## Mutation

Mutation er til for at lave små ændringer i skemaet. Den skal kunne sørge for at der er mangfoldighed, som sørger for at skemaet ikke ender i et lokalt maksimum.

Måden hvorpå dette er lavet er ved at tage et individ. Finder to helt tilfældige timer på hele skemaet, og bytter disse ud. Der kommer her to variabler, der kan finpudses for at finde den bedste løsning. Der er chancen for at en mutation kan ske, og antal mutationer der maksimalt kan ske pr individ.

void mutation(individual individuals[]){

int i = 0, j = 0, ran1Day = 0, ran1Week = 0, ran2Day = 0, ran2Week = 0, chance = 0, mutations = 0, temp = 0;

for(i = 0; i < NUMBER\_OF\_INDIVIDUALS; i++){

chance = rand()% 100;

mutations = rand()% MAX\_MUTATIONS\_PER\_INDIVIDUAL;

for (j = 0; j < mutations; j++){

if (chance > CHANCE\_OF\_MUTATION){

do {

ran1Week = rand()% SCHOOL\_DAYS\_IN\_WEEK;

ran1Day = rand()% LESSONS\_PER\_DAY\_MAX;

ran2Week = rand()% SCHOOL\_DAYS\_IN\_WEEK;

ran2Day = rand()% LESSONS\_PER\_DAY\_MAX;

} while ((ran1Week == ran2Week) && (ran1Day == ran2Day));

temp = individuals[i].individual\_num[ran1Day][ran1Week];

individuals[i].individual\_num[ran1Day][ran1Week] = individuals[i].individual\_num[ran2Day][ran2Week];

individuals[i].individual\_num[ran2Day][ran2Week] = temp;

}

}

}

}