the minimum and the maximum value of $\hat{\xi}_i^n$, respectively. We partition $[\underline{\xi}_i, \overline{\xi}_i]$ into subintervals $\mathcal{B}_1^i, \ldots, \mathcal{B}_{K_i}^i$, resulting in a finite and discrete support $\Xi_i = \{\mathcal{B}_k^i\}_{k=1}^{K_i}$ for ξ_i. The empirical probability mass function f_0^i of each random ξ_i, $i \in \mathcal{S}$ is given by

$$f_0^i(\mathcal{B}_{k_i}^i) = \frac{\sum_{n=1}^{N_\text{o}} \mathbb{I}\{\hat{\xi}_i^n \in \mathcal{B}_{k_i}^i\}}{N_\text{o}}, \; k_i = 1, \ldots, K_i, \tag{36}$$

where $\mathbb{I}(\cdot)$ is an indicator function that returns value 1 if $\cdot$ is true and 0 otherwise. We derive the set Ξ and function f_0 based on their marginal counterparts as follows. First, Ξ is the Cartesian product of the supports Ξ_i, $i \in \mathcal{S}$. Consider that the support set Ξ is partitioned into K subregions, denoted by $\Xi = \{\mathcal{B}_1, \ldots, \mathcal{B}_K\}$. Then, $\Xi = \bigtimes_{i \in \mathcal{S}} \Xi_i$, where $K = \prod_{i \in \mathcal{S}} K_i$ and each subregion $\mathcal{B}_k$ is given by an S-tuple $(\mathcal{B}_{k_1}^1, \ldots, \mathcal{B}_{k_S}^S)$ that chooses one $\mathcal{B}_{k_i}^i$ from the support Ξ_i for each $i \in \mathcal{S}$. We use the tensor product of the marginal probabilities f_0^i, $i \in \mathcal{S}$ to derive f_0. That is, for any $\mathcal{B}_k \in \Xi$, we have

$$f_0(\mathcal{B}_k) = \frac{\sum_{n=1}^{N_\text{o}} \mathbb{I}\{\hat{\boldsymbol{\xi}}^n \in \mathcal{B}_k\}}{N_\text{o}} = \prod_{i \in \mathcal{S}} f_0^i(\mathcal{B}_{k_i}^i). \tag{37}$$

A.2. Configuring parameter d

Parameter d indicates the risk preference of a decision maker, and can be set to any nonnegative value. If $d = 0$, the decision maker only plans against the true distribution $f = f_0$, and trusts the empirical distribution completely. When $d > 0$, a smaller d-value leads to a tighter set $\mathcal{F}_\phi(d)$ based on the same ϕ-function, and thus we can obtain less conservative DR solutions. On the other hand, the value of d needs to be sufficiently large to maintain a desired confidence level for satisfying the DR chance constraint for any possible true distribution $f \in \mathcal{F}_\phi(d)$. Thus, more data (i.e., larger N_o), smaller confidence level, smaller dimensions of decision variables and uncertain parameters will all lead to smaller d. In this paper, we follow suggestions by Ben-Tal et al. (2013) to derive appropriate d values used in our DR-CCSP model.

Let $\phi''(1)$ be the second derivative of function ϕ evaluated at 1; m is the cardinality of the support of the random vector; $\chi^2_{m-1,1-\alpha}$ is the $1-\alpha$ percentile of the χ^2_{m-1}-distribution, i.e., $\mathbb{P}(X \geq \chi^2_{m-1,1-\alpha}) = \alpha$, of which X follows a χ^2_{m-1}-distribution. We propose to use

$$d = d_\text{J} = \frac{\phi''(1)}{2N_\text{o}} \chi^2_{m-1,1-\alpha}, \tag{38}$$

based on statistical inference results (see Pardo 2005), to *asymptotically* guarantee that set $\mathcal{F}_\phi(d_\text{J})$ contains the true probability distribution function f at $(1-\alpha)$ confidence when N_o goes to infinity. Note that $m = K = \prod_{i \in \mathcal{S}} K_i$ in our problem, dependent on the number of surgeries S and the size K_i of every ξ_i. This could lead to a very large d_J and subsequently a conservative ambiguity set $\mathcal{F}_\phi(d_\text{J})$.

