Online Supplement

Arden Baxter[1,2], Pinar Keskinocak[1,2], Mohit Singh[1]
[1]H. Milton Stewart School of Industrial and Systems Engineering and [2] Center for Health and Humanitarian Systems, Georgia Institute of Technology, Atlanta, GA, 30332, abaxter@gatech.edu

---

Appendix A: Alternate Formulation for $mRmD$

Table 1 Notation for IPM-Alt

D	Set of demands	
R	Set of resource types	
S^r	Set of starting locations for $r \in R$	
l^r_s	Number of resources of type $r \in R$ starting at location $s \in S^r$	
$\mathcal{K}^r$	Number of resources of type $r \in R$	
f_{ij}	Travel time between $i, j \in \{\cup_{r\in R} S^r\} \cup D$	
M_d	Subset of resources required by $d \in D$	
τ_d	Service start time for $d \in D$	
Δ_d	Service duration for $d \in D$	
w_d	Reward for meeting $d \in D$	
T	Last time period for demand to be served	
$\boldsymbol{y_d}=$	$\begin{cases} 1, & \text{if demand } d \text{ is satisfied} \\ 0, & \text{otherwise} \end{cases}$	$d \in D$
$\boldsymbol{x^{k,r}_{it}}=$	$\begin{cases} 1, & \text{if resource } k \text{ of type } r \text{ is at location } i \text{ at time } t \\ 0, & \text{otherwise} \end{cases}$	$i,t,k,r : i \in \{\cup_{r\in R} S^r\} \cup D, t \in \{0,\dots,T\}, k \in \{1,\dots,\mathcal{K}^r\}, r \in R$
$\boldsymbol{z^{k,r}_{d}}=$	$\begin{cases} 1, & \text{if demand } d \text{ is met by resource } k \text{ of type } r \\ 0, & \text{otherwise} \end{cases}$	$d,k,r : k \in \{1,\dots,\mathcal{K}^r\}, r \in M_d, d \in D$

We present an alternative integer programming model (IPM-Alt) for $mRmD$. For an overview of the notation and description of the decision variables, please refer to Table 1.

$$\max \sum_{d\in D} y_d w_d \tag{18}$$

s.t.

$$\sum_{k=1}^{\mathcal{K}^r} x_{s0}^{k,r} = l_s^r \qquad s\in S^r, r\in R \tag{19}$$

$$\sum_{s\in S^r} x_{s0}^{k,r} = 1 \qquad k\in\{1,\ldots,\mathcal{K}^r\}, r\in R \tag{20}$$

$$\sum_{i\in\{\cup_{r\in R} S^r\}\cup D} x_{it}^{k,r} \le 1 \qquad t\in\{0,\ldots,T\}, k\in\{1,\ldots,\mathcal{K}^r\}, r\in R \tag{21}$$

$$x_{it}^{k,r} + x_{jt'}^{k,r} \le 1 \qquad t'>t|f_{ij}>t'-t, i,j\in\{\cup_{r\in R} S^r\}\cup D, \; t\in\{0,\ldots,T-1\}, k\in\{1,\ldots,\mathcal{K}^r\}, r\in R \tag{22}$$

$$x_{dt}^{k,r} \ge z_d^{k,r} \qquad t\in\{\tau_d,\ldots,\tau_d+\Delta_d\}, k\in\{1,\ldots,\mathcal{K}^r\}, r\in M_d, d\in D \tag{23}$$

$$\sum_{k=1}^{\mathcal{K}^r} z_d^{k,r} \ge y_d \qquad r\in M_d, d\in D \tag{24}$$

$$y_d\in\{0,1\} \qquad d\in D \tag{25}$$

$$x_{it}^{k,r}\in\{0,1\} \qquad i\in\{\cup_{r\in R} S^r\}\cup D, t\in\{0,\ldots,T\}, k\in\{1,\ldots,\mathcal{K}^r\}, r\in R \tag{26}$$

$$z_d^{k,r}\in\{0,1\} \qquad k\in\{1,\ldots,\mathcal{K}^r\}, r\in M_d, d\in D \tag{27}$$

The objective function (18) of IPM-Alt is to maximize the overall reward obtained from satisfying demands. Constraints (19) ensure that the number of resources at starting location s at time 0 is equal to the number of resources available at that location. Constraints (20) assign each resource to a starting location. Constraints (21) ensure that a resource can only be at at most one location at each time index. Constraints (22) maintain that the movement of each resource from location to location is valid for the given time indices. Constraints (23) enforce that a demand is met by a given resource only if that resource is at the demand's location for its entire service duration. Constraints (24) state that a demand can only be satisfied if it has been served by a resource of each of its required resource types. Domain constraints for the variables are given in (25)-(27).

