Supplemental Material for Article "Submodular Dispatching with Multiple Vehicles"

Appendix A: Proofs for Formulations (2) and (3)

A.1. Formulation 2

Proof: We prove the equivalence between feasible solutions for MILP (1) and (2). Assume we have a solution (x^1, t^1, z^1) for MILP (1); we construct a solution (x, t, z, y, w) for MILP (2). Set $x,\ y$ and w as zero vectors, then:

- For all $S \subseteq N, k \in M$ such that $x^1_{S,k} = 1$, let $i = \max\{j : j \in S\}$ and set variables $x_S = 1$, $y_{i,k} = 1$ and $w_{i,k} = f(S)$.
- For all $i \in N, k \in M$ set $t_{i,k} = t^1_{i,k}$ and $z = z^1$

As x^1 forms a partition, then x satisfies constraint (2g). As it is a partition, for each $i \in N$ there is at most one set $S \in \mathcal{N}_i$ such that $x_S = 1$; therefore (2f) $\sum_{k=1}^m y_i \leq 1$ holds. From construction, equality (2d) is met, and by monotonicity inequality (2e) holds. Finally, from the definition of w and feasibility of t^1, our new solution (x, t, z, y, w) satisfies constraints (2a), (2b) and (2c). From construction, the makespan is equal to the makespan of solution (x^1, t^1, z^1) for MILP (1). Now consider a feasible solution (x, t, z, y, w) for MILP (2); we construct a solution (x^1, t^1, z^1) for MILP (1):

- For all $i \in N, k \in M$ set $t^1_{i,k} = t_{i,k}$, then set $z^1 = z$ and x^1 as a zero-valued vector.
- Find all subsets $S \subseteq N$ such that $x_S = 1$. For each of those sets S, compute its maximum index order. Because of non-negativity of f and (2e), there is exactly one machine k such that $y_{i,k}$ is equal to one and $w_{i,k} = f(S)$. Set $x^1_{S,k} = 1$.

By definition of w, inequalities (2a), (2b) and (2c) correspond to inequalities (1a), (1b) and (1c) in MILP (1). As x is a partition, then x^1 is as well and (x^1, t^1, z^1) is feasible and has the same makespan as solution (x, t, z, y, w) for MILP (2).

Now we focus on the linear relaxation of this formulation with non-negative (instead of binary) domains for x and y. We consider the dual of the linear relaxation of (2) and let α be the dual variable of (2a), β be the dual variable of (2b) and (2c), ε be the dual variable of (2d), ϕ be the dual variable of (2e), π be the dual variable of (2f) and finally γ be the dual variable of (2g). The dual constraints corresponding to the (relaxed) x variables are $-\varepsilon_i f(S) + \sum_{j \in S} \gamma_j \leq 0, i \in N, S \in \mathcal{N}_i$, and therefore the separation problem is $\min_{S \in \mathcal{N}_i} \left\{ \varepsilon_i f(S) - \sum_{j \in S} \gamma_j \right\}$. Constraint (2d) can be replaced with a greater-than-or-equal constraint without loss of optimality, and thus we may assume $\varepsilon \geq 0$. Therefore, the separation problem corresponds to submodular minimization, and the linear relaxation of (2) is solvable in polynomial time. ■

A.2. Equivalence Between the LP Relaxations of Formulations 1 and 2 for MSMD

Proof: Assume we have a feasible solution (x^1, t^1, z^1) for the LP relaxation of MILP (1); we construct a feasible solution (x, t, z, w, y) for the LP relaxation of MILP (2):

- Let $x_S = \sum_{k \in M} x^1_{S,k}$ for each $S \subseteq N$; which implies $x_S \geq 0$ from feasibility of x^1, but also that $x_S \leq 1$.
- Set $z = z^1$, and $t_{i,k} = t^1_{i,k}$ for all $i \in N,\ k \in M$.
- Set $w_{i,k} = \sum_{S \in \mathcal{N}_i} x^1_{S,k} f(S)$ for all $i \in N,\ k \in M$; from the feasibility of x^1 and f, then $w_{i,k} \geq 0$.
- Define $y_{i,k}$ as the minimum number such that $w_{i,k} \leq y_{i,k} f(\{1, \ldots, i\})$; this implies $0 \leq y_{i,k}$, as $w_{i,k} \geq 0$, but also that $y_{i,k} \leq 1$ as $w_{i,k} = \sum_{S \in \mathcal{N}_i} x^1_{S,k} f(S) \leq 1 \times f(\{1, \ldots, i\})$, where the last inequality follows from no order being dispatched more than once (over all the fractional dispatches including that order).

We show that constraints (2a)-(2g) are satisfied by (x, t, z, w, y). As (1a) holds for x^1 then (2a) holds. Moreover, $\sum_{k \in M} w_{i,k} = \sum_{k \in M} \sum_{S \in \mathcal{N}_i} x^1_{S,k} f(S) = \sum_{S \in \mathcal{N}_i} f(S) \sum_{k \in M} x^1_{S,k} = \sum_{S \in \mathcal{N}_i} f(S) x_S$, and so (2d) is satisfied. From the definitions of $w_{i,k}$ and $t_{i,k}$ plus the feasibility of x^1 and t^1, constraints (2b) and (2c) hold as well. Moreover, by construction (2e) holds and from the choice of $y_{i,k}$ we have that $\frac{\sum_{k \in M} w_{i,k}}{f(\{1, \ldots, i\})} = \sum_{k \in M} y_{i,k}$, and as $\sum_{k \in M} W_{i,k} \leq f(\{1, \ldots, i\})$, then (2f) is

satisfied. Finally, (2g) follows from the choice of x. For every feasible solution (x^1,t^1,z^1) for the LP relaxation of (1), we can thus create a feasible solution (x,t,z,w,y) for the LP relaxation of MILP (2) and with the same makespan.

