Consensus ADMM Under Uncertainty

Aymeric Legros - École Polytechnique, IP Paris
David L. Woodruff - UC Davis

August 8, 2024

Abstract

Decomposition using consensus ADMM can be used to allow parallelization efficiencies or for reasons related to information security. In either case, the input data may be uncertain and we give a decomposition algorithm.

1 Introduction

This work focuses on a scenario decomposition algorithms for optimization under uncertainty called Progressive Hedging (PH) as proposed by Rockafellar and Wets [6]. As noted in [3], the PH algorithm is related to other decomposition algorithms such as Alternating Direction Methods of Multipliers [2] (ADMM) and we exploit that relationship to create an algorithm for ADMM decomposition with uncertain data.

1.1 Notation

1.1.1 Stochastic Program

We begin with notation for a stochastic program without considering consensus ADMM decomposition. Our notation is similar to [4]. We denote by T the number of decision stages and we use $t \in \{1, \ldots, T\}$ to index stages; however, we observe that these stages do not always correspond to time periods. Let ξ_t be a random variable, which may be vector valued, associated with each decision stage t. This random variable represents the stochastic aspect of the problem. In the case of time periods, we consider that the decisions for stage t are made once the values of the random variables for stages up to and including t are known. Hence we will mostly refer to the value ξ_t only for stage $2, \ldots, T$.

We denote by $\vec{\xi}^{\,t}$ the realized values of all ξ_t up to and including stage t. In particular,

$$\vec{\xi}^{\,T} = (\xi_t,\ t = 2, \ldots, T)$$

refers to a full scenario. We will simply use the notation ξ.

We denote by Ξ the full set of scenarios, where each scenario ξ has probability $\Pr_\xi$. We define a tree corresponding to the set of realizations ξ such that different scenarios with the same realization up to stage t share a node corresponding to that stage t. Hence, $\vec{\xi}^{\,t}$ refers also to a node in the scenario tree.

We denote by $\mathcal{G}_t$ the set of all nodes for stage t and by $\mathcal{G}_t(\xi)$ the node corresponding to scenario ξ. If $\mathcal{D}$ is a given node, we denote by $\mathcal{D}^{-1}$ the set of scenarios that define the node.

We denote by x^t the decision variable at stage $t \in \{1, \ldots, T\}$, and $\vec{x}^{\,t}$ the decisions for all stages up to and including $t \in \{1, \ldots, T\}$. Let f be the cost function. More specifically, $f_1(x^1)$ corresponds to the first stage cost and $f_t(x^t; \vec{x}^{\,t-1}, \vec{\xi}^{\,t})$ each subsequent stage. Let us note that x^t is the argument of f_t while $\vec{x}^{\,t-1}, \vec{\xi}^{\,t}$ are parameters giving the solutions and realizations up to stage t.

Thus, we can express the multistage stochastic program as follows:

$$Z^* = \min_{x,\hat{x}} \sum_{\xi \in \Xi} \Pr_\xi \left[f_1(x^1(\xi)) + \sum_{t=2}^{T} f_t \left(x^t(\xi); \vec{x}^{\,t-1}, \vec{\xi}^{\,t} \right) \right] \tag{1a}$$

$$x^t(\xi) - \hat{x}^t(\mathcal{D}) = 0, \;\; t = 1, \ldots, T-1, \; \mathcal{D} \in \mathcal{G}_t, \; \xi \in \mathcal{D}^{-1} \tag{1b}$$

$$x(\xi) \in X_\xi, \;\; \xi \in \Xi \tag{1c}$$

with X_ξ a set of constraint for each scenario $\xi \in \Xi$. The condition (1b) enforces the decision variables non-anticipativity. Indeed, it forces x^t to only consider the information available before stage t (i.e. the scenarios that define the nodes $\mathcal{D} \in \mathcal{G}_t$). The condition (1c) summarizes all other constraints. Note that in this abstract formulation, minimization of variables not subject to non-anticipativity are subsumed by the f functions, typically f_T.

1.1.2 Consensus ADMM

Regardless of whether the input data are uncertain, one may find it useful to use *consensus ADMM* to decompose a problem into subproblems $(f^{(a)}(\cdot))$, $a \in \mathcal{A}$ so that

$$f(\cdot) = \sum_{a \in \mathcal{A}} f^{(a)}(\cdot) \tag{2}$$

while retaining a requirement that specified variables take the same value in specified subproblems. For each subproblem, there is a list of variables that must be constrained to have value equal to the same variable in any other subproblem(s) in which it occurs, $\mathcal{C}_a, \; a \in \mathcal{A}$.

