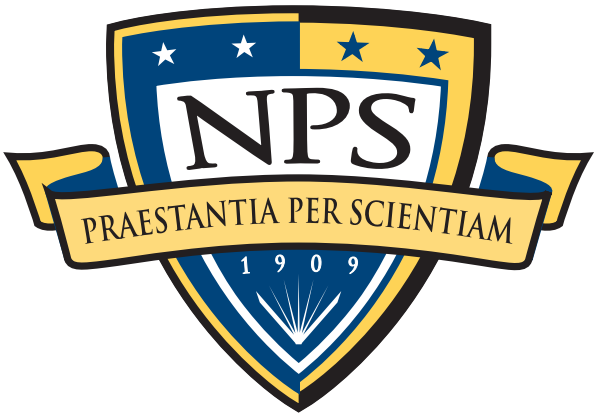

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION PROPOSAL

OPTIMIZATION VIA ADAPTIVE RESOLVENT SPLITTING

by

Peter D. Barkley, Commander, United States Navy

Sept 2025

Dissertation Supervisor:	Assistant Professor Robert Bassett
Dissertation Committee:	Professor W. Matt Carlyle
	Professor Lucas Wilcox
	Professor Javier Salmeron
	Associate Professor Emily Craparo
	Assistant Professor Louis Chen

THIS PAGE INTENTIONALLY LEFT BLANK

1 Introduction

This research introduces a broad class of distributed algorithms which extend operator splitting approaches like progressive hedging, Douglas-Rachford splitting, the Ryu splitting algorithm, and the recent Malitsky and Tam algorithm (Malitsky and Tam 2023). We parameterize this class of algorithms in terms of a set of matrices which characterize them, and use the parametrization to build semi-definite programs which provide alternative algorithms striking a middle ground between centralized hub and spoke and decentralized ring designs. We analyze their convergence properties and compare their performance across a wide variety of decentralized distributed optimization problem classes and applications. These classes include distributed coordination problems with both local and shared decision variables, multi-stage stochastic programming problems, and mixed integer variants of these. We apply the algorithms to mission optimization by distributed military systems and electrical grid coordination. We examine empirical run time and rate of convergence as well as measures of inter-node communications, particularly under degradation of the network.

2 Motivating Examples

2.1 UxS Mission Optimization

Planned DoD investments in unprecedented numbers of unmanned systems (UxS) under the Replicator program create an opportunity for the use of large scale optimization to maximize their mission utility (Garamone 2023). While a large body of work describes splitting algorithms suitable for large scale optimization at this level, little existing research does so in a decentralized manner which uses the platforms as the edge computation nodes and builds the algorithms not only for speed and memory efficiency, but also for network robustness and communication efficiency. This research tailors the algorithm to take advantage of the computing power on each node while accounting for their communication constraints and minimizing their communications while preserving minimal levels of redundancy.

Weapon Target Assignment

The (extensively studied) Weapon Target Assignment (WTA) problem matches a set of munitions which vary in kind and quantity against a set of targets which vary in targetability, kind, and quantity in order to minimize adversary success. Each munition has a particular level of effectiveness against a given target, which may vary based on the type of each and potentially the distance from the firing platform to the target as well. Standard WTA formulations then model post-engagement adversary effectiveness as the weighted sum of the remaining mission effectiveness of adversary platforms after the engagement. In the static WTA problem, all information becomes available at the beginning of the engagement. In the more difficult dynamic WTA problem, the engagement proceeds through multiple

rounds with increasing levels of information made available. In its canonical form, the static WTA problem is formulated as an integer program with a convex non-linear objective function, but several authors have found tractable relaxations which provide high quality integer solutions even for large problems (Manne 1958; Ahuja et al. 2007; Hendrickson et al. 2023). See appendix 1 for one such formulation. The methods described below can solve the continuous relaxation of this problem without any modification.

Mining

The UxS mining problem we examine consists of determining the number of mines n platforms will drop in m planned minefields. The objective function penalizes under- and over-shooting the target number of mines for each field and includes a secondary cost penalty for each platform-minefield pairing. The cost penalty is a function of endurance cost, survivability cost, and mission cost. The distance between a given platform and minefield and its speed determine the endurance cost penalty for a given pairing. This endurance penalty may have a multiplier which expresses the alternative utility of the platform for other missions per unit time. Given the possibility of non-permissive environments and challenging environmental conditions, the cost function also incorporates a survivability cost which captures the risk to the platform incurred by executing the mission. This risk is a function of the platform value and the likelihood of platform loss. The utility of the platform for other mining missions (and the difficulty of re-arming it) determine the mission cost of a given pairing. See appendix 1 for a sample formulation.

