

UNIVERSITÀ DEGLI STUDI DI TORINO
Econophysics Course A.A. 2016/2017 Prof. Pietro Terna
ABM for the Last Mile Problem

Federico Cinus & Francesco Surano

Abstract

In this work is introduced the Last Mile problem and the Agent-Based modelling process. A solution for the Last Mile problem is proposed suggesting an active engagement of the population. Some estimation has been done in order to make the simulation suitable for representing the city of Turin and actual cost of the delivering process. Results show the robustness of the proposed solution both in economical and user's engagement terms, discussing also extreme situation.

Introduction

The Last Mile Problem is rooted in telecommunication theory and, informally speaking, it expresses the idea of data transfer between the principal telecommunication line and the user in the last (physical) mile of the network. This can be translate in transportation engineering's in the problem of goods delivery through the last part of the journey between producer and customer, which this work is about.

In the last decades the competition among telephone companies and, as in this case, transportation companies has determined an “investment challenge” that contributed in focusing the attention on the optimization aspect in the branch of the operations research. Indeed with urbanization and globalization the goods exchange’s network is exponentially grown and the use of more resources and methods of transportation has become essential in the transportation’s chain, in particular for the efficiency of the delivery system.

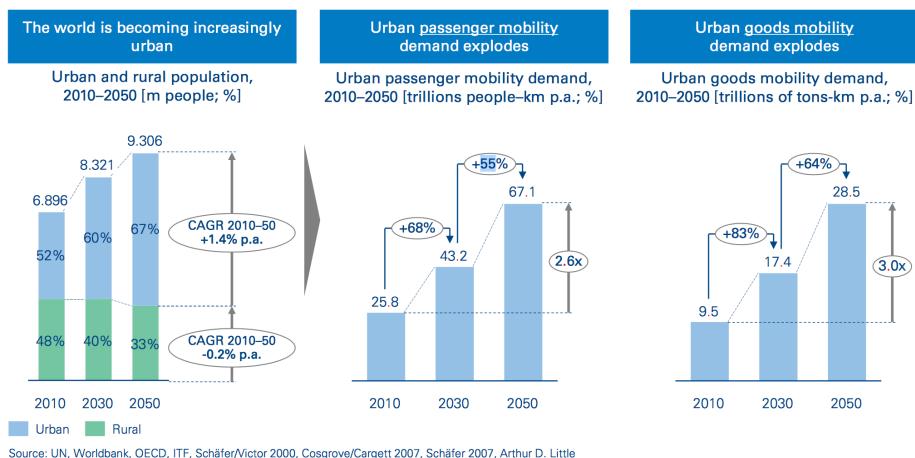


Figura 1: Forecast of urban growth and goods mobility demand

While a lot of work has been done until now for the optimization of long journey transportation, the delivery from the last warehouse to the customer still has a central role in this branch of research; in fact this part of the route contributes with the 28%¹ to the total cost of transportation.

This particular fact expresses the proportionality between today cities complexity and transport costs in them, outlining the purpose this work wants to pursuit: optimization and simulation of new delivering methods. Classical optimization system are not adequately solving the problem and a paradigm change could bring better or interesting results for this sector. Moreover we underline the fact that even if the most important stakeholders (public authorities and transportation companies) may have diverging interests and objectives, it is possible to go further finding a better solution in terms of ecological sustainability, valorization of urban ecosystem and services to the citizen.

¹Wikipedia: [https://en.wikipedia.org/wiki/Last_mile_\(transportation\)](https://en.wikipedia.org/wiki/Last_mile_(transportation))

1 ABM

1.1 ABM approach

The history of Agent Based Model can be traced back to 1940s and the Von Neumann cellular automata, a theoretical machine capable of self-reproduction following simple rules. In the 1970s and 1980s we see the birth of Thomas Schelling's segregation model and Robert Axelrod's tournament of Prisoner's Dilemma algorithms. By the 1990s, with the creation of NetLogo and Swarm software, the agent-based approach became widespread in social science and study of complex system.

An agent-based model (ABM) is one of a class of computational models based on the Agent-oriented programming (AOP), which simulates the actions and interactions of autonomous agents with a view to assessing their effects on the system as a whole. In other words the ABM's aim is to re-create and predict the appearance of complex phenomena simulating operations and interactions of multiple agents, only defining them and not the phenomena; the analogy to statistical mechanics is to define the system through its micro-scale details and then study the complex macrostate. Combining elements of complex systems, computational sociology, multi-agent systems and Monte Carlo methods (to introduce randomness) the complex behavior emerges from the lower (micro) level of systems, i.e. simple behavioral rules, to a higher (macro) level. This principle, known as K.I.S.S. ("Keep it simple, stupid")², is extensively adopted in the modeling community. Individual agents are typically characterized as boundedly rational, i.e. they act pursuing their own interests, but from the point of view of an external observer they express heuristic decision-making rules, learning and adaptation that can be synthesized as: "the whole is greater than the sum of the parts", in the sense that we get more information than those we put in the system.

From a theoretical point of view there are three central concepts that distinguish ABM from other computational model.

- [Emergence] The Agent-based model explore the complex system equilibrium deriving from simple rules of the agents instead of studying the system's equilibrium.
- [Agent-Object] Agent-base models consist of dynamically interacting rule-based agents which can create a real-world-like complexity. Agents are typically situated in space and time, and their location and behavior are encoded in algorithmic form in computer programs. In some cases, though not always, the agents may be considered as intelligent and purposeful.
- [Complexity] ABMs complement traditional analytic methods in the sense that the last characterize the equilibria of a system, while the first allow the possibility of generating those equilibria. Agent-based models can explain the emergence of higher-order patterns and non-linear systems for which we do not have an analytic solution or the numerical one is too expensive from a computational point of view.

