

Article

The “ReadyPark” Collaborative Parking Search Strategy

Jean-David Collard ^{1,*}, , Erick Stattner ^{2,†} and Panagiotis Gergos ^{1,†}¹ AngelsBayTech SAS, 06640 Saint Jeannet, France; panos@angelsbaytech.com² LAMIA Lab., Campus de Fouillole, Université des Antilles, 97154 Pointe à Pitre, France; erick.stattner@univ-antilles.fr

* Correspondence: jdc@angelsbaytech.com

† These authors contributed equally to this work.

Abstract: Many of us have experienced this when driving in the city, we had to “turn for hours” to find a parking space. To attenuate this drawback, this paper proposes a collaborative parking search strategy based on the reality of a drivers network in which, depending on his/her role at a given time, each one can help the others to find public on-street parking spots. To highlight a situation of parking spot scarcity, we assume the extreme case where there is no free space and the only solution to park is to swap with a parked car that leaves its location. Under this assumption, agent-based modelling and simulations allow to compare the ReadyPark solution with the classical approach that consists, by luck, in taking the place of a nearby car that is leaving its spot.

Keywords: smart city mobility; parking spot search; collaborative strategy; agent-based modelling and simulation



Citation: Collard, J.-D.; Stattner, E.; Gergos, P. The “ReadyPark” Collaborative Parking Search Strategy. *Smart Cities* **2021**, *4*, 1130–1145. <https://doi.org/10.3390/smartcities4030060>

Academic Editor: Pierluigi Siano

Received: 26 June 2021

Accepted: 2 August 2021

Published: 12 August 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A problem shared by most drivers in cities is the time spent looking for a parking spot. Apart from the owners, tenants or beneficiaries of a parking space, this wastes everyone’s time and has even a deleterious effect on the society. Indeed, cruising is not only bothersome but in addition the riding time generates a series of cascading externalities: (i) economic damages (turnover loss for retailers located in critical areas, time subtracted to work, higher fuel consumption, delays, etc.); (ii) pollution increase (3 to 15 extra min per trip per car of useless emissions); (iii) traffic congestion (30% of congestion generated by cruising in San Francisco) and consequent reduced efficiency of public transport systems; (iv) negative impact on the well-being of the driver (time invested in parking instead of meaningful or productive activities, stress accumulation and more time sit); and (v) negative impact on the well-being of any citizen (emission and sound pollution levels, traffic, and pedestrian safety).

In European big cities, cruising time generates an average 30% additional traffic [1]. For smaller cities the rate is 5–10%. Overall, 70 million hours are lost per year, which represents an economic loss of 600 M€ per year [2]. In New York City, the hardest-hit U.S. urban area, drivers on average waste 107 h a year, with 2.243 \$ in wasted time, fuel, and emissions per driver, plus 4.3 billion \$ in costs to the Big Apple [3]. Cars individual use will not disappear any time soon: actually global car sales are rising. With 68% of the human population living in urban areas by 2050 (United Nations estimate), we need to act urgently to reduce the impact of mass car circulation on society and the environment, with solutions implementable in the span of a few months, cost-effective and with no need of specific infrastructural interventions.

In the general context of the smart city, the previous issue makes sense as mobility was identified by Guedes et al. [4], along with 7 out of 20 others, as having the “highest priority for the development of smarter and more sustainable cities”. More specifically, Munhoz et al. [5] carried out a broad study on the main factors favouring intelligent urban

mobility; among these, the authors highlighted “cooperation between stakeholders”, “technological innovation” and “smart parking”, all characteristics that fit well to the ReadyPark solution presented in this paper.

1.1. Parking Spots Supply & Demand

If we follow the supply/demand metaphor, the resources here are the parking spots. A *giver* is a driver whose vehicle is parked and will eventually leave from the parking space to “give” his/her spot, and a *taker* is a driver who is driving on the streets and will eventually look for a parking space to “take” the spot. Each driver is alternately *producer* (i.e., giver) and *consumer* (i.e., taker) of resources. We are therefore in a *producer-consumer* schema and the problem is to balance at any time supply and demand. The difficulty to match supply and demand is that (i) the numbers of takers and givers may markedly differ and may even vary over time and that (ii) the driving time and the parking time are in most cases independent.

In order to balance supply with demand, and so to favour swaps, it is necessary that the two frequencies of occurrence, search for a spot by a taker and disposal of his spot by a giver, are close, that is the Supply/Demand ratio (hereinafter *SD_ratio*) (Equation (1)) is close to 1.

$$SD_ratio = \frac{|givers|}{parking_time} \times \frac{driving_time}{|takers|} \quad (1)$$

where *parking_time* is the time during which a vehicle is parked and *driving_time* is the time between two consecutive parking of a vehicle.

Of course, in all generality, there may be an imbalance between supply and demand but the problem addressed here is the lack of available parking spaces for the number of drivers looking for a spot, that is, when demand exceeds supply, i.e., *SD_ratio* < 1.

For there to be an exchange of places between a taker and a giver, there must be a *space-time coincidence*; that is, the two drivers must be *in the same area at the same time*. It is worth noting that this statement makes sense only in situations without intelligent parking allocation. *Time coincidence* depends both on the density of vehicles in the city and the number of parking spots. Without intelligent parking allocation, *spatial coincidence* requires the taker and the giver to be very close to each other, and perhaps even the former on the street and the latter in the adjacent parking spot. In order to relax this last constraint, we propose the *ReadyPark* solution based on a collaborative parking search strategy.

1.2. Previous Works

The field of smart cities has experienced strong growth in recent years [6]. In this context, one of the most significant issues addressed relates to the management of urban transports [7,8]. The challenges addressed in this area are traditionally classified according to three axes: *Parking spaces*, *Traffic information*, and *Trajectories*. The common point of all these works is the need for efficient data sensors to improve the quality of services. This has been pointed out by Illari et al. [9] who compared traditional *wireless sensor networks*, for which sensors are statically deployed on the ground to *collaborative sensing*, assuming mobile crowdsensing devices or participatory sensing, to highlight how the collaborative dimension provides key benefits in urban transportation.

In this paper, we focus on the problem of *parking space* management that is an important challenge in big cities for reducing the traffic, limiting vehicle energy consumption and decreasing greenhouse gas emissions [10]. As shown in an article published in 2016 [11], this problematic is a major issue in urban management, since Aliedani et al. studied the impact of searching for parking spaces and shown that approximately 730 tons of carbon dioxide (CO₂), 95,000 h of time and 47,000 gallons of fuel are used per year in finding of parking in Los Angeles (USA). Such impacts had already been observed in a 2006 study [12].

Thus, numerous work have been proposed for intelligent management of parking spaces [13]. Pioneering works in the field have exploited hardware sensors that were able to capture raw data directly from the environment. The best-known projects concern sensor networks located on sidewalks [14], fixed or mobile cameras [15,16] capturing parking spaces or ultra-sonic sensors on parking lots [10]. The information captured is sent to a central server which has got an overall information on free parking spaces.

With the development of the IoT (Internet of Things), some approaches have proposed IoT-based architecture to address the parking space problem. For instance, Lanza et al. [17] propose an IoT platform able to provide real-time data about available parking places both indoor and outdoor. The available parking spots can then be visualised on a mobile app.

