Revisiting Urban Taxis: Optimal Real Time Management And Performance Appraisal By A Discrete Event Simulation Model

Eugénie Lioris, Guy Cohen, Arnaud de La Fortelle

***Abstract*—Cities are complex spatial structures presenting dense economical and cultural activities where an efficient transportation system is essential to meet all the challenges encountered when considering a competent industrialization, massive production, other activities and an improved quality of life. Based upon the fact that car travel is still the consumer's first choice, we are interested in the challenging idea of creating an intelligently administered new system called "Collective Taxis". The service quality provided will be comparable with that of conventional taxis (system operating with or without reservations, door-to-door services, well adapted itineraries following the current demand, controlling detours and waits, etc.), with fares set at rates affordable by almost everyone, simply by utilizing previously wasted vehicle capacity. With the aim of achieving optimal functioning of such an elegant but also complex structure, a made to measure discrete event computer simulator has been developed as a tool to evaluate and fine tune real time decision algorithms, optimize resources, study the influence of various factors pertaining to demand (level, geometry) upon performances and the threshold of profitability. The aim of this paper is to present such a methodology and illustrate it with results corresponding to fictitious but realistic data for a city like Paris.**

***Keywords*-Intelligent transportation system, discrete-event simulation, Monte Carlo simulation, Poisson processes, routing algorithms, Pareto optimality, performance evaluation, parameter optimization, queueing network model.**

I. Merits Of Collective Taxis

A preference for automobile use in urban areas is obviously related to a variety of advantages, regarding comfort, convenience and speed when conditions permit. Many attempts have been made to reduce individual automobile dependency with car sharing, car pooling (such structures often accompanied by poor spatial distribution, greater concentration on high-demand destinations and additional constraints requiring users to return vehicles to specific stations) and even schemes to combine the use of private and public transport, but despite all the encouragement given, they remain only partial solutions and cannot be considered to be realistic transport alternatives geared to a broad panel of customers.

Conventional taxis are potentially an alternative to private vehicles. However their high cost prohibits daily use for most commuters, and fuel shortages, parking restrictions, difficult road conditions and pollution issues all contribute to reducing the operational efficiency of the classical taxi system, which has become outmoded and in need of development. A collective taxi system intelligently associating more than one passenger with each vehicle and operating in urban areas to provide a comparable service quality to conventional taxis at an affordable price could be an interesting solution to the growing transportation problem. Indeed, it would offer a service quality equivalent to individual cars without the need to drive in a hazardous environment and the consternation of being unable to find available parking on arrival. Additional advantages in terms of cost, environmental impacts and traffic conditions can also be expected by simply raising the occupancy rate of vehicles in cities and dynamically optimizing the vehicle itineraries.

The idea of collecting several clients with the same vehicle has already a long history, especially under the terminology of "demand responsive transport" or "dial a ride", which mostly implies systems operating with preliminary reservations, but such systems are often geared to particular groups of customers (disabled persons for travels between home and health centers, passengers between hotels and airports, etc.), and also frequently confined to particular areas or itineraries. Our aim here is to study the most open and less constrained transportation system operating in a whole city, and possibly its suburbs, with or without preliminary reservations, and addressed to all categories of customers, from the usual driver of a private car, to the aged person who can no longer drive in crowded cities. But such an open and flexible system is very difficult to operate, and the need to first assess its performances and economic profitability, which depend on the care with which it is designed and operated, and also on the characteristics of the potential demand, should be obvious.

The main goal of this paper, a preliminary version of which was presented at a IARIA conference [1], is to propose a simulation tool with such an objective in mind, and to demonstrate how it can be used to answer various questions. At this stage of our study, no real life implementation has been achieved yet, and no real data have been collected. Numerical experiments have been conducted with fictitious but hopefully realistic data inspired by a city like Paris. Hence, the reader is not asked to adhere to any definite conclusion, but he is invited to see how the proposed tool can be used to shade some light on various issues raised by such a system (actually, at this stage, only the so-called decentralized mode of operation, that is, without preliminary reservations, have been studied).

E. Lioris is with Imara Team, INRIA, Le Chesnay, France and with CERMICS, École des Ponts-ParisTech, Marne-La-Vallée, France, jennie.lioris@cermics.enpc.fr

G. Cohen is with CERMICS, guy.cohen@mail.enpc.fr

A. de La Fortelle is with CAOR, École des Mines-ParisTech and with Imara Team-INRIA, arnaud.de_la_fortelle@mines-paristech.fr

The rest of this paper is comprised of six sections: Section II presents the three modes of operation we have in mind for the exploitation of collective taxis to provide a door-to-door service to clients wishing to travel with and/or without reservations. The simulation tool we have designed is ready for those three modes, but as already said, only the mode without reservations has been investigated in some detail yet. Section III justifies the need of simulations for learning the system behavior. Moreover it discusses the simulation technique employed, presenting the different entities composing the system and their relationship. Next, the architecture of the constructed simulator is discussed, its different parts being introduced as well as the necessary input and the results issued. The different possibilities of performing a simulation are established according to the needs of the study as well as the exploitation of the results. Section IV examines the optimization problems encountered in the real time management and in the decentralized management of each vehicle, and then it provides a brief description of the utilized algorithm to decide upon client acceptance.

In §V, after introducing the necessary input data, it is demonstrated how the simulation tool allows for a very sharp microscopic analysis and the possible detection of some abnormal functioning in some parts of the system, and how to remedy it. Section VI presents a more macroscopic (or statistical) analysis of the results and, based on it, a methodology for strategy selection and resource optimization. Finally, §VII concludes the study introduced in this paper and discusses the challenge of a future study on the centralized and mixed managements.

II. The System As We See It

The principle of associating more than one passenger with each taxi already exists in many countries but the system is not always optimized. In general with such systems of so-called *shared taxis*, it is mostly the driver and/or, less frequently, the existing passengers who decide if a new prospective client will be accepted, what the vehicle itinerary will become, and what the fare should be. This type of transport system mostly exists in developing countries [2], [3, pp. 472–474]. In general the taxi routes start and finish in central town locations such as taxi ranks, lorry parks, train or bus stations etc. [2], and invariably the principal routes served are subject to common delays.

However the system has started to be developed in the US (New York [4], San Francisco [5], Los Angeles [6], [7]), Europe: Netherlands [8], Brussels [9], UK [10], Germany [11], etc.

In the above structures, the proposed services do require advanced reservations; some suggest fixed or semi-fixed routes, whilst others offer door-to-door services. The majority covers a large but still limited area (the suburbs of the town are not always included, others cover specified avenues, etc.). Sometimes the service hours are restricted (late night journeys are often rare and sometimes impossible). Others combine public transport with a taxi service where, depending on passenger demand, the bus driver calls a company to reserve a vehicle and the pick up point will be at a bus stop. Others restrict their services to females, the elderly or people with restricted mobility. Often it is the driver and/or the clients who decide if they want to enter and what the cost should be.

Demand Responsive Transport (DRT) is getting increasingly popular and the need to provide an optimized service clearly stands.

Unlike the existing shared taxi systems we have previously referred to, for which studies and/or implementation are conducted principally for systems requiring prior reservations, and fixed or only slightly variable routes and combined taxi/bus services [12], [13], [14], [15], we are interested in the optimization of a more flexible and open system. The DRT structure envisaged should impose the minimum of constraints to both clients and drivers whilst offering a service quality comparable to conventional taxis in terms of detours, initial waiting times. More precisely we are aiming at a system providing a door-to-door service with minimal waiting times, independence and optimized itineraries (almost direct itineraries) at a low cost for both vehicles and clients.

Given the novelty of the system, it is hard to predict the proportion of potential customers it will attract in a given area, and this demand will anyway depend on the service quality offered and on the fares. Hence, our approach is to propose an optimized configuration for any demand level and geometry. In other words the outcome of this study is to provide a tool to construct the *offer curve* (in terms of performances and costs) as a function of demand. Specialists in the field of transportation should be able to return the *demand curve* by comparison with other transportation means in a given city: as always in Economy, the intersection of both curves will determine the part this system may take.

Three operating modes of the “Collective Taxis” are envisaged:

- The decentralized management where clients appear randomly in the network seeking a vehicle for an immediate departure. Such a structure requiring no prior bookings was initially studied by [16]; our study here is an advancement of that work to include matters of simulation techniques, results and conclusions about its relevance compared to the next modes of operation in any specific situation.
- The centralized management dealing with clients who must make an advanced reservation of a seat in a vehicle.
- The mixed approach combining the management of both previous types of clients.

Within this paper, we mainly consider the decentralized approach and discuss the technical issues related to this mode, and we demonstrate a methodology for an optimized management. Nevertheless, the simulator described hereafter is ready to accommodate the others modes of operation too.

Since there have been very few previous studies conducted on this form of management system, we are unable to make comparisons with other similar structures at present. The study of the other two approaches will form part of our future research and will not be discussed in this paper.

III. A Discrete-Event Simulator

Such a new system raises many questions, which require scientific answers. This section introduces the tool to help us providing them.

A. Multiple Questions in Need of Precise Answers

If the claimed advantages of such an open and flexible system are not merely wishful thinking, this must be substantiated by quantitative responses to the following issues.

- How many vehicles would be required to operate such a system?
- Will that number vary during the day? In which case, by how many and at what times?
- When a client meets a taxi, what criteria would be employed in deciding if he/she is accepted or refused and what would be the optimum journey for the passengers and vehicle efficiency?
- What should be done with an idle vehicle? Etc.

