Online Supplement

Appendix A Proofs of Lemmas

A.1 Proof of Lemma 1

Suppose the number of opened facilities is m, i.e., $\sum_{j\in J}\hat{y}_j = m$. Since the worst-case scenario always occurs at the extreme points, there will be exactly k disrupted facilities in a worst-case scenario. Figure A1 gives an example where $m > k$. We index the facilities so that the first m are opened and the rest are closed. Without loss of generality, we assume that the first k facilities are disrupted in scenario $\mathbf{z}^1$ and that facilities 2, ..., k and $m+1$ are disrupted in scenario $\mathbf{z}^2$. Since more opened facilities are disrupted in scenario $\mathbf{z}^1$, the customers have more options in scenario $\mathbf{z}^2$, leading to $B_2 \le B_1$. Therefore, we have $B_1 \ge B_2$ for $m > k$. When $m \le k$, all the demand will be left unsatisfied and we have $B_1 = B_2$.

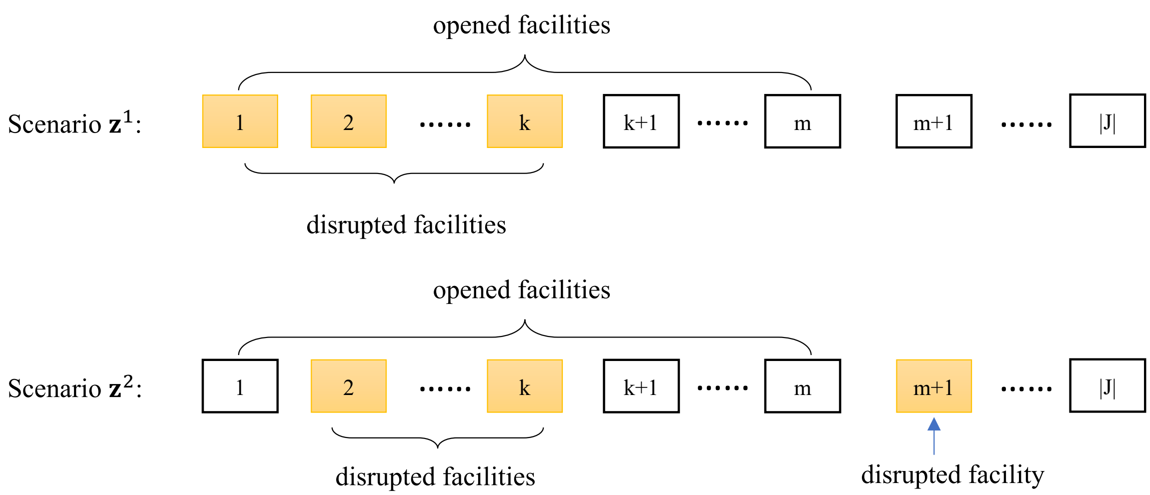

Figure A1 **Illustration for the proof of Lemma 1**

A.2 Proof of Lemma 3

To prove $\mathbb{Z}'(k)$ is equivalent to $\mathbb{Z}(k)$, we need to show that even though we relax the variables $\mathbf{z}$ to be continuous, the optimal values of $\mathbf{z}$ in a worst-case scenario still take integer values (0 or 1). This can be proved through the subproblems of the duality-based C&CG algorithms (refer to B.1 and C.1). Specifically, for the UFLP, we can rewrite the objective function (B1) as

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{j\in J}\left(\sum_{i\in I}\hat{y}_j\beta_{ij}(1-z_j)\right). \tag{A1}$$

Constraints (B2)–(B7) indicate that variables $\mathbf{z}$ are not linked to other variables (i.e., $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$). Since the subproblem is a maximization problem, the third term subtracted from Equation (A1) is expected to be as small as possible. To achieve this, we can rank $\sum_{i\in I}\hat{y}_j\beta_{ij}(\geq 0)$ for each j, and for the first k largest values, z_j would take the value of 1 (even though z_j is relaxed to be continuous) at optimum. For other values, z_j would be 0. Similarly, for the CFLP, we can rewrite the objective function (C1) as

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{j\in J}\left(\left(\sum_{i\in I}\beta_{ij} - C_j\gamma_j\right)\hat{y}_j(1-z_j)\right). \tag{A2}$$

If $\left(\sum_{i\in I}\beta_{ij} - C_j\gamma_j\right)\hat{y}_j \leq 0$, z_j would be 0 at optimum. For those with $\left(\sum_{i\in I}\beta_{ij} - C_j\gamma_j\right)\hat{y}_j > 0$, the same method describe above for the UFLP applies.

