Declarative Programming Project:
Delivery Planning

2016-2017

Presentation by Steven Adriaensen

The Delivery Planning Problem

A company:

- A set of depots storing goods.
- A fleet of vehicles for transporting them
- A set of orders that must be delivered

The Delivery Planning Problem

Problem:

- When should which orders be shipped?
- From what depot?
- Using which vehicle?
- In what sequence should they be delivered?

… to maximize profit.

➔ **Delivery Plan**

The Delivery Planning Problem

- Making good delivery plans is challenging:
  - Large # orders with tight delivery deadlines.
  - Depots have a restricted inventory.
  - Finite # vehicles with limited capacity & speed.
  - Tradeoff between
    - Revenue from delivering orders, timely.
    - Costs of delivering orders (gasoline, wages drivers)

- Automating this task is an active research area.

Problem Instances

- We provide you with 10 problem instances, varying in size and difficulty.

#	name	# orders	# product types	# depots	# vehicles	# working days (hours)	optimal profit
1	single_small	10	3	1	1	1 (12)	€1392.8
2	multi_depots_small	10	3	3	1	1 (10)	€878.0
3	multi_vehicles_small	10	3	1	3	1 (6)	€1128.2
4	multi_days_small	10	3	1	1	3 (18)	€900.4
5	multi_small	10	3	2	2	2 (12)	€804.2
6	single_large	100	4	1	1	1 (12)	???
7	multi_depots_large	100	4	5	1	1 (12)	???
8	multi_vehicles_large	100	4	1	5	1 (10)	???
9	multi_days_large	100	4	1	1	5 (50)	???
10	multi_large	100	4	5	5	5 (50)	???

- Your solver should be able to solve **any** instance.

A Concrete Problem Instance...

Types of products sold:

```
%2 types of products p1 and p2:

product(p1,10,0.1).
%items of product p1 weighs 100gr and cost €10.

product(p2,80,25).
%items of product p2 weighs 25kg and cost €80.
```

A Concrete Problem Instance...

Orders for products:

```
% 2 orders o1 and o2

order(o1,[p1/2],location(2,2),1).
% order o1 consists of 2 items of p1 and must be
% delivered before day 2 of the planning period.

order(o2,[p2/1],location(2,3),2).
% order o1 consists of 1 item of p2 and must be
% delivered before day 3 of the planning period.
```

A Concrete Problem Instance...

Depots storing goods:

```
% 2 depots d1 and d2:

depot(d1,[p1/7],location(4,3)).
% depot d1 has 7 items of p1.

depot(d2,[p2/5,p1/1],location(1,1)).
% depot d2 has 5 items of p2 and 1 of p1.
```

A Concrete Problem Instance...

Vehicles transporting goods:

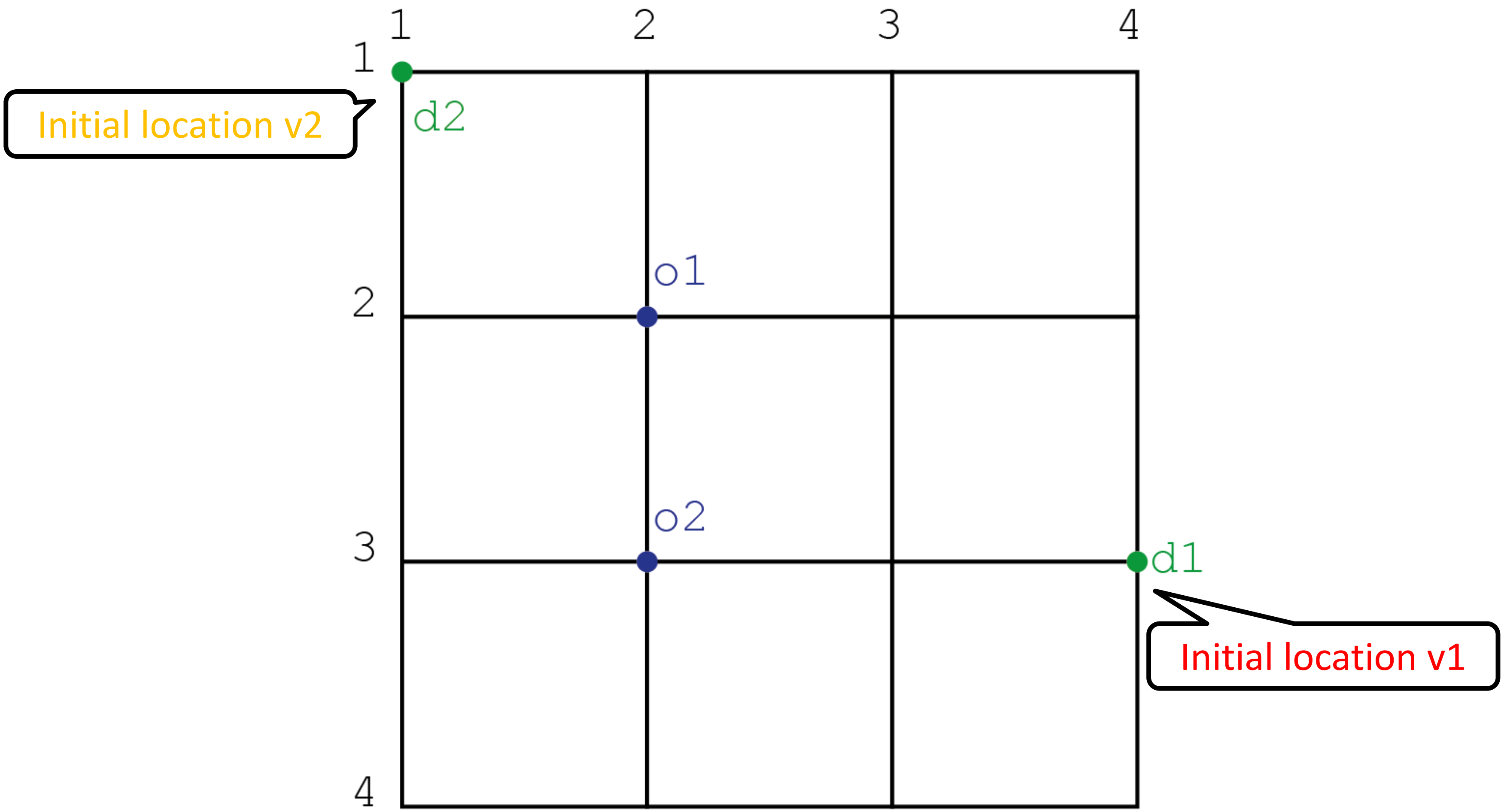
```
% 2 vehicles v1 and v2

vehicle(v1,d1,100,0.75,50,0.5).
% vehicle v1, is at the beginning of the planning
% period located at depot d1. It has a capacity
% of 100kg and moves at a pace of 0.75 min/km
% (80 km/h). It costs €50 for each day of use,
% and €0.50 per km driven.

vehicle(v2,d2,200,1,50,0.5).
% vehicle v2 differs from v1 in that it is
% initially located at depot d2, has a capacity
% of 200 kg and moves at 1.0 min/km (60 km/h).
```