We test a small instance of $mRmD$ with 10 demands, 2 resource types (with 3 resources of each type), and demand start times randomly chosen in the interval [0,100]. All other computational details remain the same as described in Section 8. Table 2 presents the solution times for 10 runs of this instance using both IPM and IPM-Alt. As shown, even with this very small instance, IPM outperforms IPM-Alt significantly. When attempting to use IPM-Alt to solve the smallest sized instance type described in Section 8 (e.g., 2 resource types, 100 demands, 10 resources), construction of the model did not finish in under an hour due to a large number of variables and constraints.

Table 2 **Comparison of run times (in seconds)**

Instance	1	2	3	4	5	6	7	8	9	10
IPM	0.044	0.04	0.083	0.027	0.045	0.032	0.044	0.036	0.045	0.051
IPM-Alt	4.47	4.15	6.33	3.57	3.58	5.82	7.22	4.20	6.19	3.19

Appendix B: Proofs of Theorems and Observations

B.1. Proof of Theorem 1

We prove that $2RmD$ is NP-hard by reduction using Numerical 3-Dimensional Matching (N3DM). The construction is similar to the proof in Keskinocak and Tayur (1998).

Instance of N3DM

Integers t, d and a_t, b_t, c_t for $i = 1, \dots, t$, satisfying the following relations:

$$\sum_{i=1}^{t}(a_i + b_i + c_i) = td \text{ and } 0 < a_i, b_i, c_i < d \text{ for } i = 1, \dots, t$$

The goal is to find permutations ρ and σ of $\{1, \dots, t\}$, such that:

$$a_i + b_{\rho(i)} + c_{\sigma(i)} = d \text{ for } i = 1, \dots, t$$

N3DM is shown to be NP-Complete in Garey and Johnson (1990). Given an instance of N3DM, we define the following:

$A_i = i$ for $i = 1, \dots, t$

$B_j = t + j$ for $j = 1, \dots, t$

$C_{ij} = 2t + (i-1)t + j$ for $i, j = 1, \dots, t$

$S = t^2 + 2t$

$T = S + 2d + 1$

An instance of $2RmD$ with $t^2 + t$ resources and travel times between demands equal to zero can be built as follows: resources $1, \dots, t$ are of type 1 and the remaining t^2 resources are of type 0. Let demands be of the form (u_j, v_j), where u_j is the start time, v_j is the end time, and $v_j - u_j$ is the service time of demand j. Define the reward for demand j to be the length of service times the number of resource types required.

The following demands require resource type 0 and 1:

$$(0, A_i) \text{ for } i = 1, \dots, t$$

$$(A_i, C_{ij}) \text{ for } i, j = 1, \dots, t$$

$$(S + d - c_k, T) \text{ for } k = 1, \dots, t$$

The following demands require resource type 0:

$$t - 1 \text{ times } (0, B_j) \text{ for } j = 1, \dots, t$$

$$(B_j, C_{ij}) \text{ for } i, j = 1, \dots, t$$

$$(C_{ij}, S + a_i + b_j) \text{ for } i, j = 1, \dots, t$$

$$(S + a_i + b_j, T - 1) \text{ for } i, j = 1, \dots, t$$

$$t^2 - t \text{ times } (T - 1, T)$$

The following demands require resource type 1:

$$(C_{ij}, S + a_i + b_j) \text{ for } i, j = 1, \dots, t$$

We now show that there exists a feasible solution to N3DM if and only if all resources are serving demands during the interval [0,T] and there is no idle time.

Suppose that there exists a feasible schedule for a subset of demands, such that all resources are busy serving demands during the interval [0,T]. Demands $(0, A_i)$ must be scheduled to the first t resources (since there are t of these trips and they each require resource type 1) and resources $t+1, \dots, 2t$ of type 0. These demands must be followed by some (A_i, C_{ij}) demands. Similarly, the $t^2 - t$ demands $(0, B_j)$ must be scheduled to the remaining $t^2 - t$ resources of type 0, followed by some (B_j, C_{ij}) demands.

