Chance Constraint Learning: A Hybrid Approach

Madeline Colbert[1], Shunyu Yao[1], Justin Dumouchelle[2], Beste Basciftci[1], Thiago Serra[1]

[1]Department of Business Analytics, University of Iowa, [2] Department of Mathematics & Statistics, University of Calgary

Background

Chance-Constrained optimization problems are often **computationally intractable** to solve.

A $MIP_{CCO} = \min\{c^{\top}x : x \in \mathcal{X}, \mathbb{P}(\tilde{A}x \leq \tilde{b}) \geq 1 - \varepsilon\}$

SAA is a **good estimate,** but **prohibitively slow** and **overfits** on smaller scenario sets. Define S as the set of indices of the scenarios (i.e. $S = [n]$).

B $MIP_{SAA}(\mathcal{S}) = \min\{c^{\top}x : x \in \mathcal{X},$
$y \in \{0,1\}^{|\mathcal{S}|}, A^i x \leq b^i + M_i(1 - y_i)\forall i \in \mathcal{S}, \Sigma_{i \in \mathcal{S}}(1 - y_i) \leq n\boldsymbol{\alpha_{SAA}}\}$

Chance Constraint Learning

Instead of using the scenarios in a MIP, use them to train a **neural network!**

Neural networks present a **fast, flexible** framework. However, there are **no guarantees** on performance

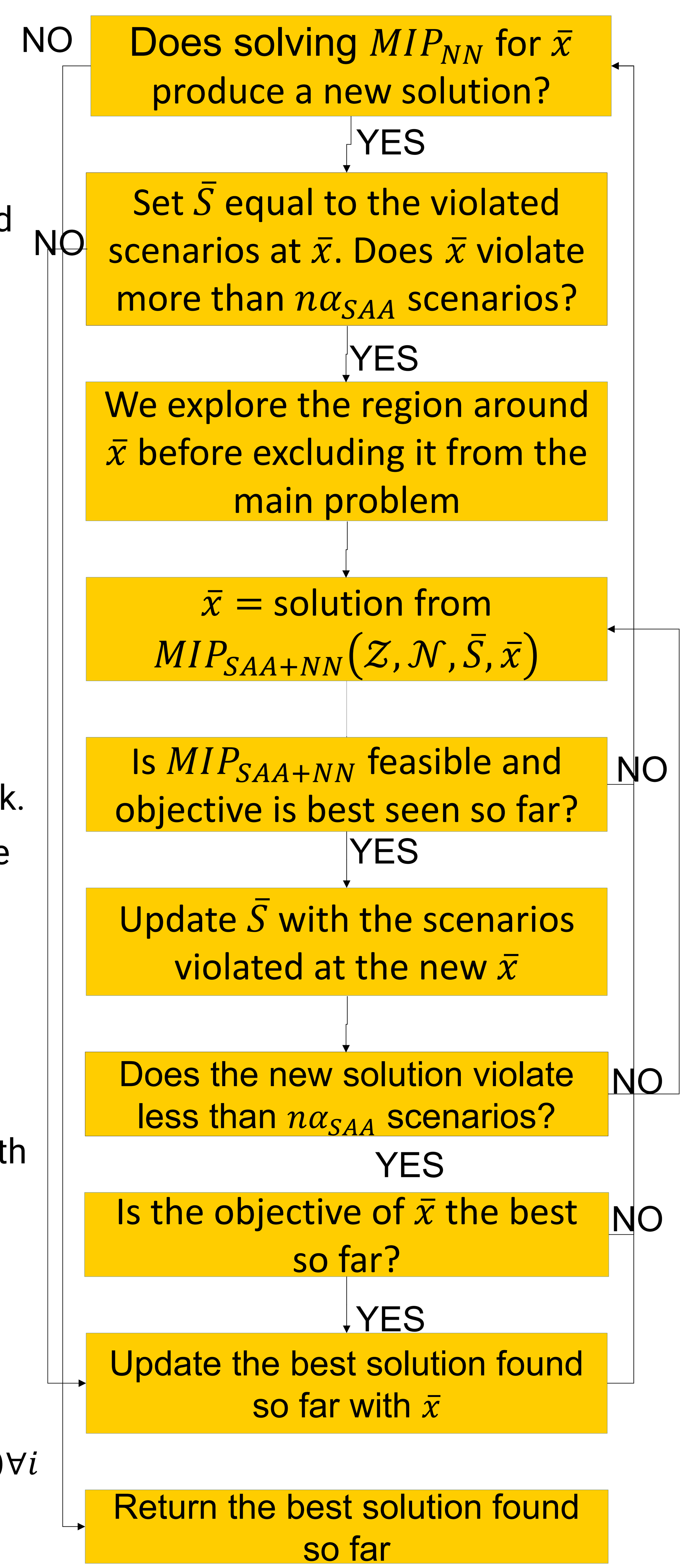
C $MIP_{NN}(\mathcal{N}, \mathcal{Z}) = \min\{c^{\top}x : x \in \mathcal{X},$
$\mathcal{N}(x) \leq \boldsymbol{\alpha_{NN}}, \mathcal{Z}(\mathrm{x}) \in \{0,1\}^{\Theta}\}$