Assume now that we have a feasible solution (x,t,z,w,y) for the LP relaxation of MILP (2); we create a feasible solution (x^1,t^1,z^1) for the LP relaxation of MILP (2) by setting $z^1=z$, vector x^1 as zero, and $t^1_{i,k}=t_{i,k}$ for all $i\in N,\ k\in M$. Then for values $x^1_{i,k}$ do as follows: for each $i\in N$, we initialize an auxiliary vector V_i that has the same values as $w_{i,k}$ for each k, i.e. $V_{i,k}=w_{i,k}$. Then, for each S such that $x_S>0$, we find its maximum index $i=\max\{j: j\in S\}$ and do as follows: find the first index k such that $V_{i,k}>0$; if $V_{i,k}>x_Sf(S)$, then set $x^1_{S,k}=x_S$, subtract $x_Sf(S)$ from $V_{i,k}$ and set x_S to zero. On the other hand, when $V_{i,k}\le x_Sf(S)$, we set $x^1_{S,k}=\frac{V_{i,k}}{f(S)}\le x_S$, set $V_{i,k}=0$ and reduce x_S by $x^1_{S,k}$, continuing until $x_S=0$. Because of feasibility of (x,t,z,w,y) (constraint (2d)) the final result of this procedure is a vector V_i that is zero-valued for all $i\in N$.

We prove that (x^1,t^1,z^1) is feasible for the LP relaxation of MILP (1). From the definition of t^1 (1a) holds, and by construction of the vector x^1 so do constraints (1b) and (1c) (auxiliary vectors V_i are zero at the end of the procedure above). Furthermore, for all $S\subseteq N$ we have $\sum_{k\in M}x^1_{S,k}=x_S$, and (1d) holds. Every feasible solution (x,t,z,w,y) for the LP relaxation of MILP (2) can thus be mapped to a feasible solution (x^1,t^1,z^1) for the LP relaxation of MILP (1).

To conclude the proof, if we are given an extreme point solution for one of the LP relaxations, then that solution has a polynomially bounded number of variables x with non-zero value, and hence the number of operations needed to go from the solution of one LP relaxation to the solution for the other is polynomially bounded too. ■

A.3. Formulation 3

Proof: We prove that MILP (3) solves MSMD by showing that each feasible solution for (3) is feasible for MSMD, and that each optimal solution for MSMD is feasible for MILP (3). Consider a feasible solution (x,y,t) for MILP (3); there must be m indices $i\in N$ such that $y_{0,i}=1$. We denote those indices as $i^1_1,\dots,i^1_m$, and will map index i^1_k to vehicle k for $k\in M$. We construct a solution $\{v_k\}_{k=1}^m$ for MSMD:

- Assign z as the makespan for our solution in MSMD, then for each vehicle $k\in M$, we find the set of indices $i^1_k,i^2_k,\dots,i^\ell_k$ such that $y_{i^1_k,i^2_k}=\dots=y_{i^\ell_k,n+1}=1$; the set exists because of constraint (3f).
- For each vehicle $k\in M$, initialize vector v_k as $v_k=(0,\varnothing,0,\varnothing,\dots,0,\varnothing)$, and set $i=1$. If i is one of the values $i^1_k,\dots,i^\ell_k$, then because of constraint (3e) there must be a subset $S_i\in\mathcal{N}_i$ with $x_{S_i}=1$; assign $v_k[2i]=t_i$ and $v_k[2i+1]=S_i$; otherwise assign $v_k[2i]=\max\big(r_i,v_k[2(i-1)]\big)$. Increase i by one and repeat until $i=n$.

Because x is a partition (3d), our solution for MSMD is as well. Moreover, because of construction and feasibility of vector t (constraints (3a) and (3g)), our vectors v_k also have a feasible schedule for the vehicles. As the makespan z is feasible given this schedule, we just constructed a solution for MSMD, with equal makespan. Now, let $\{v_k\}_{k=1}^m$ be an optimal solution for MSMD; we construct a solution for MILP (3) by first initializing vectors x,y,t as zero; then:

- For each $k\in M$, find all the indices $i_1,i_2,\dots,i_\ell$ such that $v_k[2i_1+1],v_k[2i_2+1],\dots,v_k[2i_\ell+1]$ are non-empty sets; we denote those sets as $S_{i_1},S_{i_2},\dots,S_{i_\ell}$. Set $y_{0,i_1}=y_{i_1,i_2}=\dots=y_{i_\ell,n+1}=1$ and $x_{i_1}=x_{i_2}=\dots=x_{i_\ell}=1$. Furthermore, for $j=i_1,\dots,i_\ell$, set $t_j=v_k[2j]$.
- Let J be the set of indices that have not been modified; i.e. $j\in J$ if t_j was not modified. For $j\in J$ set $t_j=\max\{\max_{i<j}(t_i),r_j\}$, with $t_0=0$. Finally, set t_{n+1} as the minimum value that meets constraints (3g).

We show that (x,t,y) is feasible: from construction and feasibility of the MSMD solution we know that constraint (3a) holds. By construction of values y, we know that constraints (3b), (3c), (3e) and (3f) hold. As $\{v_k\}_{k=1}^m$ induces a partition, then constraint (3d) is also satisfied. Finally, with respect to constraint (3g) we have two cases. Consider j in J, then for all $i<j$ we have $y_{i,j}=0$, hence $t_j \geq t_i + \sum_{S\in\mathcal{N}_i} x_S f(S) - f([1,i])$. As $f([1,i]) \geq \sum_{S\in\mathcal{N}_i} x_S f(S)$, then our choice of t_j satisfies (3g). On the other hand, every order $j\notin J$ must be associated to a dispatch done by a vehicle k:

- Consider all indices $i\in J$, with $i<j$; if $t_i = r_i$ then constraint (3g) holds; otherwise $t_i = t_k$ for some $k<i$, and because no batch in $\mathcal{N}_i$ is dispatched, then the inequality that involves indices j and k is stronger.
- Consider all indices $i<j$ that are associated to the same vehicle k; because of feasibility of the original solution for MSMD, then constraint (3g) holds for all pairs (i,j).
- Consider all indices $i_1^h, i_2^h, \ldots, i_\ell^h$ associated to vehicle $h\neq k$ and batch $S_1^h, S_2^h, \ldots$, such that $i_1^h < \ldots < i_\ell^h < j$. Because of optimality of the MSMD solution, there exists a largest index, say i_a^h such that $t_{i_a^h} = r_{i_a^h}$; and for every index $b: a\leq b\leq \ell$ we have that $t_{i_b^h} + f(S_b^h) = r_{i_a^h} + \sum_{p=a}^b f(S_p^h) \leq r_{i_b^h} + f(\cup_{p=a}^b S_p^h)$, otherwise the MSMD solution is not optimal. Moreover, for all $b\leq\ell$, we have $f([1,i_b^h]) \geq f(\cup_{p=a}^b S_p^h)$, thus for $b: a\leq b\leq \ell$,

$$\begin{aligned} t_j \geq r_j \geq r_{i_b^h} &\geq r_{i_b^h} + f(\cup_{v=a}^b S_v^h) - f([1,i_b^h]) \\ &= r_{i_b^h} + f(\cup_{v=a}^b S_v^h) - (1-y_{i_b^h,j}) f([1,i_b^h]) \\ &\geq t_{i_b^h} + f(S_b^h) - (1-y_{i_b^h,j}) f([1,i_b^h]). \end{aligned}$$

Therefore, constraint (3g) holds for indices $i_a^h, \ldots i_\ell^h$. Because of the choice of a, $r_{i_a^h} \geq t_{i_c^h} + f(S_c^h)$ for $c<a$, so the inequality holds for indices $i_1^h, \ldots, i_{a-1}^h$. The three cases include all values i such that $i<j$, hence (3g) holds for all pairs (i,j), with $j\in N$. In the case of $j=n+1$ (sink node), then the values $y_{i,n+1}=1$ enforce the correct computation of the makespan, and constraints with $y_{i,n+1}=0$ are redundant. It follows that solution (x,t,z) for (3) is feasible. As we already showed that every feasible solution for (3) is feasible for MSMD, and that every optimal solution for MSMD is feasible for (3); then formulation (3) solves MSMD. With respect to the linear relaxation of (3), the separation problem is very similar to that for (1) and (2), and can again be solved as a series of submodular minimization problems. ■

Appendix B: Proofs for the Set Cover Formulation

B.1. Complexity of Separation Problem

Proof: For each $k\in M$ the separation problem is $\min_{S\subseteq N}\{\alpha_k \,\mathrm{SMD}(S) - \beta - \sum_{i\in S}\gamma_i\}$. As $\alpha\geq 0$, we just need to optimize the separation problem for $k^* = \operatorname{argmin}_{k\in M}\{\alpha_k\}$. As f is modular, we can reformulate the separation problem as the following maximization problem, where $z = \mathrm{SMD}(S)$:

$$\max_{t,x,z} \; -z\alpha_{k^*} + \sum_{i\in N}\gamma_i x_i \tag{6a}$$

$$\text{s.t. } t_i \geq r_i \qquad i\in N \tag{6b}$$

$$t_{i+1} \geq t_i + x_i\tau_i \qquad i\in N\setminus n \tag{6c}$$

$$z \geq t_n + x_n\tau_n \tag{6d}$$

$$x_i\in\{0,1\} \qquad \forall i\in N \tag{6e}$$

$$t,z\geq 0 \tag{6f}$$

Constraints (6b), (6c) and (6d) are the feasibility constraints for SMD, and binary variable x_i represents the choice of adding order i to the subset S in the separation problem (5).

To complete the proof we reduce the knapsack problem to formulation (6). Consider the following knapsack problem with integer numbers $v_i, w_i > 0$ for all $i = 1,2\ldots,n-1$ (representing value and weight of item i, respectively) and integer knapsack capacity $W \geq w_i$; for all $i = 1,\ldots,n-1$:

$$\max_{x} \sum_{i=1}^{n-1} v_i x_i \tag{7a}$$

$$\text{s.t.} \sum_{i=1}^{n-1} v_i x_i \leq W \tag{7b}$$

$$x_i \in \{0,1\} \qquad \forall i = 1,\ldots,n-1 \tag{7c}$$

We reduce this knapsack problem to MILP (6) by using the following transformation: we set $r_i = 0$ for $i = 1,\ldots,n-1$ and $r_n = W$. Moreover, we define $\tau_i = w_i$ for $i = 1,\ldots,n-1$ and $\tau_n = 1$. Then we set $\gamma_i = v_i$ for $i = 1,\ldots,n-1$ and $\gamma_n = 1+\sum_{i=1}^{n-1} v_i$ and finally $\alpha_{k^*} = \sum_{i=1}^{n-1} v_i$. The corresponding instance for separation problem (6) is:

$$\max_{t,x,z} -z\left(\sum_{i=1}^{n-1} v_i\right) + \sum_{i=1}^{n-1} v_i x_i + \left(1+\sum_{i=1}^{n-1} v_i\right) x_n \tag{8a}$$

$$\text{s.t. } t_n \geq W \tag{8b}$$

$$t_{i+1} \geq t_i + x_i w_i \qquad \forall i = 1,\ldots n-1 \tag{8c}$$

$$z \geq t_n + x_n \tag{8d}$$

$$x_i \in \{0,1\} \qquad \forall i \in N \tag{8e}$$

$$t, z \geq 0 \tag{8f}$$

As $r_i = 0$ for $i = 1,\ldots,n-1$, by repeatedly using constraints (8c) and starting from $t_1 = 0$, we can set $t_2 = 0 + x_1 w_1$, then $t_3 = t_2 + x_2 w_2 = x_1 w_1 + x_2 w_2$ and continue until $t_{n-1} = \sum_{i=1}^{n-2} x_i w_i$. Then (8) is equivalent to:

$$\max_{t_n,x,z} -z\left(\sum_{i=1}^{n-1} v_i\right) + \sum_{i=1}^{n-1} v_i x_i + \left(\sum_{i=1}^{n-1} v_i\right) x_n + x_n \tag{9a}$$