²Wikipedia: https://en.wikipedia.org/wiki/Agent-based_model

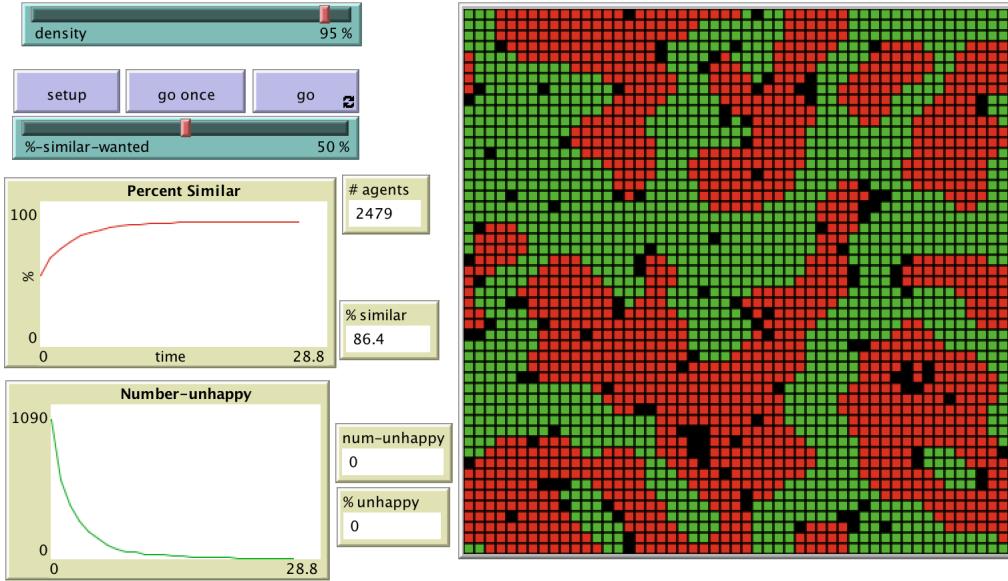


Figura 2: Thomas Schelling’s Segregation model (1978); Micromotives and Macrobbehavior; New York: Norton. Wilensky, U. (1997). NetLogo Segregation model. <http://ccl.northwestern.edu/netlogo/models/Segregation>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL

1.2 ABM approach to the LMP

In order to try solve the Last Mile Problem with an ABM, we have to forget the classical and analytical approach that involve finding a system of equation to be solved either exactly or numerically as a whole.

Instead each agent will act independently and the commissioner of the shipping will be only responsible of optimizing the cost, finding which agent can perform the task at the lowest price. Unpacking the problem in such way it can be easily translate in terms of agent-based programming, indeed we can define a set of behavioral rules which bring the model to an overall stability situation through agent interactions. However, as we said before, the solution independently emerges thanks to the computational representation of the problem complexity.

Generally speaking, each courier can be seen as an instance of a class of agents which have as objective to minimize the journey and get much money as possible; on the other hand we have to define the environment where the agent can move, it could be as complex as a city or only a cartesian coordinate system.

1.3 NetLogo



NetLogo is a multi-agent programmable modeling environment. It is used by tens of thousands students, teachers and researchers worldwide, it powers HubNet participatory simulations, is authored by Uri Wilensky and developed at the CCL.³ NetLogo was designed in order to teach children the programming concepts using agents in the form of turtles, patches, links and observer. The variegated audience brought this language to be used in many scientific published articles. NetLogo allows to explore emergent phenomena with a graphics interface consisting of switches, sliders, choosers, inputs, and other human-friendly interfaces, it has an extensive model library including models in various domains, such as economics, biology, physics, chemistry, psychology, system dynamics allowing the user to create new models and modify the existing ones.

BehaviorSearch⁴ is a software tool to help automating the exploration of agent-based models (ABMs), by using genetic algorithms and other heuristic techniques to search the parameter-space. It interfaces with the popular NetLogo ABM development platform, to provide a low-threshold way to search for combinations of model parameter settings that will result in a specified target behavior. The model's parameters space exploration act in four steps:

1. Design a quantitative measure of the behavior you are interested in.
2. Choose parameters to vary and the ranges are allowed into.
3. Choose a search algorithm and run it.
4. Examine the results (what parameters most affect the chosen behavior?)

In this work we used NetLogo 6.0 and Behaviour Search 6.0.

³NetLogo website: <https://ccl.northwestern.edu/netlogo/>

⁴Behavior Search website: <http://www.behaviorsearch.org>

2 Solution

This section's aim to introduce a possible solution to the Last Mile Problem that we investigated and modeled. The approach to this problem comes from the analysis of the city of Turin, in particular about the urban situation and the world of the sharing economy. First of all, as we shown in the introducing section, nowadays every city in the world is growing and becoming (in our terms) more complex, our case study will be Turin. This complexity causes inefficiency of transportation in urban area, the lack of mobility, and an increase of pollution. All forecasts about those problems affecting cities are not optimistic and a lot of research is focused on this aspect. Starting from those premises we turn our interest on the world of sharing economy watching to:

- BlaBlaCar (cars)
- Enjoy (cars)
- TOBike (bikes)
- Foodora (bikes)

In particular there are 116 TOBike stations with about 10 bikes for each station and 186 Enjoy's cars in Turin. Morover there is 1 subway, 10 trams routes and 112 bus routes used by about 1200 vehicles; adding all the private vehicles for a population of 887,101 citizens we discover a great complexity about the transportation system in Turin.

The main innovation of our idea is to convert this problem in our solution, finding a way to delivery goods all around the city increasing the efficiency as the complexity of a city grows. Basic idea is to deal with the delivery process by entrusting the last mile of the package's journey to people that are moving in town to their destination; this lays the bases for the inclusion of citizens, and their urban movements, in the transportation process transmuting them from passive user to active actor and ensuring the ecology of the process.

We create a model that simulates people traveling with different vehicles in a space to their destination; each one could change his journey, diverting from the original one, in order to deliver a pack to its destination. Every person has a different "inner" cost that depends on his vehicle and personal interest, this brings to an auction in which the delivery company aim at minimizing the expenses, corresponding to the product of the inner cost and the length of the deviation. In order to have a more efficient transportation, packs are located in different magazines in the space, taking inspiration to the 20 Amazon's lockers in Turin; obviously they are put in the nearest magazine to the pack's destination. The proportionality between deviation and expense needs an incentives analysis in order to reconcile the minimization of the company and the maximization of the ordinary citizen's benefits while guaranteeing the overall ecological benefits to the city.

2.1 Calibrating the model

We aim at creating a simulation suitable for the chosen context, the city of Turin.

First we have to estimate the number of possible users interest in participating in a gig-economy-like project, so we analyzed the number of users of EnJoy, a nation-wide car-sharing service. These are roughly 5'000'000 in Italy, the 8% of the population. Being almost 900'000 the citizen of Turin, estimating a similar percentage of people willing to engage in our project, we end up with 7000 possible candidate. For the purpose of our simulation we will scale down this number to 700, in order to contain the computation cost of the simulation.

Then we have to estimate the number of packs delivered in our city daily. We base our calculation on the number of packs delivered yearly by Amazon Inc., roughly 5'200'000 packs for the city of Turin alone, meaning about 14'000 packs delivered daily, supposing a uniform distribution all year along⁵. Again for the sake of computational cost we will scale this number to 1'400, consistently with the proportion used before.

Considering that Amazon charges approximately 6€ for a “same-day” pack delivery and the last mile’s contribution to the entire journey cost (from producer to customer) is about 30%, we defined a maximum threshold of 1.75€ for the simulated delivery.