A model and all the resources of Optimization and Operations Research are necessary in order to predict the system performances. Obviously, we are handling a quite complex spatio-temporal decision making problem and it is almost impossible to write a mathematical model describing it with great precision. So what could the answers be and how do we know if the correct ones are being confronted?

Gaining direct experience on a "trial and error" basis would require much time to learn the system behavior and to produce an optimized structure, not to mention the client dissatisfaction and financial risk. However, a simulation is a reliable means of exploring the many aspects of a decision making problem, reproducing precisely all stochastic features and occasional unforeseen random events that periodically occur, allowing us to assess and master the potential of collective taxis at a minimal cost without the real time consequences of poor decision making. Moreover simulation is the only way to reproduce various scenarios with a single factor modified at each run. This is a fundamental property required to search for optimal policies.

B. Stochastic Discrete Event Simulations

The methods of simulation for studying many complex structures is becoming extremely diversified and used very commonly in many fields (air taxis [17], [18], Biology, [19], Physics [20], naval simulations [21], and other fields too [22].)

Types of scientific computer simulation are derived from the underlying mathematical descriptions of the problems studied:

- *numerical simulation* of differential equations that can not be solved analytically;
- *stochastic simulation* commonly used for discrete systems where events occur probabilistically and can not be described with differential equations; a special type of discrete event simulation is the *agent-based* simulation (MAS), effectively used in Ecology, Sociology, Economy, Physics, Biology and other disciplines too; indeed, many studies have been conducted on shared taxi systems using the MAS technique ([23], [24], [14], [15]).

Many open source simulation platforms ([25], [26], [27], [28], [29], [30]) and commercial ones ([31], [32], [33], [34], [35]) based on many different programming languages (C++, Java, Python etc.) are available for almost every simulation type, and they alleviate the burden of proof by their users when designing their simulation tool. Most of them treat each simulation entity as a separate thread allowing the developer to focus only on simulation specifics.

Despite all the difficulties, we chose to develop our simulation tool without the use of any particular simulation environment. Some of the implied advantages are that we can master the entire framework, having the liberty and direct access to make any necessary modifications in order to improve the tool functionalities according to the needs of the study. Consequently, we adopt a classical (not a MAS) discrete event simulation technique for learning the behavior of the "Collective Taxis" under multiple strategies.

Within this context, the system evolution is represented as a chronological sequence of the form $\{s_i, e_i, t_i\}$ where s_i is the system state at time t_i and e_i is the event happening at time t_i bringing the system to the new state s_{i+1} and so forth.

C. Modeling The System

A delicate, and also one of the most difficult, factor to establish when modelling by the Discrete Event Simulation technique is to define the set $\{(E_i, P_i)\}_{i=1,\dots,N}$ where E_i is the set of types of events describing the system evolution, and P_i is a special procedure responsible for the treatment of any event of type E_i when activated by the event accession. Care must be taken to choose the number N of different types of events to be employed. When a very complex structure is designed, there is a high risk of producing complicated results which are difficult to understand and deal with, and consequently the value of such simulations will be considerably reduced. Conversely, if too many simplifications are taken into consideration, we shall end up with unrealistic situations and we risk coming to unreliable conclusions.

In the following, we briefly describe some types of events related to one or more of the entities involved in the system: clients, vehicles, dispatchers (in the case of a centralized or mixed mode of operation in which reservations are made through a dispatching center). For clients, we will skip some events related to clients making reservations (to save space). Clients and vehicles travel in a network and events are supposed to take place at the network nodes only.

1) Events Initiated By The Decentralized Type Of Clients:

a) Client Appears At A Node: When a client appears at the roadside seeking an available vehicle for immediate departure.

b) Client Quits The Node: When a client has found no vehicle before the waiting time limit has been reached and quits the node.

c) Client Enters The Vehicle: When a client embarks on his associated vehicle in case of a positive decision.

2) Events Prompted By Vehicles:

a) Vehicle Commences Service: At this moment the vehicle, in its initial location, becomes available to clients.

b) Vehicle Concludes Its Period Of Service: From this time onwards, the vehicle ceases to be at the disposal of clients, the event only taking place when a vehicle has an empty itinerary. If that is not the case, the vehicle will have to complete all its tasks before quitting the system.

c) Vehicle Arrives At A Node: Whenever a vehicle arrives at a node, it starts by checking if there are clients to disembark. Next, for both a mixed or centralized management, it checks if there are potential appointments with centralized types of client at this location. Then, for a mixed or decentralized management, the vehicle looks for new clients waiting at the roadside. Finally, if the vehicle is left with an empty itinerary, a station node has to be selected together with the corresponding maximum waiting period.

d) Passengers Disembark From A Vehicle: The present location of the vehicle happens to be the destination of some of its passengers and at this moment those clients alight from the vehicle.

e) Vehicle Quits Its Station: Either a vehicle responds to the request of a client, or remains stationed at a node with an empty itinerary waiting for a client until the maximum waiting time is reached, and then it must relocate to another station node, if so instructed.

f) Vehicle Ceases To Wait For Absent Clients: When a vehicle searches for its client appointments and finds one or more are absent, the vehicle checks whether and for how long it should wait. At this moment, the vehicle stops waiting for the absent client(s) and either leaves the node if its itinerary is not empty (centralized management), or it checks to examine for new candidate clients available to pick up.

g) Vehicle Completes Its Dialogue With A Decentralized Client: At this stage, it is decided whether the vehicle can accept the corresponding client or not. In the case of a positive answer, the client will embark onto the vehicle. If the client is refused, the vehicle will check if he can seek another client.

h) Vehicle Refills Its Batteries: This event concerns only systems utilizing electrical vehicles and takes place after having checked that it needs to charge its battery.

3) Events Associated With Dispatchers (Or Servers):

a) Dispatcher Starts His Service: From this time, the dispatcher is at the disposal of clients. If there are calls in the waiting queue, he selects the first one on a "first come, first served" priority basis.

b) Dispatcher Ceases His Service: If the dispatcher is not occupied with a call, he ceases his service. Otherwise, he first has to finish the task he is engaged in before ending his service.

c) Dispatcher Ends A Call: At this stage a dispatcher has given an answer to a client's request and if he has still not completed his shift, he searches for new calls waiting in the queue. He stops working when his period of duty and any call he is engaged in is complete.

D. Chain Of Events

The treatment of an event modifies the system state and it may be associated with the generation of new future events, which are then added to the event list. In Figure 1, nodes represent the event types for the decentralized management and the edges starting from each node and pointing to another one indicate the necessary creation of the event (solid line) or its possible generation (dotted line).

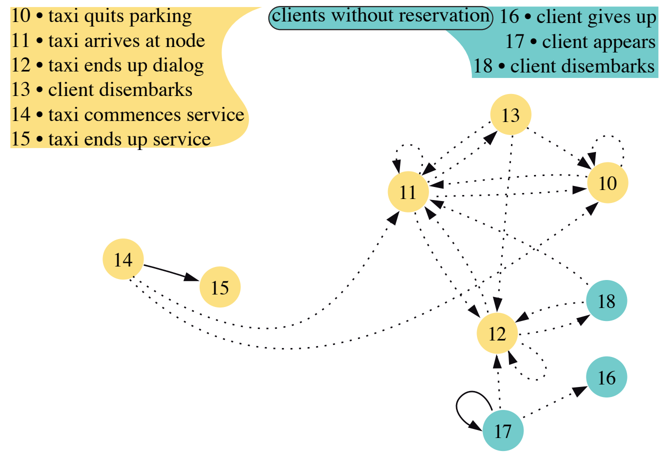

Fig. 1. Event Sequence

As an example, let us consider event type 17 corresponding to the appearance of a client at a network node. Since the client appearance is modeled by a Poisson process (that is to say, the time elapsed between two successive appearances of clients at a node is a random variable following an exponential law — see §V-A), as soon as one client appears, we have to define when the next client appearance will take place at the same node. Consequently it will be necessary to generate a new event of type 17 at this node (solid line). If the former client finds a vehicle immediately at the node, a dialogue will start between the vehicle and the client and in that case an event of type 12 will be created in order to plan the end of the dialogue. If no vehicle is found, the client decides for how long he will wait. In that case, an event of type 16 will be generated in order to put an end to the client waiting at some date in the future.

E. Decision Modeling

There is a two level set of decisions to be defined.

At the design and dimensioning stage, we have to

- model the network in which the collective taxi system will be operated (its topology, probability laws defining travel times during the entire day, ability to define congestion periods and off peak hours, client appearance and demand construction during the simulation period, definition of the client maximum waiting time at each node etc.);
- choose the operating mode (decentralized, centralized or mixed management);
- define the number of resources (number of vehicles in service, of dispatchers, etc.);
- define the service duration of each resource and its starting service time as well.

At the real time operating stage, we have to design all the control laws ruling the system. At this level, we must define decision algorithms for

- assigning vehicles to centralized types of clients;
- the acceptance or refusal of decentralized types of clients to the vehicles they meet;
- the management of the idle vehicles.

F. Simulator Design

Our goal is to conceive intelligent policies, assessing their effectiveness on a realistic virtual system closely representing a real-life structure, followed by fine tuning of all the real-time control algorithms associated with an off-line resource optimization. Evaluating the system performance for each such scenario through statistical analysis will enable us to provide an optimal configuration for any level and type of demand (offer curve).

Our initial challenge is how to construct a simulation model capable of achieving our goals. A first step in this direction consists in separating the nature of the tasks and defining the two major parts of the simulator.

