END OF SEMESTER PROJECT

DRONE DELIVERY [a]

Polytech Nice - Sophia-Antipolis
Computer Sciences - Third year

Benjamin Benni

January, 2017

[a] this document is an extension of the *problem statement for online qualification round* of the google hashcode 2016 contest. the complete description is available via `https://hashcode.withgoogle.com/2016/tasks/hashcode2016_qualification_task.pdf`. it has been freely modified to suit our needs in terms of pedagogical goals and project duration.

1 Background

The Internet has profoundly changed the way we buy things, but the online shopping of today is likely not the end of that change; after each purchase we still need to wait multiple days for physical goods to be carried to our doorstep. This is where drones come in - autonomous, electric vehicles delivering online purchases, never stuck in traffic. This new way of shipping goods is different from the actual one: trucks can't fly[1] and road traffic is too complex to have fully autonomous trucks therefore it is in another class of vehicle routing problem. As drone technology improves every year, it remains a major issue: how do we manage and coordinate all those drones? Here at AmIdone we aim to fix this and overpass all state-of-art, old-fashioned delivery companies[2]. We want to provide an AmIdone air prime premium plus delivery option that uses drones to deliver orders faster.

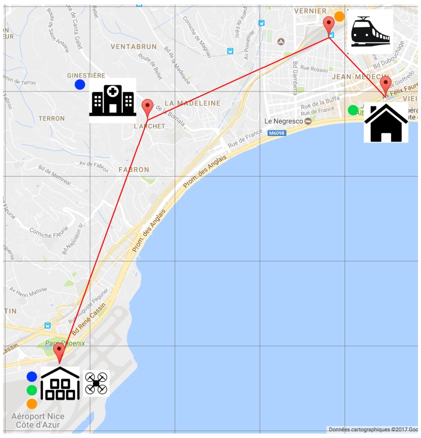
This is where **you** come in: given a fleet of drones, a list of customer orders and availability of the individual products in warehouses, schedule the drone operations so that the orders are completed as soon as possible. Effectiveness of the shipping process is the most valuable key indicator of performance in our industry since it is tight to customer satisfaction[3]. We need to find to way to deliver goods faster and faster. As a KPI of the effectiveness of a delivery strategy and of customers' satisfaction, we can take the time when an ordered was fully delivered: the earliest the customer received her entire order, the best it is.

2 Example

We give an example of drone routing in Fig.1. A drone loads goods at the Nice airport and will dispatch loaded goods to customers. In its tour, the drone will have to deliver a battery to a traveler at the Nice central train station, deliver phone parts at the house of a customer and supply drugs to the Archet Hospital. If you know road traffic in Nice, you see here why using drones instead of trucks is a huge improvement in terms of reducing delivering time. The Fig.1 displays a possible route to fulfill such request but many other routes exist. In fact, when we scale up and have to deliver 71 orders, we have $5 * 10^{80}$ different paths possible which is approximately the number of atoms in the entire known universe.

The simulation takes place on a two dimensional grid composed of cells identified by their coordinates. To simplify things, we define products as following: a product item has a product type (*e.g.* a Samsung Galaxy Note 7 item has Samsung Galaxy Note 7 type). No product items will be added or removed during the simulation. Every items of a product type weight the same. A product item has a weight inferior to the maximum payload that a drone can carry. A drone has an unlimited battery autonomy, can't be hunt down not fail in any way. Product items are stored in warehouse inventories that are located on a specific location on the map. Product items are meant to be delivered via drones to customer locations, also defined with an ID, and a location on

[1] You don't say.
[2] Who uses trucks and humans to deliver goods nowadays?
[3] If we listen to our junior quality manager



(a) Picture of the map, items, orders, and a path between them

Warehouse is at the airport (4, 0), with a single drone
Item Blue must delivered to Archet Hospital (1, 1)
Item Green must delivered at the city center (1, 4)
Item Orange must delivered to the train station (0,3)

(b) Description of the context

Figure 1: Example of deliveries

the map. No order nor warehouses can have the same location. Drones, since they fly, can overlap warehouses, orders or other drones location. In the given example, the map is 5x5, the warehouse #0 is located a (4, 0), orders #0, #1, and #2 are respectively located at (1, 4), (0, 3) and (1, 1) on the grid. Every drones start at warehouse #0 therefore drone #0 is located at (4, 0). Let say that the maximum payload of a drone is 150 units and that the blue, green and orange items weight respectively 25, 75, and 50 units. The drone can load every

Load Blue and Green
Deliver Blue at (1,1)
Deliver Green at (1,4)
Load Orange at (4,1)
Deliver Orange at (0,3)

