## **Supplementary Materials**

## **A.3 Solution method: The steps of proposed LR for each iteration of AUGCON**

## **A.3.1 Generating the lower bound of LR**

We decompose the model into two sub-problems. The first sub-problem is defined as follows:

(Sub1)

|  |  |
| --- | --- |
|  | (A-42) |
| Subject to: |  |
|  | (A-43) |
| (3) - (5)  (7) - (14)  (24) - (25)  (A-22) - (A-25) |  |
|  | (18) |
|  | (19) |
|  | (20) |

Note that certain variables are not determined during this phase: , , , thus the constraints including those variables are not included in sub-problem 1. Moreover, the term is excluded from the cost objective in this phase, and will be computed in the repair phase as a residual. Similarly, for the risk objective function, the term will be calculated during the repair phase as a residual and will be added to the risk objective at that phase.

To improve the quality and efficiency of the lower bound, we introduce a cut (Eq (26) in the main document) to the x-related model.

## **A.3.2 Repairing the lower bound of LR: the residual values**

Using the optimal values of and from previous phase, a feasible value for is calculated. Subsequently, the values of , , are determined based on the values of , , and and using Eqs (15)-(17). This process is carried out without running an optimization model since the required variables are already determined. Following this, the two terms and are computed and their calculated values are added to the relevant objective functions as residual values.

## **A.3.3 Generating the upper bound**

To generate a feasible upper bound, the outputs of Sub1 model are utilized meaning that all x-related variables are invoked after solving Sub1. The value of, and other variables of Sub1 are used as input to run the following model (FUB):

(FUB)

|  |  |
| --- | --- |
|  | (A-44) |
| Subject to:  (6)  (23) ­– (25)  (A-15.2) – (A-16.2)  (A-26) – (A-41) |  |
|  | (18) |
|  | (19) |
|  | (21) |

## **A.3.4 Subgradient optimization**

At each iteration, the Lagrangian multipliers are updated. A common approach to do so is to utilizing a subgradient optimization. In this method, subgradient direction is obtained by minimizing the dual function. Validation and theoretical convergence properties of this method can be found in Held et al. (1974). For the dualized constraint (Constraint set (6)), a penalty term (A-45) is added to the objective function, as the subgradient of the dual function:

|  |  |
| --- | --- |
|  | (A-45) |

Note that the penalty value is calculated at each iteration using the values of , and in that iteration. The Lagrange multiplier (at iteration *itrL*) is then updated as follows:

|  |  |
| --- | --- |
|  | (A-46) |

Where *stepsizeitrL*, step size in iteration *itrL*, is calculated as:

|  |  |
| --- | --- |
|  | (A-47) |

and denote the best upper and lower bounds, respectively, for the original model (MIP), obtained via the LR algorithm. is a control parameter, where

If the algorithm fails to improve the lower bound for a predefined number of consecutive iterations, is halved. The algorithm is terminated when falls below a predetermined value. In addition to a predefined number of iterations and CPU runtime, the convergence is used as another stopping criterion, calculated as follows:

|  |  |
| --- | --- |
| *Convergence* *gap* = | (A-48) |

Where *ε* is a small value, and the convergence gap is compared against it. The algorithm terminates if the convergence gap is less than or equal to *ε*. The steps of this heuristic algorithm are outlined as a simplified pseudo code in Figure 2, in the main document.

**Reference**

Held, M., Wolfe, P., & Crowder, H. P. (1974). Validation of subgradient optimization. *Mathematical Programming, 6*(1), 62-88. doi:10.1007/bf01580223