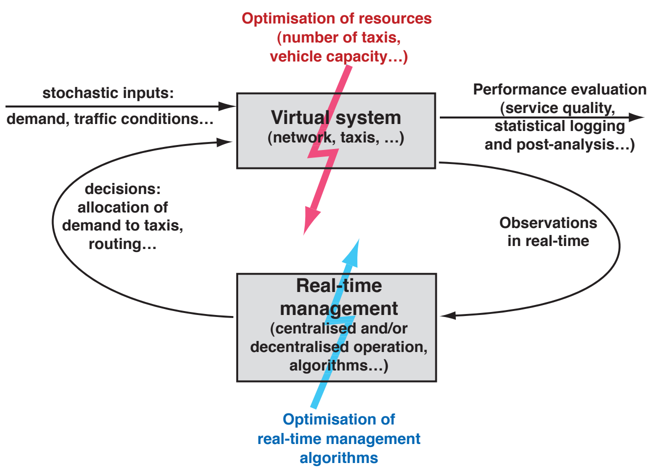

Fig. 2. The Simulator In Two Parts

The mechanical part is a virtual representation of the real system (e.g., detection of a vehicle arriving at a node). This part receives two types of data:

- *exogenous deterministic input data* related to the operational issues such as resources utilized (number of vehicles and their capacities, number of dispatchers or servers, their service durations etc.), as well as *probability laws* defining the stochastic phenomena (e.g., traffic conditions, demand, etc.);
- *decision inputs* delivered by *feedback rules* defining the real time management, which constitutes the second part of the simulator.

The mechanical part of the simulator is responsible for the evolution of the state of all agents present in the system and is in charge of interrogating the real time management when needed. Thus it detects the different states of a vehicle (arrival at a node, vehicle with empty itinerary, detection of client appointments or of clients disembarking from the vehicle, questioning of possible new clients, vehicles ceasing their service, etc.).

The real time management is responsible for all the on-line decisions required by the mechanical part. It is comprised of a set of algorithms ruling the system.

The advantages of such an architecture is, on the one hand, to allow for various experiments and comparisons of algorithms for fixed environmental data such as the demand intensity level and geometry, the topology of the network on which the system operates, the corresponding traffic conditions etc. On the other hand, for given management strategies, we intend to measure the influence of those external conditions upon the system performances.

It is well known that modeling the demand is an extremely complicated task, in particular to collect reliable data. Various methods have already been established and many new ones may be proposed in the future by specialists. Our main purpose here is to propose a consistent methodology for optimizing collective taxi performances, which would function under all potential exogenous factors. Consequently, all that information has to be modeled separately and presented to the simulation tool. The same considerations apply to traffic conditions, which may vary with the time of day and with changes of circumstance (holidays, special events, public transport strikes, road works, accidents), and to the system dimensioning concerning the number of employed resources. Meanwhile, the system control forming an individual part of the simulator, allows for a convenient evaluation of the system behavior according to various developed strategies and schemes.

Thus, for example, in order to compare two different policies concerning the vehicle management, we can proceed with two separate series of simulations, using the same simulation tool and configuration by simply modifying the real-time part related to vehicles and keeping unchanged the mechanical part and all input, including the history followed by random factors such as the client arrivals at nodes.

Similarly, if we are interested in the system behavior under some variation in resources (for example the number of the servers in the centralized or mixed managements, or the number of vehicles in service), we only modify the associated data information, everything else remaining the same.

To illustrate the respective roles of the two simulator parts, we will examine the following example. Let us consider a vehicle, which has just found a potential client waiting at the roadside. For his acceptance or refusal, the mechanical part will interrogate the real time management. This last one, by calling the appropriate decision algorithm, will reply whether the client can be accepted by the given vehicle and, in the case of a positive answer, the optimal vehicle itinerary will also be provided.

In this kind of decision, if one attempts to keep detours for passengers, with respect to their most direct itinerary, at a very low level, then this quality of service for the existing passengers is likely to penalize new candidates since rejection will be more frequent, and consequently waiting times will be extended, leading to a greater client abandonment rate. This consideration shows that such a decision algorithm must include parameters which will be fine tuned by repeated experiments. This is a first level of optimization regarding the on-line management.

The second optimization level involves the mechanical part, and more specifically the system dimensioning (number of vehicles and/or dispatchers in service according to the system management, their duration of service, the vehicle capacity, etc.).

G. Simulator Features And Options

1) Simulation Time: The simulation keeps track of the present simulation progress, which is the time of the current processed event. The treatment of each event is instantaneous and the simulation time advances as we pass from one event to another. For our studies, the time unit considered is one second.

2) Event Stack: The simulation maintains a pending event set comprised of all the events that are not yet processed. More precisely, the event list is organized as a priority queue, sorted by the event time. Thus the event list is of the form $\{(e_1, t_1), (e_2, t_2), (e_3, t_3), \dots\}$ where e_i is the event occurring at time t_i with $t_i \le t_{i+1}$. In our system the event list is not entirely sorted. It suffices to have the earlier event at the head of the list whilst the rest are not necessarily ordered.

3) Simulation Duration: Before any implementation, the simulation duration must be specified. This value expresses the real time period during which we wish to explore the system behavior.

For the experiments reported later on in this paper, we have observed that simulations of 8 hours (of real time) with steady data (fixed probability laws of random inputs, fixed resources) yield statistical estimators with a very low variance. Therefore, we adopted this value for the simulation duration.

The simulator can be used in various ways according to the needs of each particular study. We describe these options now.

4) Specifying The Initial Condition Of The System: Before any simulation run, we have the opportunity to specify precisely the initial state of the system. We may want to start a “new” simulation, which implies that the initial system state is empty and consequently the event list contains only events concerning system initializations. More precisely, all the system resources must be put in service (vehicles have to be positioned at nodes, servers or dispatchers should be at clients’disposal, etc.) and clients should start to appear in order that the interaction between these two types of agents (clients and vehicles) will start taking place.

Alternatively, it is possible to define a desired initial system state. We may need to continue a previous simulation run if we consider that its duration was not long enough to form valid conclusions. In that case, the initial state of the system is given and it is the final one of the previous simulation. The event list will contain all the events that were not processed during the previous simulation (since their times were greater than the allocated simulation time).

5) Specifying The Clients Utilized: One possibility for client appearance is to generate new clients dynamically during the implementation according to the demand probabilistic model.

But we can also employ previously generated clients (during another run) subject to the duration of the new simulation not exceeding the duration of the particular run from which we wish to re-use the clients. It turns out that this is an indispensable option for the optimization of the system performances when we are interested in analyzing how the same client history behaves under various policies or even simply under different system conditions.

6) Simulation Loop: To summarize, a simulation run requires an initialization phase which consists in

- defining the simulation duration;
- specifying the nature of the simulation (new simulation or continuation of a previous one) as well as the type of clients utilized (generation of new ones or using a record);
- setting the clock to the starting time (this is an input in case of a new simulation, or it is the time at which the previously completed one ended, detected by the simulator engine);
- fixing resource parameters and initializing the system state variables (number of resources utilized, the duration of their services, vehicle positioning, etc.);
- scheduling the event list (with the necessary bootstrap events if a new simulation is desired, or by collecting the previously unresolved events remaining in the event list).

Then the simulation proceeds by

- handling the first event of the list; by the end of its treatment, it will be removed from the list;
- updating the clock to the time of the next event in the list. . .
- or stopping the simulation if the next event takes place after the specified terminal date.

H. Recording And Analyzing Results

The purpose of simulations is to accurately reproduce the system behavior according to any possible scenario. The original architecture of the simulator, separating the nature of tasks (controls from mechanical schemes) and the simulation technique employed (independent of threads and techniques utilized on multi agent simulations) within our study, requires us to face different types of problems from those listed in many other papers such as [23], [24], [14]. The evaluation of the performances of the applied strategy and all the numerical results and conclusions are not forming part of the simulator. On the contrary, once more, we prefer to separate these tasks, which take place during a following stage. More precisely the simulator registers *all* treated events, which are then stored in a database for later exploitation. This represents an enormous volume of information provided by each run, but any question that may be raised a posteriori can in principle be answered as long as an exhaustive record of all events has been kept. Exploiting that information is a complicated work, which is one of the reasons why we prefer to disconnect it from the main simulation run.

For this second phase of analysis, several scripts have been developed with two main objectives in mind.

- Either one is willing to perform a microscopic analysis of a specific run by tracking particular sequences of events or pinpointing seemingly abnormal functioning: this is in particular the case at the debugging stage in the

development of the program, or when wrong decision rules have been implemented.
- Or one is interested in a macroscopic, rather statistical, analysis of one, or possibly a series of, run(s) in order to compare several scenarios

Examples of both types of analysis will be given at §V and VI, respectively. A microscopic analysis requires interactive scripts whereas the macroscopic analysis is rather a batch process.

I. Programming Language

For the collective vehicles simulator, we chose to develop the program in "Python", which is a dynamic object-oriented language offering strong support for integration with other languages (C, C++ etc.) and software tools, encouraging an easily maintained, clear and high quality development offering multi-platform versatility (Mac OS X, Linux/Unix, Windows). Thus we are employing programming techniques such as encapsulation, inheritance, modularity to design all the particularly complex applications to be managed and implementing all the involved objects and their interactions within a relatively simple environment. These features become especially useful since the goal is to be able to reuse code for various approaches to collective taxis.

IV. Client Acceptance Algorithm (Decentralized Mode)

As previously explained in §III-F, every time there is a decision to be taken, the mechanical part asks the real time management for the necessary answer. Hereafter we focus on the decentralized approach in which the main real time decisions concern the question of what to do with empty vehicles and when to accept clients on board of taxis.

Among all the decisions composing the real time management, an extremely important part is the one referred to as the client acceptance by vehicles. In this paper, we shall consider a simple algorithm with a parameter to be tuned with the help of the simulator. However, there are various other possible algorithms for which the simulator can also prove useful to optimize them and assess their performances.

