Caixa Mágica Software

SCHEDULE OPTIMIZER

Summer Internships 2017
Final Project Presentation

Trainee: Daniel Ramos

Mentor: Afonso Ribeiro

Co-mentor: Vitor Martins

CONTENTS

- Context;
- Defining the problem;
- Approach used to solve it;
- Demo (video).

CONTEXT

- Service of installation and maintenance of solar panels (and more);

- Currently, the algorithm used to assign teams to each appointment has some limitations.

- Develop an optimizer to overcome these limitations.

DEFINING THE PROBLEM

- A daily schedule has appointments.
- Each appointment must have an assigned team and a start time.
- A **solution is an assignment of a pair (team, start time) for each appointment.**
- The 'best solution' is the one that maximizes/minimizes a given objective function.

- **Example:**
  - **Daily Schedule for 08/09/2017:**
    - **Objective function**: least amount of teams used.
    - **Appointments**:
      - Appointment (1) starts between 10 and 12am
      - Appointment (2) starts between 10 and 12am
      - Appointment (3) starts between 8 and 11am
- **Solution**:
  - Appointment (1) = (Team1, 10:13 am);
  - Appointment (2) = (Team2, 10:15 am);
  - Appointment (3) = (Team3, 11:00 am).

ALGORITHMS

- **Integer Linear Programming:**
  - NP-Hard;
  - Hard to solve in a timely manner (optimally);
- **As a search problem:**
  - Classical Search;
  - Local Search;

CLASSICAL SEARCH

- **Assumptions:**
  - There are on average 20 appointments per day;
  - For each appointment there are at least 4 teams which can be assigned to it;
  - If an appointment has been agreed to start within the interval [8:00,10:00] it can only start at 8:00, 8:30, 9:00, 9:30, 10:00.
- **Conclusion:**
  - 4 * 5 = 20 possible assignments / pairs of (team, start time)
  - 20^{20} = 104 857 600 000 000 000 000 000 000 different schedules
  - 10 000 000 schedules $3,17 * 10^{-8}$ years
  - 20^{20} schedules 316 900 000 000 years
  - Age of the universe 13 800 000 000 years

LOCAL SEARCH

"Local search algorithms move from solution to solution in the space of candidate solutions (the search space) by applying local changes, until a solution deemed optimal is found or a time bound is elapsed. (Wikipedia)"

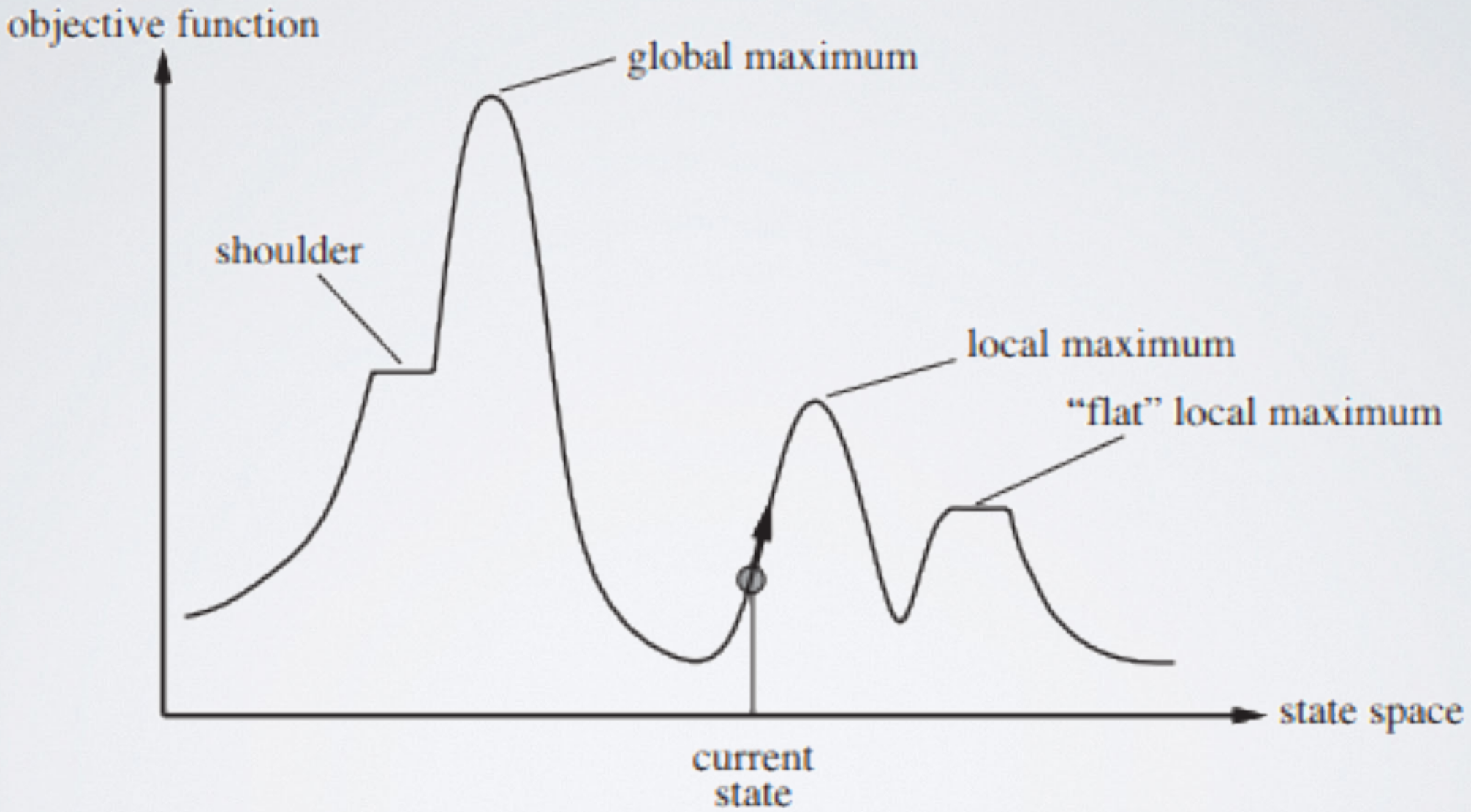
- Can stop at any time;
- Not optimal but close to optimal;

- For a predefined number of steps, given a schedule, **s**, and a set of actions, **A**, the algorithm chooses the action, $\mathbf{a} \in \mathbf{A}$, that best improves the schedule for the selected objective function, **f**;
  - **for a number of steps:**

    s ← **apply**(f, a_k), where a_k is the action such that

    f(**apply**(f, a_k)) = **max**(**f**(**apply**(s, a_1)), …, **f**(**apply**(s, a_n)))

- There's more to it (the solution above is naive).


objective function
global maximum
shoulder
local maximum
"flat" local maximum
state space
current
state

DEMO