# DYNAMIC PATIENT ADMISSION SCHEDULING PROBLEM

CS 365: Artificial Intelligence

Instructor : Amitabha Mukerjee

Students : Rohitangsu Das
Xavier Valcarcel

## SUMMARY

- Presentation of the problem
- Motivations
- Problem constraints
- Work already done
- Our goal
- References

## PRESENTATION OF THE PROBLEM

- Patient arriving in hospital :

$$\left.\begin{array}{l}\Rightarrow \text{Duration of stay}\\[1em] \Rightarrow \text{Hospital department}\\[1em] \Rightarrow \text{Room number}\\[1em] \Rightarrow \text{Equipment required}\\[1em] \Rightarrow \text{etc...}\end{array}\right\} \text{Managing}$$

## Motivations

- Hospitals have limited resources
- Very complex to manage
- Time can save lives (e.g. disaster)
- Algorithms already exist :

⇒ Resolve only a part of the problem

## PROBLEM CONSTRAINTS

- Hard constraints : Necessary to device a reasonable algorithm according to hospital process
  - Ex : - 2 patients can't be same bed, same time
    - Male and Female separated
    - etc…
- Soft constraints : Use to improve the quality
  - Ex : - The patient can choose the room
    - Assign in a room with specific equipment
    - etc…

## WORK ALREADY DONE

- Hans et al. : optimize the time taken in an operation.
- Oguluta : optimize the workload of a psychoterapist.
- Marinagi : maximize the examination time of a patient and utilization of hospital resources.
- Harper and Shahani : Bed occupancy and patient's refusal were calculated.
- Akcali : Determined the optimal bed capacity in hospitals.

## OUR GOAL

- Use the Tabu search algorithm a to create a patient scheduling algorithm for this problem, so as to satisfy as many soft constraints as possible but adapted with constraints of a disaster :
  - Real-time scheduling
  - Performance
  - Uncertainty in which department the patient has to be assigned
  - Possibility to drop some hard constraints (male and female separated room)

## References

- http://www.sciencedirect.com/science/article/pii/S0933365712001169
- http://satt.diegm.uniud.it/uploads/Papers/CeSc11.pdf
- https://cs.uwaterloo.ca/~jchampai/papers/7283770962056173435.pdf
- http://en.wikipedia.org/wiki/Tabu_search

## THANK YOU

## DISCUSSION.

### Generation of test data.

1. Patients are denoted Pi , withi ¼ 1;...;P, withP the total number of patients. There are F female patients andMmale patients, with P = F + M.
2. Nights are denoted Nk , with k ¼ 1;...;T, with T the number of nights in the planning period of the time horizon.
3. Departments are denoted as Dm , with m ¼ 1;...;D, with D the number of departments. Departments can support one or more specialisms Sl, with l ¼ 1;...;S, with S the total number of specialisms. A department Dm can enforce that assigned patients have a specific age
4. A specialism Sl can enforce that rooms satisfy specific room properties RPv.

## DISCUSSION

### Generation of test data.

Really hard to get test data,privacy issues.

Data based on experienced ‘human’  pateint admission schedulers.

A planning period of 2 weeks,with a total of 6 departments,each having one minor and two major specialism.

Each department has a total of 25 rooms,which is divided into 3 categories.

Each room can have 0, 1 or 2 room properties. Per specialism a random number (with a maximum of five) of subspecialisms is generated. With each subspecialism a length-of-stay is associated that is generated randomly based on a normal distribution with mean 5 and variance 3.

## DISCUSSION

### Constraint weights.

Not all the constraints of Section 4.2 are equally important. The weights we attribute to the constraints determine their mutual relative importance.

### The representation of a solution

Depending on how you represent the solution,one or more hard constraints will get satisfied,leaving the rest constraints on the algorithm.