As for the cost of the basic infrastructure needed, namely a number of short-term deposit in which packages are stored around the city, we set 700€/month the cost for a 24sqm room, comprehensive of maintenance, rent ad basic bills. This scales to about 3€/month for a 50x40x30cm box, about 0.2€/pack for a permanence of two days. Fixed cost are considered negligible in the context of our simulation, being considered as long-term investment.

Lastly we have to map the size of the city to the patch-size of our model. Turin is roughly 12km radius, a distance that can be covered on average⁶ in 75 minutes. This means that the length of a workday, from 8.30 am to 9.00 pm, being 1 patch/tick the speed of our agents, will be such that each agent is capable of going back and forth from one end to the city to the other about 10 times a day; being the radius of our city proportional to the map’s diagonal. This can be translate in NetLogo’s language assuming 1600 ticks equals a day.

⁵Amazon Innovation Awards 2017, University of Turin, Torino

⁶35mins by car, 150mins by foot

3 Code

The basic idea of this work is to give the possibility to every one is traveling in the city to transport a pack to its destination. This is a simple concept that, as we shown in the previous section, has some implications. In order to introduce the code to the reader it is useful to present the simulation through its evolution, i.e. we want to show how the code has changed starting from the first idea/solution for the Last Mile Problem to its generalization. We remind that the code is written in NetLogo 6.0 and it is possible to freely download it and run the simulation.

3.1 Moving agents in a cartesian coordinate system

First of all we have to implement a code that simulates the movement of a defined number of agents in a environment characterized by an empty cartesian coordinate system.

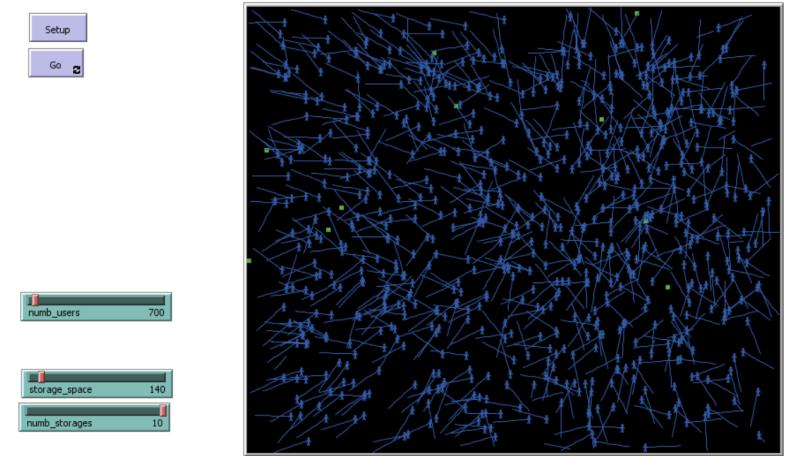


Figura 3: Moving agents and fixed storages

Basic Agents

The basic agent in our simulation will be a breed called “user” moving freely. They will have a set of own variable, at this point only the destination. Agents are created in random patches ad are assigned other random patches as destination. Each agent will then move toward their destination in a straight line and die once reached it. To avoid the extinction of users, a new one is created in the patch the previous one died.

Special Patches

A special patchset is created to represent the storages around the city. These patches will be identified by a boolean value, indicating the role of storage or not, and, if so, they will also own a variable “space”, that will be further used to represent the physical space in a storage room.

3.2 Moving packs

Once create the basic structure we need to simulate the transportation of packages around the city done by users.

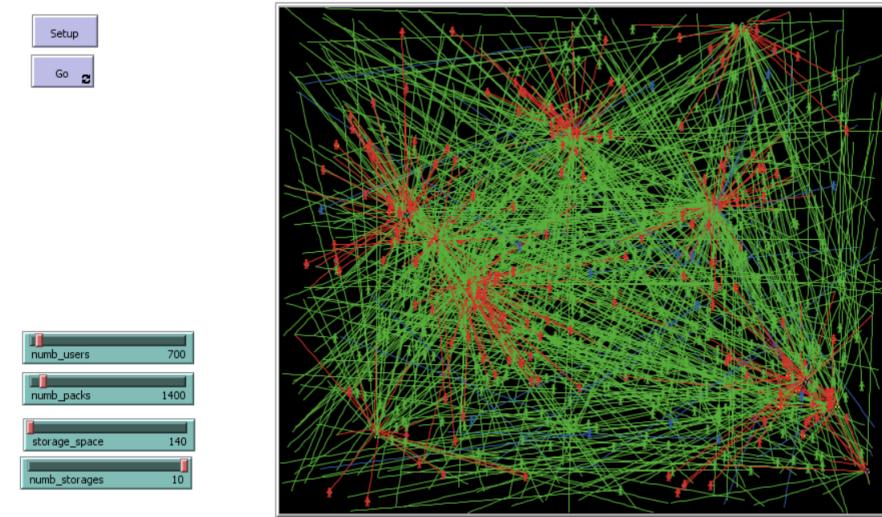


Figura 4: Agents delivering packs

Packages

A breed called “packages” is created in order to simulate the goods transported in the space. Each package will own two variable, one for the destination and another, boolean, to indicate if they are assigned to a user or not. Each packages is placed in a locker-patch ad assigned a random point on the map as destination.

Pick-up

In order to be picked up a package needs to be put in relation with a user. At first this will be done randomly, the user will go from his position to the position of the package, then do the destination of the package and lastly to his own destination. To properly keep track of what a user is doing we need to add two new variables to the user breed, “busy” and “bag”. Bag will contain the id of the assigned pack and busy will be true if the pack is being transported to its destination by the user.

3.3 Optimizing

Since now all operation are done randomly and with no regards to the optimization of the process. We need to introduce a way to properly choose the package-user couple such that the distance traveled is minimal.