More recent works have addressed this issue as a complex system [18] in which agents interact to facilitate the management of parking spaces. The very strong potential of collaborative approaches to address parking space problem had already been highlighted in [9], where the authors showed how *“collaborative sensing can provide key benefits in urban transportation, contributing to higher travel efficiency, safety, and reduced pollution, through innovative applications that benefit from the data sensed collaboratively”*. For instance, Higuchi et al. [19] present a collaborative approach, called *Co-Park*, in which a group of connected vehicles communicate, forming a *vehicular micro cloud* cluster, in order to find available parking spots in a timely fashion. The approach was evaluated with a simulation that highlighted the reduction of the trip time for participating vehicles against non-participating vehicles.

On the principle of the collaborative model, *GaParking* [20] proposes an approach to share private residential parking spaces between residents and public users. In this approach, parking spot owners are supposed to register their vacant parking space in the model which can be reserved by a driver. By using numerical experiments, the authors show how the proposed sharing parking strategy makes use of private resources to benefit the community. Popular applications based on this principle are now available such as *JustPark* [13] or *MobyPark* [21].

While these approaches are popular, they nevertheless assume that available parking spaces and their duration are known in advance. However, in most cases, drivers turn in the streets searching for a free parking space. To allow a more dynamic management of parking spaces between those who leave a parking space and who are looking for one, some studies have adopted a multi-agents based models to simulate various kinds of driver collaborative behaviours. For instance, Chalamish et al. [22] use a multi-agent based model to simulate different parking search strategies. This approach has thus allowed to highlight the evolution of the average search time according to the search strategy adopted.

Other works have attempted to take into account requirements on some parking attributes. Di Napoli et al. [23] use a multi-agent system to address the parking space allocation as the result of an agreement between parking providers and parking requestors. The approach takes into account user requirements on a parking space in terms of its location and cost, and the vendor requirements in terms of income and city regulations to obtain an efficient parking allocation and traffic redirection.

1.3. ReadyPark Collaborative Parking Search Strategy

ReadyPark is an urban parking assistance service which facilitates in real-time the search for a parking spot. It is based on a collaborative principle, enabling users peer-to-peer swaps of on-street parking spots.

With *ReadyPark*, the key point is that a strict spatial coincidence is not required to swap spots. More concretely, a swap is based on pairing one giver with one taker: when a giver is available-to-leave-spot, a taker searching-for-spot is selected by *ReadyPark* on the basis of relevant criteria such as proximity to the giver, car attributes, sharing activity level, but also specific characteristics of the taker, such as people with disabilities, medical profession, home care workers, etc., and then is guided to the spot via a turn-by-turn navigation solution provided by a mobile application. In the end, the giver is gratified if he waited to swap with the selected taker, for instance, by increasing his “sharing indicator”.

At this point, one might think that the ReadyPark strategy leads a giver to wait for a long time for the arrival of the taker and thus lower the availability of places. This statement calls for two remarks: (i) the simulations, especially those on the minimal model (Section 3.2), show that the waiting time is in fact very short compared to the parking time and driving time; (ii) the exchanges are in a collaborative mode in which the givers will in turn be takers, and vice versa, so it can be assumed that everyone will behave altruistically by agreeing to “lose a few seconds now to get a parking spot more quickly later”; nevertheless, in real operational situations, the acceptability of waiting will naturally be regulated as givers will not wait beyond a reasonable time.

To sum up, ReadyPark proposes a collaborative parking spot search strategy. The objective of the paper is to study its dynamics and its impact through agent-based modelling and simulations in order to refine the implementation of future operational versions. The present study is the result of a joint effort between an IT company, founded by two engineers, and a public research laboratory specialised in agent-based modelling and complex systems. The company developed a proof-of-concept in the form of a digital collaborative service based on a mobile application using Galileo technology (The Europe’s Global Satellite Navigation System (<https://www.euspa.europa.eu/european-space/galileo/What-Galileo>, accessed on 26 June 2021)) and a collaborative platform (web services) to implement the parking spot search strategy. This POC has won a European competition organised by the European Space Agency at the end of 2019 and is deployed in South-East France.

1.4. Paper Organisation

The remainder of this paper is articulated into three main parts. Section 2 focuses on the *No-Free-Spot* mobility model and its parameters. In Section 3, we conduct computational simulations of the model and presents the results. Finally, we conclude with a summary of our findings and discuss future work.

2. No-Free-Spot Mobility Model

Agent-based modelling has become a widespread simulation modelling technique that has provided numerous applications to real problems as, for instance, traffic, evacuation, stock market and diffusion of innovation [22–25].

The *No-Free-Spot mobility model* (hereinafter *NFS*), based on the eponymous hypothesis, proposes an agent-based modelling approach where agents are drivers moving in the spatial street network of a virtual city. The problem being to find an on-street parking spot, beyond the standard solution which requires two drivers to interact *directly*, the model proposes a solution which allows, through an *indirect* interaction, to pair a “giver” with a “taker” so that they can ultimately exchange their respective position.

Perhaps because of advances in computer performance, parsimonious models are not common in research on traffic; the preference is for detailed models that describe individual behaviour very well [26]. Despite this, the *NFS* model does not seek to take into account most of the complications inherent in real-life; rather, we tend to use a *parsimonious* modelling approach that, in relation to the issue, balances appropriate level of detail with relevance of the simulation results. Of course, at a later stage, more detailed approaches can be deployed to refine the ReadyPark strategy to achieve future versions.

2.1. No-Free-Spot Hypothesis

The No-Free-Spot hypothesis states that the only way for a driver to park is to swap with a parked car that is leaving from its spot. The term “free spot” is used here in the meaning of a *vacant space* and not of a parking spot that can be used *free of charge*.

In line with the issue of finding a parking space, it is important to note that this assumption reflects a more difficult situation than those where there are some unoccupied parking spots. Nevertheless it should be noted that such a situation is frequent in saturated city centres.

Based on the observation of drivers' habits in everyday life, both the driving time and the parking time can be set at reasonable values, let us say to fix ideas 20 min for the former and 90 min for the latter. According to the *NFS* assumption, the number of parked cars ($|givers|$) is fixed and equals to the number of parking spots and therefore depends on the topology of the city only.

It should be noted that the number of parking spots in a city may depend on many factors, not just the topology of the city; nevertheless, any modelling approach requires a simplification of reality, and, here, it is the very problematic addressed and examined under the *NFS* hypothesis (a rather difficult configuration to find parking spots) that leads us to consider the topology only. As a consequence, the supply/demand ratio (Equation (1)) can be monitored by the choice of the number of drivers on the streets (i.e., $|takers|$).

In the following, we present the different components of the *No-Free-Spot* mobility model. We detail, first the city sub-model with its street network, then how vehicles move through this network, and finally the dynamic process that leads to parking spot exchanges. A key point of this paper is that the simulations will make it possible to know if, in a situation where the supply of parking spots is low and therefore cannot satisfy a naturally increasing demand, a community of collaborating drivers, using a solution such as *ReadyPark*, can improve for itself the number of swaps without disadvantaging other drivers.

2.2. Manhattan Model

The “world” is modelled by a 2-D grid that represents an idealised city with a street network structured according to the *Manhattan model*, i.e., a uniform network of horizontal and vertical streets that delimit building blocks of similar sizes. A grid cell is either in a building, a street, a crossroad or a parking spot (Figures 2 and 3).