$$\text{s.t. } t_n \geq W \tag{9b}$$

$$t_n \geq \sum_{i=1}^{n-1} x_i w_i \tag{9c}$$

$$z \geq t_n + x_n \tag{9d}$$

$$x_i \in \{0,1\} \qquad \forall i \in N \tag{9e}$$

$$t_n, z \geq 0 \tag{9f}$$

We prove the equivalence between solutions for (9) and solutions for the original knapsack problem. Assume we have an optimal solution for (9), it is clear that $x_n = 1$ because adding order n increases the makespan z by just 1 (constraint (9d)) and therefore increases the total objective value by 1. It follows that the optimal solution is completely described by the orders maximizing $-z\left(\sum_{i=1}^{n-1} v_i\right) + \sum_{i=1}^{n-1} v_i x_i$. An optimal solution is such that $\sum_{i=1}^{n-1} x_i w_i \leq W$, otherwise by integrality of values w_i and W we would have $\sum_{i=1}^{n-1} x_i w_i \geq W+1$, and therefore $z \geq W+2$, implying that $-z\left(\sum_{i=1}^{n-1} v_i\right) + \sum_{i=1}^{n-1} v_i x_i \leq -(W+2)\left(\sum_{i=1}^{n-1} v_i\right) + \sum_{i=1}^{n-1} v_i x_i \leq -(W+1)\left(\sum_{i=1}^{n-1} v_i\right)$, the objective when we do

not dispatch to any of the $n-1$ first orders, a contradiction with optimality. Therefore, the optimal solution of (9) is such that $z = W+1$, and maximizes $\sum_{i=1}^{n-1} v_i x_i$ subject to binary variables x_i for $i = 1,\dots,n-1$ and $\sum_{i=1}^{n-1} x_i w_i \leq W$; it is precisely a feasible optimal solution for the knapsack problem. Furthermore, (9) is a feasible problem (by setting all $x_i = 0$, $t_i = 0$ for $i = 1,\dots,n$, and $t_n = z = W$) and its objective is bounded by above (by $2(\sum_{i=1}^{n-1} v_i)+1$), thus (9) is guaranteed to have an optimal solution. The knapsack problem is NP-Hard, and we did a transformation with a linear number of steps to get MILP (9); a special case of the separation problem (5) that considers a modular function f. ■

B.2. Complexity of Separation Problem for Modular Function with Setup Time

Proof: We start by remembering that the separation problem (5) is: $\min_{S\subseteq N}\{\alpha_k \,\mathrm{SMD}(S) - \beta - \sum_{i\in S}\gamma_i\}$, and we assume all values τ_i are integer. The separation problem can be solved by computing the shortest path between a source and a sink node. We describe the directed acyclic network with source node T_0 and sink node T_1 as follows:

- States (Ψ,ℓ,h); for $\Psi = 0,\dots,U$, $\ell \in N$ and $h \in \{\ell+1,\dots,n\}\cup\{0\}$. Ψ denotes the partial makespan after the end of stage ℓ, stage ℓ decides if order ℓ will be dispatched or not; and h represents the order with largest index that will be a part of the current batch; if $h = 0$, then there is no current batch.
- First we describe the arcs leaving node T_0:

  — Arcs $T_0 \to (\Psi,1,0)$ with distance 0 for $\Psi \leq \tau_1 + \tau_0 - 1$ (i.e. order 1 is not dispatched to); and with distance $-\gamma_1$ for $\Psi \geq \tau_1 + \tau_0$ (order 1 is dispatched as a singleton; this is our feasibility condition).

  — Arcs $T_0 \to (\Psi,1,h)$ with distance $-\gamma_1$ for $\Psi \geq r_h + \tau_1 + \tau_0$ (i.e. order 1 is in a batch that has as largest order h; this is the feasibility condition); for all $1 < h \leq n$.
- Now we describe the arcs leaving node $(\Psi,\ell,0)$ for $\ell = 1,\dots,n-1$ and $\Psi \leq U$:

  — Arcs $(\Psi,\ell,0) \to (\Psi,\ell+1,0)$ with distance 0 (order $\ell+1$ will not be dispatched).

  — Arcs $(\Psi,\ell,0) \to (\Psi+\tau_{\ell+1}+\tau_0,\ell+1,0)$ with distance $-\gamma_1$ ($\ell+1$ dispatched as a singleton); if $\Psi \geq r_{\ell+1}$.

  — Arcs $(\Psi,\ell,0) \to (\Psi+\tau_{\ell+1}+\tau_0,\ell+1,h)$ with distance $-\gamma_1$ (order $\ell+1$ will be dispatched in a batch that has h as its largest index order); if $\Psi \geq r_h$ (feasibility condition), and $\ell+1 < h$.
- Arcs leaving nodes (Ψ,ℓ,h) for $\ell = 1,\dots,n-1$, $\ell < h$ and $\Psi \leq U$:

  — If $\ell+1 = h$, then need to end the dispatch; i.e. arc $(\Psi,\ell,\ell+1) \to (\Psi+\tau_{\ell+1},\ell+1,0)$ with distance $-\gamma_{\ell+1}$.

  — Arc $(\Psi,\ell,h) \to (\Psi,\ell+1,h)$ with distance 0; not adding order $\ell+1$ to the current batch.

  — Arc $(\Psi,\ell,h) \to (\Psi+\tau_{\ell+1},\ell+1,h)$ with distance $-\gamma_{\ell+1}$; adding order $\ell+1$ to the current batch.
- Arcs $(\Psi,n,0) \to T_1$ with distance $\Psi\alpha_k - \beta$; for all $\Psi \geq \min_{i\in N}(r_i+\tau_i)+\tau_0$; which guarantees the optimal solution includes at least one order being dispatched.

The total number of arcs departing states $(\Psi,\ell,0)$ is $O(Un^2)$; and the total number of arcs departing states (Ψ,ℓ,h) is $O(Un^2)$; hence the total number of arcs in this directed acyclic graph is $O(Un^2)$ and solving the separation problem has complexity $O(Un^2)$. When $\tau_0 = 0$ we do not need the h in the states and have just $O(Un)$ arcs. ■

B.3. Complexity of Separation Problem for Family Setups

Proof: The proof in Appendix B.2 corresponds to the case where $Q = 1$; we generalize the shortest path problem needed to solve the separation problem in (5). Assume that vectors $\tau, \sigma,\ r$ are integer, and that values U_q are known for $q = 1,\dots,Q$. We use the notation $F_{\{i\}}$ to denote the family that contains $i \in N$. The directed acylic graph is as follows:

- States $(\Psi, \ell, \psi_1, h_1, \psi_2, h_2, \ldots, \psi_Q, h_Q)$; for all $\Psi \leq U$, $\ell \in N$, $0 \leq \psi_q \leq U_q$ and $\ell < h_q$ with $h_q \in F_q$ or $h_q = 0$ for all $q \leq Q$. Ψ represents the partial makespan, ℓ is the decision stage (whether to dispatch to order $\ell+1$ or not). For all $q \leq Q$, if $h_q > 0$, then h_q corresponds to the largest index of family q that is included in the current batch of that family and ψ_q is the accumulated dispatching time of that batch; otherwise, if $h_q = 0$ then there is no current dispatch for family q, with $\psi_q = 0$.