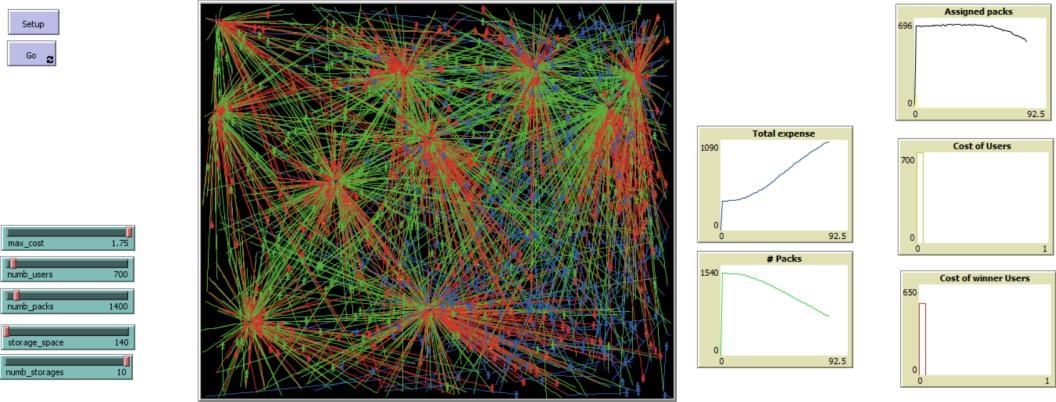


Figura 5: Cost evolution after auction

Auction

To optimize the travel we will use an auction for each package. Each user will have a new variable, “cost”, that will be the amount of a virtual money needed to be paid in order to deviate one patch off the straight line connecting the origin and the destination of the user. Each package will ask all the user the amount they will need in order to deviate from their original route, pick them up and bring them to destination: the lower bidder will be assigned the pack and will proceed with the delivery.

```

;; Auction
to auction
  ask packs with [assigned = false] [
    let t_dist distance destination
    let t_pack self
    let t_dest destination
    let t_candidate users with [bag = NOBODY and trader? = true]
    ; Finding the user that corresponds to the smallest expense
    let t_best min-one-of t_candidate [
      cost (
        ((distance t_pack) + (t_dist) + ([distance t_dest] of destination)) - (distance destination)
      )
    ]
    ; refusing if the expense is bigger than a defined threshold (max_cost)
    if t_best != NOBODY [if [
      cost (
        ((distance t_pack) + (t_dist) + ([distance t_dest] of destination)) - (distance destination)
      )
      ] of t_best > max_cost [set t_best NOBODY]
    ]
    ; adding to the total expense
    if (t_best != NOBODY) [
      set expenses (expenses + [
        cost (
          ((distance t_pack) + (t_dist) + ([distance t_dest] of destination)) - (distance destination)
        )
        ] of t_best)
    ]
    ; assigning packs to the winner of the auction
    if (t_best != NOBODY) [
      set assigned true
      ask t_best [set busy false set bag t_pack]
    ]
  ]
end

```

3.4 Making it real

At this point we already have all the essential part of our simulation: agent moving around, delivering packs, in a sort of optimized process. Yet we are far from real and some elements need to be taken into account.

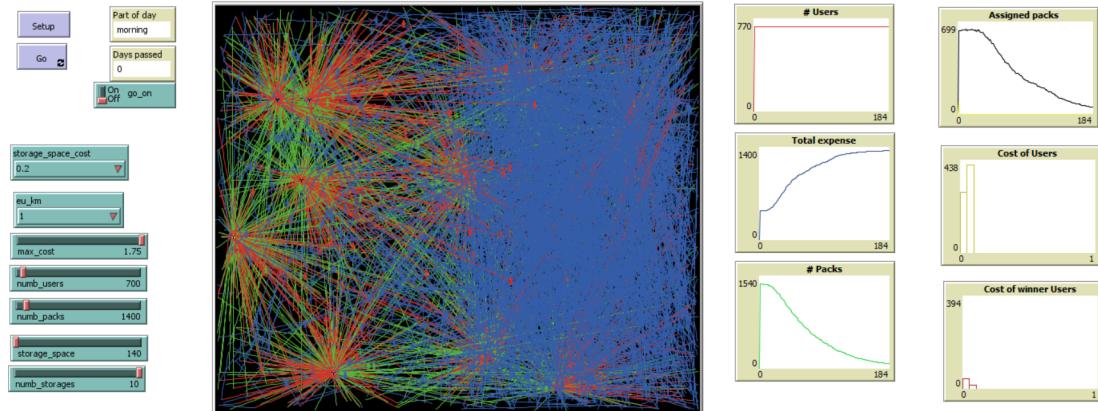


Figura 6: Complex movement behaviour

Highest cost

The commissioner will not be willing to pay any amount to have his packages delivered, a maximum cost per package is fixed, if the lowest bidder ask for a total amount higher than the fixed maximum cost, the package is not assigned and the auction run again.

Goods placement

Packages are not placed randomly in the storages patches but, if the space is enough, each package will be placed in the storage nearest to its destination.

```
; Placing pack in the nearest storage to its destination
ask packs [
  let t_pack self
  let t_dest destination
  let t_storage patches with [storage? = true and space > 0]
  let t_best min-one-of t_storage [distance t_dest]
  ifelse t_best != NOBODY [move-to t_best][move-to one-of patches with [storage? = true and space > 0]]
  ask t_best [set space (space - 1)]
]
```

Day and night

People fluxes are not constant during the day, in a city they follow the home-work path one way or the other in the morning and in the evening. To simulate that we suppose that after a certain amount of ticks we switch from morning to evening. The cartesian space is divided in two section, the left and the right one respectively along the xcor, during one part of the day agent will be assigned a destination on one side with an higher probability, the other way round during the other part of the day.

```
;User has a probability of 75% to follow the people's morning/evening flux
ask users [ifelse random-float 1 < 0.75
  [ifelse time = "morning" [set destination one-of [other patches with [pxcor > 0]] of patch-here ]
   [set destination one-of [other patches with [pxcor <= 0]] of patch-here ]
  ]
  [set destination one-of [other patches] of patch-here]
 ]
```

Randomness

Random behaviour are created for user: they may, with a small probability, not accept to deliver a package, their internal cost is uniformly distributed around a mean value instead of being fixed.

Maintenance cost

As we can expect the maintenance of all storages around the city will have a cost, more and bigger storages will cost more and add up to the expenses of the commissioner. A value can be chosen to simulate this cost.

3.5 City of Turin

In order to scale the simulation to the city of Turin, a series of metric are calculated with the same ratio we can find in the real life, as explained in the previous chapter.

```
set diagonal (((max-pxcor * 2) ^ 2) + ((max-pycor * 2) ^ 2)) ^ 0.5
set eu_pt (eu_km * 12) / diagonal
set halfday ceiling (((((max-pxcor * 2) ^ 2) + ((max-pycor * 2) ^ 2)) ^ 0.5) * 10) / 2
print "halfday:" print halfday
```

4 Results

In this section we present the results obtained from the Agent-Based simulation. We discovered that the solution is feasible i.e. people traveling in the city can deliver all packs on time and with a total expense that is lower than the upper-bound (1.75€ per pack). Moreover, being the Foodora's employee salary about 0.5€/km, we can see that in our estimated range of parameters a person can be paid more than a common delivery man.

Using genetic algorithms implemented in Behavior Search 6.0 we optimized the number of storages and their space, keeping all other estimated parameters fixed such as the number of users, the number of packs and the cost of each box in the storage.

Figure below show the fitness minimization (expenses) with the following parameters: 1400 packs, 700 users, 1€/km for users' deviation, 1.75€ as maximum expense for a pack delivery, 0.2€ cost storage's box.

