Suppose we have n observations $x_1, \ldots, x_n$, and L null distributions $F_1, \ldots, F_L$. For x_i we want to test the composite null

$$H_i : x_i \sim \text{ one of } F_s,$$

with *no* information on the alternatives or the mixture of the distributions.

For simplicity, assume the p-value of x_i under F_s is $p_{i,s} = F_s(x_i)$ and the observations are already sorted in increasing order, so that $p_{1,s} \le p_{2,s} \le \ldots \le p_{n,s}$. Constrained p-values to test the composite nulls in the above setup can be constructed in many ways via maximization under linear constraints, for example,

$$\begin{aligned} p_{i,\text{glb}} = \sup\Bigg\{ \sum_{s=1}^{L} c_s p_{i,s} \;\Bigg|\; & C \le \sum_{s=1}^{L} c_s \le 1,\ c_s \ge 0,\ \sum_{s=1}^{L} c_s p_{j,s} \le u_j \text{ for } j \ge 1, \\ & \mathbb{F}(t_2) - \mathbb{F}(t_1) + \epsilon_n \ge \sum_{s=1}^{L} c_s [F_s(t_2) - F_s(t_1)],\ \text{for all } t_1, t_2 \in T \\ & \text{with } t_1 < t_2 \Bigg\}, \end{aligned} \tag{1}$$

$$\begin{aligned} p_{i,\text{seq}} = \sup\Bigg\{ \sum_{s=1}^{L} c_s p_{i,s} \;\Bigg|\; & C \le \sum_{s=1}^{L} c_s \le 1,\ c_s \ge 0,\ \sum_{s=1}^{L} c_s p_{j,s} \le u_j \text{ for } j \ge i, \\ & \mathbb{F}(t_2) - \mathbb{F}(t_1) + \epsilon_n \ge \sum_{s=1}^{L} c_s [F_s(t_2) - F_s(t_1)],\ \text{for all } t_1, t_2 \in T \\ & \text{with } x_i \le t_1 < t_2 \Bigg\}. \end{aligned} \tag{2}$$

In the above formulas,

- $C \ge 0$ is a lower bound on the total population fraction of true nulls;
- u_j is defined as

$$u_i = \begin{cases} \bar{\Gamma}^* \left(1/n; i\right) / (\beta n), & \text{if } i \le m_n, \\ i/n + \epsilon_n & \text{otherwise}, \end{cases} \tag{3}$$

  where $\bar{\Gamma}^*(1/n; i)$ is the $100(1 - 1/n)$-th quantile of Gamma$(i, 1)$, $\beta \in (0, 1)$ a constant, and $m_n \ll n$;
- $\mathbb{F}$ is the empirical distribution $\mathbb{F}(t) = \#\{i \,|\, x_i \le t\}/n$;
- ϵ_n is a margin of error; and
- T is a set of values sorted in increasing order, or "check points", that are different from $x_1, \ldots, x_n$;

The constrained p-values of $x_1, \ldots, x_n$ in (1) can be computed by

```
p=lp.global.pval(n.obs, n.mix, obs.pvals,
                 grid.pvals=c(), grid.cumcnt=c(),
                 subset=seq(1,n.obs), coeff.sum.lbd=0.0)
```

and those in (2) by

```
p=lp.sequen.pval(n.obs, n.mix, obs.pvals,
                 grid.pvals=c(), grid.cumcnt=c(),
                 subset=seq(1,n.obs), coeff.sum.lbd=0.0)
```

where

- `n.obs` stores n, `n.mix` stores L;
- `obs.pvals` is an $n \times L$ matrix storing $p_{i,s}$, $i = 1, \ldots, n$, $s = 1, \ldots, L$;
- if T contains k values $t_1 < t_2 < \ldots < t_k$, then `grid.pvals` is a $k \times L$ matrix storing $F_s(t_i)$, $i = 1, \ldots, k$, $s = 1, \ldots, L$, and `grid.cumcnt` is a vector storing $\#\{j \,|\, x_j \leq t_i\}$ for $i = 1, \ldots, k$;
- `coeff.sum.lbd` stores the lower bound C on $\sum c_s$ in (1) and (2);
- `subset` specifies the indices of x_i for which the constrained p-values are to be computed; and
- the output `p` has two objects:
  - `lp.pval` returns the constrained p-values of $x_1, \ldots, x_n$; and
  - `coefficients` returns the primal coefficients obtained from the linear programming.

Note. 1. For both routines, the observations and check points should be sorted in increasing order beforehand. On the other hand, once `obs.pvals`, `grid.pvals` and `grid.cumcnt` are computed, there is no direct need for the values of observations or check points.

2. ϵ_n and u_i, $i = 1, \ldots, n$ in formula (3) are evaluated by

```
slack(n.obs)
```

and

```
cdf.upper.bound(i, n.obs).
```

They are called within `lp.global.pval` and `lp.sequen.pval`.