The first t resources of type 1 and the next $t+1, \dots, 2t$ resources of type 0 have the following schedules:

$$(0, A_i)(A_i, C_{ij})(C_{ij}, S + a_i + b_j)(S + d - c_k, T)$$

where each i, $1 \le i \le t$ occurs exactly once.

The remaining $t^2 - t$ resources of type 0 have schedules of the form

$$(0, B_j)(B_j, C_{ij})(C_{ij}, S + a_i + b_j)(S + a_i + b_j, T - 1)(T - 1, T)$$

where each j, $1 \le j \le t$ occurs exactly $t-1$ times. Thus, among the demands $(C_{ij}, S+a_i+b_j)$ that are served by the first t resources, each i and each j occurs exactly once.

From the schedules of the first t resources, we have $S+a_i+b_j = S+d-c_k$, i.e., $a_i+b_j+c_k = d$. So we define $\rho(i) = j$ and $\sigma(i) = k$ whenever demand $(C_{ij}, S+a_i+b_j)$ is served followed by $(S+d-c_k, T)$ by resource $i, i = 1, \ldots, t$.

Conversely, given a feasible solution to N3DM, a feasible solution for this instance of $2RmD$ can be found, which is clearly optimal, since the resources are serving demand throughout the interval [0,T] with no idle time. $\square$

B.2. Proof of Lemma 1

We first show that IPM-R can be modeled and solved as a MCF with node capacities. Define $G = (N, A, l, \mu, c, b, v)$ as follows, where l, μ are the lower and upper bound functions on arc capacity, c is the arc cost function, b is the node supplies function, and v is the node capacities function:

Nodes

$N := D^r \cup S^r \cup \{s^*, t\}$

Arcs:

(s^*, t) at cost 0 with capacity $(0, \infty)$

(s^*, s) at cost 0 with capacity $(0, l_s^r)$ $\forall s \in S^r$

(d, t) at cost 0 with capacity $(0, 1)$ $\forall d \in D^r$

(s, d) if $B_{sd} = 1$ at cost 0 with capacity $(0, 1)$ $\forall s \in S^r, d \in D^r$

(i, j) if $A_{ij} = 1$ at cost 0 with capacity $(0, 1)$ $\forall i, j \in D^r$

Supplies (b):

$b(i) = 0$ $\forall i \in S^r \cup D^r$

$-b(t) = b(s^*) = \sum_{s \in S^r} l_s^r$

Node Capacities (v):

$v(i) = y_i$ if $i \in D^r$

To convert this MCF with node capacities into a general MCF, we can perform the following transformation (Ciupală (2009)). We redefine G to be a MCF $G' = (N', A', l', \mu', c', b')$.

Nodes:

$N' = N_1 \cup N_2$ where $N_1 := \{i' | i \in N\}$ and $N_2 := \{i'' | i \in D^r\}$.

Arcs:

$A' = A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5$ where

$A_1 := \{(i', j') | (i, j) \in A, i, j \notin D^r\}$

$A_2 \coloneqq \{(j', i') | (j, i) \in A, j \notin D^r, i \in D^r\}$

$A_3 \coloneqq \{(i'', j') | (i, j) \in A, i, j \in D^r\}$

$A_4 \coloneqq \{(i'', j') | (i, j) \in A, i \in D^r, j \notin D^r\}$

$A_5 \coloneqq \{(i', i'') | i \in D^r\}$

Note that for any arc $(i, j) \in A_1 \cup A_2 \cup A_3 \cup A_4$, the cost and capacities remain the same as previously defined for the corresponding arcs in A. The new arcs $(i', i'') \in A_5$ have cost 0 and capacity $(v(i), v(i))$.

Supplies (b'):

$b'(i') = b(i)\ \forall i' \in N_1$

$b'(i'') = 0\ \forall i'' \in N_2$

Then we have the following MCF formulation:

$$\min \quad 0$$

s.t.

$$\sum_{j:(i,j)\in A'} x_{ij}^r - \sum_{j:(j,i)\in A'} x_{ji}^r = b'(i) \qquad i \in N'$$

$$l'(i,j) \le x_{ij}^r \le \mu'(i,j) \qquad (i,j) \in A'$$

Thus, we have shown that IPM-R can be modeled and solved as a MCF. $\square$

B.3. Proof of Theorem 3

We prove that $mRmD$ cannot be approximated within $O(\min\{|R|^{\frac{1}{2}-\epsilon}, |D|^{1-\epsilon}\})$, where $|R|$ is the number of resource types and $|D|$ is the number of demands, by reduction using Maximum Set Packing.