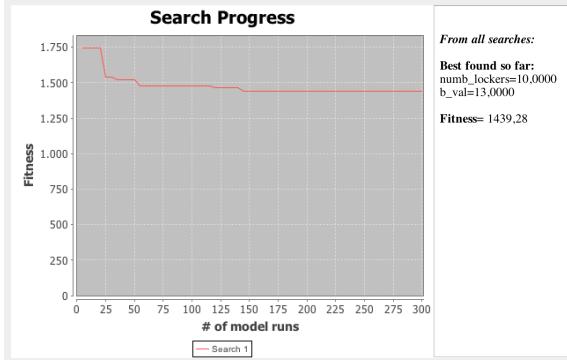


Figura 7: Behavior Search result: 10 storages with 304 boxes each, total expense 1439.28

Considering 0.8€ as storage's box cost, we find again that the number of boxes in each storage is greater than the minimum needed. This implies that spending money for having bigger storage is useful in case a pack has to be delivered in the neighborhood.

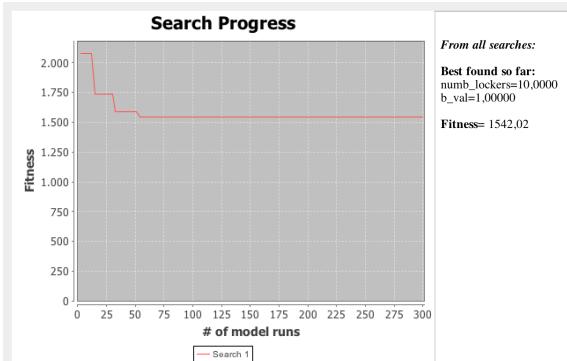


Figura 8: Behavior Search result: 10 storages with 153 boxes each, total expense 1542.02

May be found interesting for the reader to have a deeper look at two of the most important curves in our simulation: number packs vs. time and total expenses vs. time.

The figure below show that the number of packs exponentially decreases with time as expected.

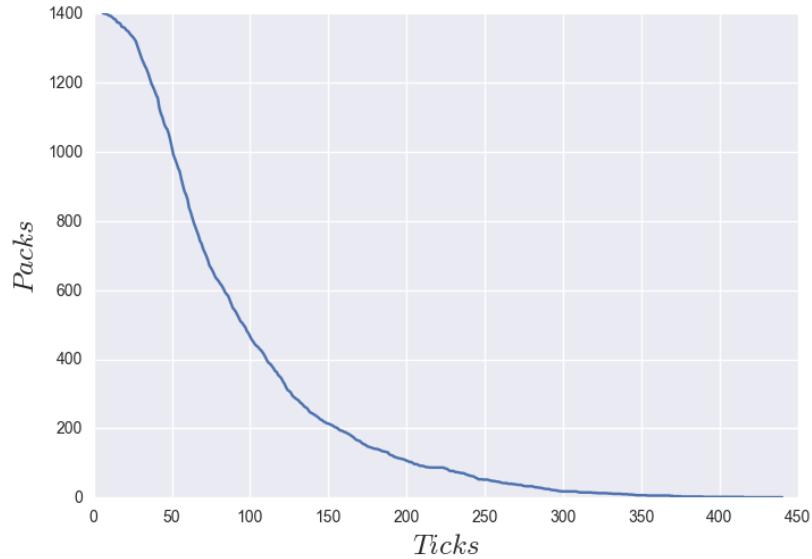


Figura 9: Numb packs vs. ticks (10 storages with 304 boxes each, 1400 packs, 700 users)

In order to roughly estimate the curve decrease we did a linear regression with 0.07 mean squared error.

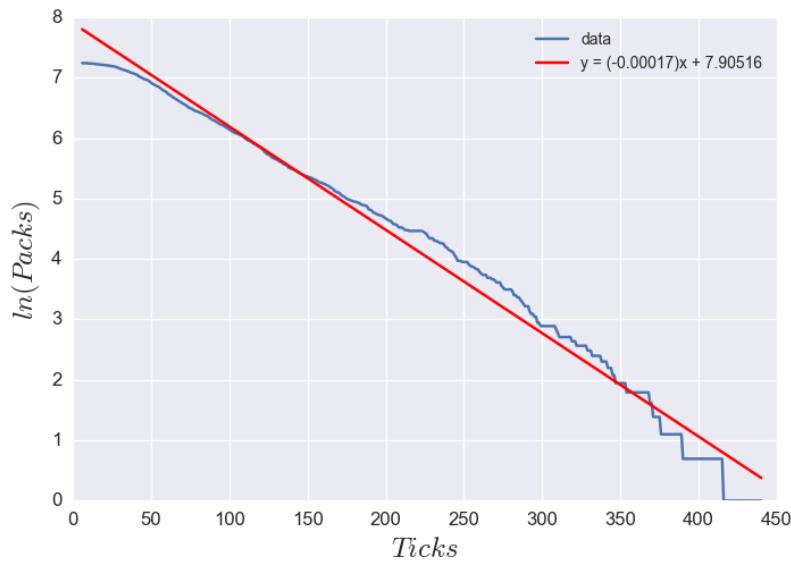


Figura 10: Fit: numb packs vs. ticks (10 storages with 304 boxes each, 1400 packs, 700 users)

On the other hand the total expenses approximately increase as a logarithm, which compensates the number of packs trend.

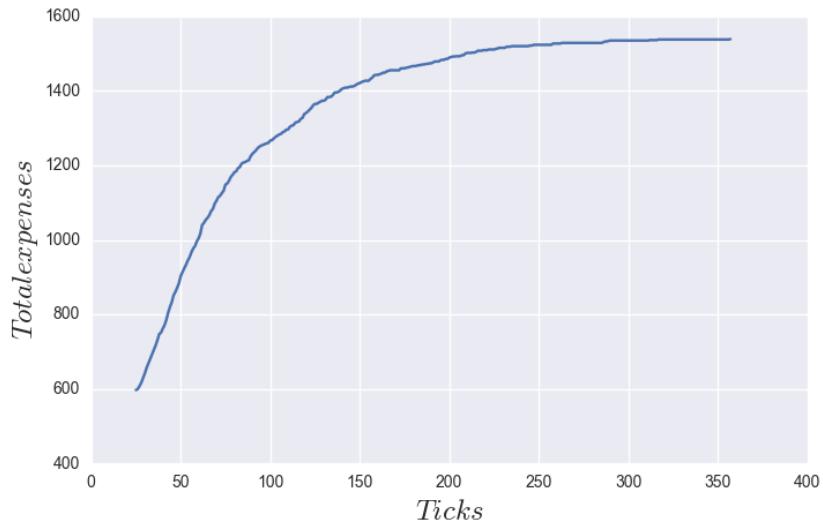


Figura 11: Total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 700 users)

Even if the mean squared error is considerably worse than the first fitting, we can have an estimate of the curve slope.

