

# Calculation of processing and overprocessing

Ilya Verenich

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

## 1 For Bondora log

Minimal possible number of checks, averaged over 11 test sets :

```
round(sum(dat$minimum_check_number)/length(filenamees))  
## [1] 21563
```

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

```
round(sum(dat$nr_checks_our)/length(filenamees))  
## [1] 21828
```

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

```
round(sum(dat$nr_checks_Wil)/length(filenamees))  
## [1] 22393
```

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

```
round(sum(dat$nr_checks_rand)/length(filenamees))  
## [1] 22800
```

Average overprocessing (in %) with our ordering:

```
round(100*(sum(dat$nr_checks_our) - sum(dat$minimum_check_number))/(length(filenamees)*sum(dat$minimum_check_number)))  
## [1] 1.23
```

Average overprocessing with Wil ordering:

```
round(100*(sum(dat$nr_checks_Wil) - sum(dat$minimum_check_number))/(length(filenamees)*sum(dat$minimum_check_number)))  
## [1] 3.85
```

Average overprocessing with random ordering:

```
round(100*(sum(dat$nr_checks_rand) - sum(dat$minimum_check_number))/(length(filenamees)*sum(dat$minimum_check_number)))  
## [1] 5.74
```

**Distribution of the number of checks performed**

```
print(tt)

##           1    2    3
## count_checks_our  974 261 6777
## count_checks_Wil  642 359 7011
## count_checks_rand 413 410 7189
## minimal          1237   0 6775
```

## 2 For Environmental permit log

Minimal possible number of checks, averaged over 12 test sets :

```
round(sum(dat$minimum_check_number)/length(filenamees))

## [1] 416
```

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

```
round(sum(dat$nr_checks_our)/length(filenamees))

## [1] 577
```

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

```
round(sum(dat$nr_checks_Wil)/length(filenamees))

## [1] 576
```

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

```
round(sum(dat$nr_checks_rand)/length(filenamees))

## [1] 657
```

Average overprocessing (in %) with our ordering:

```
round(100*(sum(dat$nr_checks_our) - sum(dat$minimum_check_number))/(length(filenamees)*sum(dat$minimum_check_number)))

## [1] 38.89
```

Average overprocessing with Wil ordering:

```
round(100*(sum(dat$nr_checks_Wil) - sum(dat$minimum_check_number))/(length(filenamees)*sum(dat$minimum_check_number)))

## [1] 38.49
```

Average overprocessing with random ordering:

```
round(100*(sum(dat$nr_checks_rand) - sum(dat$minimum_check_number))/(length(filenamees)*sum(dat$minimum_check_number)))

## [1] 58.16
```

### Distribution of the number of checks performed

```
print(tt)

##           1    2    3
## count_checks_our  2.6000 155.5  87.90000
## count_checks_Wil  2.5000 157.3  86.20000
## count_checks_rand  1.5000  77.6 166.90000
## minimal          161.1667   0.0  84.83333
```