Instance of Maximum Set Packing

Given a universe U, a family S of subsets of U, and an integer n, is there a subfamily $C \subseteq S$ of sets such that all sets in C are pairwise disjoint and $|C| \ge n$?

Maximum Set Packing is well known to be NP-Complete and cannot be approximated within $O(N^{1-\epsilon})$ unless $NP \subseteq ZPP$ (Håstad (1996)). Given an instance of Maximum Set Packing, we can create an instance of $mRmD$ with $|U|$ resources and $|S|$ demands in polynomial time as follows:

Define r_u as a resource of type $u\ \forall u \in U$.

Define d_s as a demand that requires resource type i if $i \in s\ \forall s \in S$.

Let all demands and resources be at the same location (i.e., zero travel times) and all demand intervals be equivalent. Further, set all demand rewards equal to 1.

We now show that there exists a feasible solution to Maximum Set Packing if and only if there exists a feasible solution to our problem with an objective value greater than or equal to n.

Suppose that we have a feasible solution to Maximum Set Packing. That is, $\exists C \subseteq S$ such that $|C| \geq n$ and all sets in C are pairwise disjoint. Then, for each $x \in C$, d_x can be served by our resources since they are disjoint in the resources that they require. Therefore, we can serve at least n demands and since all rewards are equal to 1, our objective must be greater than or equal to n.

Conversely, suppose we have a feasible solution to our problem such that the objective is greater than or equal to n. Since all rewards are equal to 1, this implies that we have met at least n demands. Further, since we only have one resource of each type, these n demands must have disjoint resource requirements. For each demand d_x that was served, the corresponding set of resource requirements $x \in S$ can be added to C. Therefore, we have created a subfamily $C \subseteq S$ of disjoint sets where $|C| \geq n$ and so Maximum Set Packing is feasible.

Since the transformation of Maximum Set Packing to an instance of *mRmD* preserved the objective value, then approximation results are also preserved. That is, *mRmD* cannot be approximated within $O(\min\{|R|^{\frac{1}{2}-\epsilon}, |D|^{1-\epsilon}\})$ unless $NP \subseteq ZPP$ (for every $\epsilon > 0$), where $|R|$ is the number of resource types and $|D|$ is the number of demands. □

B.4. Proof of Theorem 4

We prove that the special case of $mR\{1 \text{ or } 2\}D$ where there are an arbitrary number of resource types R, a special resource $r' \in R$, and each demand requires either r', r or $\{r', r\}, r \in R - r'$ is NP-hard by reduction using N3DM. The construction is similar to the proof in Keskinocak and Tayur (1998). Given an instance of N3DM, we define the following:

$A_i = i$ for $i = 1, \ldots, t$

$B_j = t + j$ for $j = 1, \ldots, t$

$C_{ij} = 2t + (i-1)t + j$ for $i, j = 1, \ldots, t$

$S = t^2 + 2t$

$T = S + 2d + t + 1$

An instance of the special case of $mR\{1 \text{ or } 2\}D$ with $t^2 + t$ resources and travel times between demands equal to zero can be built as follows: resources $1, \ldots, t$ are of type $i, i = 1, \ldots, t$ and the remaining t^2 resources are of type t'. Let demands be of the form (u_j, v_j), where u_j is the start time, v_j is the end time, and $v_j - u_j$ is the service time of demand j. Define the reward for demand j to be the length of service times the number of resource types required.

The following demands require resource type t' and i:

$$(0, A_i) \text{ for } i = 1, \ldots, t$$

$$(A_i, C_{ij}) \text{ for } i, j = 1, \ldots, t$$

$$(S + d - c_k, T - k - 1) \text{ for } i, k = 1, \ldots, t$$

The following demands require resource type t':

$$t - 1 \text{ times } (0, B_j) \text{ for } j = 1, \ldots, t$$

$$(B_j, C_{ij}) \text{ for } i, j = 1, \ldots, t$$

$$(C_{ij}, S + a_i + b_j) \text{ for } i, j = 1, \ldots, t$$

$$(S + a_i + b_j, T - 1) \text{ for } i, j = 1, \ldots, t$$

$$t^2 - t \text{ times } (T - 1, T)$$

$$(T - k - 1, T) \text{ for } k = 1, \ldots, t$$

The following demands require resource type i:

$$(C_{ij}, S + a_i + b_j) \text{ for } i, j = 1, \ldots, t$$

$$(T - k - 1, T) \text{ for } i, k = 1, \ldots, t$$

We now show that there exists a feasible solution to N3DM if and only if all resources are serving demands during the interval [0,T] and there is no idle time.