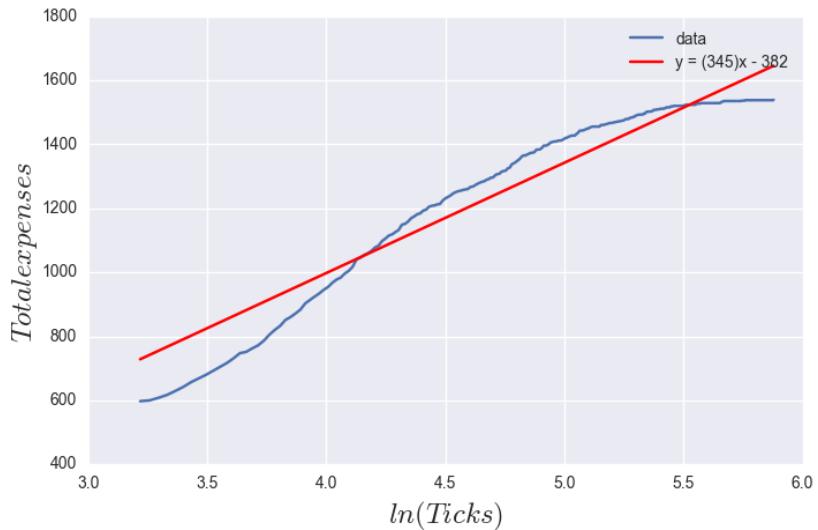


Figura 12: Fit: total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 700 users)

At any instant we have a different average cost of delivery, so we can plot the ratio between expenses and packs vs. time. As shown in the figure below the initial costs are relevant but they decrease rapidly with time.

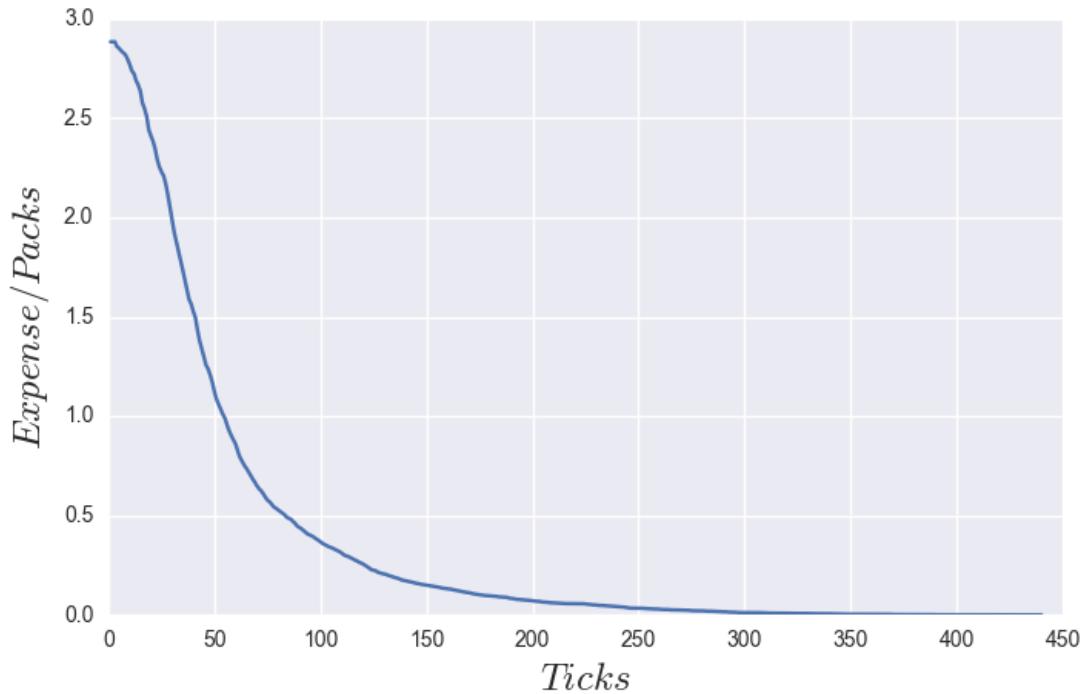


Figura 13: Total expenses/numb packs vs. ticks (10 storages with 304 boxes each, 1400 packs, 700 users)

5 Extreme Cases

As a proof of concept we decided to analyze some extreme cases in order to explore if our solution is robust enough to survive real-life challenges.

Too few

What would happen if there are too few users? Keeping all other parameters as found we set the number of users to one tenth the original value, down to 70 agents total. We observe that the delivery process takes up to two days, instead of the half day in standard situation and the total expenses increase up to 28% more.

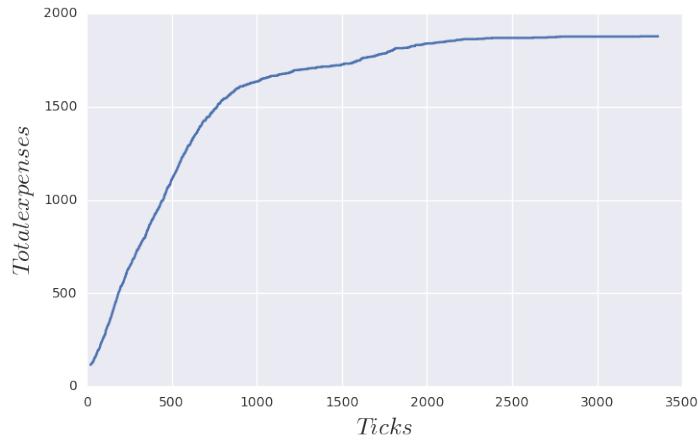


Figura 14: Total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 70 users)

We can have an estimate of the curve slope in log-lin scale.

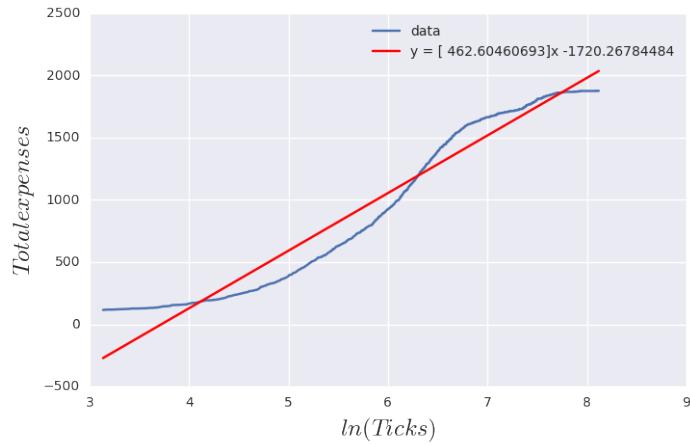


Figura 15: Fit: total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 70 users)

Too many

On the opposite we explore what happens if there are too many users, up to 7000 agents total. As can be expected the delivery process is completed in a risible time and the expenses are down 70%, proving the benefit for the commissioner in expanding the participation.