The Manhattan model is commonly used to simulate both the street network of a city and the movements of drivers on this network. We chose this model because it refers to the spatial dependency of a real city, it is able to capture many mobility scenarios and it is easily extensible and adaptable to a particular issue [27–29].

2.3. Mobility Model

Mobile agents, immersed in the Manhattan city model, represent drivers that are either moving on the street or motionless in a parking spot. As soon as it is positioned on a street, a car moves at a constant speed—this is the same for all cars. The choice of the new direction in a crossroad is either the shortest route to the giver or a random way (see below).

2.3.1. Driver Status and Role

Recall that a *giver* is a driver who is parked and will eventually leave his parking spot and give it, and a *taker* is driving on the street and will eventually look for a parking spot to take the place. Each driver is alternately giver then taker.

As *ReadyPark* is both *open* to everyone and *optional*, drivers may or may not be *ReadyParkers*; therefore, a driver is foremost described by his/her *status*, either *ReadyParker* or *noReadyParker*, which remains constant over time. In addition, a driver has a *role*, either giver or taker, which may vary over time.

Therefore, for each driver, at each time step, there are two independent features, one fixed and one variable, which lead to the definition of four cases. The set of *ReadyParker* taker (resp. giver) will be denoted by *RPtaker* (resp. *RPgiver*) and the set of *noReadyParker* taker (resp. giver) by *noRPtaker* (resp. *noRPgiver*) (Table 1).

Table 1. Driver attributes.

Name	Time Dependent?	Values
<i>status</i>	no	ReadyParker or noReadyParker
<i>role</i>	yes	taker or giver
<i>state</i>	yes	see Table 2
<i>paired?</i>	yes	boolean
<i>paired_taker</i>	yes	driver
<i>paired_giver</i>	yes	driver
<i>close_to_pair?</i>	yes	boolean

2.3.2. Dynamic Process

Depending on his/her *role*, a driver has an internal *state* whose value changes in the course of his lifetime; the state can take on seven different values (Table 2). Below, we present the possible state values and especially the conditions of transition from one state to the next either triggered by an external event or a condition, or by time elapsed.

Table 2. Drivers states.

Taker	Giver
[1'] DRIVING-ON-STREET	[1"] PARKED
[2'] SEARCHING-FOR-PARKING-SPOT	[2"] AVAILABLE-TO-LEAVE-SPOT
[3'] DRIVING-TO-THE-PAIRED-GIVER	[3"] WAITING-FOR-THE-PAIRED-TAKER
[4] SWAPING	

A natural and didactic way to represent the dynamics of such individuals with their states and transition between states, is to use a *finite state diagram*. Although this type of representation was originally used for modelling computer systems, it is common to use it to represent current state and dynamics in complex systems. The finite state diagram specifies the sequences of states taken by a driver during his lifetime. A state is a situation during which he fulfils a certain condition, performs a certain activity or expects a certain event. A transition is a correspondence between two states meaning that the driver in the first state enters the second state as soon as one event occurs and/or some conditions are met [30–32].

Figure 1 shows the driver state diagram with states represented by rectangles with rounded corners and transitions by directed lines. There are seven states and eight transitions. In the following we will describe each of these transitions; the ones for takers are numerated [1'], [2'], [3'] and [4], while for givers numerates are [1''], [2''], [3''] and [4].

Initially, a driver is either moving on-street or parked in a spot since some time. In the first case, his/her current role is that of a potential “taker” and in the second a potential “giver”. More formally, at time step $t = 0$, *state* is DRIVING-ON-STREET [1'] or PARKED [1''] and *role* is, respectively, taker or giver.

States transitions for takers (Figure 1: left side of the diagram)

[1' → 2']

The transition from DRIVING-ON-STREET to SEARCHING-FOR-PARKING-SPOT is triggered by the time elapsed parameter *driving_time*. In order to desynchronise the drivers, at time $t = 0$, we use a *random uniform* distribution to assign to each one a current driving duration in the range $[0; driving_time]$ (At the very beginning of state [1'], if $t \geq 1$, the current driving duration is set to zero.). Let us note that, in states [1'] and [2'] the choice of the new direction in a crossroad is random.

[2' → 3']

As soon as the taker finds a giver AVAILABLE-TO-LEAVE-SPOT (i.e., in state 2"), his state moves from SEARCHING-FOR-PARKING-SPOT to DRIVING-TO-THE-PAIRED-GIVER. In the same time, the giver is paired with the taker. Of course, the pairing strategy depends on the status of the drivers, ReadyParker or noReadyParker (see Section 2.4).

[3' → 4]

This transition occurs as soon as the taker is close to his paired-giver. To get closer to his/her pair, at each crossroad, the taker chooses one of the shortest routes.

[4 → 1"]

The physical exchange of positions between the taker and his paired-giver is performed and the taker is parked and becomes a potential giver.

State transitions for givers (Figure 1: right side of the diagram)

[1" → 2"]

Transition from PARKED to AVAILABLE-TO-LEAVE-SPOT is triggered by the time-elapsing parameter *parking_time*. To desynchronise the drivers, at time $t = 0$, we use a *random uniform* distribution to assign to each one a current parking duration in the range $[0; parking_time]$ (At the very beginning of state [1"], if $t \geq 1$, the current parking duration is set to zero.).

[2" → 3"] For the giver, transition to WAITING-FOR-THE-PAIRED-TAKER occurs as soon as a taker is paired with him (see Section 2.4).

[3" → 4]

The transition occurs as soon as the giver is close to his/her paired-taker.

[4 → 1']

The physical exchange of positions between the giver and his paired-taker is performed and the giver is driving and becomes a potential taker.

During a run, a ReadyParker successively takes the following states that represents what we call a *basic cycle*.

$$[1' \rightarrow 2' \rightarrow 3' \rightarrow 4 \rightarrow 1'' \rightarrow 2'' \rightarrow 3''] \quad (2)$$

The entire process for one driver is a succession of such cycles. Let us note that a swap involves two paired drivers and state [4] is the state in which the taker and the giver, synchronise, that is, each one enters state [1'] or [1"] at the same time.

2.3.3. Duration of a Transition

It is instructive to compare the relative duration of the transitions to determine which would be predominant and which would be negligible. This will be useful in emphasising the applicability of the approach.

- Transitions [1' → 2'] and [1" → 2"] depend on the global parameters *driving_time* and *parking_time*, respectively. Their duration is therefore fixed.
- Transitions [2' → 3'] and [2" → 3"], depend mainly on the Supply/Demand ratio with unrelated duration which results from the dynamic process.
- Transitions [3' → 4] and [3" → 4] are coupled also and their duration is the time taken by a taker to reach his paired giver by the shortest route. Again, this time results from the dynamic process.

To sum up, the duration of transitions [1' → 2'] and [1" → 2"] are input parameters of the model and, *a priori*, in an order of magnitude greater than that of the other transitions. For these latter it is hoped that duration is small, or even, in real situations, negligible compared to the former as the experimental results will show (Section 3).