Appendix B Duality-Based C&CG Algorithm and Benders Decomposition Method for Adjustable Robust UFLP

B.1 Duality-Based C&CG Algorithm

Master Problem. The master problem is defined by Equations (9a)–(9g).

Subproblem. Let λ and β be the dual variables of constraints (5b) and (5c), respectively. The dual problem of the inner minimization problem can be written as

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(1-z_j)\beta_{ij}, \tag{B1}$$

$$\text{s.t.}\quad \lambda_i - \beta_{ij} \leq h_i d_{ij} \qquad \forall i\in I, j\in J, \tag{B2}$$

$$\lambda_i \leq p_i h_i \qquad \forall i\in I, \tag{B3}$$

$$\sum_{j\in J} z_j \leq k, \tag{B4}$$

$$z_j \in \{0,1\} \qquad \forall i\in I, \tag{B5}$$

$$\lambda_i \geq 0 \qquad \forall i\in I, \tag{B6}$$

$$\beta_{ij} \geq 0 \qquad \forall i\in I, j\in J. \tag{B7}$$

Since the nonlinear term $z_j\beta_{ij}$ is the product of a binary variable z_j and a continuous variable β_{ij}, we can linearize it by introducing a new variable $\pi_{ij} = z_j\beta_{ij}$ and adding the following constraints:

$$
\begin{aligned}
&\pi_{ij} \geq 0 && \forall i \in I, j \in J,\\
&\pi_{ij} \leq \beta_{ij} && \forall i \in I, j \in J,\\
&\pi_{ij} \leq \mathbb{M}_{ij} z_j && \forall i \in I, j \in J,\\
&\pi_{ij} \geq \beta_{ij} + \mathbb{M}_{ij}(z_j - 1) && \forall i \in I, j \in J,
\end{aligned}
\tag{B8}
$$

where $\mathbb{M}_{ij} = \max\{h_i(p'_i - d_{ij}), 0\}$. Therefore, the full subproblem is

$$
\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\pi}} \sum_{j\in J} f_j \hat{y}_j + \sum_{i\in I} \lambda_i - \sum_{i\in I}\sum_{j\in J} \hat{y}_j(\beta_{ij} - \pi_{ij}),
\tag{B9}
$$

subject to constraints (B2)–(B8).

B.2 Benders Decomposition Method

Master problem. The master problem of the Benders decomposition method is

$$
\begin{aligned}
&\phi = \min_{\mathbf{y},s} s,\\
&\text{s.t.} \quad s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I} \hat{\lambda}_i^l - \sum_{i\in I}\sum_{j\in J} \hat{\beta}_{ij}^l y_j (1 - \hat{z}_j^l) \qquad \forall l \in \{1, ..., n\},\\
&\qquad y_j \in \{0,1\} \qquad \forall j \in J,
\end{aligned}
\tag{B10}
$$

where $\hat{\lambda}^l, \hat{\beta}^l$, and $\hat{z}^l$ are obtained at the lth iteration by solving the subproblem.

Subproblem. The subproblem is defined by Equations (B2)–(B9).

Appendix C Duality-Based C&CG Algorithm and Benders Decomposition Method for Adjustable Robust CFLP

C.1 Duality-Based C&CG Algorithm

Master problem. The master problem is defined by Equations (9a)–(9g) and (11).

