

1 E.A.5.1

1.1 Implementazione

È bastato usare il framework che avevo già scritto, e riscrivere il CSP nei termini del framework per risolvere il problema con `steepest_descent()`.

```
use ai::problem::{Heuristic, Local, Problem, Transition};

pub struct Constraints;

impl Problem for Constraints {
    type State = [i64; 5];
}

impl Transition for Constraints {
    type Action = (usize, i64); // assign val to var

    fn new_state(&self, state: &Self::State, &(var, val):
&Self::Action) → Self::State {
        let mut new_state = *state;
        new_state[var] = val;
        new_state
    }
}

impl Heuristic<usize> for Constraints {
    fn heuristic(&self, x: &Self::State) → usize {
        (0.max(-x[0] + x[2] + 1)
         + 0.max(x[1] - x[2])
         + 0.max(x[2].pow(2) + x[3].pow(3) - 15)
         + 0.max(-x[4] + 3)
         + 0.max(-x[0] - x[4] + 3)) as usize;
    }
}

impl Local for Constraints {
    fn expand(&self, state: &Self::State) → impl
Iterator<Item = Self::Action> {
        let mut actions = vec![];
        for (var, curr_val) in state.iter().enumerate() {
            for val in 0..5 {
                if &val ≠ curr_val {
                    actions.push((var, val));
                }
            }
        }

        actions.into_iter()
    }
}
```

```

fn main() {
    let solution =
        steepest_descent(&Constraints, [3, 1, 4, 5, 2]);

    println!(
        "{:?} - {:?}",
        solution,
        Constraints.heuristic(&solution)
    );
}

```

Non ho disegnato l'albero, ma di seguito sono elencate le azioni intraprese, i vicini e i loro costi. Manca la relazione d'ordine per stati che hanno costi equivalenti, ma significherebbe modificare l'implementazione di `steepest_descent()` in modo che richieda una relazione d'ordine totale sugli stati (*si potrebbe fare, ma non vale la pena modificare l'algoritmo*).

```

stato iniziale: [3, 1, 4, 5, 2] - 129

azioni: [(0, 0), (0, 1), (0, 2), (0, 4), (1, 0), (1, 2), (1,
3), (1, 4), (2, 0), (2, 1), (2, 2), (2, 3), (3, 0), (3, 1),
(3, 2), (3, 3), (3, 4), (4, 0), (4, 1), (4, 3), (4, 4)]

vicini:
[0, 1, 4, 5, 2] - 133
[1, 1, 4, 5, 2] - 131
[2, 1, 4, 5, 2] - 130
[4, 1, 4, 5, 2] - 128
[3, 0, 4, 5, 2] - 129
[3, 2, 4, 5, 2] - 129
[3, 3, 4, 5, 2] - 129
[3, 4, 4, 5, 2] - 129
[3, 1, 0, 5, 2] - 112
[3, 1, 1, 5, 2] - 112
[3, 1, 2, 5, 2] - 115
[3, 1, 3, 5, 2] - 121
[3, 1, 4, 0, 2] - 4
[3, 1, 4, 1, 2] - 5
[3, 1, 4, 2, 2] - 12
[3, 1, 4, 3, 2] - 31
[3, 1, 4, 4, 2] - 68
[3, 1, 4, 5, 0] - 131
[3, 1, 4, 5, 1] - 130
[3, 1, 4, 5, 3] - 128
[3, 1, 4, 5, 4] - 128

azioni: [(0, 0), (0, 1), (0, 2), (0, 4), (1, 0), (1, 2), (1,
3), (1, 4), (2, 0), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2),
(3, 3), (3, 4), (4, 0), (4, 1), (4, 3), (4, 4)]

```

vicini:

```
[0, 1, 4, 0, 2] - 8
[1, 1, 4, 0, 2] - 6
[2, 1, 4, 0, 2] - 5
[4, 1, 4, 0, 2] - 3
[3, 0, 4, 0, 2] - 4
[3, 2, 4, 0, 2] - 4
[3, 3, 4, 0, 2] - 4
[3, 4, 4, 0, 2] - 4
[3, 1, 0, 0, 2] - 2
[3, 1, 1, 0, 2] - 1
[3, 1, 2, 0, 2] - 1
[3, 1, 3, 0, 2] - 2
[3, 1, 4, 1, 2] - 5
[3, 1, 4, 2, 2] - 12
[3, 1, 4, 3, 2] - 31
[3, 1, 4, 4, 2] - 68
[3, 1, 4, 0, 0] - 6
[3, 1, 4, 0, 1] - 5
[3, 1, 4, 0, 3] - 3
[3, 1, 4, 0, 4] - 3
```

azioni: [(0, 0), (0, 1), (0, 2), (0, 4), (1, 0), (1, 2), (1, 3), (1, 4), (2, 0), (2, 2), (2, 3), (2, 4), (3, 1), (3, 2), (3, 3), (3, 4), (4, 0), (4, 1), (4, 3), (4, 4)]

vicini:

```
[0, 1, 1, 0, 2] - 4
[1, 1, 1, 0, 2] - 2
[2, 1, 1, 0, 2] - 1
[4, 1, 1, 0, 2] - 1
[3, 0, 1, 0, 2] - 1
[3, 2, 1, 0, 2] - 2
[3, 3, 1, 0, 2] - 3
[3, 4, 1, 0, 2] - 4
[3, 1, 0, 0, 2] - 2
[3, 1, 2, 0, 2] - 1
[3, 1, 3, 0, 2] - 2
[3, 1, 4, 0, 2] - 4
[3, 1, 1, 1, 2] - 1
[3, 1, 1, 2, 2] - 1
[3, 1, 1, 3, 2] - 14
[3, 1, 1, 4, 2] - 51
[3, 1, 1, 0, 0] - 3
[3, 1, 1, 0, 1] - 2
[3, 1, 1, 0, 3] - 0
[3, 1, 1, 0, 4] - 0
```

azioni: [(0, 0), (0, 1), (0, 2), (0, 4), (1, 0), (1, 2), (1, 3), (1, 4), (2, 0), (2, 2), (2, 3), (2, 4), (3, 1), (3, 2),

(3, 3), (3, 4), (4, 0), (4, 1), (4, 2), (4, 4)]

vicini:

[0, 1, 1, 0, 3] - 2
[1, 1, 1, 0, 3] - 1
[2, 1, 1, 0, 3] - 0
[4, 1, 1, 0, 3] - 0
[3, 0, 1, 0, 3] - 0
[3, 2, 1, 0, 3] - 1
[3, 3, 1, 0, 3] - 2
[3, 4, 1, 0, 3] - 3
[3, 1, 0, 0, 3] - 1
[3, 1, 2, 0, 3] - 0
[3, 1, 3, 0, 3] - 1
[3, 1, 4, 0, 3] - 3
[3, 1, 1, 1, 3] - 0
[3, 1, 1, 2, 3] - 0
[3, 1, 1, 3, 3] - 13
[3, 1, 1, 4, 3] - 50
[3, 1, 1, 0, 0] - 3
[3, 1, 1, 0, 1] - 2
[3, 1, 1, 0, 2] - 1
[3, 1, 1, 0, 4] - 0
[3, 1, 1, 0, 3] - 0
[3, 1, 1, 0, 3] - 0