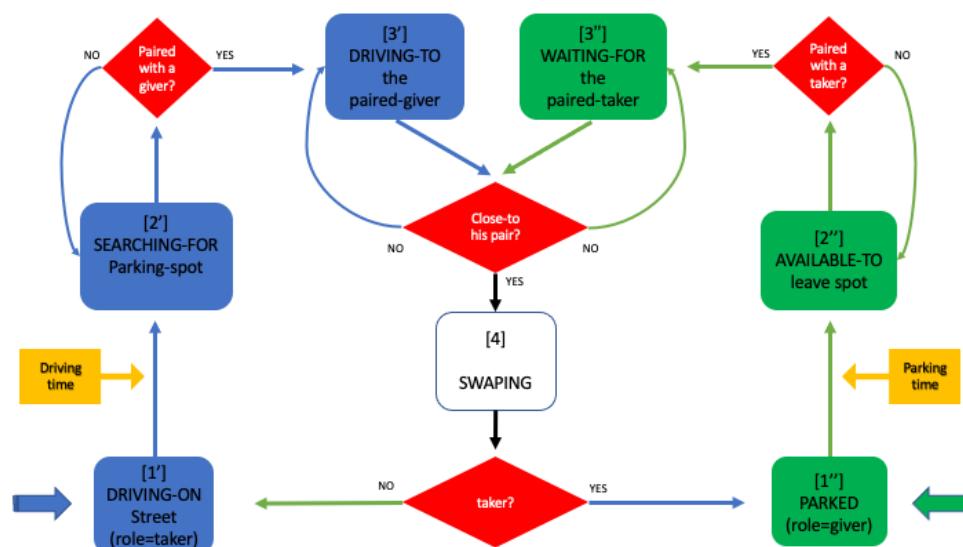


Figure 1. Drivers state diagram.

2.4. Pairing Strategy

Here, we examine the pairing strategy between a taker and a giver depending on the status of the driver, ReadyParker or not.

Each RPgiver prioritises pairing with RPtaker; however, to keep the process fair for noRPtakers, the *priority_time_for_RPtaker* parameter limits in time this priority status (Table 3). Of course, in all cases, to be paired, a giver must be in state [2''] AVAILABLE-TO-LEAVE-SPOT and not already paired.

Table 3. Global parameters.

Name	Meaning	Values
<i>speed</i>	car speed	$\mathbb{N}^*$
<i>driving_time</i>	duration of a street trip	$\mathbb{N}^*$
<i>parking_time</i>	duration of a parking period	$\mathbb{N}^*$
<i>priority_time_for_RPtaker</i>	priority time for RPtaker	$\mathbb{N}^*$
<i>cycle_time</i>	basic cycle time	$\mathbb{N}^*$
$ givers $	number of givers	$\mathbb{N}^*$
$ takers $	number of takers	$\mathbb{N}^*$
<i>SD_ratio</i>	Supply/Demand ratio	[0; 1]
<i>RP_ratio</i>	ReadyParkers proportion	[0; 1]

2.4.1. No-ReadyParker Pairing Strategy

Under the *NFS* assumption, the solution for a noRPtaker to find a spot is to move around the streets until a nearby giver leaves his current spot. If the giver is a ReadyParker, the time during which it gave priority to ReadyParkers to pair up must have expired (see Algorithm 1).

Depending on the Supply/Demand ratio, the duration of transitions [2' → 3'] and [2'' → 3''] may be excessively long. In contrast, the duration of transitions [3' → 4] and [3'' → 4] are close to zero.

Algorithm 1 Paired algorithm for noRPtaker

```

G ← givers with [(distance myself ≤ 1) and (not paired?) and (state =
AVAILABLE-TO-LEAVE-SPOT)]
G1 ← G with [status = noRP]
G2 ← G with [status = RP and (current_priority_time ≥ priority_time_for_RPtaker)]
if (G1 ∪ G2) ≠ ∅ then
    paired? ← true
    paired_giver ← min-one-of (G1 ∪ G2) [distance myself]
    ask paired_giver [paired_taker ← myself; paired? ← true]
end if

```

2.4.2. ReadyParker Pairing Strategy

For a RPtaker, the way to find a spot is to move around the streets until a giver leaves from his current spot; such a giver is either a nearby noReadyParker or a more distant ReadyParker (Algorithm 2).

Depending on the Supply/Demand ratio, the duration of transitions [2'→3'] and [2"→3"] may be long but quiet less than the ones for noReadyParker. In contrast, the duration of transitions [3'→4] and [3"→4] are negligible.

Although not mentioned in the RPtaker pairing algorithm, to reinforce the *social responsibility* of the ReadyPark solution, in future developments, the pairing strategy between a RPtaker and a RPgiver will take into account as a priority criterion not only the distance between the taker and the giver, but also, for instance, specific characteristics of the taker, such as people with disabilities, home care providers, parents or even specific times of day and specific locations as school start and end time.

Algorithm 2 Paired algorithm for RPtaker

```

G ← givers with [(not paired?) and (state = AVAILABLE-TO-LEAVE-SPOT)]
G1 ← G with [status = noRP and distance myself ≤ 1]
G2 ← G with [status = RP]
if (G1 ∪ G2) ≠ ∅ then
    paired? ← true
    paired_giver ← min-one-of (G1 ∪ G2) [distance myself]
    ask paired_giver [paired_taker ← myself; paired? ← true]
end if

```

3. Simulations and Results

This section, devoted to the simulation of the NFS model, aims to present and discuss the experiments with their results.

Experimental simulations are performed with the *NetLogo* multi-agent programmable environment [33–35]. As all the algorithms are provided and the source code is available on request from the author, computational results can be easily replicates. The results can be interpreted via qualitative data and/or plots obtain from significant runs or averaged over 50 independent runs.

The world is a 99 × 99 grid of *NetLogo patches*. A building block consists of 10 × 10 patches surrounded by streets. A crossroad corresponds to a patch at the intersection of 4 streets. To ensure a fix numbers of cars in the city, the world wrap horizontally and vertically. Driver-agents are represented by *NetLogo turtles* with square shape for ReadyParker and triangle shape for noReadyParker (Figures 2 and 3).