Subproblem. Let γ be the dual variable of constraints (6). The resulting dual problem can be written as follows:

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(1-z_j)\beta_{ij} - \sum_{j\in J} C_j\hat{y}_j\gamma_j(1-z_j), \tag{C1}$$

$$\begin{aligned} \text{s.t.}\quad & \lambda_i - \beta_{ij} - h_i\gamma_j \le h_i d_{ij} && \forall i\in I, j\in J, \\ & \gamma_j \ge 0 && \forall j \in J, \\ & \text{and (B3)–(B7).} \end{aligned} \tag{C2}$$

There are two nonlinear terms in the objective function (C1), i.e., $z_j\beta_{ij}$ and $\gamma_j z_j$. We can use the technique in B.1 to linearize the term $z_j\beta_{ij}$. For the term $\gamma_j z_j$, we introduce a new variable $\zeta_j = \gamma_j z_j$ and add the following constraints:

$$\begin{aligned} & \zeta_j \ge 0 && \forall j\in J, \\ & \zeta_j \le \gamma_j && \forall j\in J, \\ & \zeta_j \le \mathbb{M}_j z_j && \forall j \in J, \\ & \zeta_j \ge \gamma_j + \mathbb{M}_j(z_j-1) && \forall j\in J, \end{aligned} \tag{C3}$$

where $\mathbb{M}_j = \max\{\max_i(p_i - d_{ij}), 0\}$.

Therefore, the full subproblem is

$$\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\zeta}} \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(\beta_{ij}-\pi_{ij}) - \sum_{i\in J} C_j\hat{y}_j(\gamma_j - \zeta_j), \tag{C4}$$

subject to constraints (B8) and (C2)–(C3).

C.2 Benders Decomposition Method

Master problem. The master problem of the Benders decomposition method is

$$\begin{aligned} \phi = \min_{\mathbf{y},s}\ & s, \\ \text{s.t.}\quad & s \ge \sum_{j\in J} f_j y_j + \sum_{i\in I}\hat{\lambda}_i^l - \sum_{i\in I}\sum_{j\in J} y_j(1-\hat{z}_j^l)\hat{\beta}_{ij}^l - \sum_{j\in J} C_j y_j\hat{\gamma}_j^l(1-\hat{z}_j^l) && \forall l\in\{1,\dots,n\}, \\ & y_j \in\{0,1\} && \forall j\in J, \end{aligned} \tag{C5}$$

where $\hat{\lambda}^l, \hat{\beta}^l$, and $\hat{z}^l$ are obtained at the lth iteration by solving the subproblem.

Subproblem. The subproblem is defined by Equations (B8) and (C2)–(C4).

Appendix D Reformulation of the AARC Models

The AARC model for the robust UFLP can be reformulated as

$$
\begin{aligned}
\min_{\substack{\mathbf{y},\mathbf{W},\mathbf{w},\mathbf{A},\mathbf{a},s,\delta,\\ \boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\theta},\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{\varsigma},\boldsymbol{\pi},\boldsymbol{\nu}}} \quad & s, \\
\text{s.t.} \quad & s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} w_{ij} + \sum_{i\in I} p_i h_i a_i + k\delta + \sum_{e\in J}\alpha_e, \\
& \delta + \alpha_e \geq \sum_{i\in I}\sum_{j\in J} h_i d_{ij} W_{ije} + \sum_{i\in I} p_i h_i A_{ie} && \forall e\in J, \\
& \sum_{j\in J} w_{ij} + a_i - k\xi_i - \sum_{e\in J}\eta_{ie} \geq 1 && \forall i\in I, \\
& \xi_i + \eta_{ie} \geq -\sum_{j\in J} W_{ije} - A_{ie} && \forall i\in I, e\in J, \\
& -k\theta_{ij} - \sum_{e\in J}\mu_{ije} \geq -y_j + w_{ij} && \forall i\in I, j\in J, \\
& \theta_{ij} + \mu_{ije} \geq W_{ije} + y_j && \forall i\in I, j\in J, e\in J, e=j, \\
& \theta_{ij} + \mu_{ije} \geq W_{ije} && \forall i\in I, j\in J, e\in J, e\neq j, \\
& w_{ij} - k\sigma_{ij} - \sum_{e\in J}\varsigma_{ije} \geq 0 && \forall i\in I, j\in J, \\
& \sigma_{ij} + \varsigma_{ije} \geq -W_{ije} && \forall i\in I, j\in J, e\in J, \\
& -k\pi_i - \sum_{e\in J}\nu_{ie} + a_i \geq 0 && \forall i\in I, \\
& \pi_i + \nu_{ie} \geq -A_{ie} && \forall i\in I, e\in J, \\
& y_j \in \{0,1\} && \forall j\in J, \\
& \delta, \alpha_e, \xi_i, \eta_{ie}, \theta_{ij}, \mu_{ije}, \sigma_{ij}, \varsigma_{ije}, \pi_i, \nu_{ie} \geq 0 && \forall i\in I, j\in J, e\in J.
\end{aligned} \tag{D1}
$$