Every time there is a dialogue between a client and a vehicle, there is a binary decision to be taken concerning the client acceptance by the given vehicle. This decision is closely related to a second one concerning the possible new itinerary of the vehicle. The aim of the decision algorithm is to decide whether the new client should be accepted and, if so, to provide the new itinerary.

A. Notations

- Node n_0 is the present position of the taxi, t_0 is the present time, and $L = \{n_1, n_2, \ldots, n_m\}$ is the *sorted list* of nodes of its itinerary.
- We are searching for an optimal order for list $\ell = L \cup \{n_c^d\}$ where n_c^d is the destination of a candidate the taxi just met (M is the number of distinct nodes in ℓ).
- $\delta(n_i^o, n_i^d)$ is the duration of the direct travel (shortest path computed using average travel times on all arcs of the network) from origin n_i^o to destination n_i^d of client i (matrix δ is precomputed as explained later on).
- We denote by $\ell_1 = \{x(1), \ldots, x(|\ell|)\}$ any considered tour to visit *all* nodes in ℓ. In addition, $x(0) = n_0$.
- $p\big(x(k)\big)$ is the number of passengers disembarking at node $x(k)$.
- $u(k), k = 0, \ldots, M-1$, is the chosen next visited node when the vehicle is at node $x(k)$; consequently $x(k+1) = u(k)$.
- $t(k)$ is the predicted arrival time at node $x(k)$, which is easy to compute using matrix δ: $t(0) = t_0$ and $t(k+1) = t(k) + \delta(x(k), u(k))$.
- $\mathcal{E}(k)$ is the set of all visited nodes at step k; we consider that $\mathcal{E}(0) = \emptyset$; obviously $\mathcal{E}(k+1) = \mathcal{E}(k) \cup \{u(k)\}$.
- s is the *detour threshold* with respect to direct travel: *this parameter is to be tuned by repeated simulations* as discussed later on.
- $t^{\lim}(j)$ is the deadline for arrival at node $n_j \in \ell$:

$$t^{\lim}(j) = \begin{cases} t_0 + s \times \delta(n_0, n_c^d) \\ \qquad\qquad \text{if } n_j = n_c^d \text{ and } n_c^d \notin L, \\ \max\Big(\min\limits_{i \in d^{-1}(j)} \big(t_i^o + s \times \delta(n_i^o, n_i^d)\big), t_j^p\Big) \\ \qquad\qquad \text{if } n_j \in L. \end{cases} \tag{1}$$

The idea behind this formula is that, for each passenger, the detour threshold s should not be exceeded, as a proportion of the likely duration of his direct travel (the one he would have made if he had chosen a classical taxi or used his own vehicle). However it may be already impossible to satisfy that constraint due to past vehicle delays for doing various operations at nodes, stochastic travel times, etc. In that case, the retained deadline associated with node j will be its predicted arrival time t_j^p *before* the new candidate is accepted (that is, t_j^p is computed using the *original* sorted list L).

B. Problem Statement

The purpose is to minimise

$$\sum_{k=1}^{M} p\big(x(k)\big)\, t(k)$$

by choosing ℓ_1, that is, an order for the unsorted list ℓ, subject to the constraints:

$$\begin{aligned} & x(k+1) = u(k),\ k = 0, \ldots, M-1,\ x(0) = n_0, \\ & \mathcal{E}(k+1) = \mathcal{E}(k) \cup \big\{u(k)\big\},\ k = 0, \ldots, M-1,\ \mathcal{E}(0) = \emptyset, \\ & t(k+1) = t(k) + \delta(x(k), u(k)), k = 0, \ldots, M-1, t(0) = t_0, \\ & u(k) \notin \mathcal{E}(k),\ k = 0, \ldots, M-1, \\ & t(k) \le t^{\lim}(x(k)),\ k = 1, \ldots, M, \end{aligned}$$

where $t^{\lim}$ can be precomputed independently of the problem solution using (1).

C. Resolution

Such a problem can be solved by Dynamic Programming [36] using the state vector $(x, \mathcal{E}, t)$. However, when vehicles have a moderate capacity (say, 5 passengers maximum), the enumeration of all possible orders ($5! = 120$ at most) is feasible. For every order, we have to recursively compute the $t(k)$ (and the cost function simultaneously) and check whether they exceed the deadline, and consequently stop exploring that order as soon as the deadline constraint is violated.

Finally, we keep the best feasible order according to the cost function. In case of greater vehicle capacities, another (suboptimal) alternative is to try inserting the candidate node at each possible position of the given itinerary without changing the order provided by the original list L.

V. Implementing The Simulator And Scanning The Results

We now proceed to implementation and explain how results can be analyzed. For the time being, our study has been limited to the decentralized management although the simulator is ready for the other two operation modes. The main difficulty is to design real time decision algorithms, a task which is more difficult and which requires further research in the case of the centralized and mixed management.

We start by showing the data utilized in this study. At this stage of our research, for reasons that we briefly discuss hereafter, we had no access to real data, hence the data used for our experiments have been built up with the objective of looking realistic enough for a city like Paris (except for the network depicted in Figure 3 whose description is somewhat sketchy). Indeed, our main goal for the time being is to establish a methodology to evaluate such a collective taxi system. Different sets of data will provide different numerical results, but they should not affect the method of producing results and exploiting them.

Throughout the world, legislation governing the use of taxis is very restrictive, especially regarding the decentralized management considered here, in which clients are picked up in the streets. This has the unfortunate effect of leaving little room for intelligent systems with potentially successful results and the implied benefits for customers in terms of cost, service quality, and environmental advantages (parking, congestion, fuel consumption and its impact on pollution). This is probably one of the main reasons why we had to launch this initiative without waiting for partners ready to consider a real life implementation, hoping that this step would contribute to remove some of the present locks in the future.

A. Data Utilized

1) Network: The area in which the system operates (a city possibly including the suburbs) is represented as a set of nodes and edges with their associated properties. All events take place at nodes while vehicles travel on edges. With each edge, there is an associated random travel time following a shifted log-normal distribution (the shift ensures a minimum positive trip duration). These laws may vary during the simulation period to represent changing traffic conditions.

In general decisions are based on mean values of those quantities for future events, but, during simulation, the actual movements of vehicles are based on pseudo-random values following the given laws. For speeding the simulation up, a matrix δ of the durations of shortest paths for all pairs of nodes is calculated in advance, using the average traveling times of arcs. Another matrix S, calculated simultaneously and also stored, provides the route for following the shortest path (namely, if $k = S(i, j)$, then k is the next node to go to when the vehicle is at node i and is willing to join node j by the shortest path; at k, it will go to $S(k, j)$ and so forth). The computation of matrices is performed using a program in C provided by S. Gaubert [37] and based on Howard's algorithm [38]. This program can be interfaced with the script language of SCILAB software [39] and its network analysis toolbox METANET [40] which we used to manipulate our network data.

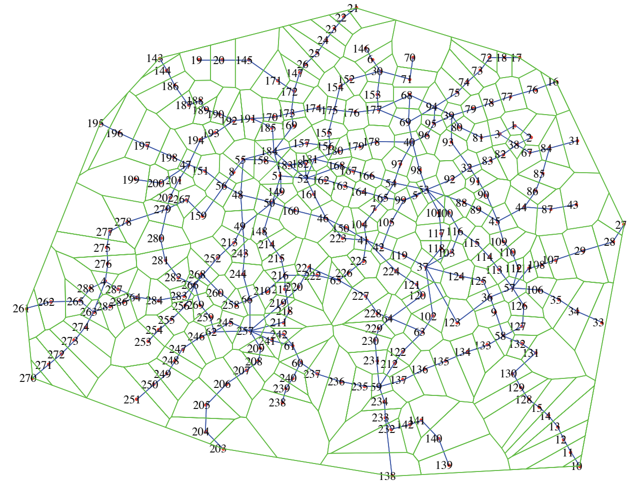

Fig. 3. The Network

The network we have used is comprised of 288 nodes and 674 edges (indeed, it is inspired by the Paris metro plan — see Fig. 3).

2) Demand: Clients arrive at nodes according to a Poisson process: the elapsed time between successive arrivals at node i follows an exponential law of parameter λ_i.

Moreover every such client receives a randomly chosen destination node. This choice follows the conditional probability law of destinations, given the origin, the so-called O-D (origin-destination) matrix.

The parameters of Poisson processes at nodes determine the demand intensity, and this may vary within the time (as also will the O-D matrix) to represent fluctuations of the demand in volume and geometry.

For the next reported results, the average number of clients per hour on the whole network is about 15,400, whereas the flow from origin to destination is mostly centripetal (clients are moving mostly from the suburbs towards the centre of the town). But we are able to modify those characteristics at will and to reoptimize the system for any given demand as demonstrated at §VI.

3) Vehicles: A number of vehicles operate at each period of the day. Each vehicle is characterized by its seating capacity and service time. For a vehicle with an empty itinerary, a decision must be made indicating what must be done with it (either it will park for some limited time at its present position, or it will be directed toward another node for parking, unless it meets new clients in the meanwhile).

For the following run, we initially positioned 13 vehicles per node (3,744 overall), but this distribution changes rapidly as the the simulation progresses. This figure has not been chosen at random of course, but after some experiments. It must be adapted to the demand level and/or geometry as explained at §VI-B.

The capacity of each vehicle is 5 passengers and we will study its influence at §VI-C.