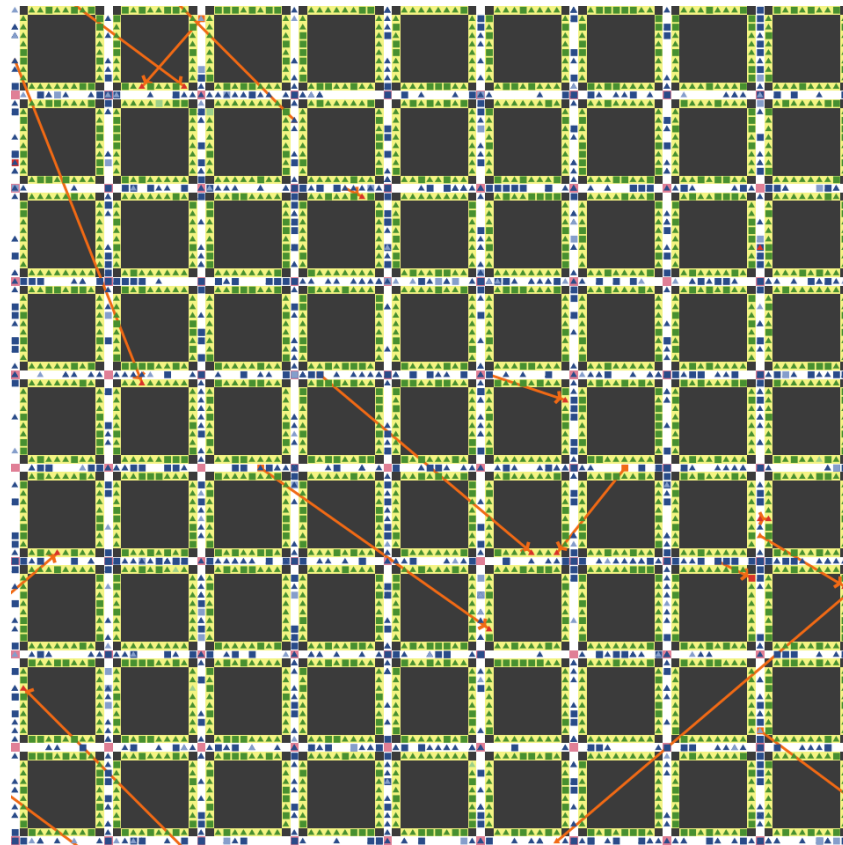


Figure 2. *ReadyPark* space: Building (black); street (white); parking (yellow); crossroad (pink); ReadyParker (square); noReadyParker (triangle); pairing (red link).

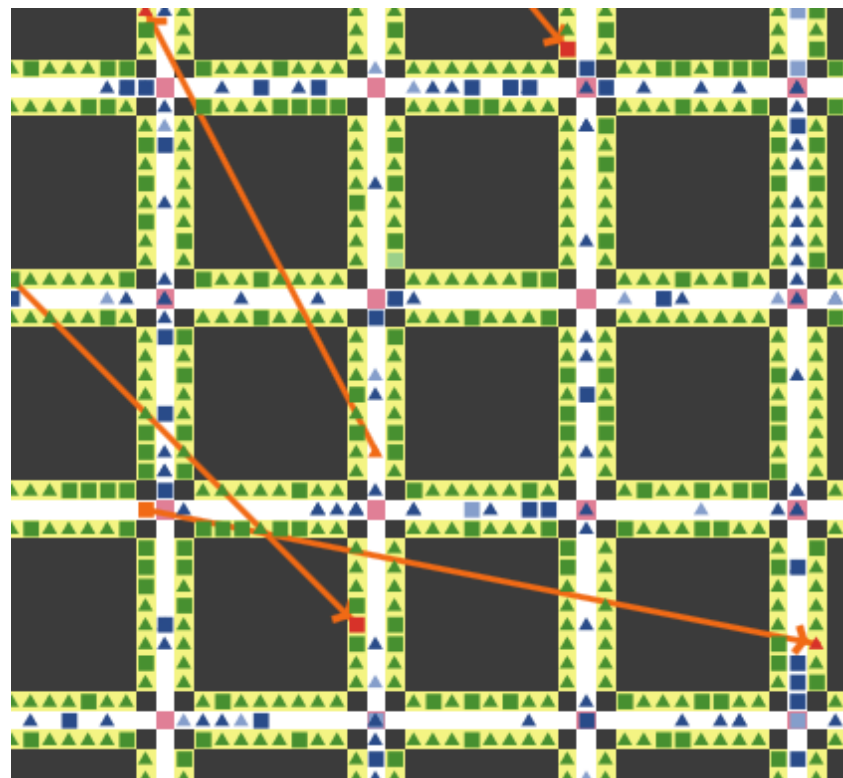


Figure 3. Focus on the *ReadyPark* space: Building (black); street (white); parking (yellow); crossroad (pink); ReadyParker (square); noReadyParker (triangle); pairing (red link).

3.1. Calibration

A NetLogo *patch* corresponds to a square space of 5×5 m. Therefore, the city is made up of building blocks of 50×50 m. The length of a parking spot is 5 m and between two consecutive crossroads there are 10 parking spots, or 50 m.

Time is discretized and all cars move at the same speed. A car travels at a speed of 15 km/h, so it covers 50 m (the distance between two consecutive crossroads) in 12 s. As during a NetLogo *tick* (i.e., a time step for the simulator) a car moves forward one patch (i.e., 5 m), one tick corresponds to 1.2 s; this is the frequency of updating the drivers state and position.

Based on the observation of drivers' habits in everyday life, the *driving_time* and the *parking_time* are set to 20 m (time to drive 5 km) and 90 m, respectively. Let us remember that each RPgiver prioritises pairing with RPtakers during a limit in time, the *priority_time_for_RPtaker* parameter. We set it at a reasonable short time of 2 m (Table 4).

Table 4. Global parameters (calibration for simulation)

Name	Values
<i>speed</i>	15 km/h
<i>driving_time</i>	20 mn
<i>parking_time</i>	90 mn
<i>priority_time_for_RPtaker</i>	2 mn
<i> givers </i>	2592 or 1 for the minimal model
<i> takers </i>	? or 1 for the minimal model
<i>SD_ratio</i>	[0; 1]
<i>RP_ratio</i>	[0; 1]

3.2. Simulations of the Minimal NFS Model

In a first step, we simulate a minimal version of the NFS mobility model with only two drivers in the city (one taker and one giver). We consider different situations depending on the status of the drivers, ReadyParker or not, and the supply/demand imbalance. The interest of such a minimal model is to (i) illustrate the drivers interdependence, (ii) highlight the interest of a collaborative strategy and (iii) visualise the influence of the supply/demand ratio. Let us note that this ratio depends here on the *driving_time/parking_time* ratio only (Equation (1)).

Three diagrams are plotted where time is the *x*-axis and the *y*-axis represent the current state number (from 1 to 3) with *positive* values for takers and *negative* values for givers.

As state number 4 is represented at $y = 0$, one swap corresponds to a crossing between a vertical line and the *x*-axis. Such diagrams give a dynamic view of the NFS model (Figures 4–6).

Assuming that demand matches supply (i.e., *driving_time* = *parking_time*), (i) for two ReadyParkers, swaps occur regularly according to a period close to the common value of the parking and driving times (Figure 4); while, (ii) for two noReadyParkers, swaps occur when a taker, by chance, is close to his paired-giver only (Figure 6).

Figures 4 and 5 allow to compare the dynamics of two ReadyParkers assuming that supply is either equal (*SD_ratio* = 1) or lower than demand (*SD_ratio* = 0.7). In the latter case, we observe that the swap period is longer because the taker in state [2'] has to wait for the giver in state [1"] to become available to leave from his spot (Figure 5).

To sum up, (i) the number of swaps is clearly increased with ReadyParkers compared to noReadyParkers, (ii) and, obviously, a weak supply has a negative impact on the overall performance.

