Topic 6: Traffic Jams

Authors

- Andro Irina
- Borgovan Andreea
- Doica Rareş Andrei
- Hărăguş Andreea
- Miron Alexandru Bogdan
- Morodan Georgiana Tatiana
- Popa Irina

Affiliation

Colegiul Național "Emil Racoviță"

Problem statement

In order to simplify the problem of traffic jams, let's start with a simple case:

- a single line of cars
- the cars are all identical and move at the same speed
- two possible positions: stop or go
- a car moves forward one square when the space in the front of it is empty
- a car stays in place when the space in front of it is occupied.
- by placing a number of cars, study the evolution of traffic

Different Velocities Approach

The flow of traffic can be affected by number of lanes, each cars' speeds. For a simple case, with one lane and same speed, it's the ideal situation because it prevents the eventually accidents. Another simple case is with different speed, where it can results many clusters and the flow of straffic is slowing down because of varying acceleration and deceleration of vehicles.

In a complex case, the vehicles have the occasion to change lanes, but it can result a sequence of events that all the cars behind of the relevant car are slowing down (Butterfly effect).

Mathematical Approach

Convention :

- 1 = car/occupied space
- 0 = free space
- n = number of cars not divided by free spaces
- m = number of free spaces not divided by cars

We consider that the traffic flows freely if every car (1) is placed between two free spaces (0).

- most basic case: 1111...1 (n) fluidises in n-1 moves
- complicated case: 111..100..011...11

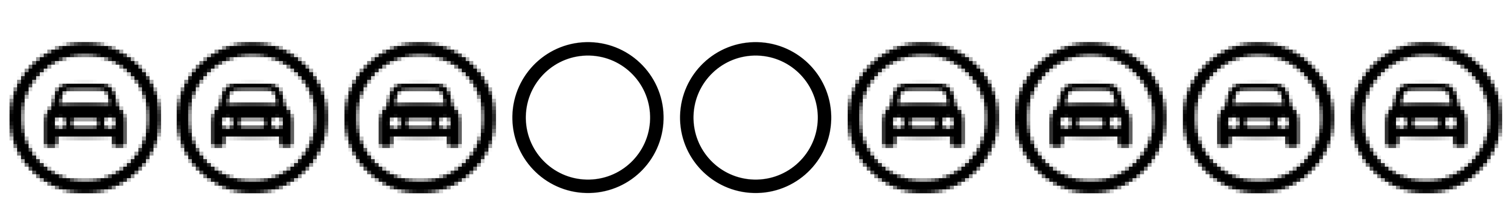
Computer Science Approach

We developed a computer science algorithm in order to efficiently compute the number of steps required for the traffic jam to reach a fluid state. The algorithm is based on an iterative approach in order to simulate the basic operations that a car can perform:stagnate and move forward.