A difference between standard formulations in stochastic programming and consensus ADMM is that in the latter the function $f^{(a)}$ can sometimes include a different set of variable for each subproblem. For a variable, v, (i.e., an element of the vector x) let δ_{av} be 1 if $v \in \mathcal{C}_a$ and 0 otherwise. Let $K_v = \sum_{a \in \mathcal{A}} \delta_{av}$ be the number of subproblems in which v appears (so if $K_v > 0$, that means v is a consensus ADMM variable).

2 Progressive Hedging

2.1 The Basic Progressive Hedging Algorithm

The Progressive Hedging (PH) algorithm was proposed by Rockafellar and Wets [6] and has been described in many places such as [7]. It decomposes stochastic programs along scenarios in several sub-problems in order to solve them. We begin with a standard version for multistage stochastic programs (such as (1)) as described in Algorithm 1, which is the same as in [4]. For this version of the algorithm there are no consensus ADMM subproblems.

We use the notation developed so far to write the conditional probability of scenario ξ at node $\mathcal{D}$ as

$$\mathrm{Cp}_{\mathcal{D}}(\xi) = \frac{\mathrm{Pr}_\xi}{\sum_{\upsilon \in \mathcal{D}^{-1}} \mathrm{Pr}_\upsilon}$$

Algorithm 1 Progressive Hedging

```
1: Initialization: Let $\nu \leftarrow 0$ and $w^{(t,\nu)}(\xi) \leftarrow 0,\ \forall \xi \in \Xi,\ t = 1,\dots,T$. Compute
   for each $\xi \in \Xi$:
       $x^{(\nu+1)}(\xi) \in \mathrm{argmin}_x\, f_1(x^1) + \sum_{t=2}^{T} f_t\left(x^t;\, \vec{x}^{\,t-1}(\xi),\, \vec{\xi}^{\,t}\right)$
2: Iteration Update: $\nu \leftarrow \nu + 1$
3: Aggregation: Compute for each $t = 1,\dots,T-1$ and for each $\mathcal{D} \in \mathcal{G}_t$:
       $\bar{x}^{(t,\nu)}(\mathcal{D}) \leftarrow \sum_{\xi \in \mathcal{D}^{-1}} \mathrm{Cp}_{\mathcal{D}}(\xi) x^{(t,\nu)}(\xi)$
4: Price Update: Compute for each $t = 1,\dots,T-1$ and for each $\xi \in \Xi$
       $w^{(t,\nu)}(\xi) \leftarrow w^{(t,\nu-1)}(\xi) + \rho\left[x^{(t,\nu)}(\xi) - \bar{x}^{\nu}(\mathcal{G}_t(\xi))\right]$
5: Decomposition: Compute for each $\xi \in \Xi$
       $x^{(\nu+1)}(\xi) \in \mathrm{argmin}_x\, f_1(x^1) + \sum_{t=2}^{T} f_t\left(x^t;\, \vec{x}^{\,t-1}(\xi),\, \vec{\xi}^{\,t}\right)$
       $+ \sum_{t=1}^{T-1}\left[w^{(t,\nu)}(\xi)^\top x^t + \frac{\rho}{2}\|x^t - \bar{x}^{(\nu)}(\mathcal{G}_t(\xi))\|^2\right]$
6: Termination: If a criterion is met, Stop. Otherwise go to step 2.
```

2.2 Progressive Hedging Algorithm to Accommodate Consensus ADMM

The intuition is as follows: from an algorithmic perspective we treat the consensus variables in the algorithm as if they were a special sort of non-anticipative variable and treat the admm subproblems as if they are special scenarios. We need to deal with the fact that not all consensus variables are in all subproblems and we need to combine the two sources of scenarios.