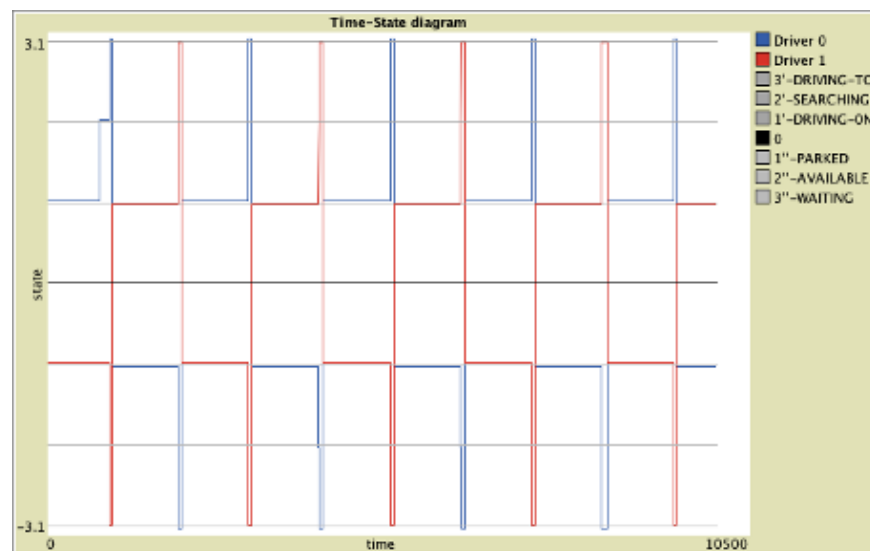


Figure 4. Minimal NFS Model with two ReadyParkers and $supply = demand$.

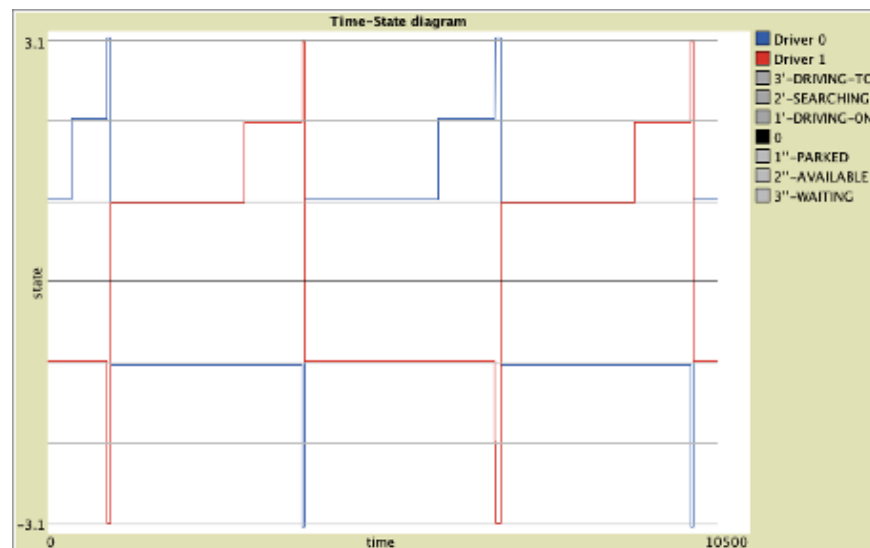


Figure 5. Minimal NFS Model with two ReadyParkers and $supply < demand$.

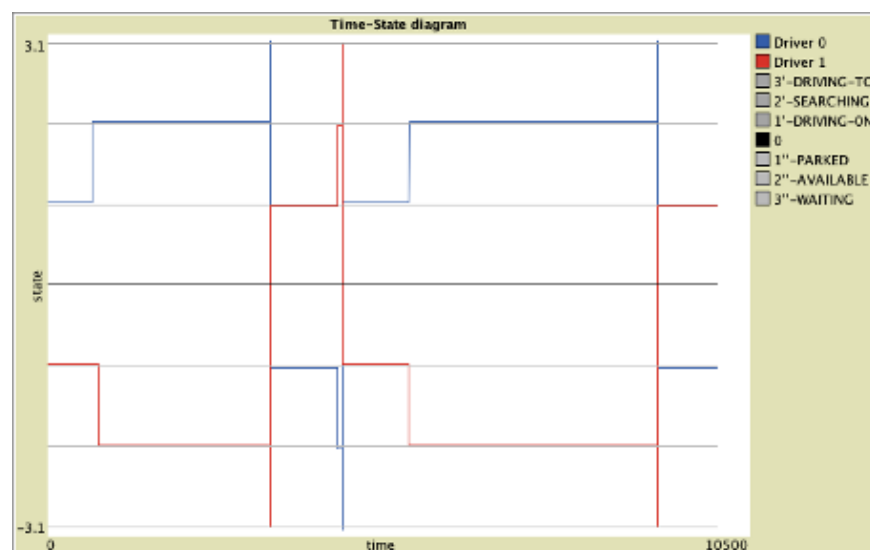


Figure 6. Minimal NFS Model with two noReadyParkers and $supply = demand$.

3.3. Simulations of the NFS Model

To simulate the full NFS model we assume that all the parking places on both sides of the streets are available to park a car, that is, the number of cars on a parking spot is 2592 (Figures 2 and 3). The overall performance of the system is measured by the number of swaps *per driver* and *per basic cycle* (see Section 2.3.3). In the following, we study the influence of the ratio of ReadyParkers in the driver pool on the overall performance.

3.3.1. Supply = Demand

Here, we assume that supply in parking spot is equal to demand ($SD_ratio \approx 1$). Concretely, we set the number of takers to 575 and the number of givers to 2592.

A. priori, this situation should not be so bad for noReadyParkers and, indeed, we observe on Figure 7 that without ReadyParkers (i.e., $RP_ratio = 0$) the performance is 0.960, thus close to the maximum of 1. However, in a mixed environment (i.e., $RP_ratio > 0$), the performances of a network of collaborative ReadyParkers are better; moreover, as RP_ratio increases, (i) the network performance increases from 0.960 to 0.975 and (ii) the noReadyParkers performance decreases to reach the 0.84.

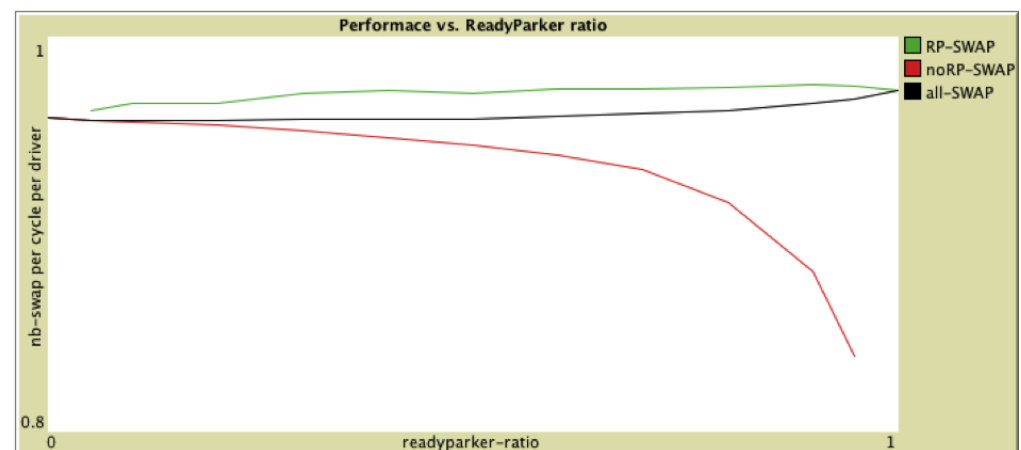


Figure 7. NFS Model: Performance vs. RP_ratio with supply = demand. 24,000 time steps for each RP_ratio value (eq. 8 h).