The AARC model for the robust CFLP can be reformulated with (D1) and the constraints

$$
\begin{aligned}
& k\rho_j + \sum_{e\in J}\Gamma_{ej} + \sum_{i\in I} h_i w_{ij} \leq C_i y_j && \forall j\in J, \\
& \rho_j + \Gamma_{ej} \geq \sum_{i\in I} h_i W_{ije} + C_j y_j && \forall e\in J, j\in J, e=j \\
& \rho_j + \Gamma_{ej} \geq \sum_{i\in I} h_i W_{ije} && \forall e\in J, j\in J, e\neq j \\
& \rho_j, \Gamma_{ej} \geq 0 && \forall e\in J, j\in J.
\end{aligned} \tag{D2}
$$

Appendix E Statistics of CPU Time for Master and Sub Problems

Table E1 presents the statistical results of the CPU time spent on the master and sub problems respectively. The *time reduction factor* is calculated as the time spent by the C&CG-D algorithm divided by the time spent by the C&CG-E algorithm. A time factor equals to 2 indicates that the problem spends only half the time using the C&CG-E compared to using the C&CG-D. Table E1 shows that the main time reduction comes from the computing time for the master problem, which indicates that the cuts generated from the second-worst scenarios can improve the convergence of the C&CG algorithm. When $k = 3$ and $k = 4$, the time factor of the master problem is only a bit larger than 1, this is because some instances are not solved to optimality and the time limit is reached. We also notice that the solution process of the subproblem of the C&CG-E consumes little time although we need to solve a large number of MCFPs. This process is efficient because we use the network simplex algorithm (via the NetworkX package) which works well for the MCFP model (Király and Kovács 2012).

Table E1 Statistics of average CPU time for master and sub problems

			C&CG-E		C&CG-D		Time reduction factor					C&CG-E		C&CG-D		Time reduction factor	
Model	k	p	Master	Sub	Master	Sub	Master	Sub	Model	k	p	Master	Sub	Master	Sub	Master	Sub
UFLP	2	$p^{0.8}$	5295.7	0.5	9463.1	12.2	1.79	23.76	CFLP	2	$p^{0.8}$	3306.0	1.6	4050.0	9.0	1.23	5.82
		p^{max}	1298.4	0.4	3803.0	10.1	2.93	23.37			p^{max}	4650.7	1.9	5967.4	10.8	1.28	5.67
	3	$p^{0.8}$	43919.4	1.1	44583.3	21.2	1.02	18.55		3	$p^{0.8}$	29998.2	6.7	31490.8	21.6	1.05	3.21
		p^{max}	46193.4	1.8	46689.0	24.6	1.01	13.89			p^{max}	28908.2	8.9	29623.3	22.7	1.02	2.56
	4	$p^{0.8}$	57341.0	1.0	58148.3	24.9	1.01	24.18		4	$p^{0.8}$	41155.5	13.1	42011.5	24.6	1.02	1.88
		p^{max}	52229.5	1.7	53508.4	26.5	1.02	15.80			p^{max}	39355.7	22.7	40930.1	31.7	1.04	1.39

Appendix F Results of Adjustable Robust UPMP and CPMP

In this section, we apply the C&CG-E to the adjustable robust uncapacitated/capacitated PMP (UPMP/CPMP). We first introduce the models and then report the numerical results.