- We now proceed to describe the arcs for this acyclic directed graph. Between stage ℓ and $\ell+1$, the only coordinates of the states vector that can change are Ψ and the coordinates of the family of order $\ell+1$; i.e. $F_{\{\ell+1\}}$; so our notation for states will be $(\Psi, \ell, \ldots, \psi_{F_{\{\ell+1\}}}, h_{F_{\{\ell+1\}}}, \ldots)$. We start with the arcs departing T_0:

  — Arcs $T_0 \to (\Psi, 1, \ldots, 0, 0, \ldots)$ with distance 0; for all $\Psi < \tau_1 + \sigma_{F_{\{1\}}}$ (order 1 is not dispatched).

  — Arcs $T_0 \to (\Psi, 1, \ldots, 0, 0, \ldots)$ with distance $-\gamma_1$; for all $\tau_1 + \sigma_{F_{\{1\}}} \leq \Psi \leq U$ (order 1 dispatched as singleton).

  — Arcs $T_0 \to (\Psi, 1, \ldots, \tau_1 + \sigma_{F_{\{1\}}}, h_{F_{\{1\}}}, \ldots)$ with distance $-\gamma_1$; for all $\Psi \leq U$, and $1 < h_{F_{\{1\}}}$ with $h_{F_{\{1\}}} \in F_{\{1\}}$ (i.e., order 1 is dispatched in a batch that has as maximum index $h_{F_{\{1\}}}$, feasibility conditions are enforced later).

- For each $\ell = 1, \ldots, n-1$, and $\Psi \leq U$; when $h_{F_{\{\ell+1\}}} = 0$ then $\psi_{F_{\{\ell+1\}}} = 0$ and the arcs are:

  — $(\Psi, \ell, \ldots, 0, 0, \ldots) \to (\Psi, \ell+1, \ldots, 0, 0, \ldots)$ with distance 0 (order $\ell+1$ is not dispatched).

  — $(\Psi, \ell, \ldots, 0, 0, \ldots) \to (\Psi + \tau_{\ell+1} + \sigma_{F_{\{\ell+1\}}}, \ell+1, \ldots, 0, 0, \ldots)$ with distance $-\gamma_{\ell+1}$ (order $\ell+1$ is dispatched as a singleton), if $\Psi \geq r_{\ell+1}$ (feasibility condition).

  — $(\Psi, \ell, \ldots, 0, 0, \ldots) \to (\Psi, \ell+1, \ldots, \tau_{\ell+1} + \sigma_{F_{\{\ell+1\}}}, h, \ldots)$ with distance $-\gamma_{\ell+1}$ for $h > \ell+1$ and $h \in F_{\{\ell+1\}}$ (order $\ell+1$ is dispatched in a batch where the largest order is h, feasibility condition is enforced later).

- For each $\ell = 1, \ldots, n-1$, and $\Psi \leq U$; when $h_{F_{\{\ell+1\}}} = \ell+1$ then the only possible arc (that represents the dispatch being finalized) is $(\Psi, \ell, \ldots, \psi_{F_{\{\ell+1\}}}, h_{F_{\{\ell+1\}}}, \ldots) \to (\Psi + \psi_{F_{\{\ell+1\}}} + \tau_{\ell+1}, \ell+1, \ldots, 0, 0, \ldots)$ and exists only if $\Psi \geq r_{\ell+1}$ (feasibility condition for the batch with largest index $\ell+1$).

- For each $\ell = 1, \ldots, n-1$, and $\Psi \leq U$, when $h_{F_{\{\ell+1\}}} > \ell+1$ then arcs are:

  — $(\Psi, \ell, \ldots, \psi_{F_{\{\ell+1\}}}, h_{F_{\{\ell+1\}}}, \ldots) \to (\Psi, \ell+1, \ldots, \psi_{F_{\{\ell+1\}}}, h_{F_{\{\ell+1\}}}, \ldots)$ with distance 0 ($\ell+1$ not dispatched).

  — $(\Psi, \ell, \ldots, \psi_{F_{\{\ell+1\}}}, h_{F_{\{\ell+1\}}}, \ldots) \to (\Psi, \ell, \ldots, \psi_{F_{\{\ell+1\}}} + \tau_{\ell+1}, h_{F_{\{\ell+1\}}}, \ldots)$ with distance $-\gamma_{\ell+1}$ (order $\ell+1$ is added to the current batch of family $F_{\{\ell+1\}}$; feasibility conditions are enforced at $\ell+1 = h_{F_{\{\ell+1\}}}$).

- Finally, we have the arcs $(\Psi, n, \ldots, 0, 0, \ldots) \to T_1$ with distance $\Psi\alpha_k - \beta$; for $\Psi \geq \min_{i \in N}(r_i + \tau_i + \sigma_{F_{\{i\}}})$; condition that guarantees any shortest path dispatches to at least one order.