Decentralized optimization application to UxS

In each of these cases, the challenge lies in optimizing the mission in a timely manner across a large number of assets which have their own onboard computers. Solving – and even approximating – these optimization problems in time may not be possible on a single platform given the onboard processing power available. Splitting methods offer an approach which instead splits the sum of individual functions in the objective across the computing nodes, creating decision variables on each node which must then be brought to final a consensus value which is common across the nodes. In cases where the individual functions depend on variables which impact only that function, splitting also benefits from separability. Other approaches to decentralized optimization include gossip algorithms, which rely on stochastic point to point communications governed by a doubly stochastic mixing matrix (Boyd et al. 2006), primal-dual gradient based methods (including PDHG) (Condat 2013; Hendrickson et al. 2023), and P-EXTRA, a splitting method which relies on mixing matrices across multiple time steps for consensus (Shi et al. 2015). Recent work by Tam (2023), however, has shown promising results for the frugal splitting methods described below relative to P-EXTRA and PDHG.

2.2 Large Scale Parallel Computing

It is also possible that bespoke frugal splitting methods could speed up parallel optimization in high performance computing (HPC) clusters by tailoring the consensus process and subproblem sequence to the architecture of the given system and problem. This portion of the research focuses on the stochastic unit commitment problem as a test case for exploring the benefits of tailored frugal resolvent splittings in HPC settings and for stochastic programs in general.

Stochastic Unit Commitment

The power generation unit commitment problem provides another well-studied and economically valuable problem which has a finite sum objective function amenable to splitting. In it, a utility system operator decides which generation units to operate, and at which level, to meet power demand over a number of discrete time periods while minimizing overall cost. Typical instances optimize over hundreds of generating units and tens of time periods. Non-stochastic versions of the problem can be solved relatively quickly (Knueven et al. 2020), but stochastic instances are so large that extensive form solutions become intractable (Håberg 2019). Decentralized optimization methods using progressive hedging and Benders decomposition have made significant progress in finding good solutions in reasonable time frames (Ryan et al. 2013). Some of these centralized approaches also incorporate asynchronous updates between nodes to remove bottlenecks in the parallel computation (Aravena and Papavasiliou 2015). It is possible that designing the splitting algorithm in light of the parallel computing architecture as is proposed below could yield further improvements in solution speed. See appendix 1 for a sample stochastic unit commitment formulation.

3 Splitting Algorithms

In this section we review a number of concepts which support our work, primarily following the exposition in Bauschke et al. (2011) unless otherwise annotated.

3.1 Definitions and Notation

A (potentially set-valued) operator A on a Hilbert space $\mathcal{H}$ is monotone if $(x-y)(u-v) \geq 0$ for all $(x,u),\ (y,v)$ in its graph. A is maximally monotone if there is no monotone operator that properly contains it.

The resolvent of an operator A on some $x \in \mathcal{H}$ is given by $J_A(x) = (\mathrm{Id} + A)^{-1}(x)$ where Id is the identity operator. Equivalently, if $J_A(x) = z$, we have $x \in (\mathrm{Id} + A)(z)$ or $x \in z + A(z)$. Finding the stationary point $x = J_A(x)$ is therefore equivalent to finding a zero of A.

The prox of a function f at x is defined as $\text{prox}_f(x) = \arg\min_u f(u) + \frac{1}{2}||u - x||^2$. The resolvent of the subdifferential of a convex, closed, and proper (ccp) function is the prox operator of that function.

Proof 1 *Let* $z = (\text{Id}+\partial f)^{-1}(x)$ *for some ccp function* f. *This implies that* $x \in z+\partial f(z)$. *This means* $0 \in \partial f(z)+(z-x)$. *Since* $(z-x) = \frac{\partial}{\partial z}\frac{1}{2}||z-x||^2$, *we have* $z = \arg\min_u f(u)+\frac{1}{2}||u-x||^2$, *so* $z = prox_f(x)$ *(Ryu and Yin 2022).*

The decomposition methods examined below typically rely on separate instances of the decision variables on each of the n computation nodes, which are then brought into consensus in some way. To formalize the examination of these, we defined the *lifted* version of a vector of decision variables $x \in \mathcal{H}$ from $\mathcal{H}$ into $\mathcal{H}^n$ as $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{H}^n$ where each $x_i \in \mathcal{H}$. To concisely describe linear operations on this lifted vector, for a matrix $W \in \mathbb{R}^{n\times m}$, we then define $\mathbf{W} := W \otimes \text{Id}$ where Id is the identity on $\mathcal{H}$ and $\otimes$ is the Kronecker product. Then in the product $\mathbf{Wx}$, each coefficient in W acts on a single element of $\mathcal{H}$. For a vector of decision variables x where $\mathcal{H}$ is $\mathbb{R}^d$, a coefficient W_{ij} then linearly operates on the entirety of the j^{th} instance of the decision variables as a portion of the i^{th} element of the result (i.e. $\mathbf{W}$ is a bounded linear operator from $\mathcal{H}^m$ to $\mathcal{H}^n$). For some $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, we have consensus if $x_1 = x_2 = \cdots = x_n$ (Malitsky and Tam 2023).