3.3.2. Supply < Demand

Here, we assume that supply cannot meet demand ($SD_ratio = 0.5$). Concretely, we set the number of takers to 1150 and the number of givers to 2592. Such a situation is obviously more critical for noReadyParkers, and the question is whether collaboration between drivers can be effective.

Indeed, we can see on Figure 8, that without ReadyParkers (i.e., $RP_ratio = 0$) the noReadyParkers performance is downgraded (0.84 compared to 0.96), indicating that drivers on-street have to wait for a long time to find a parking spot. Moreover, as the RP_ratio increases, the noReadyParkers performance decreases, first slowly and then more quickly, to reach a value close to zero.

In contrast, the performances of a collaborative ReadyParkers network remain high and relatively constant. For instance, the performance is above 0.97 if $RP_ratio < 0.7$. More, if the RP_ratio is less than 0.20, the noReadyParkers performances are slightly affected; this result shows that for an optimal mix, the collaborative network should not be more than about 20% of the total population.

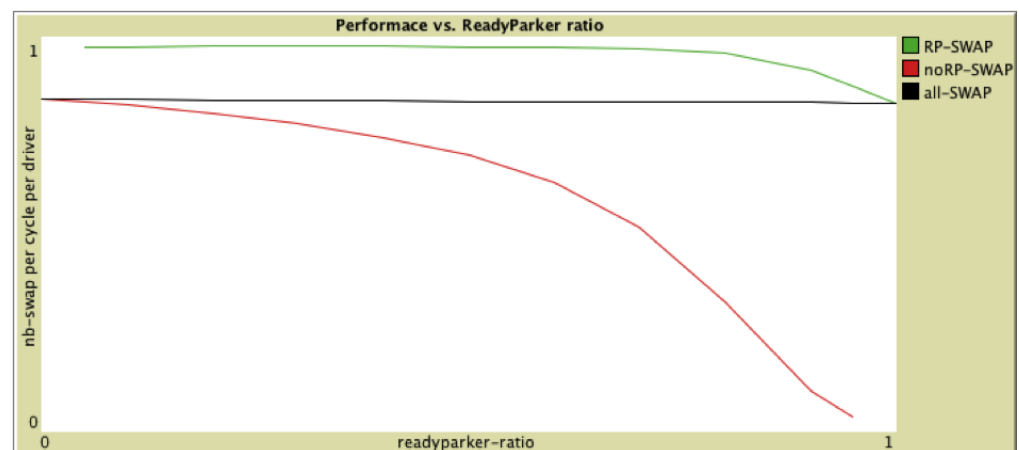


Figure 8. NFS Model: Performance vs. RP_ratio with $supply < demand$ ($SD_ratio = 0.50$). 24,000 time steps for each RP_ratio value (eq. 8 h).

4. Conclusions and Future Work

In this paper, we have presented the solution provided by *ReadyPark* to manage the finding of a parking spot in urban areas in a collaborative sharing mode. Knowing that, without intelligent parking allocation, for there to be a swap between a taker and a giver, there must of course be a spatio-temporal coincidence, the key idea of the *ReadyPark* collaborative solution is to relax the spatial constraint by allowing drivers to exchange their respective positions *via* a digital collaborative service based on a mobile application using Galileo technology and a collaborative platform (web services) to implement the parking spot search strategy.

Through the NFS assumption, we have considered the extreme, but nevertheless frequent situation, in which no parking spot is available, so that the only way to find a spot for a driver is if it is vacated by another driver and reachable by the given driver. The finding process has been modelled and simulated according to an agent-based approach implemented in the Netlogo multi-agent programmable environment. The streets network is the Manhattan-like grid model and drivers are supposed to belong or not to the *ReadyPark* collaborative community. ReadyParkers are informed when and where a spot is about to become available and so have the advantage to reach it more easily. A minimal model has been first simulated in order to highlight the impact of the supply/demand ratio on the dynamics and to show the potential interest of the collaborative approach. Computational simulations on the full model have confirmed these preliminary results and show that, if the collaborative community remains in the minority (say below 20%), the other drivers are not or only slightly affected.

Beyond the numerous previous works in the field of urban parking modelling requiring infrastructures based on sensors or cameras [9,10,13–16], many works deal with specific issues that need to take into account particular characteristics such as giver/taker agreement [23], knowledge of places availability [21], giver/taker asymmetry [20], smart parking availability [13], outdoor/indoor parking [17], private/public parking, free/paid parking, etc. All these approaches therefore differ from the present work which, on the contrary, proposes a purely software solution supported by a parsimonious modelling approach dedicated to city community members searching and leaving public on-street parking spots with giver/taker symmetrical role. However, most of these approaches, including *ReadyPark*, operate in the same general context of urban parking assistance by offering intelligent management and promoting collaborative dimension.

In summary, *ReadyPark* offers a collaborative parking spot sharing solution that is (i) *lightweight*: it does not require the implementation of heavy hardware infrastructure such as sensors or cameras; (ii) *flexible*: it can adapt to the context of a given city or time of year or day of the week or time of day; (iii) *configurable*: rather to cover all users, at all times and in all places, the pairing strategy allows intelligent priority control for the well-being

of the community; (iv) *socially responsible*: the pairing algorithm can take into account as a priority criterion specific characteristics of individuals (as doctors or home care providers) or specific times or even specific locations (as school start and end time); (v) *open*: it is not niche solution reserved for a private club; anybody can collaborate although no one is obliged to do so; it is up to the municipality to fix priority rules in the common interest; (vi) *user-friendly*: by facilitating contacts and mutual assistance, it strengthens the links between the citizens.

In future works we plan to extend the current approach in the following ways: (i) Release the constraint defined by the NFS model to make it more realistic, for instance, by allowing the switch from the Manhattan-like grid model to more complex and realistic city model using a geographical information system (GIS) or by integrating free parking spots in the model. (ii) Optimise simulation accuracy and relevance by developing and integrating a Hotspot Affluence Model. (iii) Consider different types of independent collaborative networks, for example, study the effects on all drivers of one connected cars network. (iv) Study the acceptance by the drivers of the collaborative dimension for instance, the time during which a ReadyParker prioritises other ReadyParkers is a sensitive parameter that will be interesting to study to better understand its impact on driver involvement. (v) Partition the population according to some priority levels (e.g., people with disabilities, medical profession, etc.) and then study the influence of the number of levels and each part size on the overall performances.