For each $\ell < n$, if $h_{F_{\{\ell+1\}}} \neq 0$, then state $(\Psi, \ell, \ldots, \psi_{F_{\{\ell+1\}}}, h_{F_{\{\ell+1\}}}, \ldots)$ has either one or two departing arcs; so there are $O\left(U\left[\Pi_{q=1}^{Q} U_q \times |F_q|\right]\right)$ arcs departing from these states. On the other hand, if $h_{F_{\{\ell+1\}}} = 0$, then there are up to $|F_{\{\ell+1\}}|+2$ departing arcs; however as $\psi_{F_{\{\ell+1\}}} = 0$ and $|F_{\{\ell+1\}}| < U_{F_{\{\ell+1\}}}$, then the total number of arcs departing from these states is also $O\left(U\left[\Pi_{q=1}^{Q} U_q \times |F_q|\right]\right)$. By adding all arcs over $\ell \in N$, there are $O\left(nU\left[\Pi_{q=1}^{Q} U_q \times |F_q|\right]\right)$ arcs. ■

Appendix C: Interval-Solvable Functions

Proof: We prove the claim for each of the functions separately:

(1) As shown by Erazo and Toriello (2024), we may assume without loss of optimality that the τ_i are in non-increasing order. Consider a solution for the MSMD and order all batches according to their *minimum index*; let $B_1, \ldots, B_h$ be the ordered batches. We set $i = 1$ and construct iteratively a new solution as follows: Let l_i and u_i be the minimum and maximum indices (respectively) of B_i; we construct new interval batch $B_i' = \{l_i, l_i+1, \ldots, u_i\}$ and

assign it to the same vehicle to which B_i is assigned to; then update all the batches $B_{i+1},\dots,B_h$ by removing the new orders that were just assigned to B'_i, and increase i by one. If B_i is empty, then we add B'_i as empty and increase i by one; we finish this procedure when $i = h+1$. By construction, the minimum index of B'_i is greater than or equal to the minimum index of B_i for all i where B'_i is not empty, so $f(B'_i) \le f(B_i)$ because of τ being monotone non-increasing. Furthermore, the maximum index of B'_i is smaller or equal to the maximum index of B_i for all $i = 1,\dots,h$ where B'_i is not empty, so each of those respective B'_i dispatches do not start later than when batch B_i starts its dispatch in the original optimal solution. Each vehicle has its makespan reduced or maintained and the interval solution is optimal.

(2) From Proposition 4, without loss of optimality we assume that each vehicle dispatches to batches according to an increasing value of the *maximum index* of the batch. From this observation, we create an interval solution as follows: order all the batches according to their maximum index and let $B_1,\dots,B_h$ be the ordered batches, with respective cardinalities $C_1,\dots,C_h$. We can construct the interval solution by doing $B'_1 = \{1,\dots,C_1\}$, and $B'_{i+1} = \{\sum_{j=1}^{i} C_j + 1,\dots,\sum_{j=1}^{i+1} C_j\}$ for $i = 1,\dots,h-1$, such as to get batches $B'_1,\dots,B'_h$, where by construction $|B'_i| = |B_i|$ and $\max_{j\in B'_i}\{j\} \le \max_{j\in B_i}\{j\}$. By assigning each batch B'_i to the same vehicle that batch B_i was assigned to, the new solution cannot increase its makespan; thus, the interval solution is optimal as well.

The function $f(S) = \sum_{i\in S}\tau_i$ being singleton-solvable comes from the serial-batching machine scheduling literature. ■

Appendix D: Strength of LP relaxation Lower Bounds

D.1. Worst Case for the Bound Given by the LP Relaxations of our Formulations

Proof: Consider any arbitrary number of vehicles m and positive integer h; we design an instance I_h with $n = m+2$ orders, $r_i = 0$ for all $i \in N$; $\tau_1 = 1$ and $\tau_i = 1/h$ for all orders $i = 2,\dots,m+2$. For $h \ge n-1$, the makespan of instance $z^*_{I_h}$ is equal to 1. We prove the claim for each of the LP relaxations:

- For the LP relaxation of formulation (1), set $x_{N,k} = 1/m$ for every vehicle $k \in M$; this ensures constraint (1d) holds. As all release times are zero, we set $t_1 = t_2 = \ldots = t_n = 0$ and constraints (1a), (1b) and (1c) hold. Finally, $z^{LP(1)}_{I_h} = \big((n-1)/h+1\big)/m = (m+1)/hm + 1/m$; thus $\lim_{h\to\infty} z^*_{I_h}/z^{LP(1)}_{I_h} = m$.
- For the LP relaxation of formulation (4), set $x_{\{1\},k} = x_{\{2\},k} = \ldots = x_{\{m-1\},k} = x_{[m-1,m+1],k} = 1/m$ for every vehicle $k \in M$. This choice guarantees that constraints (4b) and (4c) hold. Each constraint (4a) becomes $z \ge 1/m + (m-2)/hm + 3/hm = 1/m + (m+1)/hm$; thus $z^{LP(4)}_{I_h} = 1/m + (m+1)/hm$; and $\lim_{h\to\infty} z^*_{I_h}/z^{LP(1)}_{I_h} = m$.
- For the LP relaxation of formulation (3) set $x_i = 1$ for all $i \in N$, then (3d) holds. Moreover, set $y_{0,1} = 1$, $y_{1,2} = 2/3$, $y_{1,3} = 1/3$ (i.e. flow constraints at order 1 hold), $y_{0,2} = 1/3$, $y_{2,3} = 2/3$, $y_{2,n+1} = 1/3$ (i.e. flow constraints at order 2 hold); $y_{3,4} = 1/3$, $y_{3,n+1} = 2/3$ (i.e. flow constraints at order 3 hold), $y_{0,4} = 2/3$, $y_{4,n+1} = 1$ (i.e. flow constraints at order 4 hold) and finally, $y_{0,i} = y_{i,n+1} = 1$ for $i \in 5,\dots,n$. It follows that constraints (3e) and (3f) hold; and as $n = m+2$; then (3b) and (3c) hold as well. Now with respect to constraints (3g), starting with orders 1, 2, 3 and 4:

  —$t_1 = 0$; then $t_2 \ge t_1 + 1 - (1-2/3)\times 1 = 2/3$; thus we set $t_2 = 2/3$

  —$t_3 \ge t_1 + 1 - (1-1/3)\times 1 = 1/3$ but also $t_3 \ge t_2 + 1/h - (1-2/3)\times(1+1/h) = 2/3 + 1/h - (1/3)(1+1/h) = 1/3 + 2/(3h)$; therefore we set $t_3 = 1/3 + 2/(3h)$

  —$t_4 \ge t_1 + 1 - 1 = 0$; $t_4 \ge t_2 + 1/h - (1+1/h) = -1/3$ and $t_4 \ge t_3 + 1/h - (1-1/3)(1+2/h) = 1/3 + 2/(3h) - 2/3 - 4/(3h) = -1/3 - 2/(3h)$; so we set $t_4 = 0$

For t_i with $5 \le i \le n$ because $f([1,i]) > 1$ and the only non-zero incoming variable y is $y_{0,i} = 1$, then $t_i = 0$. For t_{n+1}:

  —$t_{n+1} \ge t_1 + 1 - 1 = 0$

—$t_{n+1} \geq t_2 + 1/h - (1-1/3)(1+2/h) = 2/3 + 1/h - 2/3 - 4/(3h) < 0$

—$t_{n+1} \geq t_3 + 1/h - (1-2/3)(1+3/h) = 1/3 + 1/h - 1/3 - 1/h = 0$

—$t_{n+1} \geq t_4 + 1/h - 0 = 1/h$.