4) Detours: The detour threshold parameter s was introduced at §IV for limiting the detours borne by passengers with respect to the expected duration of their direct travel form origin to destination. At §VI, we will study the influence of this parameter upon performances and see how it can be tuned to achieve reasonable trade-offs between conflicting indicators of the quality of service offered to customers. In the following experiment, it is set to 1.9 (again not totally at random!).

5) Durations: We allocate 30 seconds for each dialogue between a candidate client and a vehicle and 10 seconds for each embarking or disembarking of passenger.

The client maximal waiting time at his origin node is 10 minutes (that is, if no vehicle was able to accept it after 10 minutes, the customer gives up and this will be reflected in the abandonment rate reported below).

The maximal parking duration of an empty vehicle is 15 minutes: after this time is exceeded and no client showed up, a decision must be made about staying at the same node for a new parking period or departing toward another node.

Those duration values are not necessarily fixed but may be modeled as random variables to represent for example more or less patients customers.

B. Statistical Verifications

Before analyzing the results of a simulation run, we can check if the random data generated during the implementation have conformed to the corresponding probability laws. This mostly concerns the demand and also travel times.

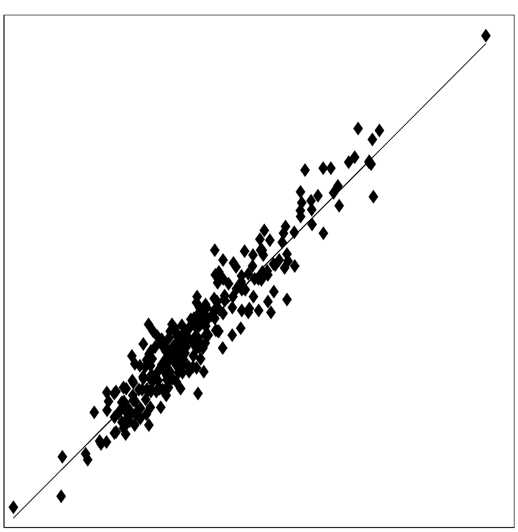

Fig. 4. Checking The Demand Intensity

Figure 4 represents the statistical verification of parameter λ_i corresponding to the intensity of client appearance at node i (see §V-A). The x-axis represents the theoretical value of λ_i and the y-axis represents its estimated value, for $i = 1, \ldots, 288$. Therefore, the plot is made up of 288 dots, which should ideally be aligned along the first diagonal.

The same visual technique can be used to compare theoretical and estimated values of the frequency of each destination given the origin (O-D matrix), and the average travel time of vehicles through each edge of the graph.

C. Microscopic Analysis

In the sequel of this section, we show the wealth of information that a simulation can provide.

1) Network Results: Figure 5 shows the number of visits of nodes by vehicles over the whole simulation run (the x-axis corresponds to the node numbers). The same kind of plot can be obtained for how many times vehicles crossed each edge of the graph. This is a quick way of discovering parts of the network which may not be well served by the decentralized management policy adopted here and under the prevailing conditions (in particular the geometry of demand).

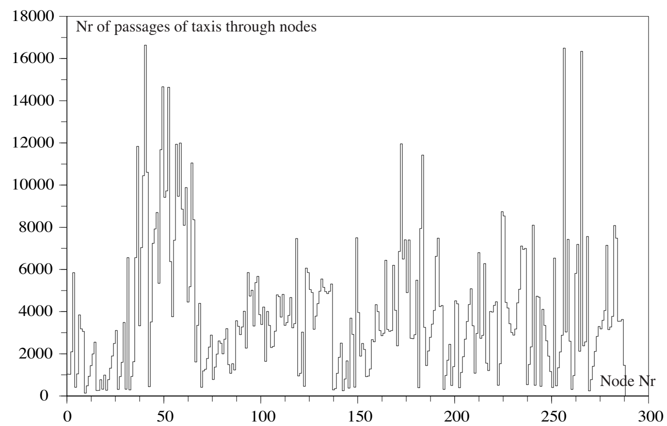

Fig. 5. Visits Of Nodes By Vehicles

2) Client Waiting Time And Abandonment Rate: Figure 6 represents the mean client waiting time ($\pm$ the standard deviation) at each network node and for the whole network (horizontal lines).

In order to detect critical network nodes, we may also examine the abandonment rate: as explained earlier, clients not served after 10 minutes give up and we measure the proportion of such clients at each node. The average client abandonment rate for the whole network is 1.33% as shown in Figure 7 by the horizontal line, but the node-per-node abandonment rate displayed in the same figure reveals that it have reached a very high value close to 45% at Node 10. In order to locate this node, it is easier to consider Figure 8 which is a colored map in which each cell is a Voronoi cell relative to a node of the network (a Voronoi cell is the set of points in the area which are closer to that node — called the “centroid” of the cell — than to any other node of the net): the shade of a cell is proportional to the abandonment rate of its centroid, and darker shades represent higher values of this rate. Node 10

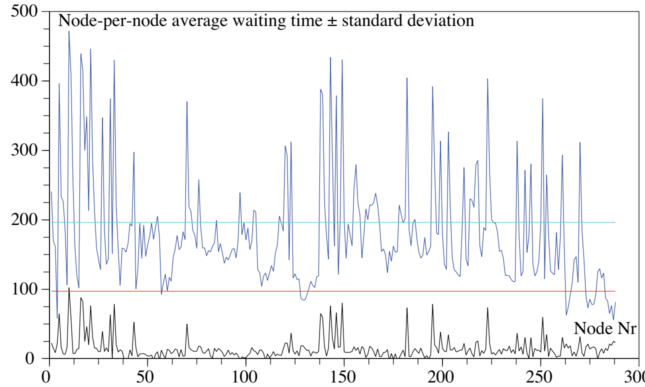

Fig. 6. Client Waiting Time At Each Node

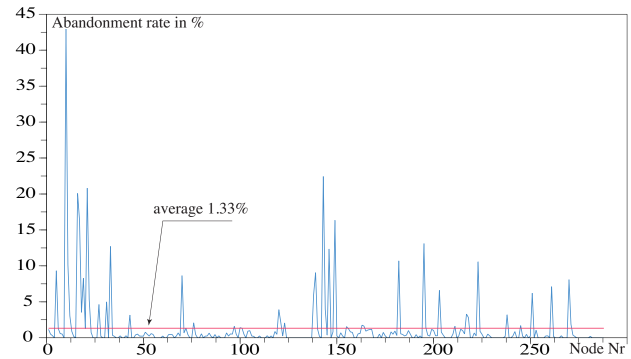

Fig. 7. Client Abandonment Rate

is located at an extreme South-East position in the area. We also notice other "bad" nodes mostly located at the periphery of the city. This observation is not surprising, given that the demand is mostly centripetal, which tends to concentrate taxis in the the city center.

However, some dark shades also appear in the city center, the most noticeable one corresponding to node 149. During our analysis session using SCILAB scripts, we can interactively ask for a plot of the evolution of the client queue length at

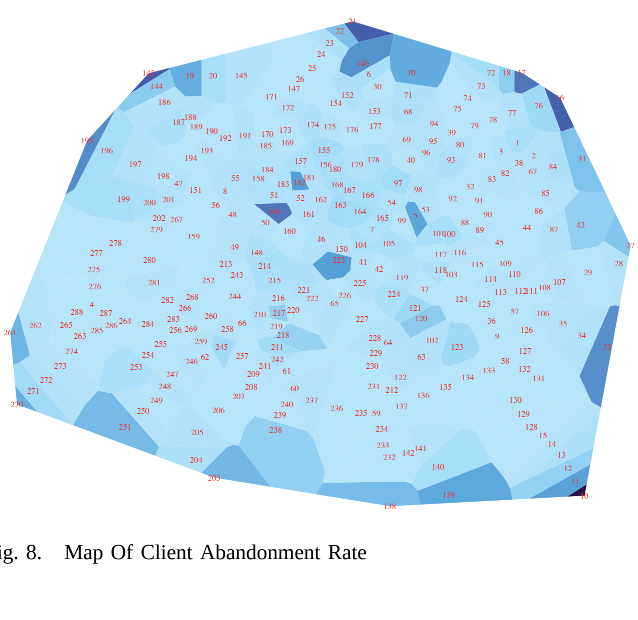

Fig. 8. Map Of Client Abandonment Rate

this node over the whole simulation, and this plot shows that this queue length quite often reaches rather high values. We may also observe the rather small number of visits of this node by vehicles in Figure 5. All those observations lead us to proceed to a deeper analysis for this particular node. Figure 9 shows a small portion of the network around Node 149 with the frequency of visit of the nodes by the vehicles (average number of passages per minute, in parentheses). All edges between nodes can be used in both directions (indeed, in the internal representation of the graph, all edges are directed, that is one-way, but for clarity of the drawing, only one line is drawn here for both directions). Obviously, from Node 50

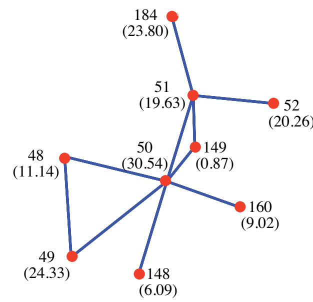

Fig. 9. Zoom Around Node 149 (figures in parenthesis are the frequency of visits by taxis in nr/min)

to Node 51 (back and forth), vehicles go mostly straight and bypass Node 149 by using a parallel route. This particular configuration of the network explains the bad situation at Node 149.

This example shows the ability of our simulation tool to quickly detect flaws and misconceptions of the system.

3) Taxi Activity: During this simulation run, taxis carried from 2.63 to 5.13 clients/hour (these are the extreme values observed over all taxis in service) with an average of 3.77 clients/hour. The average number of passengers per vehicle is 2.16 (individual values vary from 1.29 to 3.15). Recall that the capacity of taxis in this simulation is 5 passengers, that is, in the average, vehicles are not crowded.