The graph Laplacian of a weighted undirected graph G with nodes N and edges E is given by $W = D - A$ where $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the weighted degree of node i in D_{ii} and zeros elsewhere, and $A \in \mathbb{R}^{n\times n}$ is the weighted adjacency matrix, giving the weight of the edge between nodes i and j in A_{ij}. If $M \in \mathbb{R}^{|E|\times|N|}$ is a directed edge adjacency matrix, we can also find the graph Laplacian $W = M^T M$.

Every graph Laplacian has a minimal eigenvalue equal to 0, which corresponds to the eigenvector of all ones $\mathbb{1}_n$. The graph Laplacian is also therefore symmetric positive semi-definite. Its second smallest eigenvalue gives its algebraic connectivity, or *Fiedler value*. This corresponds to the level of connectedness of the graph. The Fiedler value will be greater than 0 if and only if the graph is connected. It is closely related to the edge and node connectivity of an unweighted graph, which gives the number of edges (or nodes respectively) which must be removed to form a disconnected graph (Fiedler 1973).

Frugal Resolvent Splittings

In (Ryu 2020), Ryu describes the concept of a frugal resolvent splitting algorithm. The Douglas-Rachford splitting algorithm provides the prototypical example. For a given objective function of the form $\sum_{i=1}^n f_i(x)$ where f_i are closed, convex, and proper, a resolvent splitting sequentially finds the resolvent of linear functions of the subdifferentials of f_i and any previously determined resolvents. A frugal resolvent splitting calculates each resolvent

only once in each iteration. Because the operations on the resolvents are linear, the algorithm can be expressed in terms of a set of matrices giving the scalar multiples of the terms. See appendix 1 for descriptions of the proximal point method, Douglas-Rachford splitting, Ryu splitting, the Malitsky Tam algorithm, and progressive hedging in terms of the resolvent matrices and the resulting implementation.

Let $\mathbf{F} = (\partial f_1, \ldots, \partial f_n)$. The frugal resolvent splitting with lifting from $x \in \mathcal{H}$ to $\mathbf{x} \in \mathcal{H}^n$ and $\mathbf{z} \in \mathcal{H}^d$ consists of fixed point operator $T : \mathcal{H}^d \to \mathcal{H}^d$ given by

$$T(\mathbf{z}) := \mathbf{z} + \gamma \mathbf{M}\mathbf{x}, \quad \mathbf{y} = \mathbf{B}\mathbf{z} + \mathbf{L}\mathbf{x}, \quad \mathbf{x} = J_{\mathbf{F}}(\mathbf{y})$$

where $M \in \mathbb{R}^{d \times n}$, $B \in \mathbb{R}^{n \times d}$, $L \in \mathbb{R}^{n \times n}$ are coefficient matrices for $\mathbf{M}$, $\mathbf{B}$, and $\mathbf{L}$ (the results of the Kronecker products described above), and $\gamma \in (0, 1)$.

We assume the following assumptions hold:

1. $\ker M = \operatorname{span}(\{\mathbb{1}_n\})$
2. $B = -M^T$
3. $L + L^T - 2\mathrm{Id} + M^T M \preceq 0$
4. $\sum_{i,j} L_{ij} = n$ and L is lower triangular
5. For every $\mathbf{z} \in \mathcal{H}^d$, there is a unique $\bar{x} \in \mathcal{H}$ such that for $\mathbf{x} = \mathbb{1} \otimes \bar{x}$

$$\mathbf{x} = J_{\mathbf{F}}(\mathbf{B}\mathbf{z} + \mathbf{L}\mathbf{x})$$

   For this assumption, it suffices to have one row of L sum to zero.
6. $\|L\| < 1$

With the additional assumption that L is lower triangular with zeros on the diagonal in place of assumption 5 and 6, Tam proves convergence for $\mathbf{F} = (F_1, \ldots, F_n)$ where all component operators are maximally monotone (of which the set of subdifferential operators of ccp functions is a subset). Specifically, if the set of zeros of $\sum F_i$ is non-empty, then the iterates $\mathbf{z}^{k+1} = T(\mathbf{z}^k) = \mathbf{z}^k + \gamma \mathbf{M}\mathbf{x}^k$, $\mathbf{x}^k = J_{\mathbf{F}}(\mathbf{B}\mathbf{z}^{\mathbf{k}} + \mathbf{L}\mathbf{x}^{\mathbf{k}})$ produce a sequence $(\mathbf{z}^k)$ which converges weakly to a fixed point, and a sequence $(\mathbf{x}^k)$ which converges weakly to a point $(\bar{x}, \ldots, \bar{x}) \in \mathcal{H}^n$ such that $\bar{x} \in \operatorname{zer}(\sum F_i)$.