Author Contributions: Conceptualization, J.-D.C.; funding acquisition, J.-D.C.; methodology, J.-D.C., E.S.; project administration, J.-D.C.; writing—original draft preparation, J.-D.C., E.S., P.G.; supervision, J.-D.C.; writing—review and editing, J.-D.C., E.S.; software, P.G.; resources, P.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *Parc et Immatriculations des Véhicules Routiers*; French Ministry of Ecological Transition: Paris, France, 2020.
2. *Pour une Politique du Stationnement au Service des Usagers*; French General Commission for Strategy and Forecasting: Paris, France, 2013.
3. *Drivers Spend an Average of 17 h a Year Searching for Parking Spots*; USA Today Reporter Kevin McCoy: New York, NY, USA, 2017.
4. Guedes, A.L.A.; Alvarenga, J.C.; Goulart, M.D.S.S.; Rodriguez, M.V.R.; Soares, C.A.P. Smart Cities: The Main Drivers for Increasing the Intelligence of Cities. *Sustainability* **2018**, *10*, 3121. [\[CrossRef\]](#)
5. Munhoz, P.A.M.S.A.; da Dias, C.F.; Chinelli, C.K.; Guedes, A.L.A.; dos Santos, J.A.N.; da e Silva, S.W.; Soares, C.A.P. Smart Mobility: The Main Drivers for Increasing the Intelligence of Urban Mobility. *Sustainability* **2020**, *12*, 10675.
6. Betis, G.; Larios-Rosillo, V.M.; Petri, D.; Wu, X.; Deacon, A.; Hayar, A. The IEEE smart cities initiative—Accelerating the smartification process for the 21st century cities. *Proc. IEEE* **2018**, *106*, 507–512. [\[CrossRef\]](#)
7. Semajnski, I.; Gautama, S. Smart city mobility application—Gradient boosting trees for mobility prediction and analysis based on crowdsourced data. *Sensors* **2015**, *15*, 15974–15987. [\[CrossRef\]](#)
8. Pau, G.; Severino, A.; Canale, A. Special Issue “New Perspectives in Intelligent Transportation Systems and Mobile Communications towards a Smart Cities Context”. *Future Internet* **2019**, *11*, 228. [\[CrossRef\]](#)
9. Ilarri, S.; Wolfson, O.; Delot, T. Collaborative Sensing for Urban Transportation. *IEEE Data Eng. Bull.* **2014**, *37*, 3–14.
10. Kianpisheh, A.; Mustafa, N.; Limtrairut, P.; Keikhosrokiani, P. Smart parking system (SPS) architecture using ultrasonic detector. *Int. J. Softw. Eng. Its Appl.* **2012**, *6*, 55–58.
11. Aliedani, A.; Loke, S.W.; Desai, A.; Desai, P. Investigating vehicle-to-vehicle communication for cooperative car parking: The copark approach. In Proceedings of the 2016 IEEE International Smart Cities Conference (ISC2), Trento, Italy, 12–15 September 2016; pp. 1–8.
12. Shoup, D.C. Cruising for parking. *Transp. Policy* **2006**, *13*, 479–486. [\[CrossRef\]](#)
13. Lin, T.; Rivano, H.; Le Mouél, F. A survey of smart parking solutions. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3229–3253. [\[CrossRef\]](#)
14. Bajwa, R.; Rajagopal, R.; Varaiya, P.; Kavalier, R. In-pavement wireless sensor network for vehicle classification. In Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, Chicago, IL, USA, 12–14 April 2011; pp. 85–96.

15. Huang, C.C.; Wang, S.J. A hierarchical bayesian generation framework for vacant parking space detection. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 1770–1785. [\[CrossRef\]](#)
16. Li, X.; Chuah, M.C.; Bhattacharya, S. UAV assisted smart parking solution. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1006–1013.
17. Lanza, J.; Sánchez, L.; Gutiérrez, V.; Galache, J.A.; Santana, J.R.; Sotres, P.; Muñoz, L. Smart city services over a future Internet platform based on Internet of Things and cloud: The smart parking case. *Energies* **2016**, *9*, 719. [\[CrossRef\]](#)
18. Boccara, N. *Modeling Complex Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2010.
19. Higuchi, T.; Ucar, S.; Altintas, O. A Collaborative Approach to Finding Available Parking Spots. In Proceedings of the IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
20. Shao, C.; Yang, H.; Zhang, Y.; Ke, J. A simple reservation and allocation model of shared parking lots. *Transp. Res. Part C Emerg. Technol.* **2016**, *71*, 303–312. [\[CrossRef\]](#)
21. Adamuscin, A.; Golej, J.; Panik, M. The challenge for the development of Smart City Concept in Bratislava based on examples of smart cities of Vienna and Amsterdam. *EAI Endorsed Trans. Smart Cities* **2016**, *1*. [\[CrossRef\]](#)
22. Chalamish, M.; Sarne, D.; Lin, R. Enhancing parking simulations using peer-designed agents. *IEEE Trans. Intell. Transp. Syst.* **2012**, *14*, 492–498. [\[CrossRef\]](#)
23. Di Napoli, C.; Di Nocera, D.; Rossi, S. Negotiating parking spaces in smart cities. In Proceeding of the 8th International Workshop on Agents in Traffic and Transportation, in Conjunction with AAMAS, Paris, France, 5–6 May 2014.
24. Ausloos, M.; Dawid, H.; Merlone, U. Spatial Interactions in Agent-Based Modeling. In *Dynamic Modeling and Econometrics in Economics and Finance*; Springer: Cham, Switzerland, 2015.
25. Bonabeau, E. Agent-based modeling: Methods and techniques for simulating human systems. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7280–7287. [\[CrossRef\]](#)
26. Daganzo, C.; Gayah, V.; Gonzales, E. The potential of parsimonious models for understanding large scale transportation systems and answering big picture questions. *EURO J. Transp. Logist.* **2012**, *1*, 47–65. [\[CrossRef\]](#)
27. Alenaz, M.J.F.; Abbas, S.O.; Almowuena, S.; Alsabaan, M. RSSGM: Recurrent Self-Similar Gauss–Markov Mobility Model. *Electronics* **2020**, *9*, 2089. [\[CrossRef\]](#)
28. Kour, S.; Singh, H.; Kaur, S. Performance Evaluation of Manhattan Mobility Model in Mobile Ad-hoc Networks. *Int. J. Future Revolut. Comput. Sci. Commun. Eng.* **2018**, *4*, 13–18.
29. Ramakrishnan, B.; Bhagavath Nishanth, R.; Milton Joe, M.; Shaji, R.S. Comprehensive analysis of Highway, Manhattan and Freeway mobility models for vehicular ad hoc network. *Int. J. Wirel. Mob. Comput.* **2015**, *9*, 78–89 [\[CrossRef\]](#)
30. Ransikarbum, K.; Kim, N.; Ha, S.; Wysk, R.A.; Rothrock, L. A highway-driving system design viewpoint using an agent-based modeling of an affordance-based finite state automata. *IEEE Access* **2018**, *6*, 2193–2205. [\[CrossRef\]](#)
31. Kim, N.; Shin, D.; Wysk, R.A.; Rothrock, L. Using finite state automata (FSA) for formal modelling of affordances in human-machine cooperative manufacturing systems. *Int. J. Prod. Res.* **2010**, *48*, 1303–1320. [\[CrossRef\]](#)
32. Shin, D.; Wysk, R.A.; Rothrock, L. An investigation of a human material handler on part flow in automated manufacturing systems. *IEEE Trans. Syst. Man Cybern. Syst. Hum.* **2006**, *36*, 123–135. [\[CrossRef\]](#)
33. Wilensky, U. *NetLogo*; Technical Report; Center for Connected Learning and Computer-Based Modeling, Northwestern University: Evanston, IL, USA, 2010.
34. Wilensky, U.; Rand, W. *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*; MIT Press: Cambridge, MA, USA, 2015.
35. Squazzoni, F. *Agent-Based Computational Sociology*; Wiley: Hoboken, NJ, USA, 2012.