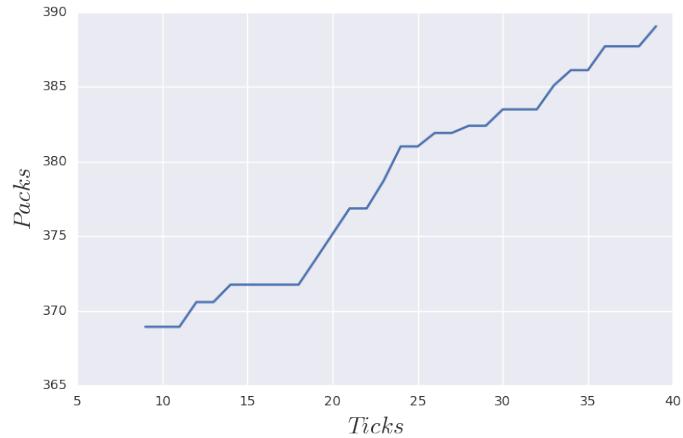


Figura 16: Total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 7000 users)

We can have an estimate of the curve slope in log-lin scale.

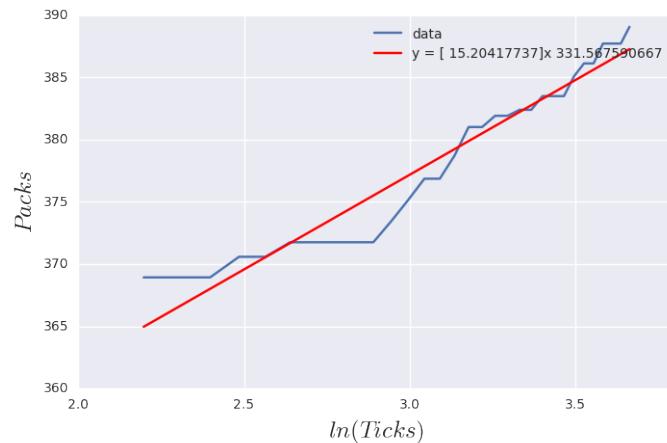


Figura 17: Fit: total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 7000 users)

Wanting more

A reasonable question may that an higher pay may be needed to incentive users to participate in the activity. We examined what would happen if the pay was up to 5€/km of deviation. Similarly to the first case, the process of delivering all packs may take up to two days, but the total expenses will not be too different than the standard situation.

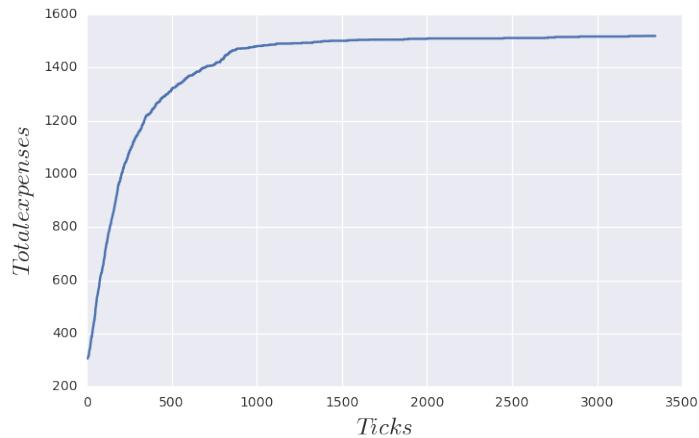


Figura 18: Total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 700 users)

We can have an estimate of the curve slope in log-lin scale.

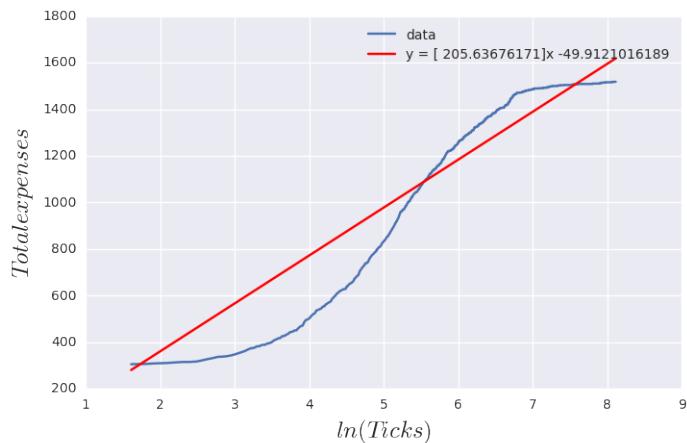


Figura 19: Fit: total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 700 users)

High Maintenance

A success of our method may have the downside of increasing maintenance costs for the storages around the city. Examining a bi-daily maintenance cost five time the standard situation we found an increase in the total expenses around 21%, like the first case examined, but without any other downside.

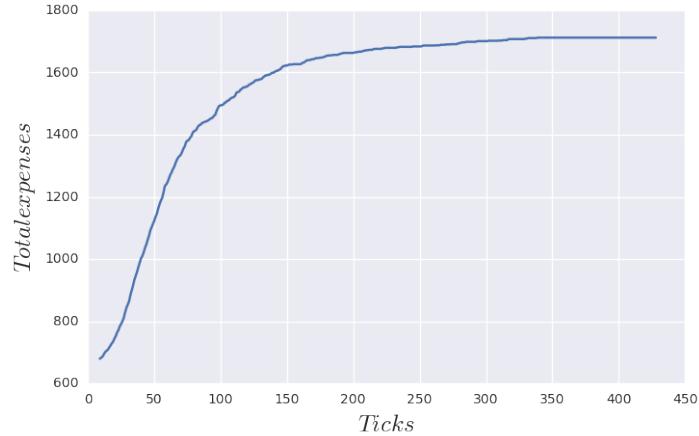


Figura 20: Total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 700 users)

We can have an estimate of the curve slope in log-lin scale.

