Problem Solving and Search in Artificial Intelligence

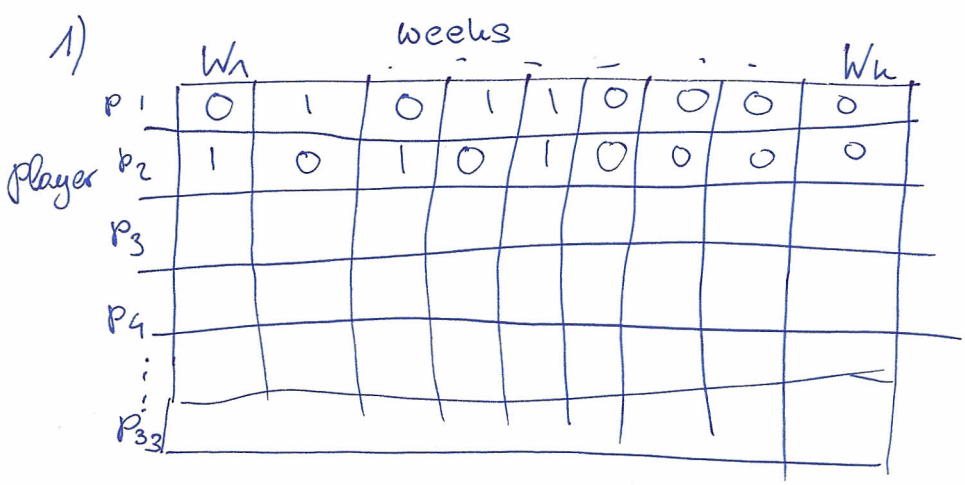
Final Exam, 27.10.2010

Student:
Matrikel Nr:

The Social Golfer Problem is defined below:
32 golfers play golf once a week, and always in groups of 4. For how many weeks can they play such that no two players play together more than once in the same group?
The problem can be generalized as a decision problem: Is it possible to schedule **n = g×p** golfers in **g** groups of **p** players for **w** weeks such that no two golfers play in the same group more than once?

1. Find an appropriate solution representation for this problem.
2. Propose an objective function for the decision problem and for the optimization problem.
3. Propose a local search technique based on tabu search to solve this problem (including an appropriate move, decision for acceptance of solution, neighborhood exploration, …).
4. Can you find an appropriate crossover operator for this problem?
5. Formulate this problem as CSP and
   a. Illustrate how you could solve this problem by forward checking (only few steps)
   b. Draw a part of constraint graph for this problem. Do you think that this graph will have small (hyper)tree width (explain your answer)

1)



0 → not playing
1 → playing

2.) Decision problem: which golfer should play?

Choose golfer who has played the fewest.
if more than one → choose random

optimization problem: fitness: max (breaktime between two games)
find golfer with longest break between two games. ~~switch timeslot~~ apply crossover between this break and switch with other players. Choose best solution.

Problem solving of AI 27.10.2010

3) Tabu Search:

1. Initialise Tabu list
2. Generate random solution
3. Generate moves: flip player x_i into 0 or 1
4. Generate neighbourhood solutions by flipping all players timeslots
5. Evaluate fitness of solution: ~~choose soluti~~
   fitness = Maximum breaktime of a player between two games
   if two solutions have the same fitness, choose ~~the one, where the second big~~ biggest ~~breaktime~~ random
6. Choose the best neighbourhood solutions by ~~evaluating~~ the minimum fitness.
7. If the cur-solution fitness > neighbour fitness then choose the neighbourhood sol.
8. All the other solutions are added to the tabu list

4) Crossover operator

1. A random solution is generated
2. Get the player with the longest break between two games
   1st timeslot t_i, 2nd timeslot of 2nd game t_k
3. The solution is cut between $(i+k)/2$ (casual rounding) and $(i+k)/2 + 1$
4. Switch 2 parts of solution strips randomly
5. evaluate fitness, choose best solution

EDIT: Better to choose number of weeks as fitness