The collection of ADMM subproblems are considered to emanate from a scenario tree node that is replicated for addition to the original scenario tree at every original leaf node. This results in a total of $|\Xi||A|$ *extended scenarios.* We denote the elements of the extended tree with a tilde. Note that for every $\widetilde{\xi}$ there is a corresponding ξ that was extended to create it, denoted as $\widetilde{\xi} \mapsto \xi$. $\widetilde{\xi}$ contains the information of ξ local to its region. These scenarios have a stage $T+1$ with, of course, no additional stochastic data. They do have variables subject to non-anticipativity constrains that we now associate with stage T, which are the ADMM consensus variables. These variables may originally appear in only some subproblems (they might not appear in the argument to function $f^{(a)}$ for all subproblems a), so in order to have various expressions make sense, we add such variables to $(f^{(a)})$ for a in which they do not appear and fix their values to something arbitrary (e.g., zero) and the result is that the vector of variables subject to non-anticipativity is the same for all extended scenarios so we refer to it simply as x in the sequel.

We replace the real-valued scenario probabilities with a vector of probabilities corresponding to variables subject to nonanticipativity, $\pi^t_{\widetilde{\xi}}$, $\widetilde{\xi} \in \widetilde{\Xi}$, $t = 1, \ldots, T$. To establish values for the elements of this vector, we use the scenario ξ for which $\widetilde{\xi} \mapsto \xi$ For variables that do not appear in $\bigcup_{a \in \mathcal{A}} \mathcal{C}_a$ the value is simply the original probability of the scenario in which they occur scaled by the number of ADMM subproblems:

$$\frac{\Pr_\xi}{|\mathcal{A}|}.$$

For variables, v, subject to ADMM consensus constraints (i.e., $K_v > 0$), the value is

$$\frac{\Pr_\xi}{K_v}$$

and the value is zero for this variable in extended scenarios for subproblems where the variable does not originally appear.

To represent the vector of unconditional probabilities for variables subject to non-anticipativity at a node in the extended scenario tree, we use:

$$\tau_{\widetilde{\mathcal{D}}}(\widetilde{\xi}) = \frac{\pi_{\widetilde{\xi}}}{\sum_{\widetilde{v} \in \widetilde{\mathcal{D}}^{-1}} \pi_{\widetilde{v}}},$$

where the calculation is done element-wise.

Algorithm 2 displays the algorithm. Bear in mind that the extended tree has $T+1$ stages. To do the price update, we are going to need to mask out the variables added to subproblems for scenarios where they did not originally appear. For that purpose, we create a vector $m^t(\widetilde{\xi})$ with elements corresponding to the variables at each stage t. It will have a zero for variables that did not originally appear in $\widetilde{\xi}$ and a one for all other variables (i.e. most variables).

Algorithm 2 Progressive Hedging To Support Consensus ADMM

```
1: Initialization: Let $\nu \leftarrow 0$ and $w^{(t,\nu)}(\widetilde{\xi}) \leftarrow 0,\ \forall \widetilde{\xi} \in \Xi,\ t = 1, \dots, T$. Compute
   for each $\widetilde{\xi} \in \Xi$:
       $$x^{(\nu+1)}(\widetilde{\xi}) \in \operatorname{argmin}_x f_1(x^1) + \sum_{t=2}^{T} f_t \left( x^t;\ \vec{x}^{\,t-1}(\widetilde{\xi}),\ \vec{\xi}^{\,t} \right)$$
2: Iteration Update: $\nu \leftarrow \nu + 1$
3: Aggregation: Compute the average vector for each $t = 1, \dots, T$ and for each
   $\widetilde{\mathcal{D}} \in \widetilde{\mathcal{G}}_t$:
       $$\bar{x}^{(t,\nu)}(\widetilde{\mathcal{D}}) \leftarrow \sum_{\xi \in \mathcal{D}^{-1}} \tau_{\widetilde{\mathcal{D}}}(\widetilde{\xi}) \circ x^{(t,\nu)}(\widetilde{\xi})$$
4: Price Update: Compute for each $t = 1, \dots, T$ and for each $\widetilde{\xi} \in \Xi$
       $$w^{(t,\nu)}(\widetilde{\xi}) \leftarrow w^{(t,\nu-1)}(\widetilde{\xi}) + \rho m^t(\widetilde{\xi}) \circ \left( \left[ x^{(t,\nu)}(\widetilde{\xi}) - \bar{x}^{\nu}(\widetilde{\mathcal{G}}_t(\widetilde{\xi})) \right] \right)$$
5: Decomposition: Compute for each $\widetilde{\xi} \in \Xi$
       $$x^{(\nu+1)}(\widetilde{\xi}) \in \operatorname{argmin}_x f_1(x^1) + \textstyle\sum_{t=2}^{T+1} f_t \left( x^t;\ \vec{x}^{\,t-1}(\widetilde{\xi}),\ \vec{\xi}^{\,t} \right)$$
       $$+ \textstyle\sum_{t=1}^{T} \left[ w^{(t,\nu)}(\widetilde{\xi})^\top x^t + \frac{\rho}{2} \| x^t - \bar{x}^{(\nu)}(\widetilde{\mathcal{G}}_t(\widetilde{\xi})) \|^2 \right]$$
6: Termination: If a criterion is met, Stop. Otherwise go to step 2.
```