Tam also notes that the matrix M functions as an edge incidence matrix linking the results of the resolvent calculations into edges which then feed the resolvent calculations on their nodes, and L allows the direct flow of resolvents between nodes. Given the fact that the graph Laplacian of a matrix can be found as $W = M^T M$ where $M \in \mathbb{R}^{|E| \times |N|}$ is the edge incidence matrix, we can move back and forth from the graph Laplacian W and its edge incidence matrix M as long as we assume there are no loops.

3.2 Adaptive Resolvent Splitting

The key insight this research explores is the extension of the set of frugal resolvent splitting algorithms by leveraging the class of matrices satisfying these requirements. This minimal description allows the construction of semi-define programs (SDP) for designing such matrices, and further allows the addition of various constraints and penalties on the network structure which support its application to decentralized optimization in real-world contexts.

To our knowledge, no existing results describe the use of an SDP to design frugal resolvent splittings. The optimization problem below lays it out in detail. Constraint 1a and the requirement for W to be positive semi-definite in constraint 1e together enforce the existence of M satisfying assumption 1 above. Constraint 1b enforces assumption 3. Constraint 1c convexly sets the minimal algebraic connectivity of the graph defined by W, since its minimal eigenvalue is 0 and its second gives the connectivity. Constraint 1d and the triangularization procedure below enforce assumption 4. One of the benefits of this construction lies in the ability offered by constraints 1f and 1g to enforce a list of prohibited communication paths across W and L, and the option to variably penalize them via the loss function.

Algorithm design problem

The adaptive frugal resolvent splitting algorithm design problem therefore lies in minimizing a penalty function on $\tilde{L}$ and W subject to the constraints in the SDP below:

$$
\begin{aligned}
\min_{\tilde{L},W} \quad & \phi(\tilde{L}, W) \\
\text{s.t.} \quad & W\mathbb{1}_n = 0 && \text{(1a)} \\
& 2\tilde{L} - 2I + W \preceq 0 && \text{(1b)} \\
& \lambda_1(W) + \lambda_2(W) \geq c && \text{(1c)} \\
& \sum \tilde{L}_{ij} = n && \text{(1d)} \\
& W \succeq 0 && \text{(1e)} \\
& W \in \mathcal{W} && \text{(1f)} \\
& \tilde{L} \in \mathcal{L} && \text{(1g)}
\end{aligned}
$$

where $\mathcal{W}$ is the convex set of feasible values for W, $\mathcal{L}$ is the convex set of feasible values for $\tilde{L}$, c is the minimum algebraic connectivity, and ϕ is a loss function which penalizes communication.

We then recover L, M, and B. Since L must be triangular, we define a *triangularization* operator $\mathrm{Ltri} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ as

$$\mathrm{Ltri}(X) = Y,$$

where

$$Y_{i,j} = \begin{cases} X_{i,j} + X_{j,i} & \text{if } i < j \\ X_{i,j} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

L is then defined as $\text{Ltri}(\tilde{L})$ and still satisfies constraints 1c and 1d, and any elements $\tilde{L}_{ij}$ and $\tilde{L}_{ji}$ which must both be zero for all $\tilde{L} \in \mathcal{L}$ will still be zero in L.

We form $B \in \mathbb{R}^{n \times d}$ as the weighted edge incidence matrix of the graph determined by the graph Laplacian W with the assumption that there are no loops. The d non-zero entries in the lower diagonal of W imply d edges in its graph and set the dimension of B. We then let $M = -B^T$, completing the algorithm parametrization.

This results in the matrices which provide the algorithm design. Entry L_{ij} determines whether a node i utilizes the resolvent of node j and how much weight to assign it. Entry B_{il} determines whether node i is incorporating the consensus value of the variables in which link l forces consensus. It need not participate in every link since the links fully connect the graph, and the values from each nodes eventually reach the entire graph.

Implementation

Once the algorithm matrices have been designed, we solve for the minimum of $\sum_{i=1}^{n} f_i(x)$ by iteratively solving the following subproblem on each node i for iteration k:

$$\min_{x_i} f_i(x_i) + \frac{1}{2}||(1 - L_{ii})x_i - \sum_{j=1}^{i-1} L_{ij} x_j^k - \sum_{l=1}^{d} B_{il} z_l^{k-1}||_2^2 \tag{2}$$

where x_i is the copy of the decision variables on node i, x_j are the decision variable values resulting from node j ahead of i in the calculation, and $z_l^{k+1} = z_l^k + \gamma \sum_{j=1}^{n} M_{lj} x_j$. If any constraints limit the feasible region for x_i, these can be thought of as indicator functions included in the definition of f_i. If any decision variables are separable across the functions, we can remove then from the set of lifted variables and use them only in the local computation, setting $f_i(x) = \inf_{y_i} f_i(x, y_i)$.