Figure 10 shows the percentage of time spent by vehicles with n passengers on board. On request, such histograms can be produced individually for particular taxis.

We are now interested in examining how busy taxis are. Figure 11 shows the percentage of time spent by vehicles on travelling (largest area), doing various operations at nodes (examining new candidate clients, embarking or disembarking passengers — medium area) and finally being idle (smallest area representing only 3% of the total time of service).

4) Quality of Service Provided: We explore now an indicator deserving a particular attention since it is an important ingredient of the *quality of service* provided by the collective taxi system.

We define the *"total detour ratio"* as the ratio of the "actual duration of client's travel plus the initial client's waiting time" over "the duration of client's direct travel" (the latter being evaluated by the shortest path from origin to destination using the average travel times on edges).

Remember that, for the considered run, the parameter s introduced at §IV (see in particular (1)) has been set to 1.9.

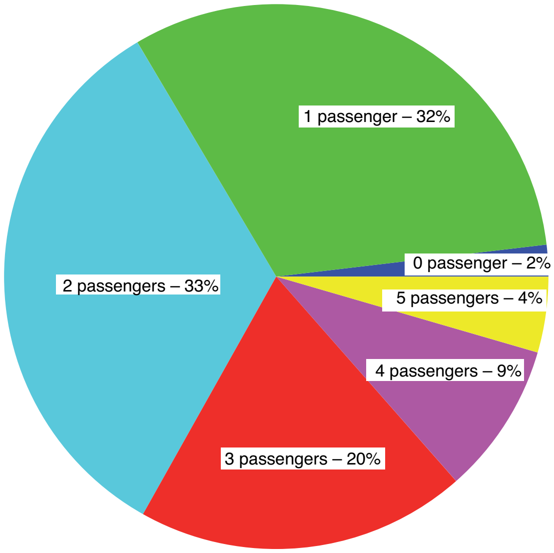

Fig. 10. Percentage Of Time With n Passengers On Board

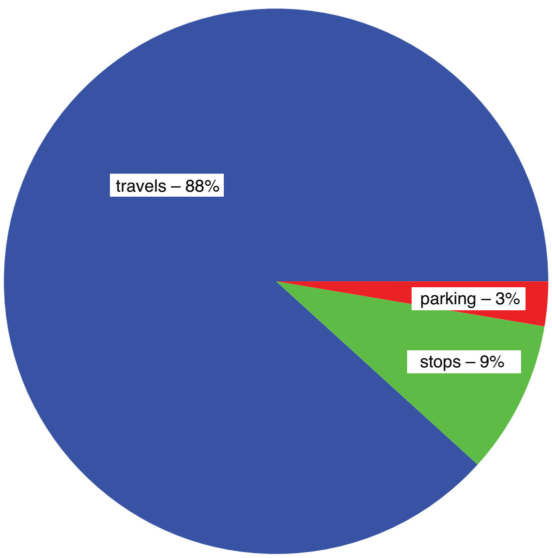

Fig. 11. How Busy Taxis Are

This parameter was introduced in the client acceptance algorithm as a means to moderate, if not absolutely constrain, the "simple detour ratio", which differs from the "total detour ratio" introduced here by the fact that the initial waiting time is not included at the numerator of the ratio (therefore, the total detour ratio is greater than the simple detour ratio). Indeed, if s is decreased, the resulting simple detour ratio will likely decrease too, but, as explained earlier, the initial waiting time is likely to increase (because of a higher probability for candidates to be rejected). Hence the total detour ratio is a nice indicator of the quality of service since it incorporates two conflicting quantities which must be balanced.

Figure 12 presents the histogram of the total detour ratio for all the served clients during the simulation. It may seem strange that a small part of the histogram lies below the value 1. However, remember that the denominator of the ratio defining the total detour ratio uses *average* travel times

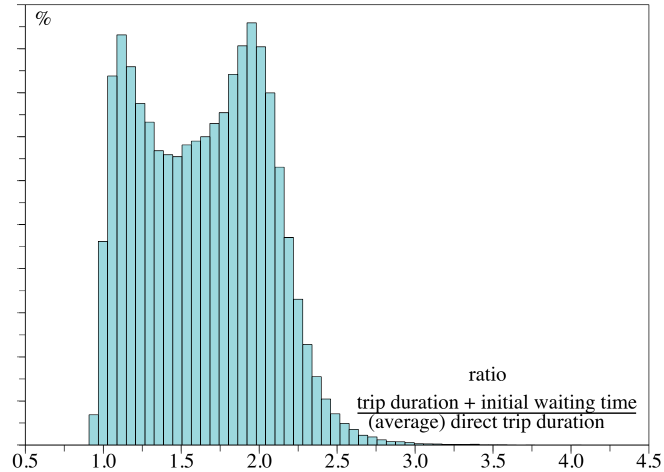

Fig. 12. Total Detour Histogram

on edges, whereas the numerator involves *actual realizations* of those travel times, which may sometimes fall under the average. For the same reason, and also because when s is used in the algorithm, it is applied to trip durations which are partly realized and partly predicted, it should not be surprising either that a part of the histogram lies beyond the value 1.9 adopted for s (plus again the fact that the total detour ratio incorporates the initial waiting time, while s limits only the simple detour ratio).

Nevertheless, it turns out that the average value of the total detour ratio is 1.64 (notably less than the value of s) with a standard deviation of 0.4.

Finally, one may have noticed the two-hump shape of this histogram, a permanent observation in all our experiments (which applies also to the histogram of the simple detour ratio). This suggests the existence of two (or perhaps more) categories of clients. Several explanations have been imagined and tested by additional investigations. We refer to [41] for such a discussion. One (albeit partial) explanation (proposed by F. Meunier) has to do with the number of passengers already on board when a candidate is accepted by a taxi. Indeed, if the histogram is split into separate histograms for clients accepted in empty taxis, clients accepted when one passenger is already on board, etc., it turns out that the left-hand side hump is predominant for clients embarking in empty taxis, and then this hump tends to disappears compared to the right-hand side one when the number of passengers already on board increases.

VI. Tuning Parameters

So far, we have shown some of the information that can be extracted by the detailed analysis of single run of the simulation program using interactive SCILAB scripts. Among the parameter which must be set off-line, let us mention

- the detour threshold s introduced by the client acceptance algorithm;
- the number n of vehicles in service;
- the capacity (maximum number of passengers) of these taxis.

The purpose of this section is to work out a methodology to analyze a series of simulation runs in which one or several of these parameters take a range of values, with the purpose of choosing "the best values". Indeed, given the huge amount of indicators of the quality of service offered to clients, and also of the cost of operation, that can be obtained from each run, the real problem is to select a small number of truly relevant indicators on which to base our choice. As we shall see, the same methodology can also be used to assess the impact of other external factors that we do not directly master, such as the demand, in intensity and geometry.

A. Relevant Indicators And Methodology

Initial waiting times of clients, queue lengths at nodes, detours with respect to direct trip, etc., are some of the numerous indicators of the quality of service offered to clients. Some of them are strongly correlated with each other (we have checked this for waiting times and queue lengths at nodes for example) so that it is sufficient to monitor one of them. Some generally vary in opposite directions under the influence of control parameters: as we observed it already, decreasing the value of s will likely also make the average detours decrease but waiting time increase. In that case, the notion of "total detour" introduced earlier is a compact way of taking both effects into account in a single indicator.

However, notice that such indicators are only relevant for clients that finally became passengers, that is, who finally succeeded to get on board of a taxi. Those who gave up after their maximum waiting time was elapsed are definitely lost: they must be specifically taken into account through the abandonment rate.

Regarding the number n of taxis in service, increasing this value will certainly impact all indicators of the quality of service positively from the point of view of clients. But the price to be paid will be a reduced commercial activity of each vehicle, meaning that fares should be raised in order to ensure the profitability of the system. By the way, it is perhaps time to say that, in the same way as we made no assumptions about the feedback between the quality/fare ratio offered by the system and the demand attracted by this system (demand is an independent input of the simulator), we did not either postulate any particular fare level or tariff structure (fixed, proportional to the direct trip length, etc.). We limit ourselves to extract from simulations the relevant information allowing to evaluate the turnover and associated operation cost for taxis. For example, taking the assumption of the simplest tariff structure, namely, a fixed fare for any passenger, we can measure the numbr of customers served by each taxi during a simulation. With a fare proportional to the direct travel time, a more relevant information would be the total of such direct travel times for all customers transported by each taxi.

Finally, the following three indicators will be especially monitored:

- x: the average client abandonment rate throughout the network;
- y: *minus* the average number of transported customers per vehicle (the *minus* sign will be justified hereafter);
- z: the average total detour ratio of all served passengers: recall that z incorporates two statistical indicators, the *client initial waiting time* and *detours* born by passengers already arrived at their destination.

Indicators x and z improve (decrease) when the number of vehicles in service increases, whereas the average number of clients served per taxi is likely to worsen (that is, also to decrease). Therefore, by choosing "minus" this number (as y is defined above), for all three indicators, *better* now means *smaller*.

In the following, we start by keeping the capacity of taxis constant throughout all runs of a series, but let the threshold detour parameter s and the number n of vehicles in service vary. Then, each run in the series corresponds to particular values given to (s, n), and this run produces a point in a 3D space with coordinates (x, y, z). This cloud of points depending on two degrees of freedom in a 3D space should draw a surface.