(a)

Load Blue and Green
Deliver Green at (1,4)
Deliver Blue at (1,1)
Load Orange at (4,1)
Deliver Orange at (0,3)

(b)

Load Blue and Orange
Deliver Orange at (0,3)
Deliver Blue at (1,1)
Load Green at (4,1)
Deliver Green at (1,4)

(c)

Load Blue and Orange
Deliver Blue at (1,1)
Deliver Orange at (0,3)
Load Green at (4,1)
Deliver Green at (1,4)

(d)

Load Green and Orange
Deliver Green at (1,4)
Deliver Orange at (0,3)
Load Blue at (4,1)
Deliver Blue at (1,1)

(e)

Load Green and Orange
Deliver Orange at (0,3)
Deliver Green at (1,4)
Load Blue at (4,1)
Deliver Blue at (1,1)

(f)

Figure 2: Schedules possible to deliver items, maximizing drone payload

items and deliver the Archet hospital (H) then the train station (T) and finally the center of the city (C). If we change the input configuration and change the weight of the blue item from 25 to 30 units, the drone has to let an item at the warehouse and deliver the other two before going back to the warehouse and delivery the third one. Therefore, different schedules, depicted in Tab.2, can be computed.

3 Expected work

We need to ensure that orders will be completed and when they will be. We also need to know the performance of our delivery system in order to change or adapt it to a given context. Therefore we strongly need to simulate our drone delivery before deploying it into the real-world. Deploying real drones with real antennas, real items, and so on, is expensive and we won't invest in this without the insurance that it will work. That is why we called your services. On the one hand we want to compute an effectiveness scheduling and compare different scheduling strategies, and on the other hand we want to visualise this schedule from the company and the end-user point of view. From this, we already identified[4] that there are 3 sections in our new drone delivery system:

- The *scheduler* part that orchestrates the drone fleet by applying a delivering strategy. It produces a sequence of commands (*e.g.* deliver, load, fly) that represent what it has scheduled,
- The *visualiser* part that allows either the local operator of our company or the end-user that ordered items, to see a given *execution* of deliveries,
- The *benchmark* part that allows the quality manager to perform comparison of delivering strategies and produces exploitable results.

The simulation's time is handled by a discrete clock that have a maximum number of turns. The simulation ends when it reaches the maximum number of turns allowed. Each action but drone moves take exactly one turn to be processed. We define a drone flight as a single movement of the drone that happens between two subsequent commands for the given drone (as described below). Each drone flight is computed via the two dimension euclidean distance. This part is described in the technical reference. As said in the introduction of this paper, scheduling effectiveness is the most important KPI [5] for us. We defined a metric that describes this scheduling effectiveness[6]. We think that the earlier an order is fulfilled the best it is. You can obviously come with your own effectiveness scoring, if you explain how compute it and why you think it is a good KPI.

[4] Actually our junior-engineer comes up with this
[5] https://fr.wikipedia.org/wiki/Indicateur_cl%C3%A9_de_performance
[6] The description of this metric is available in technical reference document

4 Milestones

4.1 Global schedule

These section describes milestones of the project. **It must not be taken as deliveries**. Milestones are here to inform you about your progression. You **can go faster** than the given milestones, but if you are behind this schedule you may be late. Each milestone will be automated. A script will clone your repository at the given time and perform the corresponding analyses, similarly of what your previously experienced in QGL. The remainder of this section describes the expectation for each milestone. The Tab.1 displays the schedule of the week [7]. This project is cut into two parts:

- from Monday to Wednesday (or less) you will be able to implement the full execution chain (as described in the milestones section below), this doesn't mean you have to implement everything described in this document, but that you have to do the minimal base, as described below.
- from Wednesday (or less) to Friday (or more) you will dig a specific axis that you will choose and that is described below.

A milestone is achieved only when at least the described conditions are met. You can do more than described in the different milestones, but you have do complete it entirely to mark it as reached. A milestone is reached only when the described conditions are met, meaning that there is also no need to do more that the milestone require to reach it. See milestones as short-term goals to achieve and let the second part of the project as extension and more complex works. Let think upside-down and start from the result to go to the start of the week. You want to deliver the ultimate solution [8] and a complete report by Sunday, so you have to ensure that you **can** do the complete execution chain at the middle of the week (to have time left to build **the ultimate** solution). So if you have to deliver the complete execution chain by the middle of the week (Wednesday), you want to be sure (and your client too) that you can do a tiny, basic working solution by Tuesday. So if you have to deliver a tiny, basic working solution by Tuesday, you want to be sure (and your client too) that you can do a working skeleton, that do the simplest things possible, focused at the core of the system, by Monday night. Start small but see big(ger).