Suppose that there exists a feasible schedule for a subset of demands, such that all resources are busy serving demands during the interval [0,T]. Demands $(0, A_i)$ must be scheduled to the first t resources (since there are t of these trips and they each require resource type $i, i = 1, \ldots t$) and resources $t + 1, \ldots, 2t$ of type t'. These demands must be followed by some (A_i, C_{ij}) demands. Similarly, the $t^2 - t$ demands $(0, B_j)$ must be scheduled to the remaining $t^2 - t$ resources of type t', followed by some (B_j, C_{ij}) demands.

The first t resources of type $i, i = 1, \ldots, t$ and the next $t + 1, \ldots, 2t$ resources of type t' have the following schedules:

$$(0, A_i)(A_i, C_{ij})(C_{ij}, S + a_i + b_j)(S + d - c_k, T - k - 1)(T - k - 1, T)$$

where each i and each k, $1 \le i, k \le t$ occur exactly once.

The remaining $t^2 - t$ resources of type t' have schedules of the form

$$(0, B_j)(B_j, C_{ij})(C_{ij}, S + a_i + b_j)(S + a_i + b_j, T - 1)(T - 1, T)$$

where each j, $1 \le j \le t$ occurs exactly $t-1$ times. Thus, among the demands $(C_{ij}, S+a_i+b_j)$ that are served by the first t resources, each i, j, and k occurs exactly once.

From the schedules of the first t resources, we have $S+a_i+b_j = S+d-c_k$, i.e., $a_i+b_j+c_k = d$. So we define $\rho(i) = j$ and $\sigma(i) = k$ whenever demand $(C_{ij}, S+a_i+b_j)$ is served followed by $(S+d-c_k, T-k-1)$ by resource $i, i = 1, \dots, t$.

Conversely, given a feasible solution to N3DM, a feasible solution for this instance of the special case of $mR\{1 \text{ or } 2\}D$ can be found, which is clearly optimal, since the resources are serving demand throughout the interval [0,T] with no idle time. □

B.5. Proof of Theorem 5

For an instance $\mathcal{I}$ of the special case of $mR\{1 \text{ or all}\}D$, let m_i be the number of resources of type i for $i = 1, \dots, |R|$. Since there is zero travel time between demands (i.e., all demands are at the same location), all demands have the same service start time, and all service times are infinite, then each resource can only be assigned a single job.

First, divide the list of demands D into subsets $D_0, D_1, \dots, D_{|R|}$ where subset D_0 is the demands that require all resource types and subsets D_i for $i = 1, \dots, |R|$ are the demands that only require resource type i. Order the demands in each subset by decreasing reward. We can find the optimal objective value $OPT(\mathcal{I})$ and its corresponding solution by Algorithm $\mathcal{E}$, which runs polynomially in number of resources. Note the solution is constructed by serving all demands whose reward was added to the objective value $OPT(\mathcal{I})$. □

B.6. Proof of Theorem 7

We prove that $1R1D$ is solvable in polynomial time. Consider m resources and n demands. Without loss of generality, assume that each resource has its own starting location and are labeled by resource number (i.e., $s = 1, \dots, m$). This problem can be modeled as a MCF on a directed acyclic graph with the following nodes and arcs:

Nodes:

For each resource there is a node b_i, $i = 1, \dots m$.

For each demand there are two nodes u_j and v_j, $j = 1, \dots n$.

There is a dummy source node s and a dummy sink node t.

Arcs:

(s, b_i) and (b_i, t) at cost 0, $i = 1, \dots, m$.

(v_j, t) at cost 0, $j = 1, \dots n$.

(b_i, u_j) if $B_{ij} = 1$ at cost 0, $i = 1, \dots, m, j = 1, \dots, n$.