For all $5 \leq i \leq n$ we get $t_{n+1} \geq t_i + 1/h = 1/h$; thus, vector t is feasible for constraints (3a) and (3g). We conclude that $t_{n+1} = z_{I_h}^{LP(4)} = 1/h$; and $\lim_{h\to\infty} z_{I_h}^{*}/z_{I_h}^{LP(1)} = \infty$. By swapping the values of τ_1 and τ_n, then $z_{I_h}^{LP(1)}$ and $z_{I_h}^{LP(4)}$ remain constant, but $z_{I_h}^{LP(3)} = z_{I_h}^{*}$; this suggests that our formulations have a complementary structure for the lower bounds. ■

D.2. The Linear Relaxation of (1) is Dominated

Proof: We start the proof by adding a constraint in the formulation, that implies we get an upper bound on the objective of the LP relaxation. We enforce $x_{S,1} = x_{S,2} = \ldots = x_{S,m}$ for all subsets $S \subseteq N$; that makes the LP relaxation to be equivalent to solving a modified instance of a single vehicle SMD, with function $f'(S) = f(S)/m = (\sum_{i\in S} \tau_i)/m$. Erazo and Toriello (2024) proved that this problem can be solved in linear time with the recursion $t_1 = r_1 = 0$ and $t_{i+1} = \max(t_i + f'(\{i\}), r_{i+1})$ for all $i \in N\backslash\{1\}$. From the recursion, we know that there exists a maximum index $j \geq 1$ such that $t_j = r_j$, and therefore we have:

$$z = t_{n+1} = t_j + \sum_{i=j}^{n} \frac{f'(\{i\})}{m} = r_j + \frac{f(\{j,\ldots,n\})}{m} \leq r_j + \frac{f(\{j,\ldots,n\})}{\min(m, n-j+1)} \leq \max_{i\in N}\left[r_i + \frac{f(\{i,\ldots,n\})}{\min(k, n-i+1)}\right].$$

The equations follow from definition, and the last expression is precisely the lower bound from Proposition 5. ■

D.3. Strong Set Cover Formulation

Proof: To prove that the strong formulation continues to solve the MSMD, we assume that we have an optimal solution z, x for MILP (4). Because that solution is optimal, then each vehicle $k \in M$ will have only one variable $x_{S_k,k}$ equal to one, and constraints (4a) are just equal to $z \geq \text{SMD}(S_k)x_{S_k,k}$ for all $k \in M$. Moreover, LB is a lower bound on the makespan; therefore, $z \geq LB$, which implies $z \geq \max(\text{SMD}(S_k), LB)x_{S_k,k}$ for all $k \in M$, thus z, x is also optimal for the strong set cover formulation. The separation problem of the strong set cover MILP is $\min_{S\subseteq N}\left(\alpha_k \max(LB, SMD(S)) - \beta - \sum_{i\in S}\gamma_i\right)$ for all $k \in M$. The proof for Theorem 2 continues to hold for any knapsack problem with $W \geq LB$; thus, the complexity result holds. For Theorems 3 and 4, the same dynamic programs work; but for the arcs arriving into the terminal state T_1 we modify the distance to be $\max(LB, \Psi)\alpha_k - \beta$ instead of $\Psi\alpha_k - \beta$. ■

Appendix E: Experiment Details

E.1. Column Generation Acceleration

We used the acceleration technique from Ben-Ameur and Neto (2007) when solving the linear relaxations of our four formulations and the strong version of LP (4). During column generation, after solving the Restricted Master Problem (RMP), the reduced costs are retrieved, and they characterize an infeasible solution for the dual problem of the given LP; we denote this infeasible dual solution as δ_m. For each iteration of column generation, this solution is either separated, and a new column enters the RMP, or the solution is proved optimal.

The acceleration technique also uses an incumbent feasible dual solution, denoted δ_f; this solution can be initialized with any feasible solution for the dual LP, such as $\delta_f = 0$. Instead of attempting to separate δ_m, we solve the separation problem for the convex combination $\hat{\delta} = \lambda\delta_f + (1-\lambda)\delta_m$. Intuitively, this solution is likelier to be feasible or at least closer to the dual feasible region. If $\hat{\delta}$ is dual feasible, it necessarily has a larger objective value than δ_f, so it becomes the new feasible incumbent and we repeat the separation for the new $\hat{\delta}$. Otherwise, it is dual infeasible, so we add dual cutting planes (i.e. columns in the primal) and solve the RMP again. This technique helps to speed up column generation by generating fewer columns. After preliminary calibration, we used $\lambda = 0.5$ in our experiments.

E.2. MILP for the Pricing Problem of the Set Cover Formulations

The separation problem for (4) requires us to compute $\min_{S\subseteq N}\left\{\alpha_k \,\mathrm{SMD}(S)-\beta-\sum_{i\in S}\gamma_i\right\}$ for all vehicles $k\in M$. Given that $SMD(S)$ and each α_k are positive, we can just solve for $\alpha=\min_{k=1,\dots,m}\alpha_k$. Similarly, we can define $\gamma_S=\sum_{i\in S}\gamma_i$ for subsets $S\subseteq N$. We define the following variables:

$x_S\in\{0,1\}$: indicates if batch $S\subseteq N$ is dispatched.

$t_i\geq 0$: departure time of the i-th dispatch, if it occurs, for $i\in N$.

$z\geq 0$: makespan.