The complicated case

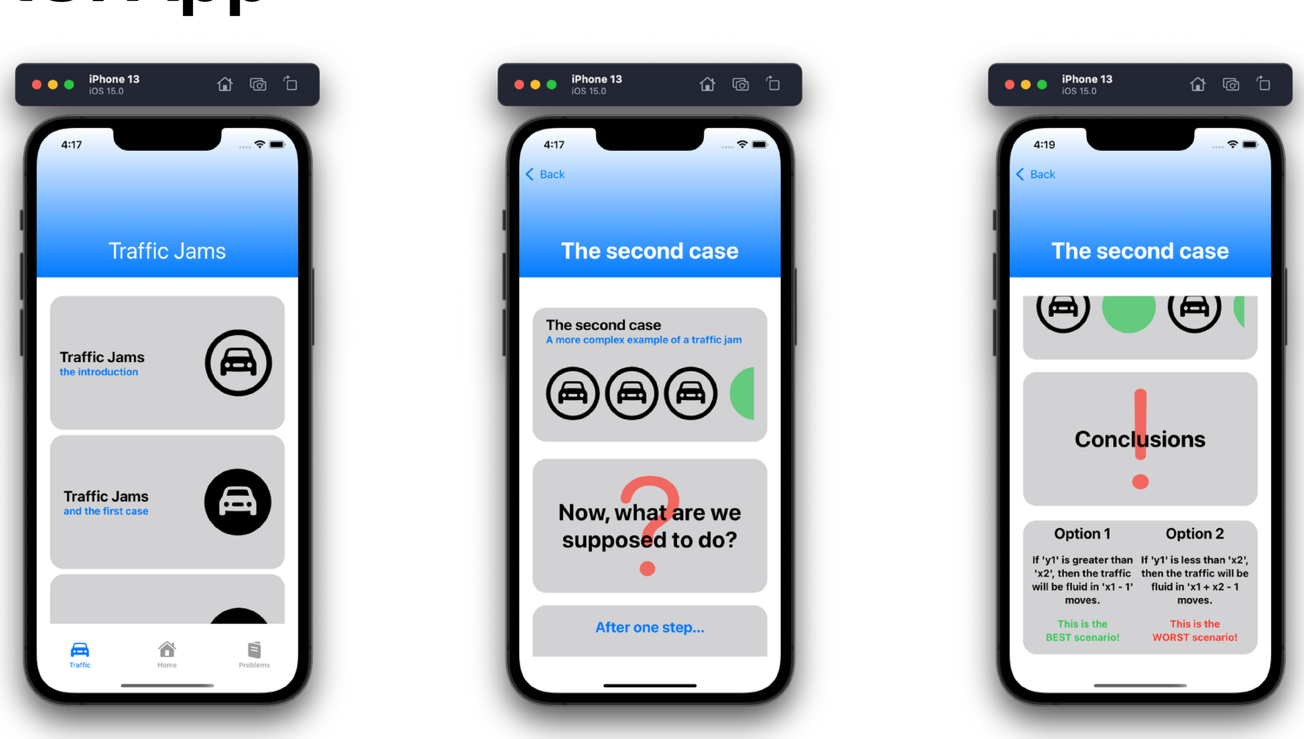
Let's take for instance n groups of cars, X_1 cars in the first one, X_2 cars in the second group and so on. We will use the notation Y for the number of empty spaces between two groups of cars.

If $Y_n \geqslant X_{n+1}$, for every n, the traffic reaches fluidization in $X_{max} - 1$ steps. On the other hand, if we take for instance two groups and they don't satisfy the condition ($Y_1 \leqslant X_2 - 1$), then the number of steps before the formation of the new block is Y_1 (and we will have a new group of $X_2 - Y_1 + 1$ cars), after that is $X_1 + X_2 - Y_1 - 1$, so the traffic reaches fluidization in $X_1 + X_2 - 1$ steps.



This traffic will be solved in 6 moves!

SwiftUI App



C++ Code

```cpp
bool isValid()
{
    for(int i = 1; i <= n; ++i)
      if(v[i] == 1)
         return false;
    return true;
}

void solve()
{
    bool isFluid = false;
    while(!isFluid)
    {
      for(int i = 1; i <= n; ++i)
         copie[i] = v[i];
      for(int i = n; i > 0; --i)
        if(v[i] == 1)
          if(copie[i + 1] == 0)
          {
             v[i + 1] = 1;
             v[i] = 0;
          }
      for(int i = 1; i <= n; ++i)
         copie[i] = v[i];
      if(isValid())
         isFluid = true;
      ++ans;
    }
    std::cout << ans - n;
}
```

Ecological Impact

The traffic jams make up a vast factor in climate change and global warming. This is, from out point of view, the factor which has the most powerful impact over our environment and we should work and fight together to stop it immediately!

Conclusion & Future Plans

To sum up, we have discovered some important mathematical laws that help us solve the problem of a traffic jam. Moreover, we have developed a C++ code and a SwiftUI app that can give us a better visualisation of the problem. in the future, we intend to discover a mathematic law for more complex cases, in order to solve, the traffic jam problem.