4.2 Common base

Milestone #0: Walking skeleton. Monday, 16 at 8PM. It is the very first milestone of your project. This is the tiny step you have to do at the beginning to be sure that the further steps will go taller. You can parse the input file 'context0.in'. The scheduler produces only $Wait$ commands for all the drones of the fleet, therefore the `scheduler.out` file contains as many $Wait$ commands (and as many lines) as drones described in the input file. A `map.csv` file is generated by the scheduler, describing the map of the input file `context0.in`. The csv file should map "line" of the grid-map to a line of the file. Each line

[7] Every hours are given on the GMT+0100 (CET) timezone. Maybe you want to check this: `http://anglais-pratique.fr/index.php/rubriques/erreursfrequentes/57-12am-12pm`

[8] Yes, you do.

Monday	Tuesday	Wednesday
8AM: Kick-off 8PM: Milestone #**0**	12:30PM: Milestone #**1**	12:30PM: Milestone #**2**
Thursday	**Friday**	**WE**
8AM-12:30PM: Demo **C. Papazian** 2PM-4PM: Office O+306 **A.M. Déry** 3:30PM-5:30PM: Office O+450 **S. Mosser** 2PM-4PM: Office O+444		Delivery

Table 1: Schedule - every hours are given GMT+0100 (CET)

is the same size as the width of the grid-map. Each element of the csv line can contains the following element: W_i, O_i, ϵ where W_i denotes the $i_t h$ warehouse (as described in the input file) and O_i denotes the $i_t h$ order (as described in the input file) and ϵ denotes the empty element (there is nothing here in the grid-map). The Fig.3 displays an example of equivalence.

Milestone #1: Walking-a-bit-more skeleton. Tuesday, 17 at 20. You have implemented 1 or 2 non-trivial strategies and can parse any well-formed input files. Therefore you have implemented different commands and their respective execution. The scheduler outputs *scheduler.out* files that contain commands else than $Wait$ commands and depict a strategy. You can score your simulation(s) according to the scoring section described in the technical document. You have a basic visualisation provided by the visualiser such as a command-line interface that takes a file and "plays" it. You may have to refactor the scheduler to match these requirements and build on top of your system. Your strategies don't have to be complex, but have to be simple. You can take into account only one drone and make it deliver everything, one item at a time ; or take into account the whole fleet but make each drone deliver one item at a time ; or take into account a single drone that will maximise its load to deliver multiple items at a time ; or anything else **simple**.

Milestone #2: Common base. Wednesday, 16 at 12:30. You achieved to run the whole execution chain, and compared at least 4 different strategies. You accept any well or ill-formed inputs, parse them correctly. The scheduler can (and will) apply 4 different delivery strategies and produce a non-empty, well-formed, non-trivial *scheduler.out* file. From that point, the Visualiser consume the generated traceand "play" the trace as a video player will do. The

W_0		O_0	
	O_1		W_1

(a) Representation of an example of grid

```
W0,,O0,
,O1,,W1
```

(b) CSV format description equivalence

Figure 3: Example of mapping from grid to csv description

Benchmark produces at least 1 complete, sound and coherent dashboard. At this milestone, the scheduler is non-trivial, the benchmark produces relevant (not necessary complex) information and the visualiser allows one to see what happened during the simulation, therefore the whole execution chain is implemented. We will call this the common base.

4.3 User-stories of common-base

We wrote down some user-stories for you. There are expression of what your user(s) needs. We introduce three characters (or personna) that will symbolically represent three different type of users:

- a customer who placed an order an wait for it to be delivered,
- a local operator that has received a set of orders from the headquarters and has to deliver them,
- a quality manager that wants to analyse each delivery process and compare different delivery strategies.

The user-stories (U.S.) below are not in any particular order and are here to help you in your development. Those can be shown as tiny milestones that must be completed[9].

U.S.1 As a local operator, I want to see the position of all drones in order to supervise the deliveries via a textual console,

U.S.2 As a local operator, I want to see the inventory (*i.e.* content) of all drones in order to supervise the deliveries via a textual console,

U.S.3 As a local operator, I want to see the position of all warehouses in order to supervise the deliveries via a textual console,

U.S.4 As a local operator, I want to see the actual inventory (*i.e.* content) of all warehouses in order to supervise the deliveries via a textual console,

U.S.5 As a local operator, I want to see the position of all orders in order to supervise the deliveries via a textual console,

U.S.6 As a local operator, I want to see the actual content of all orders in order to supervise the deliveries via a textual console,

[9] Therefore you have to decide how, why and when this tiny milestone will be complete