F.1 The Adjustable Robust UPMP and CPMP Models

For both models, the objective function optimizes the weighted sum of the nominal cost and the worst-case cost. The detailed models (An et al. 2014) are as follows.

Notation. The parameter ϱ is the weight of the worst-case cost. The variable b_{ij} is the fraction of demand from customer $i \in I$ that is satisfied by facility $j \in J$ in a disruptive scenario. The variable g_i is the unsatisfied portion of demand at customer $i \in I$ in a disruptive scenario. The definitions of the other parameters and variables are the same as in Section 3.

The UPMP under disruptions can be formulated as

$$\min_{\mathbf{x},\mathbf{y}}(1-\varrho)\sum_{i\in I}\sum_{j\in J} d_{ij}h_i x_{ij} + \varrho \max_{\mathbf{z}\in\mathbb{Z}(k)} \min_{(\mathbf{b},\mathbf{g})\in S(\mathbf{y},\mathbf{z})} \Big(\sum_{i\in I}\sum_{j\in J} d_{ij}h_i b_{ij} + \sum_{i\in I} p_i h_i g_i\Big), \tag{F1}$$

$$\begin{aligned} \text{s.t.}\quad & x_{ij} \le y_j && \forall i\in I, j\in J,\\ & \sum_{j\in J} x_{ij} = 1 && \forall i \in I,\\ & \sum_{j\in J} y_j = p, && \\ & x_{ij} \ge 0, && \forall i\in I, j\in J,\\ & y_j \in \{0,1\} && \forall j \in J, \end{aligned} \tag{F2}$$

$$\begin{aligned} \text{where } S(\mathbf{y},\mathbf{z}) = \{ & b_{ij} \le 1 - z_j && \forall i\in I, j\in J,\\ & b_{ij} \le y_j && \forall i\in I, j\in J,\\ & \sum_{j\in J} b_{ij} + g_i = 1 && \forall i\in I,\\ & b_{ij} \ge 0, && \forall i\in I, j\in J,\\ & g_i \ge 0, && \forall i \in I\} \end{aligned} \tag{F3}$$

The CPMP under disruptions is

$$\min_{\mathbf{x},\mathbf{y}}(1-\varrho)\sum_{i\in I}\sum_{j\in J} d_{ij}h_i x_{ij} + \varrho \max_{\mathbf{z}\in\mathbb{Z}(k)} \min_{(\mathbf{b},\mathbf{g})\in S^C(\mathbf{y},\mathbf{z})} \Big(\sum_{i\in I}\sum_{j\in J} d_{ij}h_i b_{ij} + \sum_{i\in I} p_i h_i g_i\Big) \tag{F4}$$

$$\text{s.t.}\quad (\text{F2}) \text{ and } \sum_{i\in I} h_i x_{ij} \le C_j y_j, \quad \forall j \in J, \tag{F5}$$

$$\text{where}\quad S^C(\mathbf{y},\mathbf{z}) = \{(\text{F3}) \text{ and } \sum_{i\in I} h_i b_{ij} \le C_j y_j, \quad \forall j\in J\}. \tag{F6}$$

F.2 Numerical Results of Adjustable Robust UPMP and CPMP

Since our preliminary experiments as well as those of An et al. (2014) have shown that C&CG-D performs better than BD, we compare only the performance of C&CG-E and C&CG-D. The parameters, the optimality tolerance ($\epsilon = 0.1\%$), and the time limit (2 hours, and we allow the last iteration before reaching the time limit to terminate) take the same values as in An et al. (2014). The results for the two models are given in Tables F2 and F3, where $J = I$ and ϱ is the weight of the worst-case cost.