4 Proposed Contributions

Our proposed contributions lie in three areas:

1. theoretical advances in the field of frugal resolvent splittings,
2. implementation of algorithm design SDPs tailored to parallel computing and decentralized optimization, and
3. application of SDP-designed splitting algorithms to the SUCP, WTA, and other relevant UxS mission optimization problems.

4.1 Theoretical

Proposed theoretical contributions focus on analyzing the impact of network structure on the rate of convergence, determining the effect of function ordering on that rate, and identifying possible relaxations of the convergence assumptions.

Our proposed analysis of the impact of the graph structure will focus on the identification of:

1. specific permissible graphs which facilitate the efficient parallelization of subproblems,
2. ideal consensus structures striking a balance between connectedness and communication efficiency, possibly building on the analysis of exponential graphs in Ying et al. (2021), and
3. graphs which provide robustness to disruption (if possible).

Given existing work (Bauschke and Moursi 2016) demonstrating path dependence in splitting iterations based on the ordering of functions, this work will seek to determine an optimal function ordering based on the constituent functions' smoothness, strength of convexity, and the complexity of their resolvent calculations.

Finally, we may be able to expand the space of known convergent algorithms by identifying relaxations of the assumptions stated in Tam (2023). Possible relaxations include removing the requirement for zeros on the diagonal of L and determining whether $L+L^T-2\mathrm{Id}+M^TM$ can be permitted to have a positive eigenvalue. Given the reliance of the convergence proof on the negativity of of $\left\langle x-\bar{x},\left[M^TM-2\mathrm{Id}+L+L^T\right](x-\bar{x})\right\rangle$, this would require showing that $x^k-\bar{x}$ remained orthogonal to the eigenvector corresponding to the positive eigenvalue for all k.

4.2 Computational

Our proposed computational contribution includes formulation and implementation of algorithm design SDPs tailored to the constraints of decentralized optimization by unmanned

platforms and parallel computing environments. The use of these types of algorithms for solving large UxS optimization problems motivates the theoretical analysis of consensus structures balancing connectedness and communication efficiency. Our computational work will implement both the algorithm design and algorithm execution in a context representative of UxS coordination, including latency and potential disruption. We will build algorithms which target specific parallelization structures, function ordering, and function agglomeration to determine their relative computational efficiency and compare them with existing centralized and decentralized optimization algorithms under similar processor speed, memory, and communication constraints.

We will conduct similar implementation work in a parallel computing environment, building algorithm design SDPs which account for latency between nodes, data bandwidth limitations, and parallelization.

4.3 Application

We will then apply our implementation work to significant real-world applications, initially targeting continuous relaxations the Stochastic Unit Commitment Problem, the Weapon Target Assignment Problem, and other relevant UxS optimization problems. We will also examine its applicability to the integer versions via a distributed branch and bound aligned with the splitting. A number of valuable instances of SUCP remain too large for single compute nodes, and significant recent research continues to analyze splitting methods for it. This work will contribute to that pursuit. Likewise, in the DOD, significant levels of planning continue to focus on the use of large number of unmanned system to defeat enemy aggression. The optimal use of unmanned systems enables these efforts, and this work will demonstrate state-of-the-art options for determining that optimal employment in the WTA and mining scenarios.

We will assess our contribution by collecting the time and number of iterations until convergence of both the algorithm generating SDP and the final algorithm in each problem class. For the resulting algorithms, we will analyze convergence not only deterministically, but also under stochastic disruption. We will also examine the maximum feasible network size given time and memory constraints on the SDP, including under direct formulation (rather than via CVXPY) and using scalable sparse SDP solvers if needed (Yurtsever et al. 2021).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: 1

A.1 Weapon-Target Allocation Formulation

This formulation is derived from (Hendrickson et al. 2023)

Indices and Sets

$i \in 1 \ldots n$ weapons

$j \in 1 \ldots l$ targets

Parameters

V_j: the value of target j

Pk_{ij}: the probability that weapon i destroys target j

Decision Variables

$x_{ij} \in \{0,\ 1\}$: whether weapon i is employed against target j

$$\min_x \sum_{j=1}^{l} V_j \prod_{i=1}^{m} (1 - Pk_{ij})^{x_{ij}} \tag{A.1}$$

$$\text{s.t.} \sum_{j=1}^{l} x_{ij} \leq 1 \quad \forall i \in 1 \ldots n \tag{A.2}$$