Algorithm: $\mathcal{E}$

Input: $\mathcal{I}$, m_i, $i = 1, 2, \ldots, |R|$

$m \leftarrow \min\{m_i | i \in \{1, 2, \ldots, |R|\}\}$
$Opt(\mathcal{I}) \leftarrow 0$
for $1 \le i \le |R|$ **do**
  if $m < m_i$ **then**
    Increase $OPT(\mathcal{I})$ by rewards of first $m_i - m$ entries of D_i.
    Remove first $m_i - m$ entries from D_i.
  end
end
for $1 \le i \le m$ **do**
  Let $r(D_i[0]) =$ reward of first entry in D_i for $i = 0, 1, \ldots, |R|$.
  if $\sum_{i=1}^{|R|} r(D_i[0]) \le r(D_0[0])$ **then**
    Increase $OPT(\mathcal{I})$ by $r(D_0[0])$.
    Remove $D_0[0]$ from D_0.
  end
  else
    Increase $OPT(\mathcal{I})$ by $r(D_i[0])$ for $i = 1, 2, \ldots, |R|$.
    Remove $D_i[0]$ from D_i for $i = 1, 2, \ldots, |R|$.
  end
end

Output: $OPT(\mathcal{I})$

(v_j, u_k) if $A_{jk} = 1$ at cost 0, $j, k = 1, \ldots, n$.

(u_j, v_j) at cost $-w_j$, $j = 1, \ldots, n$.

All arcs have capacity 1. Let the supply at node s be m, supply at node t be $-m$ and supply for all other nodes be zero. Further, note that since the graph is directed and acyclic, we can transform the arc costs to be non-negative. Because of the capacities of the arcs, each arc will have either no flow or flow of 1 unit in the optimal solution. It is clear that there is a one-to-one correspondence between the MCF from s to t on this constructed graph and the optimum allocation for a single type of resource. If there is a positive flow on the arc (b_i, t), this means that resource i was not used to meet demand. If there is a positive flow on the arc (b_i, u_j), this means that demand j is the first demand met by resource i. Positive flow on the arc (v_j, u_k) means that demand k is met immediately after demand j by the same resource. Finally, positive flow on the arc (u_j, v_j) means that demand j was met. □

B.7. Observation 1

Note that if travel times are zero, then $1R1D$ can be formulated as a maximum weighted coloring problem on an interval graph. Consider m resources and n demands. We can now create an (undirected) interval graph $G = (N, A)$ as follows:

Nodes:

For each demand, create a node $d_i, i = 1, \ldots, n$.

Arcs:

(d_i, d_j) if $\tau_i + \Delta_i > \tau_j \quad \forall i, j \in \{1, 2, \ldots, n\}$

Then, we can solve the maximum weighted m-coloring problem on interval graph G using the following integer program:

$$\max \sum_{i=1}^{n} w_i d_i$$

s.t.

$$\sum_{i \in S} d_i \leq m \qquad \forall S \text{ s.t. } S \text{ is a maximal clique of size} \geq m$$

$$d_i \in \{0, 1\} \qquad \forall i = 1, \ldots, n$$

Since the nodes versus cliques matrix of an interval graph is totally unimodular (Mertzios (2008)), then we can relax the above formulation to a linear program and still maintain integral optimal solutions.

Appendix C: Computational Study

C.1. Bicriteria Results

Figures 1 and 2 present bicriteria results to compare the trade-off between increasing the number of resources and its impact on the objective function value (i.e., demands met). These results could provide insights for decision-makers when determining resource capacity and desired levels of demand satisfaction. In Figure 1, we examine the $mRmD$ problem with 3 resource types, 400

