Auto-repair infeasible optimization models

- Approach 1: Irreducible Infeasible Subset (IIS)

- Approach 2 : Elastic Programming (EP)

	IIS	EP
PROS	Knowing the cause, repair with minimal changes	Facilitate automatic fix on bounds
CONS	Need to either drop or penalize constraints	Hard to explain, unwanted changes

- Proposed Approach: Hybrid of IIS and EP
  - Infeasibility explanation
  - Minimal changes to the original model

Irreducible Infeasibility Set

- Deletion Algorithm
  - Find one IIS each run
  - Fast for LP due to warm start
  - Slow for Quadratic and Integer Models

Example

IIS is {B,D} in {A,B,C,D}
- Delete A: {B,C,D} infeasible
- Delete B: {C,D} feasible
- Put back B and delete C: {B,D} infeasible, Stop
- Output: {B,D}

- Additive Algorithm :
  - Performance similar to Deletion Algorithm

Example

IIS is {B,D} in {A,B,C,D}
- {A}, {A,B}, {A,B,C} all feasible
- {A,B,C,D} infeasible, check {D}
- {D, A}, {D, A, B} infeasible, check {D,B}
- {D,B} infeasible, Stop
- Output: {B,D}

- Farkas Lemma
  - Used in GUROBI
  - Complicated Implementation but faster for LP
  - Slow for Quadratic and Integer Models
  - How to use Farkas' lemma to say something important about linear infeasible problems by Prof. Andersen, E.D

If Ax <= b infeasible, support of the model with dual variable y gives an IIS:
- A'y = 0,
- b'y <=-1
- y >= 0

Elastic Programming

$$\underset{x}{Maximize}\ \ c^T x$$

$$Subject\ to:\ AX \leq b$$

Original Model

```
Maximize 0
Subject To
    c1: x + y <= 0.5
    c2: x >= 1
Bounds
    y >= 0
```

$$\underset{e}{Minimize} \sum e$$

$$Subject\ to:\quad AX \leq b + e$$
$$e \geq 0$$

Elastic Model

```
Minimize e1 + e2
Subject To
    c1: x + y <= 0.5 + e1
    c2: -x <= -1 + e2
    e1, e2 >= 0
Bounds
    y >= 0
```

Solution

```
e1=0.5
e2=0
```

$$\underset{x}{Maximize}\ \ c^T x$$

$$Subject\ to:\ AX \leq b + e$$

Relaxed Model

```
Maximize 0
Subject To
    c1: x + y <= 0.5 + 0.5
    c2: x >= 1 - 0
Bounds
    y >= 0
```

Proposed Hybrid IIS/EP Algorithm

- Input: Infeasible Model, Priority for each constraint
- Step 1: Break all IISs
  - 1.1 Initialize the list of constraints to relax to empty: cons_list = []
  - 1.2 Compute IIS
  - 1.3 Break this IIS by removing the **least important** constraint, and add it to cons_list
  - 1.4 Check feasibility of the new model
    - If feasible, go to step 2
    - Else go to 1.2
- Step 2: Modify the constraints in cons_list
  - 2.1 Create an EP model by adding elastic variable to each constraint in cons_list
  - 2.2 Solve the EP model and get the optimal value of the elastic variables
  - 2.3 Modify bound of the constraints using the optimal value of elastic variables
- Output: Original Infeasible Model, Fixed Feasible Model, Changed Bounds, IISs

Example and Demo: beta_A1121_Global_20230725.lp

- Linear Programming Infeasible Model:
  - 26811 variables
  - 8945 constraints
- IIS Approach:
  - GUROBI found the first IIS of size 2: R0 and R3
  - Remove R0, and model is still infeasible
  - Re-compute IIS, and get another IIS of size 3: R3, R5 and R6
  - Remove R3, and model becomes feasible
- EP Approach:
  - Create an EP model by allowing bounds being changed for all constraints
  - Solve the EP model and get an feasible model with hundreds of changed bounds
- Hybrid Approach:
  - Create an EP model by allowing only R0 and R3 being changed
  - Solve the EP model and get an feasible model with only two modified bounds