We intend for the sum in Step 3 to be element-wise as well as the product (i.e., we are computing the average for each vector element). In Step 4 the vector m^t "masks out" zero-probability variables.

Renormalizing the objective function

In order to report objective function values, they need to be renormalized because we used faux probabilities to treat the ADMM decomposition in an algorithm designed for stochastic programs. For ADMM, the objective function for a scenario is decomposed as a sum of objective functions in the sub-problems. More precisely, let us note f the overall objective function decomposed in $(f^{(a)})_{a \in \mathcal{A}}$. Every stochastic scenario $\xi \in \Xi$ can be decomposed as $(\xi^{(a)})_{a \in \mathcal{A}}$ such that for all x we re-state Equation (2):

$$\sum_{a \in \mathcal{A}} f^{(a)}(x, \xi^{(a)}) = f(x, \xi) := f_1(x^1(\xi)) + \sum_{t=2}^{T} f_t \left(x^t(\xi);\ \vec{x}^{\,t-1},\ \vec{\xi}^{\,t} \right)$$

Therefore, knowing that the probability of an ADMM scenario $\widetilde{\xi}$ is divided by $|\mathcal{A}|$ compared to the probability of the stochastic scenario ξ it originates from, we have:

$$\begin{aligned}
Z(x) &= \sum_{\xi \in \Xi} \Pr_\xi f(x,\xi) \\
&= \sum_{\xi \in \Xi} \sum_{a \in \mathcal{A}} \Pr_\xi f^{(a)}(x,\xi^{(a)}) \\
&= |\mathcal{A}| \sum_{\xi \in \Xi} \sum_{a \in \mathcal{A}} \Pr_{\xi^{(a)}} f^{(a)}(x,\xi^{(a)}) \\
&= |\mathcal{A}| \sum_{\widetilde{\xi} \in \widetilde{\Xi}} \Pr_{\widetilde{\xi}} \widetilde{f}(x,\widetilde{\xi})
\end{aligned}$$

Cost Coefficients

Note that to make Equation (2) hold, modelers may need to consider objective function coefficients. For example, if the objective function is linear and a consensus variable v appears in the objective function, the cost coefficients must sum across the subproblems in which it appears to the original coefficient.

3 Example of maximum profit distribution problem

To illustrate possible applications of the algorithm, we present here an example for a maximum profit distribution problem.

3.1 The non-decomposed problem

The deterministic problem

First, we introduce the notations for the non-decomposed, non-stochastic maximum profit distribution problem.

Index sets

The sets of **Nodes** $\mathcal{N}$ is decomposed into **Factory Nodes** $\mathcal{F} \subset \mathcal{N}$, **Buyer Nodes** $\mathcal{B} \subset \mathcal{N}$ and **Distribution Center Nodes** $DC \subset \mathcal{N}$.
The set of **Oriented Edges** $\mathcal{E} \subset \mathcal{N}^2$ represents the directed arcs.

Parameters

- **Net supply** (S_n) positive at factory nodes, negative at buyer nodes, null otherwise
- **Production Costs** (P_f) and **scrap loss** L_f at each factory node. The scrap loss is the proportion of loss during the production.
- **Revenues** (R_b) at each buyer node
- **Capacities** (K_e) at each arc
- **Shipping Costs** (C_e) at each arc

Variables

- **Flows** $0 \leq x_e \leq K_e$ at each arc
- **Slack variables** (y_n). If $n \in \mathcal{F}, 0 \leq y_n \leq S_n$ (unused capacity), if $n \in \mathcal{B}, S_n \leq y_n \leq 0$ (opposite of unmet demand) otherwise (for a distribution center) $y_n = 0$.
- **Inventory** $0 \leq i_f \leq y_f$ for $f \in \mathcal{F}$.
  Although the model is a **one-period model**, we allow an inventory which is valued at a low price: the half of the production cost. In the deterministic version of the model, this will always be zero, but will be non-zero for some scenarios with uncertain data.