Amongst these points, we need to pay attention only to the *non dominated* ones (points for which there are no any other point which is better according to the three coordinates simultaneously — Pareto optimality). We seek values of the parameters (s, n) for which the three selected indicators become *as small as possible*, but of course no point will dominate all other points. We have to make a "reasonable" trade-off amongst the three indicators by choosing some point lying on the surface, which in turn will determine the value of (s, n) to adopt.

Nevertheless, when we compare two situations in which, for example, the demand differs, each demand assumption will provide such a surface and, if one surface is above the other one, it means that, even before choosing a suitable trade-off, we can say that the demands can be ranked as more or less favorable.

In what follows, we illustrate this methodolgy by studying the impact of some factors such as the demand and the capacity of taxis. We start with the influence of demand geometry (mostly characterized by the O-D matrix) but the reader may refer to [41] for similar results corresponding to varying the intensity of demand (characterized by the parameters λ_i introduced at §V-A2).

B. Influence Of The Demand Geometry

We have constructed three different geometries of demand :

- the *centripetal* one where clients move mostly from the periphery towards the city center;
- the *centrifugal* geometry in which the outskirts are more attractive;
- the *balanced* one in which each node emits the same number of clients that it attracts.

Those scenarios are tuned to correspond to the same demand intensity (average number of clients appearing at all nodes per time unit). We refer the reader to [41] to see how this is mathematically done by playing with the O-D matrix M and the vector λ of Poisson process parameters. It suffices to say here that, considering the centripetal and centrifugal demands for example, they intuitively correspond to the movements of people going from home to work in the morning and returning back home in the evening.

We proceed to two series of simulations when varying values of s and n for the centrifugal and centripetal demand scenarios. In Figure 13, the lower surface corresponds to the centripetal demand and the upper one to the centrifugal. Therefore, one can assert that the centripetal geometry is more favorable.

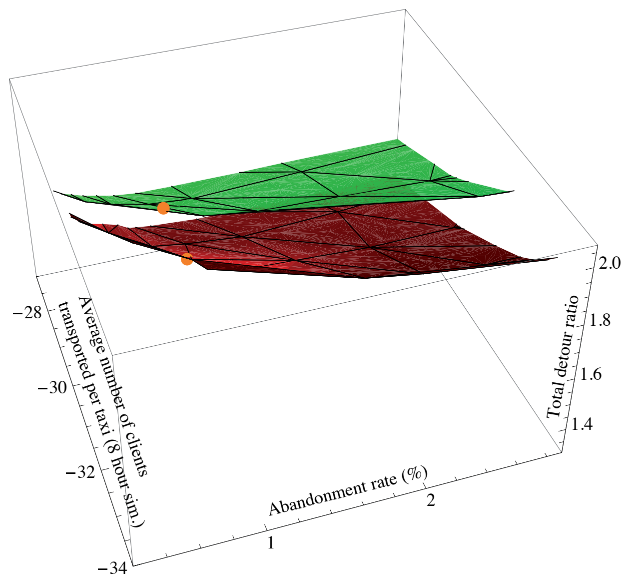

Fig. 13. Comparison Of Centripetal And Centrifugal Demands-3D

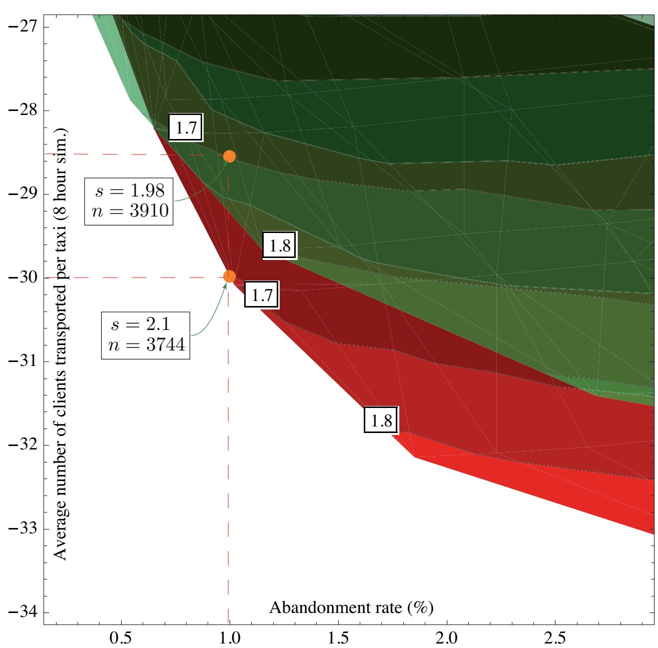

Fig. 14. Comparison Of Centripetal And Centrifugal Demands-2D

In order to quantify this advantage, for each of the two scenarios, we choose a particular point on the corresponding surface in such a way that two of the three coordinates of those points are equal, and we look at difference over the third one. For this operation, it is convenient to use the 2D representation of the surfaces in the (x, y) plane while the z coordinate is represented by its level curves. The level curves are indicated in square boxes in Figure 14. The points chosen on each surface are located on the level curve 1.7 (that is, both achieve an average total detour ratio of 1.7) and they also both achieve a global abandonment rate (x coordinate) of 1%. However, the corresponding average number of transported clients per vehicle $-y$ is greater with the centripetal geometry (about 30 clients for simulations corresponding to 8 hours of real time) than with the centrifugal (only about 28.5). This is consistent with the fact that more vehicles are needed with the centrifugal demand to achieve those performances (3,910 versus 3,744). Finally, a different value of s is also needed: 1.98 versus 2.1.

We can conclude that the centripetal geometry of demand in a network having a topology inspired by the Paris metro plan, tending to accumulate vehicles towards the city center, is more favorable than the centrifugal one, which tends to disperse taxis toward the suburbs. Of course, other topologies may lead to different conclusions.

C. Varying The Vehicle Capacity

In this section, we are interested in the following question: what is the optimum seating capacity and corresponding number of taxis for maximum efficiency? Just as a preliminary study in this direction, we compare series of simulations using either taxis with capacity 5 or 7.

As mentioned at §IV-C, when passing from capacity 5 to 7, it is no longer possible to use the exhaustive enumeration of all possible orders to search for the itinerary which solves the optimization problem of §IV-B: this is too computationally expensive. Therefore, we have used the suboptimal strategy which consists in trying to insert the new candidate at all possible positions in the existing itinerary of the taxi. We mention that experiments conducted with taxis of capacity 5 in order to compare the exhaustive enumeration and this suboptimal solution led to the following conclusion: out of the cases when there are 0 or 1 passenger already on board — since, in that case, there is actually no difference at all between the two strategies —, we observed that the solutions delivered by the two algorithms were different in only 0.5% of the cases. Of course, we do not claim that this conclusion is still valid with capacity 7.

Figure 15 represents the surfaces corresponding to simulations with 5 (upper surface) and 7 passenger seats (lower surface). In this 3D figure, the so-called upper surface indeed crosses the lower surface. Consequently along that curve of intersection, the two systems present the same performances.

Let us now have a closer look at the 2D-representation of Figure 16. As previously, the level curves display the value of z (total detour ratio, now indicated in circles). Consider a particular point at the intersection of the level curves corresponding to $z = 1.8$ (average total detour ratio): this point is obtained with about the same number of vehicles (4,104 versus 4,105 taxis, for capacities 5 and 7, respectively) and for slightly different values of s (2,02 versus 1,96). In both cases, the abandonment rate is $x = 1.85\%$, and the number of

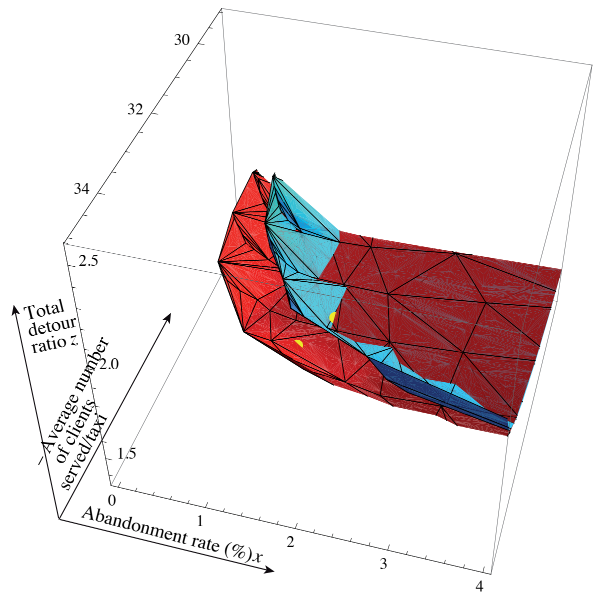

Fig. 15. Comparison Of 5 And 7 Passengers Capacities-3D

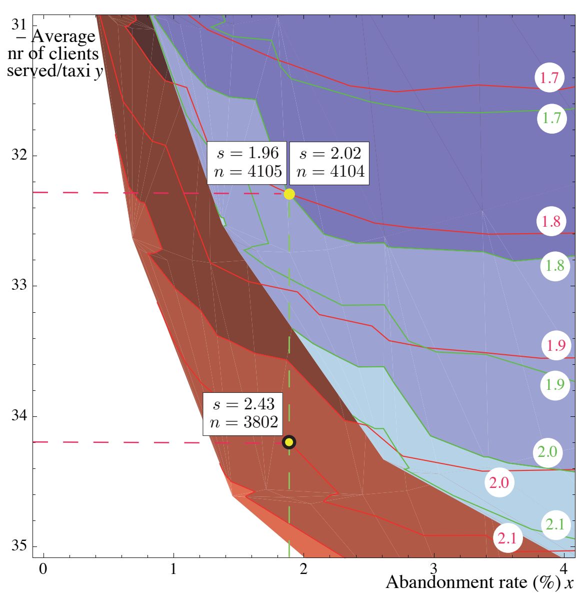

Fig. 16. Comparison Of 5 And 7 Passengers Capacities-2D

transported passengers per vehicle $-y$ is approximately equal to 32.25. Therefore, for such an operating point, it is apparent that the greater vehicle capacity does not bring any significant contribution, that is, the additional capacity is not really used.