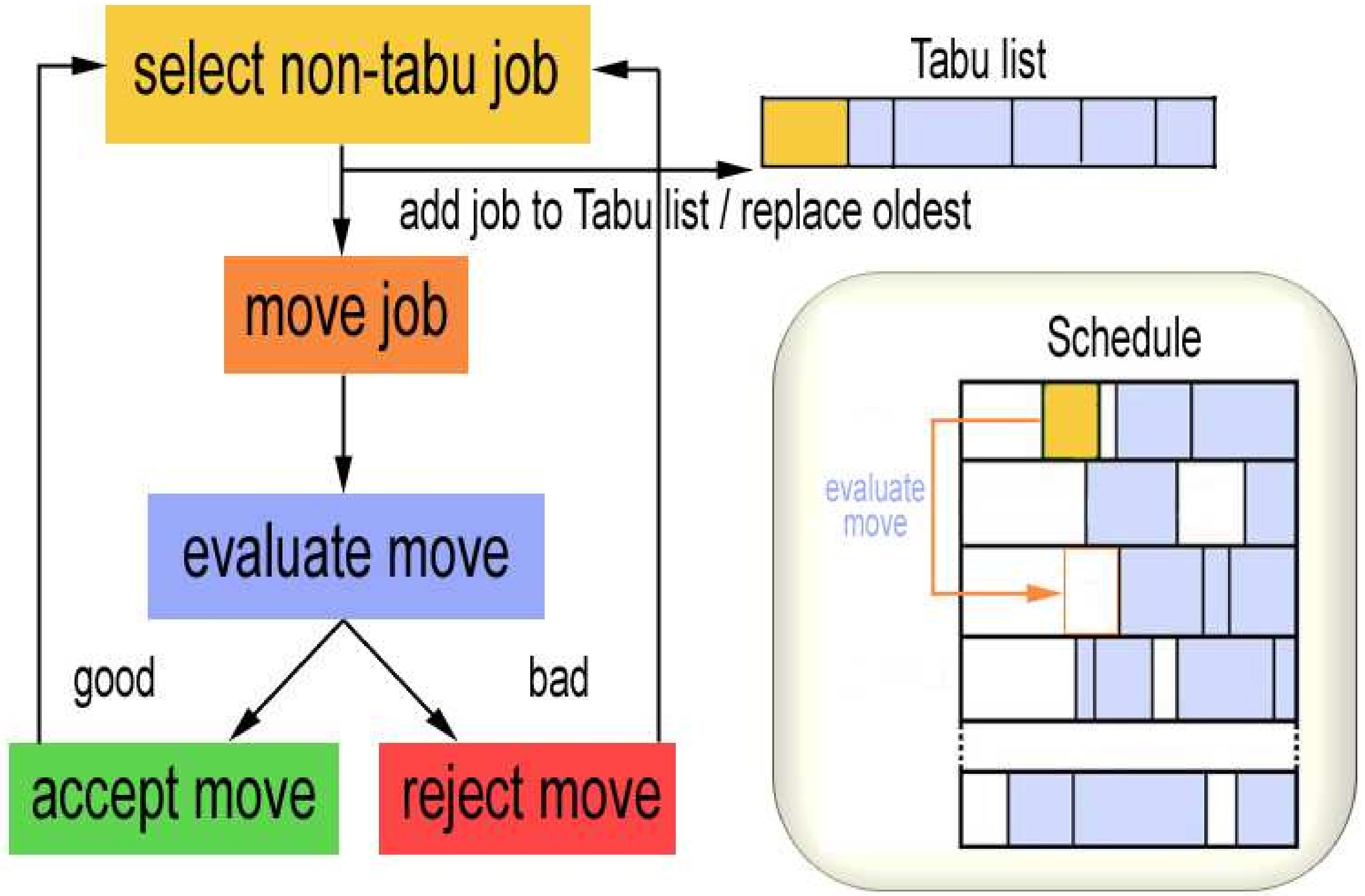
One representation.

We represent a solution as a set of two-dimensional matrices. Each row of a matrix represents a bed in a department. The columns represent the consecutive nights.

## Tabu Search Algorithm

- A Meta Heuristic search algorithm .
- Takes a potential solution  and searches its neighborhood in the hope of finding better solution
- Steps
  1. Take a initial solution.
  2. Maintain a Tabu List
  3. Look at its neighbors and keep adding solution to set known as  candidate list, if the elements of that soln isn't present in the tabu list.
  3. Choose optimal solution from the candidate list.
  4. Compare this solution from previous solution,if better make it the current best solution.
  5. Finally,add the elements of this algorithm from the new solution and add it to the Tabu List.
  6. Keep repeating this step until a user defined exit condition is encountered.

## Tabu search Algorithm.


select non-tabu job
Tabu list
add job to Tabu list / replace oldest
move job
Schedule
evaluate move
evaluate move
good
bad
accept move
reject move