- U.S.7 As a customer, I want to know the estimated time of arrival of an item, in order to be there on time,
- U.S.8 As a customer, I want to know the estimated time of arrival of my order, in order to know when it will be complete,
- U.S.9 As a customer, I want to know the status of my order, in order to know if I have to wait another item,
- U.S.10 As a quality manager, I want to analyse a scheduling (previously computed), in order to know the number of drones used for that planification,
- U.S.11 As a quality manager, I want to analyse a scheduling (previously computed), in order to establish the satisfaction score of that planification,
- U.S.12 As a quality manager, I want to analyse a scheduling (previously computed), in order to visually know the number of drones used for that planification,
- U.S.13 As a quality manager, I want to analyse a scheduling (previously computed), in order to visually know the satisfaction score of that planification,
- U.S.14 As a quality manager, I want to compare two scheduling (previously computed), in order to visualise the different KPIs on a web-page,
- U.S.15 As a local operator, I want to enter my test-list [10] of orders, in order to get the minimal planification (do nothing),
- U.S.16 As a local operator, I want to enter my test-list [11] of orders, in order to draw the map of the city,
- U.S.17 As a local operator, I want to enter a list of orders, in order to get the minimal planification (do nothing),
- U.S.18 As a local operator, I want to enter a context, in order to get a planification using a single drone,
- U.S.19 As a local operator, I want to enter a context, in order to get a planification where a drone load one item at a time,
- U.S.20 As a local operator, I want to enter a context, in order to get a planification maximizing the load of each drone,
- U.S.21 As a local operator, I want to enter a context, in order to get a planification using the whole fleet.

[10] https://github.com/ttben/2017-01-si3/blob/master/contexts/context0.in
[11] https://github.com/ttben/2017-01-si3/blob/master/contexts/context0.in

4.4 Extensions

You see this hole from Wednesday afternoon to Friday afternoon? This is not a mistake, this where **you** come in.

Show us what you got. At this point you have reached the "common base" milestone. You handled the whole execution chain and have reached the minimal requirement. Now, you have to decide what to do. You are free to dig the following axis, pick one or more that may fit more or less your team:

- Algorithm: you are definitely into digging the scheduler and providing an optimal scheduler to reach the highest score possible (if known). Christophe Papazian is an algorithm specialist and is available Tuesday, the 19, from 2PM to 4PM at his office O+306.
- Benchmark: you are definitely into extracting information from executor's trace and provide industrial-level dashboard, with professional charts computed in a scientific ways. Sébastien Mosser can help you binding the benchmark on the scheduler in a clean and efficient way and is available Tuesday, the 19, from 2PM to 4PM at his office O+444. Anne-Marie Déry is a H.C.I. specialist and is available Tuesday, the 19, from 3:30PM to 5:30PM at her office O+450, and can help you build clean and meaningful dashboards.
- Player: you are definitely into providing a high-quality-level, end-user interface that allows one to track drones, orders and warehouses status overtime, visually compare multiple strategies at once. Anne-Marie Déry is a H.C.I. specialist and is available Tuesday, the 19, from 3:30PM to 5:30PM at her office O+450.

You are free to pick one of the axis from the list above. Taking more than one is discouraged but not forbidden since digging one of the axis above is a project on its own. There is absolutely no obligation to see the people listed above. These are suggestions of people, specialist in a domain, that may help you.

Delivery. The complete project must be delivered by Sunday, the 20th at 20 GMT+0100 (CET). This includes your repository, your report and your benchmark results. Your repository will be automatically cloned at the deadline. Your report file is already is the root of your team folder and is (and must remained) named 'Rapport-TeamName' where "TeamName" must be replaced with the name of your team[12]. Your code must compile running a *mvn clean install* command, and be ran using a $mvn\ exec : java\ -Dexec.args = ...$ command. Your code must be tested and documented. Every QGL's rule apply here and must be applied.

[12] I think you won't have question about this.

5 Resources and technical reference

- A GitHub repository[13] is available, containing code samples and context samples.
- A PDF file named technical-reference contains detailed description of commands, file format, and expected architecture. You will find a lot of answers to your question in it. It also contains information about scheduling of the project, vocabulary, hypothesis, work-flow, and so on. Please read it carefully before asking any questions.
- The Project Piazza environment is still active and will be use in this project as the main way to communicate.
- Slides of the kick-off[14].

[13] `https://github.com/ttben/2017-01-si3`

[14] `https://docs.google.com/presentation/d/1ZKidmSTA6e2AWRqlEQRxlwU9mSCK0nnXYxLqBcEcBHw/edit?usp=sharing`