If we consider now the alternative performance point of Figure 16 (circled in black), which is accessible only by the system employing 7 passenger vehicles, we observe that the number of transported clients per vehicle does increase to 34.2, while preserving the same value of the client abandonment rate $x = 1.8$ as previously. However, the average total detour ratio jumps to 2.1 i,nstaed of 1.8 previously. We conclude that this configuration allows the transportation of more clients over the 8 hour simulation(therefore costs would be lower), but at the expense of providing a lower quality of service.

In conclusion, the smaller vehicle capacity is adequate to gear the system toward performances that are more "taxi oriented" (although already "collective"), whereas the larger capacity turns the system more toward a collective transportation mode.

VII. Conclusion

The Collective Taxi structure is a real world stochastic multi agent system heavily affected by random external factors. Simulation is a promising means of allowing accurate assessment of the system behavior and consequently its optimal exploitation at almost no risk.

The purpose of this paper is to present a brief overview of a methodology providing an optimal management of such a "Collective Taxi" system by discrete event simulations. With the aim of evaluating the system performances for any possible strategy applied, a discrete event simulator tool has been developed, tested and validated, which enable us to manage the three system approaches (decentralized, centralized and mixed managements). After constructing the necessary decision algorithms concerning the control of the system (management of clients and vehicles), but at this stage only for the decentralized mode, we initiated multiple experiments according to different scenarios.

We were then faced with the challenge of dealing with the enormous information output provided by each experiment. After retrieval of the simulation results, we proceeded to a statistical analysis with the aim of providing some preliminary conclusions characterizing the system performances for the chosen policy. Furthermore, we presented a brief methodology suggesting the choice of optimal values for the parameters involved or reasonable trade-offs to permit satisfactory results.

Future work will aim at the study of the centralized and mixed managements. More precisely, we are interested in developing the control algorithms for these approaches, and subsequently to proceed to a comparison of the system performances for each mode (examining if and when the additional costs of central dispatching are justified, etc.). Moreover, a study employing real data can be envisaged in order to provide persuasive answers for those who still hesitate regarding the effective productivity of "Collective Taxis".

Acknowledgment

The authors are indebted to Dr. Frédéric Meunier of LVMT-ENPC for interesting discussions about routing algorithms, and in particular for pointing out the reference [36].

References

[1] E. Lioris, G. Cohen, and A. de La Fortelle, "Evaluation of collective taxi systems by discrete-event simulation," in *The Second International Conference on Advances in System Simulation (SIMUL 2010)*, Nice, France, August 22-27 2010.

[2] "Share taxi," http://en.wikipedia.org/wiki/Share_taxi.

[3] B. Mayhew, *Central Asia*. Lonely Planet, 2007.

[4] A. Bernstein, "NYC taxi commish setting up shared cabs from U.S. open," May 2011, http://transportationnation.org/2011/05/20/nyc-taxi-commish-setting-up-shared-cabs-from-u-s-open/.

[5] "Bay area taxi cab," http://www.bayareataxi.us/services.php.

[6] "Prime time shuttle," http://www.primetimeshuttle.com/.

[7] "Super Shuttle," http://www.supershuttle.com/.

[8] "Train taxi," http://www.holland.com/uk/gettingthere/around/train/traintaxi.jsp.
[9] D. Dufour, "Shared taxi in Brussels: the missing link in urban transport?" 2008, http://www.urbanicity.org/Site/Articles/Dufour.aspx.
[10] "TaxiShareUK," http://www.taxishareuk.com/.
[11] "Rural transport 2000, transport solutions to sustain the rural economy," http://www.buergerbusse-in-deutschland.de/03_vortraege/virgil_differentiated-forms-of-service-in-germany.htm.
[12] P. Lalos, A. Korres, C. K. Datsikas, G. Tombras, and K. Peppas, "A framework for dynamic car and taxi pools with the use of positioning systems," in *2009 Computation World*, Athens, Greece, November 15-20 2009.
[13] C. Tao, "Dynamic taxi-sharing service using intelligent transportation system technologies," in *International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, Shanghai, China, 21-25 September 2007, pp. 3209–3212.
[14] F. Ciari, M. Balmer, and K. Axhausen, "Large scale use of collective taxis: a multi-agent approach," in *12th International Conference on Travel Behaviour Research*, Jaipur, India, December 2009.
[15] J. Xu and Z. Huang, "Autonomous dial-a-ride transit introductory overview," *Journal of Software*, vol. 4, no. 7, pp. 766–776, September 2009.
[16] P. Fargier and G. Cohen, "Study of a collective taxi system," in *Proceedings of the Fifth International Symposium on the Theory of Traffic Flow and Transportation*, G. Newell, Ed. University of California, Berkeley: American Elsevier, June 16–18 1971, pp. 361–376.
[17] D. W. Lee, E. J. Bass, S. D. Patek, and J. A. Boyd, "A traffic engineering model for air taxi services," *Transportation Research E: The Logistics and Transportation Review*, vol. 44, pp. 1139–1161, 2008.
[18] M. Barth and M. Todd, "Simulation model performance analysis of a multiple station shared vehicle system," *Transportation Research Part C: Emerging Technologies*, vol. 7, pp. 237–259, 1999.
[19] R. Keen and J. Spain, *Computer simulation in biology, a basic introduction*. John Wiley & Sons, 1992, 2nd edition.
[20] E. Gloaguen, C. Dubreuil-Boisclair, P. Simard, B. Giroux, and D. Marcotte, "Simulation of porosity field using wavelet Bayesian inversion of crosswell GPR and log data," in *XIIIth International Conference on Ground Penetrating Radar*, Lecce, Italy, 21-25 June 2010, pp. 598–603.
[21] "US naval academy uses simulation solutions," http://navaltoday.com/2011/03/11/us-naval-academy-uses-simulation-solutions/.
[22] "Simulation," http://en.wikipedia.org/wiki/Simulation.
[23] Y. Wu, "Agent behaviour in peer-to-peer shared ride systems," Master's thesis, Departement of Geomatics, The University of Melbourne, Australia, 2007, http://people.eng.unimelb.edu.au/winter/pubs/wu07agent.pdf.
[24] M. Mes, M. van der Heijden, and A. van Harten, "Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems," *European Journal of Operational Research*, vol. 181, no. 1, pp. 59–75, 2007.
[25] H. Klee and R. Allen, *Simulation of dynamic systems with MATLAB and Simulink*. CRC Press Inc., 2011, 2nd edition.
[26] N. Matloff, "A discrete-event simulation course based on the SimPy language," 2010, http://heather.cs.ucdavis.edu/~matloff/simcourse.html.
[27] D. Bollier and A. Eliëns, "SIM: a C++ library for discrete event simulation," 1995, http://www.cs.vu.nl/~eliens/sim/sim_html/sim.html.
[28] E. Kofman, M. Lapadula, and E. Pagliero, "Powerdevs: A devs-based environment for hybrid system modeling and simulation," FCEIA, National University of Rosario, Tech. Rep., 2003, http://www.fceia.unr.edu.ar/~kofman/files/lsd0306.pdf.
[29] "Tortuga (software)," http://en.wikipedia.org/wiki/Tortuga_(software).
[30] A. Varol and M. Günal, "SharpSim tutorial-1," May 2011, http://sharpsim.codeplex.com/.
[31] A. Borshchev, "Simulation modeling with AnyLogic: Agent based, discrete event and system dynamics methods," 2011, http://www.xjtek.com/anylogic/resources/book/.
[32] W. David Kelton, R. Sadowski, and D. Sturrock, *Simulation with Arena*. McGraw-Hill Higher Education, 2003.
[33] K. Concannon, M. Elder, K. Hindle, J. Tremble, and S. Tse, *Simulation Modeling with SIMUL8*. Visual Thinking International, 2007, 4th edition.
[34] "Simcad Pro," http://en.wikipedia.org/wiki/Simcad_Pro.
[35] "Lanner group ltd," http://en.wikipedia.org/wiki/Lanner_Group_Ltd.
[36] J. Tsitsiklis, "Special cases of travelling salesman and repairman problems with time windows," *Networks*, vol. 22, pp. 263–282, 1992.
[37] S. Gaubert, "Howard's multichain policy iteration algorithm for max-plus linear maps." [Online]. Available: http://amadeus.inria.fr/gaubert/HOWARD2.html
[38] J. Cochet-Terrasson and S. Gaubert, "Policy iteration algorithm for shortest path problems," 2000, http://amadeus.inria.fr/gaubert/PAPERS/shortestpath.ps.
[39] C. Bunks, J. Chancelier, F. Delebecque, C. Gomez, M. Goursat, R. Nikoukhah, and S. Steer, *Engineering and Scientific Computing with Scilab*. Boston: Birkhäuser, 1999.
[40] C. Gomez and M. Goursat, "METANET: a system for network problems study," INRIA, Rocquencourt, Le Chesnay, France, Tech. Rep. 124, Novembre 1990, http://hal.inria.fr/docs/00/07/00/43/PDF/RT-0124.pdf.
[41] E. Lioris, "Évaluation et optimisation de systêmes de taxis collectifs en simulation," Ph.D. dissertation, École des Ponts-ParisTech, Marne la Vallée, France, December 2010.