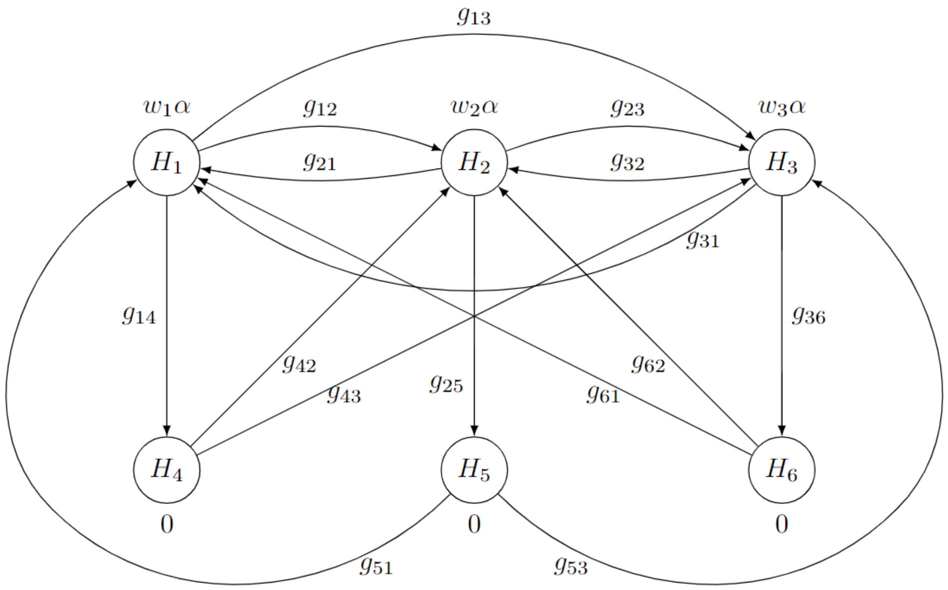
Graph Optimization

In this markdown, we demonstrate the motivating example in section 5 of our manuscript and how to implement with the function we provided. We first load the functions that are needed for the optimization.

```
source('src/optimization/w_optimization.R')
source('src/optimization/graph_optimization.R')
```

Set up



Six hypotheses are tested on 3 doses against placebo on primary and secondary endpoints (see figure above). The graphical template is defined in the graph above. Sample sizes are set to be the same for all doses and placebo as 200. The FWER is set to be controlled at 0.025 level on one-sided hypotheses. The marginal powers for H_i, $i = 1, ..., 6$ are 0.8, 0.8, 0.8, 0.9, 0.8, 0.71.

```
alpha = 0.025  # FWER
mp = c(0.8, 0.8, 0.8, 0.9, 0.8, 0.71)  # marginal power
n0 = 200  # sample size for placebo
n1 = 200  # sample size for dose 1
n2 = 200  # sample size for dose 2
n3 = 200  # sample size for dose 3
```

The correlations between test statistics for the same endpoint is determined by the sample size. The correlation between test statistics for primary hypothesis and secondary hypothesis are set to be 0.5.

```
# correlation between test statistics for the same endpoint
rho12 = sqrt(n1/(n1+n0) * n2/(n2+n0))
rho13 = sqrt(n1/(n1+n0) * n3/(n3+n0))
rho23 = sqrt(n2/(n2+n0) * n3/(n3+n0))
```

```r
A = matrix(c(1, rho12, rho13,
             rho12, 1, rho23,
             rho13, rho23, 1), nrow=3)

# Incorporate correlation between primary and secondary hypothesis
rho = 0.5

Sigma = rbind(cbind(A, A*rho), cbind(A*rho, A))
```

Optimizing hypothesis weights with constraints

In the first step, we optimize the disjunctive power of all hypotheses with following constraints:

- The initial hypothesis weights for secondary hypotheses should be 0.

To calculate the optimal weights of disjunctive power for hypotheses with correlation, we recommend using `go_optim_w_dp` function with basic argument inputs of `alpha`(FWER), `mp`(marginal powers) and `rho`(correlation matrix or constant for same correlation for all pairs). These have been defined in the set-up section.

If the weights are pre-defined, i.e. the weights should be some given constants. We set the corresponding position of constraint vector to be the numbers given. Otherwise, set NA as no constraints on current hypotheses.

