Calculation of processing and overprocessing

$$\widetilde{W_\sigma} = \sum_{i=1}^{N}(P_i^r \cdot \prod_{k=1}^{i-1}(1-P_k^r) \cdot (\sum_{j=1}^{i} E_j)) \tag{1}$$

1 For Bondora log

```
filenames <- list.files()[grep(paste("^output_bondora(?=.*\\_under.csv)",sep=''), list.files(), perl=TRUE)]
dat = c()
result = matrix(0,nrow = length(filenames),ncol = 6)
colnames(result) = c("nr_checks_our","nr_checks_Wil","nr_checks_rand",
                     "overproc_our","overproc_Wil","overproc_rand")

for (i in 1:length(filenames)) {
  foo = read.table(filenames[i],sep=",",header=TRUE)
  dat = rbind(dat,foo)
  result[i,1] = mean(foo$nr_checks_our)
  result[i,2] = mean(foo$nr_checks_Wil)
  result[i,3] = mean(foo$nr_checks_rand)
  result[i,4] = mean(foo$nr_checks_our - foo$minimum_check_number)
  result[i,5] = mean(foo$nr_checks_Wil - foo$minimum_check_number)
  result[i,6] = mean(foo$nr_checks_rand - foo$minimum_check_number)
}
```

Average number of checks that one would do if they follow **our ordering**:

```
round(mean(result[,1]),digits = 4)

## [1] 2.794
```

Average number of checks that one would do if they apply **Wil's method** (constant reject probabilities):

```
round(mean(result[,2]),digits = 4)

## [1] 2.801
```

On average we are doing **0.251** % less checks than Wil.

Average number of checks that one would do if for every case they do checks in **random order**

```
round(mean(result[,3]),digits = 4)

## [1] 2.8475
```

Average **overprocessing** - our method

```
round(mean(result[,4]),digits = 4)

## [1] 0.0978
```

Average **overprocessing** - Wil method

```
round(mean(result[,5]),digits = 4)

## [1] 0.1048
```

On average our overprocessing is **7.158** % less than Wil.

Average **overprocessing** - random ordering

```
round(mean(result[,6]),digits = 4)

## [1] 0.1512
```

```
tt = matrix(0,nrow = 3,ncol = length(unique(dat$nr_checks_rand)))
rownames(tt) = c("count_checks_our","count_checks_Wil","count_checks_rand")
tt[1,] = round(table(dat$nr_checks_our)/1)
tt[2,] = round(table(dat$nr_checks_Wil)/1)
tt[3,] = round(table(dat$nr_checks_rand)/1)
colnames(tt) = names(table(dat$nr_checks_rand))
print(tt)

##                       1    2     3
## count_checks_our   4523 7455 68142
## count_checks_Wil   6208 3524 70388
## count_checks_rand 4114 3992 72014
```

Distribution of overprocessing

2 For Environmental permit log

```
filenames <- list.files()[grep(paste("^output_envpermit(?=.*\\_under.csv)",sep=''), list.files(), perl=TRUE
dat = c()
result = matrix(0,nrow = length(filenames),ncol = 6)
colnames(result) = c("nr_checks_our","nr_checks_Wil","nr_checks_rand",
                     "overproc_our","overproc_Wil","overproc_rand")

for (i in 1:length(filenames)) {
  foo = read.table(filenames[i],sep=",",header=TRUE)
  dat = rbind(dat,foo)
  result[i,1] = mean(foo$nr_checks_our)
  result[i,2] = mean(foo$nr_checks_Wil)
  result[i,3] = mean(foo$nr_checks_rand)
  result[i,4] = mean(foo$nr_checks_our - foo$minimum_check_number)
  result[i,5] = mean(foo$nr_checks_Wil - foo$minimum_check_number)
  result[i,6] = mean(foo$nr_checks_rand - foo$minimum_check_number)
}
```

Average number of checks that one would do if they follow **our ordering**:

```
round(mean(result[,1]),digits = 4)

## [1] 2.3248
```

Average number of checks that one would do if they apply **Wil's method** (constant reject probabilities):

```
round(mean(result[,2]),digits = 4)
```

```
## [1] 2.324
```

On average we are doing -0.035 % less checks than Wil.

Average number of checks that one would do if for every case they do checks in **random order**

```
round(mean(result[,3]),digits = 4)
```

```
## [1] 2.6602
```

Average **overprocessing** - our method

```
round(mean(result[,4]),digits = 4)
```

```
## [1] 0.6654
```

Average **overprocessing** - Wil method

```
round(mean(result[,5]),digits = 4)
```

```
## [1] 0.6646
```

On average our overprocessing is -0.122 % less than Wil.

Average **overprocessing** - random ordering

```
round(mean(result[,6]),digits = 4)
```

```
## [1] 1.0008
```

```
tt = matrix(0,nrow = 3,ncol = length(unique(dat$nr_checks_rand)))
rownames(tt) = c("count_checks_our","count_checks_Wil","count_checks_rand")
tt[1,] = round(table(dat$nr_checks_our)/1)
tt[2,] = round(table(dat$nr_checks_Wil)/1)
tt[3,] = round(table(dat$nr_checks_rand)/1)
colnames(tt) = names(table(dat$nr_checks_rand))
print(tt)
```

```
##                     1    2    3
## count_checks_our   29 1603  828
## count_checks_Wil   27 1609  824
## count_checks_rand 16  804 1640
```

Distribution of overprocessing