A.2 Mining Formulation

Indices and Sets

$i \in 1 \ldots n$ platforms

$j \in 1 \ldots m$ minefields

Parameters
cf_{ij}: endurance cost for platform i to mine minefield j

cs_{ij}: survivability cost associated with risk to platform i of mining minefield j

$c_{ij} = cf_{ij} + cs_{ij}$: decision cost of mining minefield j with platform i

cm_i: per mine mission cost associated with risk to other mining missions for platform i (loss of future opportunity)

μ_j: cost of underseeding minefield

κ_j: cost of overseeding minefield

M_j: target number of mines in minefield j

p_i: payload of platform i

Decision Variables
x_{ij}: number of mines platform i mines in minefield j

y_{ij}: 1 if platform i mines minefield j, 0 otherwise

u_j: number of mines under the target number in minefield j

v_j: number of mines over the target number in minefield j

Objective Function
$\min \sum_{i=1}^{n} \sum_{j=1}^{m} (c_{ij} y_{ij} + cm_i x_{ij}) + \sum_{j=1}^{m} (\mu_j u_j + \kappa_j v_j)$

Constraints
$\sum_{i=1}^{n} x_{ij} = M_j - u_j + v_j \quad \forall j \in 1 \ldots m$ (1)

$\sum_{j=1}^{m} x_{ij} \leq p_i \quad \forall i \in 1 \ldots n$ (2)

$x_{ij} \leq p_i y_{ij} \quad \forall i \in 1 \ldots n, j \in 1 \ldots m$ (3)

$y_{ij} \in \{0, 1\} \quad \forall i \in 1 \ldots n, j \in 1 \ldots m$ (4)

$u_j \geq 0 \quad \forall j \in 1 \ldots m$ (5)

$v_j \geq 0 \quad \forall j \in 1 \ldots m$ (6)

$x_{ij} \geq 0 \quad \forall i \in 1 \ldots n, j \in 1 \ldots m$ (7)

Constraint 1 sets u_j as the amount under the target based on the planned seeding and v_j as the over shoot.

Constraint 2 requires the employment to be less than or equal to the capacity of each platform (p_i).

Constraint 3 sets x_{ij} to 0 if y_{ij} is 0.

Constraint 4 sets y_{ij} to 0 or 1.

Constraints 5, 6, and 7 sets u_j, v_j, and x_{ij} to be greater than or equal to 0.

A.3 Stochastic Unit Commitment Formulation

Following (Knueven et al. 2020), we use the following summary formulation for the UCP:

Indices and Sets

$g \in \mathcal{G} = \{1 \ldots G\}$ generators

$t \in \mathcal{T} = \{1 \ldots T\}$ time periods

Parameters

$L \in \mathbb{R}^T$: the demand schedule over the timesteps.

Π_g gives the feasible operating region across the decision variables and cost function for generator g.

Decision Variables

$p_g \in \mathbb{R}^T$ gives the generated power schedule for generator g over the timesteps.

$\bar{p}_g \in \mathbb{R}^T$ gives the maximum available power for generator g over the timesteps.

$u_g \in \{0, 1\}^T$ gives the on/off status for generator g.

θ covers other decision variables affecting the system operation (other power sources, etc).

Functions
$c_g(t)$ gives the cost of operating generator g at time t as a function of the decision variables.

$$\begin{aligned} \min \quad & \sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}} c_g(t) \\ \text{s.t.} \quad & \sum_{g\in\mathcal{G}} A_g(p_g, \bar{p}_g, u_g) + N(\theta) = L && \text{(A.3)} \\ & (p_g, \bar{p}_g, u_g, c_g) \in \Pi_g \quad \forall g \in \mathcal{G} && \text{(A.4)} \end{aligned}$$

A.3.1 Full UCP Formulation

The full formulation uses the following additional decision variables seen in the stochastic UCP below:

$v_g \in \{0,1\}^T$ gives the startup decision for generator g across the timesteps.

$f_l \in \mathbb{R}^T$ gives power flow over line l (part of the function $N(\theta)$ in the Kneuven formulation.

A.3.2 Stochastic UCP

Aravena (Aravena and Papavasiliou 2015) uses the following extension for the two stage stochastic UCP, indexing across scenarios with $s \in S$, and providing non-anticipativity for commitment and startup via w_g and z_g:

$$\begin{aligned} \min_{p,u,v,f,w,z} \quad & \sum_{s\in S}\sum_{g\in G}\sum_{t\in T} \pi_s c_{gs}(t) && (1) \\ \text{s.t.} \quad & \sum_{g\in G} p_{gst} + \sum_{l\in N} f_{lst} = L_{st}, s \in S, t \in T && (2) \\ & (p_{gs},\ \bar{p}_{gs},\ u_{gs},\ v_{gs},\ f_s) \in \Pi_{gs},\ \forall s \in S, \forall g \in G && (3) \\ & (w_g,\ z_g) \in \Pi_{gwz}, \forall g \in G && (4) \\ & u_{gs} = w_g,\ v_{gs} = v_g,\ \forall g \in G_{\text{SLOW}}, s \in S && (5) \end{aligned}$$

π_s here is the probability of scenario s.