Figure 1 **Bicriteria results for $mRmD$**

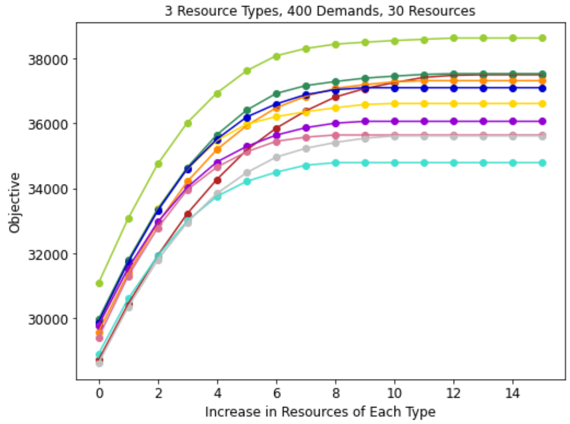
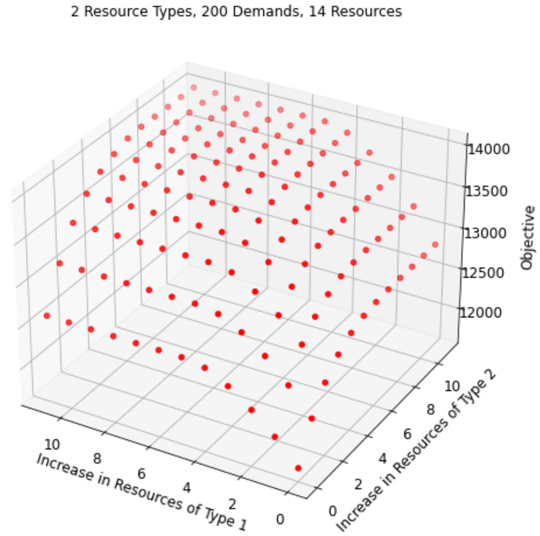


Figure 2 **Bicriteria results for $2RmD$**



demands, and 30 resources. Each line in the graph represents an instance of this problem. Starting at 10 resources of each type, we increase this amount by 0 to 15 and record the objective function value. As shown, there is a diminishing return as resources are increased. On average, after increasing the resources by about 6 or 7, we reach the maximum objective function value for each instance studied. Figure 2 looks at the $2RmD$ problem with 200 demands and 14 resources. In this case, we increase the number of resources of each type separately by 0 to 11 and present the resulting objective function value. Again, we see a diminishing return as resources are increased. It is worth noting that jointly increasing resources (i.e., increasing both resource types) is more profitable than only increasing one resource type. For example, increasing resource type 1 and type 2 by 3 and 4 resources, respectively, produced an objective of 13606 whereas only increasing resource type 1 by 8 led to an objective of 12844.

C.2. Travel Costs

Let $BFlow$ and $CFlow$ represent solutions in which the flow variables (x) are restricted to be binary and those in which the flow variables are relaxed to be continuous, respectively. Tables 3 and 4 present a comparison of the run times (in seconds) for $BFlow$ and $CFlow$ under all instances considered, where $|R|$ is the number of resource types, $|D|$ is the number of demands, and L is the total number of resources. Note that *Scaled Demands* refers to instances where reward for demand was multiplied by a factor of 100.

Computational experiments show that for small and medium sized problems (less than 5 resource types), IPM-C can be efficiently solved by Gurobi and the optimal solution can be found in reasonable time. As the size of the problem increases (i.e., greater demands or larger number of resource types), the instances become harder to solve. For 2-4 resource types, on average, $BFlow$ performs better than $CFlow$ when demands are not scaled, whereas when demands are scaled, CFlow performs better than BFlow. For 5-7 resource types, on average, $CFlow$ performs better than $BFlow$ in both scaled and unscaled demand scenarios. However, for some instances, $BFlow$ is drastically better than $CFlow$ (i.e., 7 resource types and 700/800 demands for unscaled/scaled demands, respectively). In general instances with scaled demands perform better than those in which demands are not scaled; this could be a result of less importance given to travel costs in the objective.

Table 3 Results for number of resource types equal to 2,3 and 4

$\lvert R\rvert$	$\lvert D\rvert$	L	Run time (BFlow)	Run time (CFlow)	Run time (BFlow) (Scaled Demands)	Run time (CFlow) (Scaled Demands)
2	100	10	0.11	0.09	0.16	0.10
2	200	14	0.28	0.25	0.33	0.31
2	300	18	0.71	0.72	1.10	0.69
2	400	22	1.54	2.01	2.24	1.70
2	500	26	3.47	3.96	3.84	3.47
2	600	30	5.49	6.90	6.79	6.89
2	700	34	8.35	13.77	8.05	9.80
2	800	38	12.69	19.09	15.16	14.44
3	100	12	0.09	0.09	0.15	0.10
3	200	18	0.49	0.39	0.57	0.59
3	300	24	1.01	1.15	1.48	1.10
3	400	30	3.81	3.63	3.78	3.42
3	500	36	7.29	8.23	6.72	5.90
3	600	42	9.98	13.51	13.54	13.69
3	700	48	19.40	30.17	20.65	21.99
3	800	54	43.85	65.33	25.16	61.34
4	100	16	0.28	0.14	0.24	0.13
4	200	24	0.97	0.82	1.08	0.55
4	300	32	2.63	2.19	2.29	1.56
4	400	40	6.22	5.41	5.31	4.67
4	500	48	19.49	15.28	15.39	14.26
4	600	56	26.87	30.03	23.84	22.85
4	700	64	64.84	96.78	40.41	38.23
4	800	72	89.05	125.97	52.18	77.69