Alternatively, one can use a marginal counterpart of d_J denoted as

$$d = d_\text{M} = \sum_{i \in \mathcal{S}} d_i,$$

to approximate the result, where each d_i is derived according to (38) for bounding the ϕ-divergence distance of the marginal true and empirical probability distribution functions f^i and f^i_0, such that $d_i = \frac{\phi''(1)}{2N_\text{o}} \chi^2_{K_i-1,1-\alpha}$. Comparing the values of $\sum_{i \in \mathcal{S}} \chi^2_{K_i-1,1-\alpha}$ with $\chi^2_{K-1,1-\alpha}$, the value of d_M is usually much smaller than d_J, yielding a less conservative and more computationally tractable CCSP reformulation of DR-CCSP.

Appendix B: A Branch-and-Cut Algorithm

At each branching node ν, we optimize the following relaxation problem:

$$\begin{aligned}
\mathbf{NR}(\nu): \quad \min \quad & \mathbf{c}^\mathsf{T}\mathbf{x} \\
\text{s.t.} \quad & (1)-(3),\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{1},\ \mathbf{0} \leq \mathbf{z} \leq \mathbf{1} \\
& \sum_{\omega \in \Omega} \kappa_\omega \leq L,\ \mathbf{0} \leq \boldsymbol{\kappa} \leq \mathbf{1} && (39) \\
& (\mathbf{z}, \boldsymbol{\kappa}) \in Q' && (40) \\
& x_j = 0, \forall j \in F^0_x(\nu),\ x_j = 1, \forall j \in F^1_x(\nu) && (41) \\
& z^j_{ik} = 0, \forall (j,i,k) \in F^0_z(\nu),\ z^j_{ik} = 1, \forall (j,i,k) \in F^1_z(\nu) && (42) \\
& \kappa_\omega = 0, \forall \omega \in F^0_\kappa(\nu),\ \kappa_\omega = 1, \forall \omega \in F^1_\kappa(\nu). && (43)
\end{aligned}$$

Constraints (39) yield a linear programming relaxation of $\Gamma(\beta^d)$; set Q' in (40) collects the current set of packing and scheduling cuts that aim at enforcing $(\mathbf{z}, \boldsymbol{\kappa}) \in Q$; constraints (41)–(43) specify the values of $\mathbf{x}$, $\mathbf{z}$ and $\boldsymbol{\kappa}$ at node ν. Specifically, $F^0_x(\nu)$ and $F^1_x(\nu)$ are disjoint subsets of $\mathcal{R}$, $F^0_z(\nu)$ and $F^1_z(\nu)$ are disjoint subsets of $\mathcal{R} \times \mathcal{S} \times \mathcal{S}$, and $F^0_\kappa(\nu)$ and $F^1_\kappa(\nu)$ are disjoint subsets of Ω.

If $\mathbf{NR}(\nu)$ is infeasible or it attains an optimal objective value that is no better than an incumbent upper bound $\overline{\texttt{obj}}$, we fathom node ν. Otherwise, let $(\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\kappa}})$ be the current solution, which will go through

Algorithm 1, and then go through Algorithm 2 if it passes Algorithm 1. We re-solve model $\mathbf{NR}(\nu)$ whenever a packing cut or a scheduling cut is generated into the set Q' of cuts in (40), but we do not wait until collecting all the cuts because the former approach is computationally more efficient based on our numerical experiences described later. If no cuts are derived from subroutines in Algorithms 1 and 2, then $(\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\kappa}})$ is feasible to CCSP. We use its associated objective value to update the upper bound $\overline{\texttt{obj}}$, and proceed to other branching nodes that have not been fathomed. Algorithm 3 outlines the overall branch-and-cut approach.

Algorithm 3 A branch-and-cut procedure for solving the MILP formulation of CCSP.