Constraints

The slack allows the **Flow Balance** constraint:

$$\forall n \in \mathcal{N} \backslash \mathcal{F}, \sum_{o \in \mathcal{O}_n} x_o - \sum_{j \in \mathcal{I}_n} x_j = S_n - y_n$$

$$\forall n \in \mathcal{F}, \sum_{o \in \mathcal{O}_f} x_o - \sum_{j \in \mathcal{I}_f} x_j + i_n = (S_f - y_f) L_f$$

In which $\mathcal{O}_n$ (resp. $\mathcal{I}_n$) is the set of arcs leaving (resp. entering) n.

Objective function

To maximize the profit, we will minimize the cost, given by:

$$Z(x,y) = \sum_{e \in \mathcal{E}} C_e \, x_e + \sum_{n \in \mathcal{F}} (P_n \, (S_n - y_n) - \frac{P_n}{2} i_n) + \sum_{n \in \mathcal{B}} R_n \, (S_n - y_n)$$

We seek

$$\underset{\text{x,y}}{\operatorname{argmin}} \, Z(x,y)$$

The stochastic version

We now consider the stochastic version of this problem. We treat the scrap loss as uncertain. Therefore the amount of production needs to be decided first (with y_f for $f \in \mathcal{F}$), then the amount of production L_f^{ξ} is known (depending on the scenario ξ) and finally the shipping quantities are decided. This is a two-stage problem, where the non-anticipative variable is $(y_f)_{f \in \mathcal{F}}$, for each scenario ξ the model is as given above in the non stochastic-version with the loss $L_f = L_f^{\xi}$.

3.2 The ADMM decomposition

We now assume each model of our stochastic problem is an inter-regions maximum profit distribution problem which can be decomposed in sub-problems, in the sense that there exists regions such that the flow is restricted inside the regions, except for inter-region arcs between distribution centers in different

regions.
In a stochastic problem, the regions could be solved independently if the flow between regions were imposed. That is why PH is used with the flow among regions as consensus variables.
A representation is given in Figure 1. Only the flows, costs and capacities of inter-regional arcs are represented.

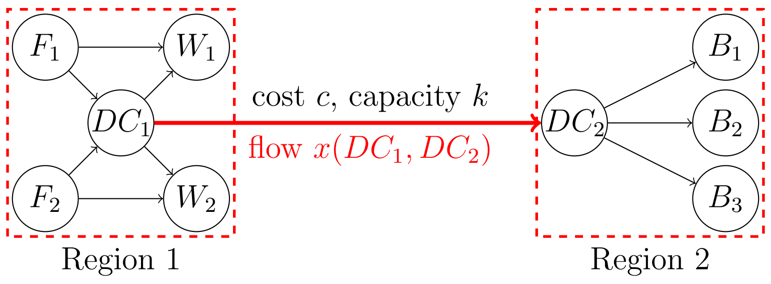

Figure 1: Representation of a stochastic scenario without ADMM decomposition

We have seen that to do an ADMM decomposition into stochastic ADMM sub-problems, we need to make sure that the overall objective function can be split into sub-problems.

$$Z(x,y) = \sum_{e\in\mathcal{E}} C_e\, x_e + \sum_{n\in\mathcal{F}} P_n\,(S_n - y_n) + \sum_{n\in\mathcal{B}} R_n\,(S_n - y_n - \frac{i_n}{2}) = \sum_{a\in\mathcal{A}} f^{(a)}(x, y, \xi^{(a)})$$

Therefore, we chose the objective function for an ADMM sub-problem $a \in \mathcal{A}$ (a region) to be:

$$\begin{aligned} f^{(a)}(x,y,\xi^{(a)}) = & \sum_{\substack{e=(n_1,n_2)\in\mathcal{E}\\ n_1,n_2\in a}} C_e\, x_e + \sum_{\substack{e=(n_1,n_2)\in\mathcal{E}\\ n_1\in a, n_2\notin a}} \lambda_e\, C_e\, x_e + \sum_{\substack{e=(n_1,n_2)\in\mathcal{E}\\ n_1\notin a, n_2\in a}} (1-\lambda_e)\, C_e\, x_e \\ & + \sum_{n\in\mathcal{F}\cap a} P_n\,(S_n - y_n - \frac{i_n}{2}) + \sum_{n\in\mathcal{B}\cap a} R_n\,(S_n - y_n) \\ = & \sum_{e=(n_1,n_2)\in\mathcal{E}\cap a} C_e\, x_e + \sum_{n\in\mathcal{F}\cap a} P_n\,(S_n - y_n - \frac{i_n}{2}) + \sum_{n\in\mathcal{B}\cap a} R_n\,(S_n - y_n) \\ & \sum_{\substack{e=(n_1,n_2)\in\mathcal{E}\\ n_1\in a, n_2\notin a}} \lambda_e\, C_e\, x_e + \sum_{\substack{e=(n_1,n_2)\in\mathcal{E}\\ n_1\notin a, n_2\in a}} (1-\lambda_e)\, C_e\, x_e \end{aligned}$$

It is therefore possible to split the stochastic scenario in extended scenario for each region.
This objective function for an extended scenario is similar to the one for a scenario but presents the extra term associated with arcs among different regions:

$$\sum_{\substack{e=(n_1,n_2)\in\mathcal{E}\\ n_1\in a, n_2\notin a}} \lambda_e\, C_e\, x_e + \sum_{\substack{e=(n_1,n_2)\in\mathcal{E}\\ n_1\notin a, n_2\in a}} (1-\lambda_e)\, C_e\, x_e$$

For each region the set of nodes in the extended scenario is the subset of the

original node set containing the nodes present in the region. The revenues, production cost and scrap loss percentage are unchanged.
The oriented edges, can be distinguished in two categories, the intra-arcs and the inter-arcs. The intra-arcs are among two nodes in the same region, their capacity and cost remain unchanged. The inter-arcs are our consensus variables, and the cost needs to be chosen so that the sum in the two regions in which it appears is the cost of the original arc, as shown above. The capacity needs to remain the same.

For every stochastic scenario ξ and for every ADMM sub-problem (region) s, the extended scenario $\widetilde{\xi} = \xi^{(s)}$ contains the same data as a usual stochastic scenario, although some arcs may be linked to a node outside the model, which will not be represented.

Figure 2 provides a representation of the two extended scenarios of the previously presented stochastic scenario, in which $\lambda = 1/2$.

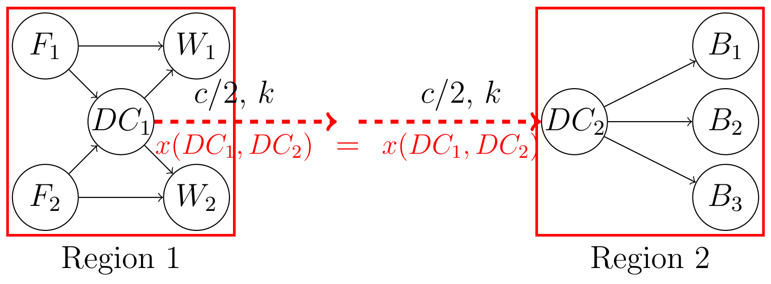

Figure 2: Representation of a stochastic scenario decomposed into extended scenarios

Note about probabilities for variables subject to non anticipitativity

In the distribution example, the consensus variables are inter-region arcs and appear in exactly two regions, i.e. $K_v = 2$. Therefore, if a consensus variable appears in a sub-problem a, i.e. if the inter-region arc has its source or target in the region, the probability of the variable in the extended scenario is $\frac{\Pr_\xi}{K_v} = \frac{\Pr_\xi}{2}$ for every scenario ξ, otherwise its probability is 0.
In the example above with two regions, the inter-region arc is in both region consequently its value is $\frac{\Pr_\xi}{2}$ for every scenario ξ.

3.3 Computational experiments

Importance of introducing slack variable for computation

Certain methods to obtain non-anticipative solutions for the stochastic problem decomposed with ADMM may not work efficiently with the model as presented above.
For instance, a method consists in finding the best solution x such that $x^t = \bar{x}^{(\nu)}(\widetilde{\mathcal{G}}_t(\widetilde{\xi}))$ at every-stage t for every scenario ξ. If there exists such x, then the method provides a non-anticipative solution at every iteration. The solution given by this method converges to the best non-anticipative solution, because PH converges.