The Hybrid Approach

Combine the **generalization** and **speed** of NN with **confidence** and **feasibility** of SAA by solving **subproblems** within different regions of the network based on violated scenarios, $\bar{S}$

D $MIP_{NN+SAA}(\mathcal{Z}, \mathcal{N}, \bar{x}, \bar{\mathcal{S}}) = \min\{c^{\top}x : x \in \mathcal{X},$
$\mathcal{N}(x) \leq \boldsymbol{\alpha_{NN}}, \mathcal{Z}(x) = \mathcal{Z}(\bar{x}), A^i x \leq b^i + M_i(1 - y_i)\forall i \in \bar{\mathcal{S}}, \Sigma_{i \in \bar{\mathcal{S}}}(1 - y_i) \leq n\boldsymbol{\alpha_{SAA}}\}$

Hybrid Approach Algorithm

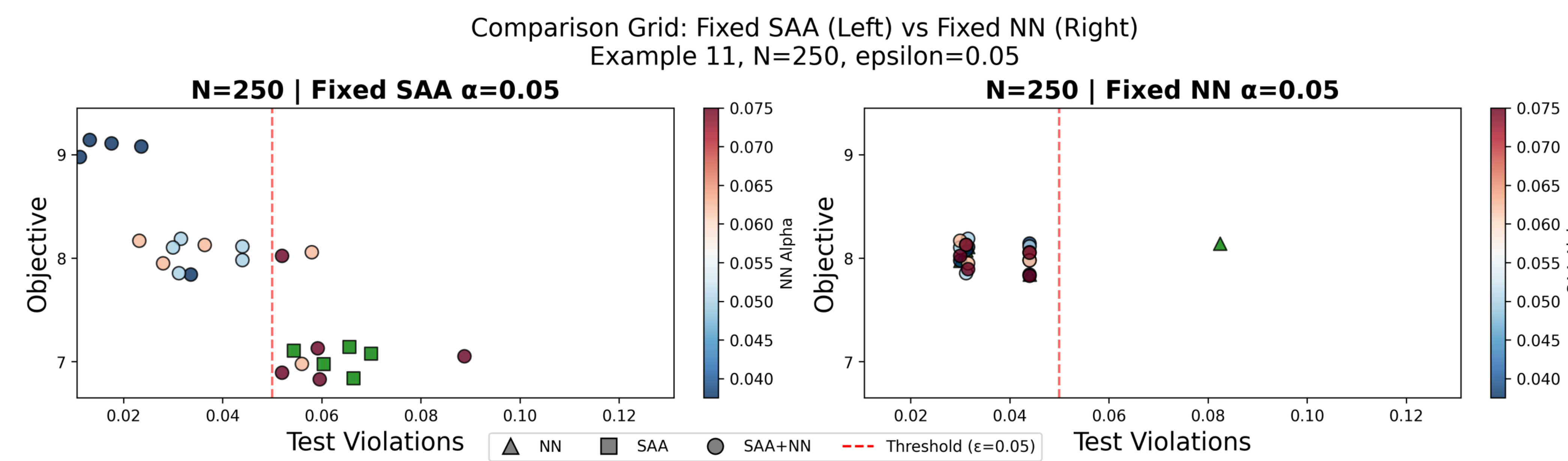


Experiments and Results

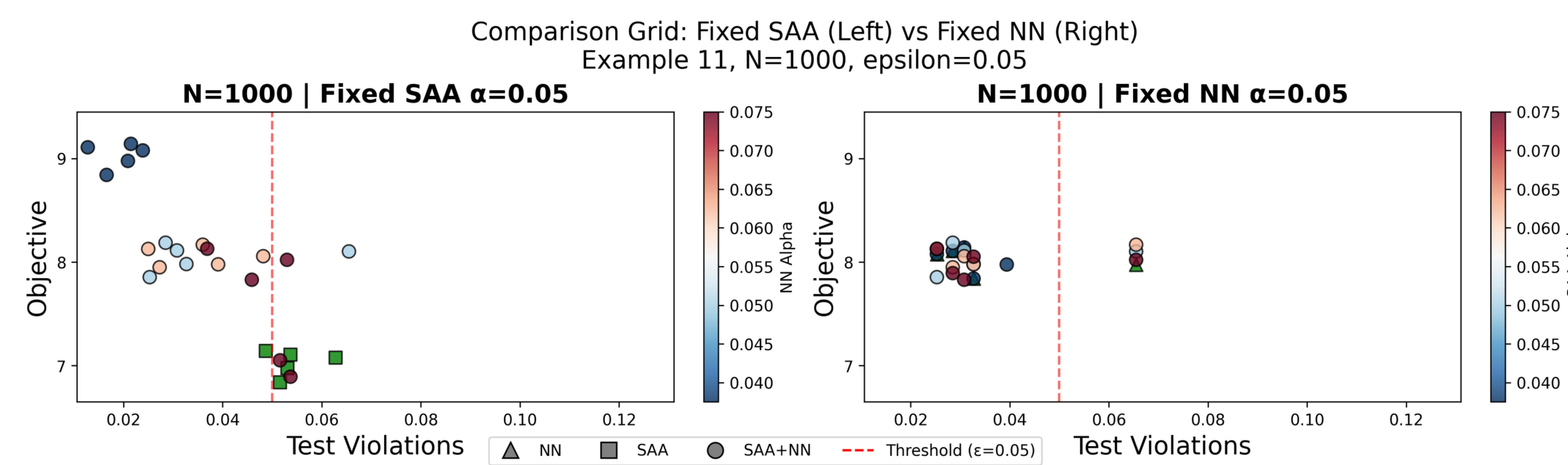
We ran experiments on a facility location problem with a single joint chance constraint, 205 demand points, and minimizing 16 potential facility locations while ensuring all have access in 95% of cases.