G_{SLOW} are the generators which require advanced decisions for unit commitment and startup (thermal generators, primarily).

A.4 Algorithms

Following the exposition in Tam (2023), we present a number of splitting algorithms here in terms of their characterization as frugal resolvent splitting methods. We then extract their characteristic coefficient matrices and note the validity of the Tam assumptions for these matrices.

A.4.1 Douglas-Rachford Splitting

Douglas-Rachford splitting addresses the problem of finding a zero of maximally monotone operators A and B on $\mathbb{R}^d$, which is equivalent to minimizing $f_1(x) + f_2(x)$ where A and B are the subgradients of ccp functions f_1 and f_2. (Tam 2023; Eckstein and Bertsekas 1992)

1. Let counter $k = 0$, tolerance ϵ as desired, $\gamma \in (0, 1)$, $\alpha > 0$, and $z^0 = 0$ for $z \in \mathbb{R}^d$
2. Let $x_1 = J_{\alpha A}(z^k)$
3. Let $x_2 = J_{\alpha B}(2x_1 - z^k)$
4. Let $z^{k+1} = z^k + \gamma(x_2 - x_1)$. If $||z^{k+1} - z^k|| < \epsilon$, go to final step.
5. Let $k = k + 1$, go to step 2.
6. Return x_1

In terms of the corresponding coefficient matrices, we have the following:

$$M = \begin{bmatrix} -1 & 1 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}.$$

Clearly, the null space of M is spanned by the ones vector, $\sum_{i,j} L_{i,j} = n$, $||L|| < 1$, and a row of L sums to zero. Steps 2 and 3 correspond to the fixed point iteration $\mathbf{x} = J_{\mathbf{F}}(\mathbf{Bz} + \mathbf{Lx})$, where $B = \begin{pmatrix} 1 \\ -1 \end{pmatrix} = -M^T$.

We also have $L + L^T - 2\mathrm{Id} + M^T M = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \preceq 0$, so all of the assumptions are satisfied by the Douglas-Rachford splitting.

A.4.2 Progressive Hedging

As noted by Ruszczyński and Shapiro (2003), progressive hedging (PH) (Rockafellar and Wets 1991) can be thought of as a special case of Douglas-Rachford splitting where the objective function is $f(x) + \iota_W(x)$ for $x \in \mathbb{R}^{S \times n}$, $n = \sum_{i=1}^{T} n_i$ across stages in $|T|$, $f(x) = \sum_{s=1}^{S} p_s f_s(x^s)$ (the expected value of the problem over scenarios in S), and ι_W is the indicator function for the subspace W defined by the non-anticipativity constraints. Note that the x we begin with in this view of PH is the full set of decision variable across all scenarios

– it is not the decision variable for a single instance which is then further lifted. While the separability of f across scenarios means its prox can be calculated separately for each scenario, that splitting is not explicitly defined by the Douglas-Rachford characterization of PH.

A.4.3 Ryu Algorithm

Ryu's algorithm (Ryu 2020) addresses the problem of finding a zero of maximally monotone operators A, B, and C on $\mathbb{R}^d$, which is equivalent to minimizing $f_1(x) + f_2(x) + f_3(x)$ where A, B, and C are the subgradients of ccp functions f_1, f_2, and f_3. In terms of the corresponding coefficient matrices, we have:

1. Let counter $k = 0$, tolerance ϵ as desired, $\theta \in (0, 1)$, $\alpha > 0$, and $z^0 = 0$ for $z \in \mathbb{R}^{2d}$
2. Let $x_1 = J_{\alpha A}(z_1^k)$
3. Let $x_2 = J_{\alpha B}(x_1 + z_2^k)$
4. Let $x_3 = J_{\alpha C}(x_1 - z_1 + x_2 - z_2)$
5. Let $z_1^{k+1} = z_1^k + \theta(x_3 - x_1)$
6. Let $z_2^{k+1} = z_2^k + \theta(x_3 - x_2)$. If $||z^{k+1} - z^k|| < \epsilon$, go to final step.
7. Let $k = k + 1$, go to step 2.
8. Return $\frac{1}{3}(x_1 + x_2 + x_3)$

$$M = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

As with Douglas-Rachford, we satisfy all required assumptions for convergence as a frugal resolvent splitting. Tam also provides an extension of this algorithm to n operators.