However, with our model there might not be any solution x such that $x^t = \bar{x}^{(\nu)}(\widetilde{\mathcal{G}}_t(\widetilde{\xi}))$ at every stage and scenario. In practice, we have seen that such a method may not find any incumbent in the first iterations. The reason is that, when taking separately the extended scenarios, the inter-arcs entering a region tend to reach the capacity upper bound k. Therefore, in the first iterations the flow on inter-region arcs, which is the average of the flow incoming and leaving will be high: $\bar{x} \geq k/2$. However, it may not be feasible for the inter-arc leaving the region to provide a flow greater than $k/2$.

To solve this issue, we added dummy slacks entering the distribution centers with extremely high costs. Thanks to this flow the solution $\bar{x}$ is always feasible, and the dummy slacks quickly converge to 0.

For the same reason, having an inventory allows to always have feasible incumbents, although here after convergence the inventory is always zero. Indeed, it brings a slack variable to the model where the non-anticipative variables for the stochastic scenarios (here the slack at the factory nodes y_f) are defined, which allows a $\bar{y}$ solution to be feasible.

Computational results

Scalable random instances of distribution models were generated to provide a benchmark for varying sizes of each region, numbers of stochastic scenarios, and the number of processors.

Experiments were run on a shared computer with 64 dual-threaded Intel(R) Xeon(R) Gold 6430 cores that run between 800 and 3400 MHz with Gubobi solver's persistent interface. The results are presented in Table 3.

region size	num. scenarios	num. processors	wall clock (sec.)	iterations	relative gap (%)
8	16	3	23.09	57	0.965
8	16	12	7.42	58	0.907
8	16	36	4.18	58	0.907
8	256	3	475.54	58	0.960
8	256	12	102.56	59	0.833
8	256	36	34.98	59	0.833
128	16	3	156.98	29	0.964
128	16	12	42.96	30	0.780
128	16	36	20.25	30	0.780
128	256	3	2845.59	30	0.809
128	256	12	583.44	30	0.809
128	256	36	232.96	30	0.809

Figure 3: Computational results for the distribution example. We used stochastic admm package in mpi-sppy [4] with the "xbar" upper bounder, the "lagrangian" lower bounder, and termination at a gap below 1%.

Based on trace output not shown here, improvements are probably possible using the "ph-ob" lower bounder instead of "lagrangian" and perhaps using the gradient based rho setter.

4 Conclusions

Decomposition using consensus ADMM can be used to allow parallelization efficiencies or for reasons related to information security (see, e.g., [5]). In either case, the input data may be uncertain and we have given a decomposition algorithm based on Progressive Hedging. It would be possible to base the algorithm on Asynchronous Projective Hedging [3] instead of PH using similar ideas.

References

[1] D. P. Bertsekas. "Multiplier methods: A survey". In: *Automatica* 12.2 (1976), pp. 133–145. ISSN: 0005-1098. DOI: `https://doi.org/10.1016/0005-1098(76)90077-7`. URL: `https://www.sciencedirect.com/science/article/pii/0005109876900777`.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122. ISSN: 1935-8237. DOI: `10.1561/2200000016`. URL: `http://dx.doi.org/10.1561/2200000016`.

[3] J. Eckstein, J.-P. Watson, and D. L. Woodruff. "Projective Hedging Algorithms for Multistage Stochastic Programming, Supporting Distributed and Asynchronous Implementation". In: *Operations Research* to appear (2023). URL: `https://doi.org/10.1287/opre.2022.0228`.

[4] B. Knueven, D. Mildebrath, C. Muir, J. D. Siirola, J.-P. Watson, and D. L. Woodruff. "A Parallel Hub-and-Spoke System for Large-Scale Scenario-Based Optimization Under Uncertainty". In: *Math. Prog. Comp.* 15 (2023), pp. 591–619.

[5] M. Ma, L. Fan, and Z. Miao. "Consensus ADMM and Proximal ADMM for economic dispatch and AC OPF with SOCP relaxation". In: *2016 North American Power Symposium (NAPS)*. 2016, pp. 1–6. DOI: `10.1109/NAPS.2016.7747961`.

[6] R. T. Rockafellar and R. J.-B. Wets. "Scenarios and Policy Aggregation in Optimization Under Uncertainty". In: 16 (1991), pp. 119–147. ISSN: 0364-765X. DOI: `10.1287/moor.16.1.119`.

[7] J. Watson and D. L. Woodruff. "Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems". In: *Comput. Manag. Sci.* 8.4 (2011), pp. 355–370. DOI: `10.1007/s10287-010-0125-4`.