Table 4 Results for number of resource types equal to 5,6 and 7

$\lvert R\rvert$	$\lvert D\rvert$	L	Run time (BFlow)	Run time (CFlow)	Run time (BFlow) (Scaled Demands)	Run time (CFlow) (Scaled Demands)
5	100	20	0.19	0.19	0.13	0.18
5	200	30	2.00	1.01	1.10	0.80
5	300	40	6.36	3.68	3.39	2.74
5	400	50	14.79	9.20	9.04	7.76
5	500	60	35.01	32.58	21.28	17.83
5	600	70	50.66	70.35	51.90	73.11
5	700	80	96.25	172.39	47.01	53.80
5	800	90	453.05	1014.62	134.52	202.69
6	100	24	0.44	0.22	0.29	0.26
6	200	36	2.49	1.25	1.23	1.40
6	300	48	7.08	5.43	5.44	4.44
6	400	60	18.03	15.78	11.36	10.40
6	500	72	53.19	55.05	30.33	33.52
6	600	84	134.02	107.60	137.94	145.15
6	700	96	195.10	237.03	114.26	181.08
6	800	108	537.75	1039.68	232.16	285.46
7	100	28	0.33	0.25	0.29	0.20
7	200	42	2.06	1.63	1.85	1.27
7	300	56	13.76	8.58	9.92	5.92
7	400	70	125.48	55.14	23.67	17.23
7	500	84	161.38	112.70	67.47	51.95
7	600	98	421.19	389.39	133.33	167.05
7	700	112	989.80	1629.80	306.81	279.56
7	800	126	1485.61	1398.86	359.35	534.93

Appendix D: Extension 2

We present an extension to $mRmD$. In this extension, resource types have both an origin and destination location, i.e., resources may not go directly from demand incident to demand incident. For an overview of the additional notation needed for this formulation, please refer to Table 5.

In order to construct feasible schedules for resources, we create 0-1 matrices for each resource type $r \in R$, A^r, and a 0-1 matrix, B. Matrix A^r has a row and column for each demand incident and $A^r_{ij} = 1$ if and only if demand incident j can be served after demand incident i by resource type r and demands i and j both require resource type r. Matrix B has a row for each resource starting location and a column for each demand incident and $B_{sd} = 1$ if and only if a resource starting at location s can serve demand d first. That is, we have the following pre-processing matrices:

$$A^r_{ij} = \begin{cases} 1 \text{ iff } \quad \tau_i + \Delta_i + f(a^r_i, b^r_i) + f(b^r_i, a^r_j) \le \tau_j, \quad r \in M_i \cap M_j \\ 0 \text{ otherwise} \end{cases} \qquad i, j \in D, r \in R$$

$$B_{sd} = \begin{cases} 1 \text{ iff } \quad f(s, a^r_d) \le \tau_d, s \in S^r, r \in M_d \\ 0 \text{ otherwise} \end{cases} \qquad s \in S^r, r \in R, d \in D$$

The formulation is the same as IPM, where variable construction relies on matrices A^r and B.

Table 5 Additional notation for Extension 2

V	Set of nodes
R	Set of resource types
$f(u,v)$	Travel time between $u, v \in V$
(a^r_d, b^r_d)	Origin-destination pair for resource type $r \in M_d$ needed by $d \in D$

References

Ciupală L (2009) About flow problems in networks with node capacities. *W. Trans. on Comp.* 8(8), ISSN 1109-2750.

Garey MR, Johnson DS (1990) *Computers and Intractability; A Guide to the Theory of NP-Completeness* (USA: W. H. Freeman & Co.).

Håstad J (1996) Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 627–636.

Keskinocak P, Tayur S (1998) Scheduling of time-shared jet aircraft. *Transportation Science* 32(3):277–294.

Mertzios GB (2008) A matrix characterization of interval and proper interval graphs. *Applied Mathematics Letters* 21(4):332 – 337.