A.4.4 Malitsky-Tam Algorithm

The Malitsky-Tam minimal lifting algorithm (Malitsky and Tam 2023) finds the zero of the sum of n maximal monotone operators $F_1 \dots F_n$, equivalent to minimizing $\sum f_i(x)$ where each f_i is ccp and $F_i(x) = \partial f_i(x)$.

1. Let counter $k = 0$, tolerance ϵ as desired, $\gamma \in (0, 1)$ and $z^0 = 0$ for $z \in \mathbb{R}^{2d}$.
2. Let $x_1 = J_{F_1}(z_1^k)$.
3. For $i \in 2 \dots n-1$, let $x_i = J_{F_i}(x_{i-1} + z_i^k - z_{i-1}^k)$.
4. Let $x_n = J_{F_n}(x_1 + x_{n-1} - z_{n-1})$.
5. Let $z^{k+1} = z^k + \gamma \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix}$. If $||z^{k+1} - z^k|| < \epsilon$, go to final step.
6. Let $k = k + 1$, go to step 2.
7. Return x_1

In terms of the corresponding coefficient matrices, we have:

$$M = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ 0 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}.$$

As with Douglas-Rachford and the Ryu algorithm, this satisfies all required assumptions.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Ahuja RK, Kumar A, Jha KC, Orlin JB (2007) Exact and heuristic algorithms for the weapon-target assignment problem. *Operations research* 55(6):1136–1146.

Aravena I, Papavasiliou A (2015) A distributed asynchronous algorithm for the two-stage stochastic unit commitment problem. *2015 IEEE Power & Energy Society General Meeting*, 1–5 (IEEE).

Bauschke HH, Combettes PL, et al. (2011) *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408 (Springer).

Bauschke HH, Moursi WM (2016) On the order of the operators in the douglas–rachford algorithm. *Optimization Letters* 10:447–455.

Boyd S, Ghosh A, Prabhakar B, Shah D (2006) Randomized gossip algorithms. *IEEE transactions on information theory* 52(6):2508–2530.

Condat L (2013) A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of optimization theory and applications* 158(2):460–479.

Eckstein J, Bertsekas DP (1992) On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical programming* 55:293–318.

Fiedler M (1973) Algebraic connectivity of graphs. *Czechoslovak mathematical journal* 23(2):298–305.

Garamone J (2023) Hicks discusses replicator initiative. URL https://www.defense.gov/News/News-Stories/Article/Article/3518827/hicks-discusses-replicator-initiative/.

Håberg M (2019) Fundamentals and recent developments in stochastic unit commitment. *International Journal of Electrical Power & Energy Systems* 109:38–48.

Hendrickson K, Ganesh P, Volle K, Buzaud P, Brink K, Hale M (2023) Decentralized weapon–target assignment under asynchronous communications. *Journal of Guidance, Control, and Dynamics* 46(2):312–324.

Knueven B, Ostrowski J, Watson JP (2020) On mixed-integer programming formulations for the unit commitment problem. *INFORMS Journal on Computing* 32(4):857–876.

Malitsky Y, Tam MK (2023) Resolvent splitting for sums of monotone operators with minimal lifting. *Mathematical Programming* 201(1-2):231–262.

Manne AS (1958) A target-assignment problem. *Operations research* 6(3):346–351.

Rockafellar RT, Wets RJB (1991) Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research* 16(1):119–147.

Ruszczyński A, Shapiro A (2003) Stochastic programming models. *Handbooks in operations research and management science* 10:1–64.

Ryan SM, Wets RJB, Woodruff DL, Silva-Monroy C, Watson JP (2013) Toward scalable, parallel progressive hedging for stochastic unit commitment. *2013 IEEE Power & Energy Society General Meeting*, 1–5 (IEEE).

Ryu EK (2020) Uniqueness of drs as the 2 operator resolvent-splitting and impossibility of 3 operator resolvent-splitting. *Mathematical Programming* 182(1-2):233–273.

Ryu EK, Yin W (2022) *Large-scale convex optimization: algorithms & analyses via monotone operators* (Cambridge University Press).

Shi W, Ling Q, Wu G, Yin W (2015) A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing* 63(22):6013–6023.

Tam MK (2023) Frugal and decentralised resolvent splittings defined by nonexpansive operators. *Optimization Letters* 1–19.

Ying B, Yuan K, Chen Y, Hu H, Pan P, Yin W (2021) Exponential graph is provably efficient for decentralized deep training. *Advances in Neural Information Processing Systems* 34:13975–13987.

Yurtsever A, Tropp JA, Fercoq O, Udell M, Cevher V (2021) Scalable semidefinite programming. *SIAM Journal on Mathematics of Data Science* 3(1):171–200.