To set up the constraints in the case, we need to set the argument `constraint.w` in function `go_optim_w_dp` to be `c(NA, NA, NA, 0, 0, 0)`, which represents that there is no constraint on hypotheses 1, 2 and 3, but the weights for hypotheses 4, 5 and 6 need to be 0.

```r
# constraints for initial hypotheses weights
constraint.w = c(NA, NA, NA,  # no constraints for primary hypotheses
                 0, 0, 0)  # 0 weights for all secondary hypotheses

# optimizing with multiple initial starting points
res = go_optim_w_dp(alpha, mp, rho=Sigma, constraint.w=constraint.w)

w.optim = res[1, ] %>%  # Select the best optima from multiple outputs
  select(starts_with("w")) %>%
  as.numeric() %>% round(4)
```

The optimal hypothesis weighs with constraints are 0.333, 0.333, 0.333, 0, 0, 0.

Optimizing transition matrix with constraints

In the second step, we optimize the transition matrix to optimize the second step disjunctive power if one hypothesis is rejected. We need to use

According to the study set up, we need the following constraints:

- The transition weight from a primary hypothesis to its descendant secondary hypothesis for the same dose-control comparison should be positive but the weight to other secondary hypotheses should be 0;
- The transition weight from a secondary hypothesis to another secondary hypothesis should be 0, because this may lead to the rejection of the latter secondary hypothesis without rejecting its corresponding primary hypothesis;
- The transition weight from a secondary hypothesis to primary hypotheses should be positive. But given the second bullet point, the edge from a secondary hypothesis to its corresponding primary hypothesis for the same dose-control comparison may be redundant. For a simpler graph, we set this transition weight to 0.

Similarly, to set up the constraints, we need to pass the argument `constraint.G` to function `optim_G_dp` with the pre-defined constraints on the graph. 0 will be set to the paths that do not exist and NA will be assigned to the paths that do not have pre-defined weights. The upper and lower bound based on previous weight assignment for the second step optimization will be captured automatically.

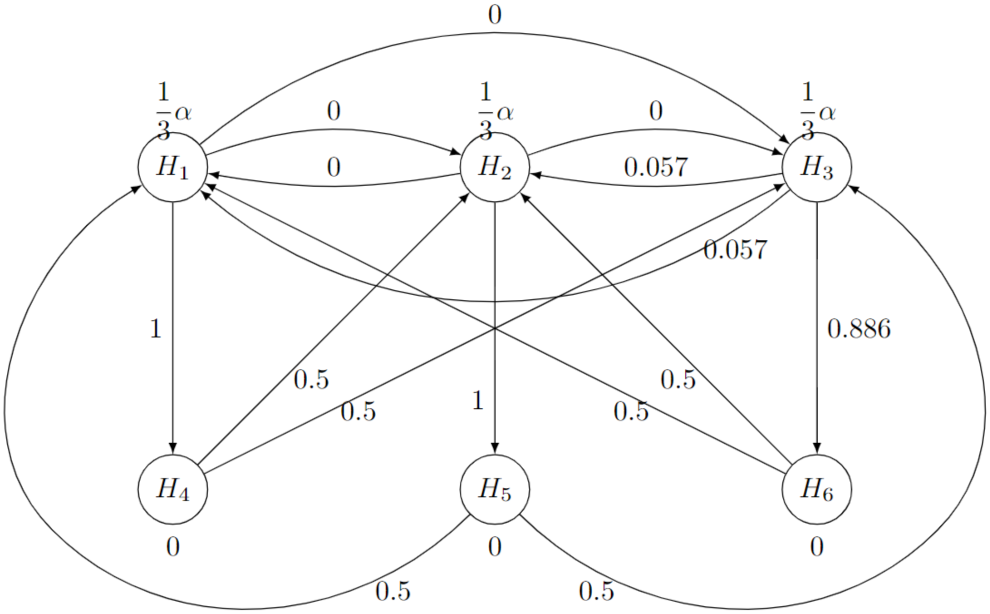
```r
# Set up constraints
constraint.G = matrix(c(0, NA, NA, NA, 0, 0,
                        NA, 0, NA, 0, NA, 0,
                        NA, NA, 0, 0, 0, NA,
                        0, NA, NA, 0, 0, 0,
                        NA, 0, NA, 0, 0, 0,
                        NA, NA, 0, 0, 0, 0),
                      nrow=6, byrow=T)

# Optimization of graph
G.optim <- optim_G_dp(alpha=alpha, w=w.optim, mp=mp, rho=Sigma,
                      constraint.G=constraint.G)
```

Optimal hypothesis weights w's are 0.333, 0.333, 0.333, 0, 0, 0, the optimal transition matrix G is

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0.057 & 0.057 & 0 & 0 & 0 & 0.886 \\ 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and the optimal graph is displayed as


0
$\frac{1}{3}\alpha$
$\frac{1}{3}\alpha$
$\frac{1}{3}\alpha$
0
0
0
0.057
H_1
H_2
H_3
0.057
1
0.886
0.5
0.5
1
0.5
0.5
H_4
H_5
H_6
0
0
0
0.5
0.5