We solve the separation problem using a prize-collecting plus makespan variant of the single-machine problem:

$$\begin{aligned}
&\min\ \alpha z-\sum_{S\subseteq N}\gamma_S x_S\\
&s.t.\ t_i\geq r_i\sum_{S\in\mathcal{N}_i}x_S && \forall i\in N && \text{(10a)}\\
&\quad t_{i+1}\geq t_i+\sum_{S\in\mathcal{N}_i}x_S f(S) && \forall i=1,\dots,n-1 && \text{(10b)}\\
&\quad z\geq t_n+\sum_{S\in\mathcal{N}_n}x_S f(S) && && \text{(10c)}\\
&\quad \sum_{\substack{S\subseteq N\\ S\ni i}}x_S\leq 1 && \forall i\in N && \text{(10d)}\\
&\quad z\geq\varepsilon && && \text{(10e)}\\
&\quad t\geq 0, t\geq 0, x\in\{0,1\}
\end{aligned}$$

The column to enter the Restricted Master Problem (if the optimal solution has an objective value below β) is obtained by computing all the orders that were dispatched, i.e. the union of subsets $S\subseteq N$ with $x_S=1$ in the optimal solution of the separation problem. Constraints (10a)-(10c) represent the feasibility conditions for the dispatches, and (10d) enforces that each order can be dispatched at most once. By using $\varepsilon=\min_{i\in N}f(\{i\})$, we ensure that the column to be added will not be an empty set; and when using the strong constraints with a lower bound LB, we only need to modify the formulation with $\varepsilon=LB$. This formulation has an exponential number of variables; however, for the problem with family setups we only need to consider subsets that have single families, which significantly reduces the feasible space.

E.3. Lower Bound for Serial-Batch Scheduling with Family Setups and Release Times

PROPOSITION 15. *Consider an instance of MSMD with m vehicles and Q families $F_1,\dots,F_Q$ that partition order set N. Each family q has a setup time $\sigma_q\geq 0$, each order $i\in N$ is associated to a positive number τ_i, and $f(S)=\sum_{i\in S}\tau_i+\sum_{q:F_q\cap S\neq\varnothing}\sigma_q$. For all $i\in N$, define $G_i:=\min(m,n-i+1)$, $L_i:=\left|\{q:F_q\cap[i,n]\neq\varnothing\}\right|$ (number of families intersecting batch $[1,n]$), and $P_i:=\{q:|F_q\cap[i,n]|>1\}$. Furthermore, define V_i as the increasing vector with the values σ_q for each $q\in P_i$, each value repeated exactly $|F_q\cap[i,n]|-1$ times. Finally, let $P_i(a)$ be the sum of the first a components of vector P_i, with $P_i(0)=0$ for all $i\in N$. Then,*

$$\max_{i\in N}\left\{r_i+\left[f([i,n])+P_i\big(\max\{G_i-L_i,0\}\big)\right]/G_i\right\}\qquad \textit{is a lower bound for the MSMD instance.}$$

Proof: For all $i\in N$, at time r_i we still have to dispatch to orders $[i,n]$, so at least $f([i,n])$ units of time need to be divided into at most m vehicles if $m\geq n-i+1$ and $n-i+1$ vehicles otherwise (one order per vehicle). Thus, $r_i+f([i,n])/G_i$ is a lower bound. For any i, if $G_i\leq L_i$, then the expression within the principal maximum function is

exactly that lower bound. On the other hand, if $G_i > L_i$ then the total dispatching load of the lower bound is divided into more vehicles than the total number of families intersecting with $[i,n]$. Therefore each vehicle needs to incur at least one family setup time, we can add the $G_i - L_i$ smallest setup times among families that intersect batch $[i,n]$ to $f(S)$ to get a lower bound on the total dispatch time done by the G_i vehicles. As $f(S)$ considers the setup of each family intersecting the batch, then only the families in P_i are eligible for their setup time to be selected; and if family q intersects the batch k times, then that setup can be added up to $k-1$ extra times. Thus, we add $P_i(G_i - L_i)$ to $f(S)$ to get a lower bound $r_i + \big(f([i,n]) + P_i(G_i - L_i)\big)/G_i$. By maximizing over $i \in N$, we find the desired lower bound. ■

E.4. Implementation Details for SDD Tactical Design

For each number of vehicles m we compute the maximum number of orders that can be dispatched given a vector of arrival times r and a maximum makespan ϕ^* by sequentially solving the minimum makespan problem (1) for an increasing sequence number of orders $n, n+1, n+2, \ldots$ until it becomes infeasible. Because of monotonicity, we can speed up computations by computing the maximum number n_k for $m = k$, then starting at $n_k + 1$ for $m = k+1$. Similarly, we do not need optimality when solving (1) for a given m_k and n_j; instead, we can just end the run once a feasible solution is found, or when the dual bound exceeds the desired maximum makespan (in which case we cannot fulfill n_j orders with m_k vehicles before the maximum deadline).

We modify MILP (1) to find the most efficient dispatches given the number of vehicles m, the vector of arrival times r, the dispatch time function f, the original makespan ϕ^* and the maximum number of orders that can be dispatched in this scenario, n_1. This new optimization problem leverages the fact that f is interval-solvable and uses the same variables as MILP (1):

$$\min \sum_S x_{S,k} f(S)$$

$$s.t.\ t_{i,k} \geq r_i \qquad \forall i = 1,\ldots,n_1,\ \forall k = 1,\ldots,m \tag{11a}$$

$$t_{i+1,k} \geq t_{i,k} + \sum_{S \in \mathcal{N}_i} x_{S,k} f_k(S) \qquad \forall i = 1,\ldots,n_1 - 1,\ \forall k = 1,\ldots,m \tag{11b}$$

$$\phi^* \geq t_{n_1,k} + \sum_{S \in \mathcal{N}_{n_1}} x_{S,k} f_k(S) \qquad \forall k = 1,\ldots,m \tag{11c}$$

$$\sum_{k=1}^{m} \sum_{S: S \ni i} x_{S,k} = 1 \qquad \forall i = 1,\ldots,n_1 \tag{11d}$$

$$t \geq 0, x \in \{0,1\}$$