Table F2 Results for uncapacitated p-median problem under disruptions

				Unit penalty cost = 15						Unit penalty cost = p^{max}					
				C&CG-E			C&CG-D in An et al. (2014)			C&CG-E			C&CG-D in An et al. (2014)		
$\|J\|$	ϱ	p	k	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU
25	0.2	8	1	0.0	1	**0.3**	0.0	4	2.6	0.0	1	**0.3**	0.0	4	2.0
			2	0.0	5	3.7	0.0	5	**2.8**	0.0	5	7.6	0.0	5	**3.2**
			3	0.0	21	**58.8**	0.0	58	578.2	0.0	35	**322.5**	0.0	70	1364.8
		10	1	0.0	1	**0.2**	0.0	1	0.6	0.0	1	**0.4**	0.0	1	0.6
			2	0.1	3	**2.4**	0.0	5	3.8	0.1	3	**2.9**	0.0	5	3.3
			3	0.0	5	**6.4**	0.0	9	10.4	0.0	7	15.0	0.0	11	**12.1**
	0.4	8	1	0.0	3	**1.0**	0.0	4	2.0	0.0	3	3.7	0.0	4	**2.2**
			2	0.0	8	**7.3**	0.0	13	12.7	0.0	8	**9.7**	0.0	13	15.7
			3	0.0	46	**492.5**	0.0	96	1718.8	**0.0**	64	**4159.7**	6.9	92	7359.1
		10	1	0.0	1	**0.3**	0.0	1	0.5	0.0	1	1.0	0.0	1	**0.6**
			2	0.0	4	**3.3**	0.0	6	3.5	0.0	4	16.4	0.0	6	**4.4**
			3	0.0	13	**27.2**	0.0	22	36.3	0.0	14	**55.6**	0.0	25	59.5
49	0.2	8	1	0.0	1	**1.2**	0.0	3	4.9	0.0	1	**3.8**	0.0	3	9.1
			2	0.0	6	**22.4**	0.0	9	29.0	**0.0**	27	**4604.8**	5.0	44	7429.3
			3	**1.6**	38	7618.4	3.6	61	7426.4	**9.0**	33	7544.5	10.2	43	7661.3
		10	1	0.0	1	**0.9**	0.0	1	1.5	0.0	1	**1.4**	0.0	1	2.2
			2	0.0	6	**36.3**	0.0	11	71.9	0.0	6	**58.9**	0.0	11	110.0
			3	0.0	27	**4280.2**	0.0	40	7241.1	**7.1**	28	7223.8	12.1	52	7272.8
	0.4	8	1	0.0	2	**4.1**	0.0	3	4.8	0.0	2	**5.5**	0.0	3	11.2
			2	0.0	17	**747.0**	0.0	29	1334.7	**8.5**	24	7661.8	17.5	46	7458.9
			3	**11.0**	32	7227.2	15.6	44	7821.3	**21.5**	27	7507.7	23.1	33	7834.5
		10	1	0.0	2	**3.8**	0.0	3	5.0	0.0	2	**4.8**	0.0	3	5.1
			2	0.0	13	**683.0**	0.0	30	1468.7	0.0	13	**788.0**	0.0	30	3069.0
			3	11.7	24	7355.0	**10.4**	39	7512.4	**15.9**	22	8057.3	18.4	44	7753.2
		Average		**1.0**$^{(3)}$	11.7	1190.9	1.2$^{(3)}$	20.7	1470.6	**2.6**$^{(5)}$	13.8	2002.4	3.9$^{(7)}$	22.9	2393.5

$^{(-)}$: indicates the number of instances (out of 24) that are not solved to optimality.