We track different solutions violations on an out-of-sample violation set that is 10 times larger than what we solve over. We run 5 different seeds for each parameter combination to get a broader scope.

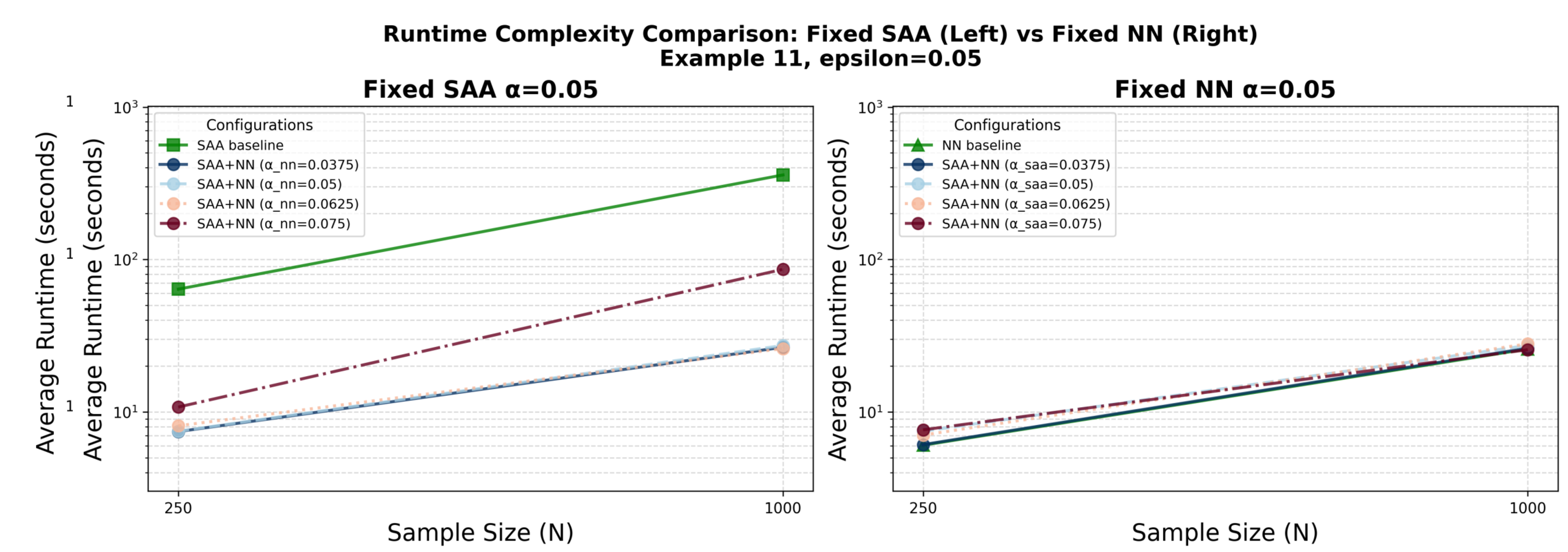
The hybrid approach is frequently **faster** than SAA, while producing more **stable** solutions, that are **similar in objective** to SAA. We show how it performs when fixing α_{SAA} or α_{NN} equal to ε while varying the other below; however, more work is being done to identify the relationship between them.



The Hybrid Approach corrects the neural network's outliers, and appears more stable than SAA



Increasing the number of scenarios used shows a similar pattern, but with slightly more stability, which is known from previous works on SAA



Our method is better than SAA in runtime, and comparable to the constraint learning formulation