1: $\overline{\texttt{obj}} \leftarrow \infty$.
2: Initialize NodeList $\leftarrow \{\nu_0\}$, where $F_x^n(\nu_0) = \emptyset,\ F_z^n(\nu_0) = \emptyset,\ F_\kappa^n(\nu_0) = \emptyset$ for $n = 0$ and 1.
3: choose a branching node $\nu \in$ NodeList.
4: solve $\mathbf{NR}(\nu)$.
5: **if** attain $(\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\kappa}})$ with $\mathbf{c}^\mathsf{T}\hat{\mathbf{x}} < \overline{\texttt{obj}}$ **then**
6:     `CUT` $\leftarrow$ GenCut_P$(\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\kappa}})$.
7:     **if** `CUT` $\neq$ `null` **then**
8:         add the packing cut `CUT` to set Q', and go to Step 4.
9:     **end if**
10:     **if** $(\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\kappa}})$ is integral **then**
11:         $\mathcal{C} \leftarrow$ GenCut_S$(\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\kappa}})$.
12:         **if** $\mathcal{C} \neq \emptyset$ **then**
13:             add every scheduling cut in $\mathcal{C}$, and go to Step 4.
14:         **else**
15:             $\overline{\texttt{obj}} \leftarrow \mathbf{c}^\mathsf{T}\hat{\mathbf{x}}$.
16:         **end if**
17:     **else**
18:         branch on a binary variable having a fractional solution value and add two new branching nodes to NodeList.
19:     **end if**
20: **end if**
21: NodeList $\leftarrow$ NodeList$\backslash\{\nu\}$.
22: **if** NodeList is empty **then**
23:     report $\overline{\texttt{obj}}$ as the optimal objective value.
24: **else**
25:     go to Step 3.
26: **end if**

Computational enhancements: When implementing Algorithm 3, we also implement the following enhancement procedures.

- We order ORs $j = 1, \ldots, R$ such that $T_1 \geq \cdots \geq T_R$. Then, the packing cut (27), the scheduling cuts (30) and (32), derived based on a particular OR j, are also valid for any $j' = j+1, \ldots, R$. We therefore propagate each cut to every OR j' that is open, which will strengthen the model $\mathbf{NR}(\nu)$ at each branching node ν.

- We store the values of $\pi_\omega(a),\ \forall \omega \in \Omega$ for all distinct surgery sequences a, so that later if we run into the same sequence again, we do not have to re-compute the result. This avoids repeatedly computing $\pi_\omega(a),\ \forall \omega \in \Omega$ for some common prefix a choices.

- We compare a few different branching rules and apply *strong branching* to improve solution quality and decrease CPU time.

Appendix C: Results of a Benchmark Model without Constraint (6)

We test a model variant of CCSP, in which we relax Constraint (6) that requires each surgery being started no earlier than its planned start time. Such a benchmark will enforce zero idle time, since we intend to start each surgery as early as possible conditional on the predecessor being completed, to reduce both waiting time and overtime. We report the overtime and idle time results for $\beta = 0.85, 0.90, 0.95, 1.00$ for this new model, which are similar to the ones of CCSP presented in Table 4 in Section 4.3. This is expected, since the idle time in Table 5 is already very short (under 5 minutes in most cases for $\beta = 0.95$ and 1.00) given by CCSP with Constraint (6). For the instances we tested, the primary driving constraints are the ones related to waiting time and overtime.

Table 13 Overtime (in minutes) of CCSP without Constraint (6)

β	$\mathtt{C}^{\text{open}}$ (# of ORs)	overtime $\mathcal{T}^{\omega}_{\text{over}},\ \omega \in \Theta$ (min)				
		Prob	mean	stdev	50%	95%
0.85	4.0 (4)	0.71	18.4	46.3	0	76.6
0.90	4.2 (4)	0.79	17.7	42.5	0	63.6
0.95	5.7 (6)	0.88	11.2	34.3	0	55.3
1.00	5.8 (6)	0.89	5.8	24.2	0	29.1