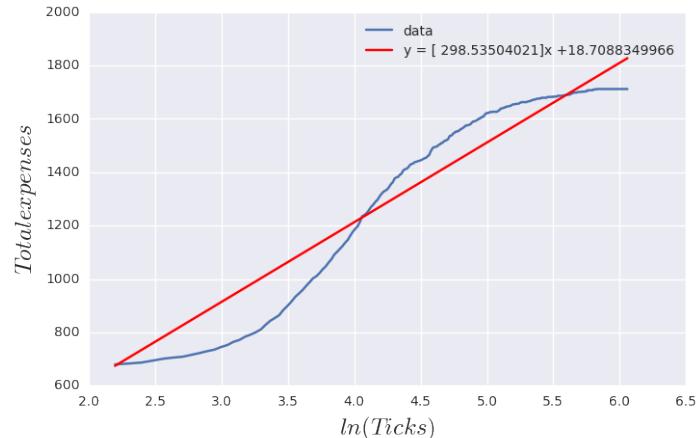


Figura 21: Fit: total expenses vs. ticks (10 storages with 304 boxes each, 1400 packs, 700 users)

)

6 Conclusions

This work has analyzed a new solution for the LMP characterized by the delivery process carried out by citizens. As we shown in the first section, the today city complexity brought a proportional increase of the transportation costs in it; indeed the presented solution, based on the sharing economy, can be ecologically and economically sustainable. Citizen becomes active actors in the transportation process delivering packs all around the city. The Agent-Based Model has been realized with a scaled map in order to express the Turin's distances and adequate (from a computational point of view) parameters:

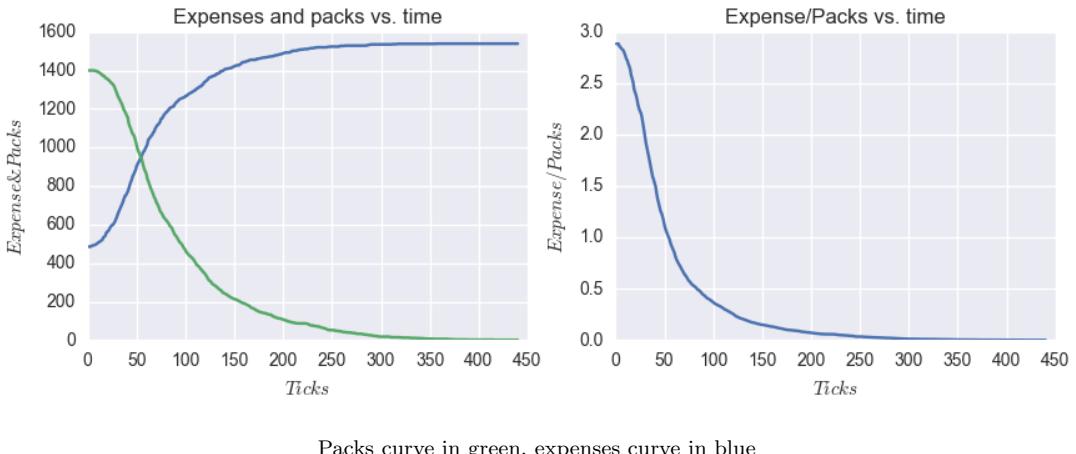
Numb. Packs	Numb. Users	Max Delivery Cost	Storage Box Cost	Users' Pay
1400	700	1.75 €	0.2 €/two days	1 €/km

The code has been written in NetLogo 6.0 and, thanks to the Behavior Search tool, brought the following results for parameters we need in order to minimize the expenses:

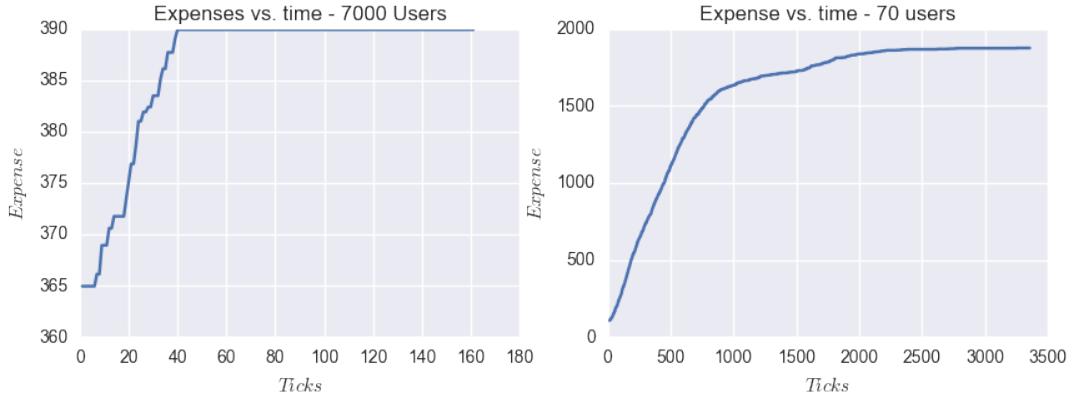
- 10 Storages.
- 304 Boxes in each storage.
- Total expenses: 1440 €

Note that the storage's capacity is greater than the minimum required for an uniform distribution of 1400 packs in 10 storages. This shows how the complex system achieves a store optimization for minimizing the expense regarding random place of delivery.

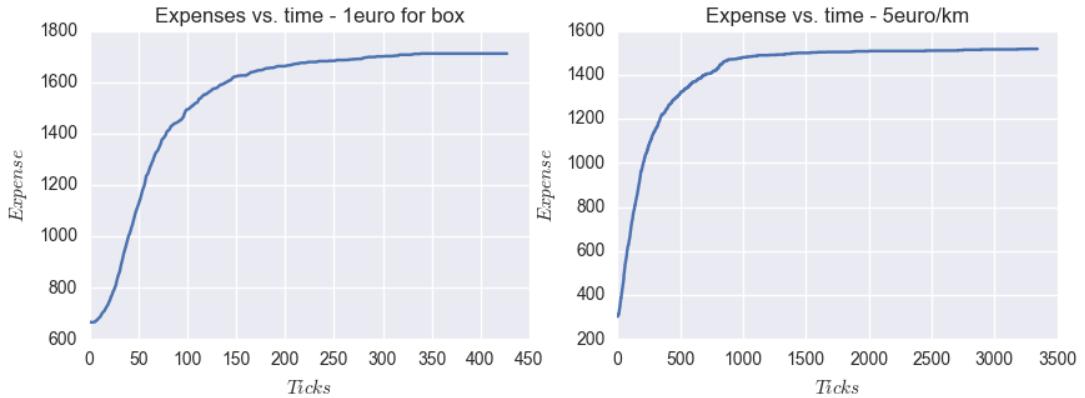
The figure below shows that the curves of expenses and number of packs vs. time have almost an opposite trend; moreover on the right we see that the expenses for each pack decreases with time.



Considering an high and a low number of users we saw that the convergence is faster in the first case and, as expected, less expensive. Note that at the end of the day (1600 ticks) in both cases all packs are delivered.



Increasing the fixed cost parameters we saw that expenses grew, but the model still succeed in delivering all packs in a day (1600 ticks) with a cost for each pack that is lower than 1.75€.



In conclusion this underlines the robustness of the proposed solution both in economical and user's engagement terms, moreover the model held interesting results also in extreme cases.

Further development

The model could be extended by considering:

- improved economic model;
- GIS map of the city of Turin;
- daily data for a real time calculation.