Tables F2 and F3 show that for both the UPMP and CPMP, C&CG-E has better performance, in terms of the average gap and the number of iterations. In particular, for the UPMP, when the unit penalty cost is p^{max}, two more instances can be solved to optimality by C&CG-E with less than 4700 seconds; however, C&CG-D produces solutions with optimality gaps over 5.0% when reaching the time limit. For the UPMP, the average CPU time of C&CG-E is also lower. For the CPMP, the average CPU time of the two algorithms is quite close, while C&CG-E provides better optimality gaps. From both tables, we observe that C&CG-E generally works better for instances with a large budget. Take $|J| = 25, \varrho = 0.4, p = 8, k = 3$ and unit penalty cost = 15 for example, for

Table F3 Results for capacitated p-median problem under disruptions

				Unit penalty cost = 15						Unit penalty cost = p^{max}					
				C&CG-E			C&CG-D in An et al. (2014)			C&CG-E			C&CG-D in An et al. (2014)		
$\|J\|$	ϱ	p	k	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU	Gap	#Iter	CPU
25	0.2	8	1	0.0	2	**1.5**	0.0	3	3.4	0.0	2	3.5	0.0	3	**3.0**
			2	0.0	5	**7.2**	0.0	7	9.4	0.0	10	**51.5**	0.0	20	104.1
			3	0.0	25	**236.5**	0.0	41	354.4	**15.4**	43	7521.6	19.8	72	7296.4
		10	1	0.0	1	**0.4**	0.0	1	0.5	0.0	1	**0.3**	0.0	1	0.5
			2	0.0	2	**2.2**	0.0	3	4.0	0.0	3	5.4	0.0	4	**4.1**
			3	0.0	7	**17.9**	0.0	12	23.2	0.0	13	111.4	0.0	19	**82.9**
	0.4	8	1	0.0	3	**3.2**	0.0	4	4.3	0.0	3	**3.4**	0.0	4	3.9
			2	0.0	8	**19.2**	0.0	15	39.8	0.0	12	**99.0**	0.0	23	168.0
			3	0.0	37	**1051.7**	0.0	74	2451.2	**28.4**	43	7352.8	36.8	63	7272.3
		10	1	0.0	1	0.8	0.0	1	**0.6**	0.0	1	**0.4**	0.0	1	0.6
			2	0.0	4	**5.5**	0.0	10	13.8	0.0	5	**9.8**	0.0	10	17.7
			3	0.0	12	**51.9**	0.0	21	63.7	0.0	21	**398.2**	0.0	35	460.6
49	0.2	8	1	0.0	1	**1.5**	0.0	3	7.5	0.0	2	**6.2**	0.0	3	9.3
			2	0.0	10	**310.5**	0.0	15	414.4	**5.8**	22	7819.2	12.4	37	7370.9
			3	**9.0**	25	7884.2	9.8	41	7676.8	24.4	16	8390.7	**24.3**	25	7396.5
		10	1	0.0	1	**1.9**	0.0	2	5.1	0.0	1	**2.1**	0.0	2	4.9
			2	0.0	8	**213.0**	0.0	21	840.2	0.0	11	**1121.3**	0.0	21	1307.6
			3	**2.2**	23	7258.5	3.0	35	8083.0	14.6	17	7292.3	**14.0**	27	9613.7
	0.4	8	1	0.0	2	8.0	0.0	3	**6.2**	0.0	2	**7.0**	0.0	3	9.6
			2	**0.7**	26	7815.3	9.5	35	7858.8	**18.3**	18	8751.8	27.0	29	7295.0
			3	**20.5**	18	8668.8	26.3	30	7347.6	**44.1**	15	7852.8	46.4	23	7379.5
		10	1	0.0	3	17.2	0.0	4	**11.5**	0.0	3	16.5	0.0	4	**14.1**
			2	**4.6**	21	8289.5	4.7	34	7257.8	10.9	19	7993.0	**6.8**	36	8071.1
			3	16.2	22	8742.2	**14.6**	31	7343.0	**28.7**	17	8477.9	29.9	27	7410.8
Average				**2.2**$^{(6)}$	11.1	2108.7	2.8$^{(6)}$	18.6	2075.8	**7.9**$^{(9)}$	12.5	3053.7	9.1$^{(9)}$	20.5	2970.7

the UPMP, C&CG-E consumes 492.5 seconds while C&CG-D takes 1718.8 seconds. Similarly, for the CPMP, the computing time is 1051.7 seconds versus 2451.2 seconds.