4) local search: 1 st: move timeslot of t_n of worker w_u to t_{n-1} or t_{n+1}

2nd move : choose 2 workers and switch timeslots t_n

crossover : generate a solution
get two workers w_u w_e and their solutions (ex. 01000100 and 0~~1~~1000001
cut both strings at random position and switch 2nd parts of the strings. If the solution is better, keep it.

5) Tabu:

1. Generate a random solution
2. Define moves: set brake of workes w_u from t_s to t_{s+1}
   moves 2 : t_s to t_{s-1}
3. Generate neighbourhood solutions using crossover operator
4. Choose best neighbourhood solution
5. If neighbourhood sol better than old sol
   old sol = new Sol
6. Actualise tabu list

6)
1. Assign randomly workers for timeslots
2. Select randomly a timeslot, which value conflicts with any constraint
3. Assign worker to this timeslot ~~with~~ with minimum conflicts
4. repeat 2-3. until solution is found

more than one worker have the same minimum [illegible]flicts the worker is chosen randomly

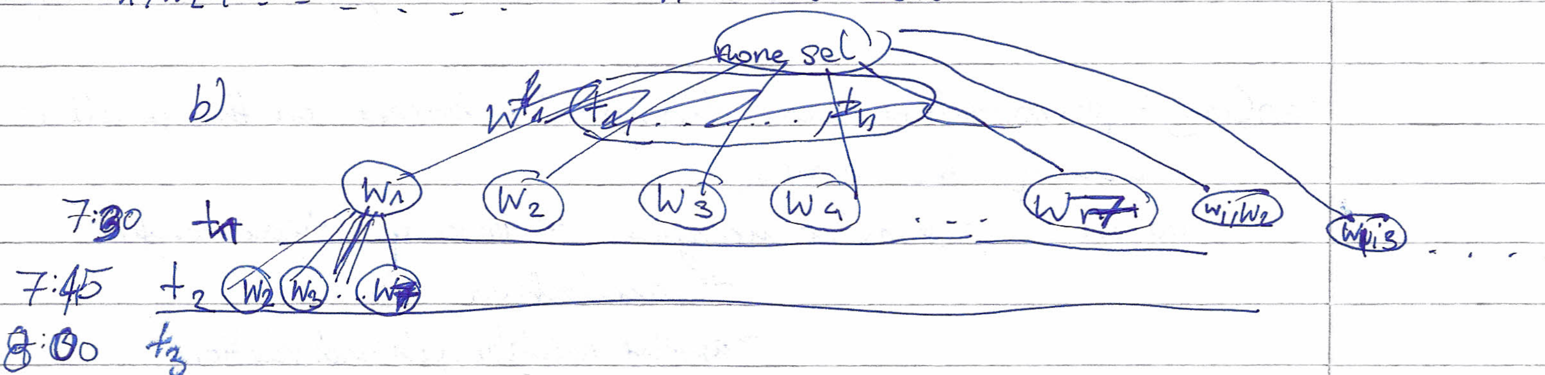
$\delta_{i,3}$ $(i,3)$ $\geq \rho x$.

II) I) CSP:

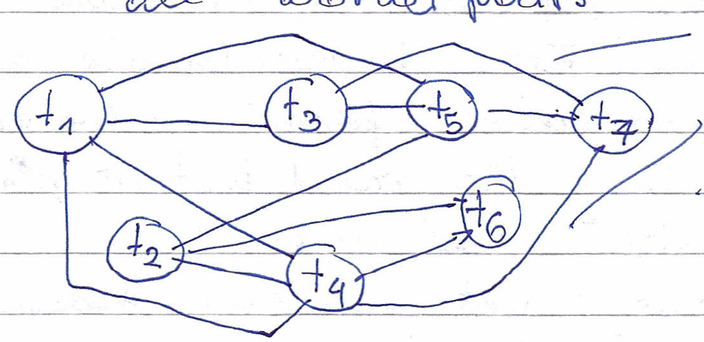
variables: all possible timeslots
domains: working / not working employees
constraints:
- workingEmployee.Size ≥ 5
- no break employee.time $< 0{:}30$ od $>$ timemax $- 0{:}30$
- maxDistance between breaks 80min
- min Distance — " — 45min
- lunch Break $11{:}00 < x < 14{:}00$

tree not graph
$t_1, t_2, \dots\dots\dots t_n$ possible timeslot
$w_1, w_2, \dots\dots\dots w_n$ workers

b)



graph: $t_1 \dots\dots t_n$
possible to contain workers $w_1 \dots w_7$ and all worker pairs



min 5. workers working constraint

t_1 kann nur zu t_3 wegen 30 min arbeiten min
t_1 kann max zu t_5 wegen max 80 min arbeiten

Graph will have a big tree width because at every tree level each node has 7-el. selected childs
so total nodes $1 + (7 + \binom{2}{7}) + (7 + \binom{2}{7} - 1) \cdot (7 + \binom{2}{7}) + \dots\dots$