A Concrete Problem Instance...

... during working days:

```
% 2 working days, the first and third of the
% planning period:

working_day(1,540,1020).
% on day 1 vehicles can leave the depot as of
% 9am and must return to a depot by 5pm latest.

working_day(3,360,1080).
% on day 3 vehicles can leave the depot as of 6am
% and must return to a depot by 6pm latest.

% no transport can take place on day 2
% (or 4,5,...).
```

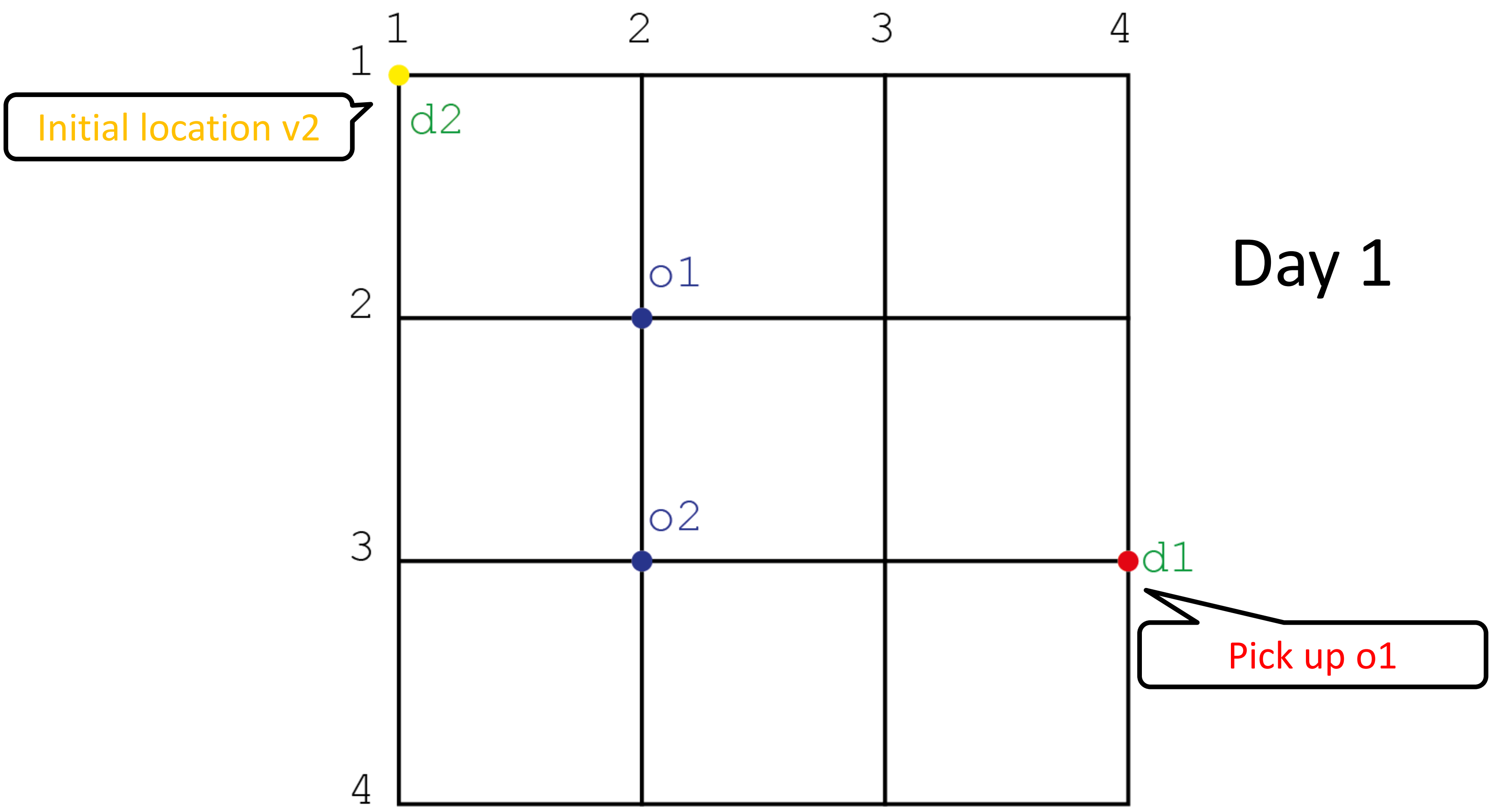
A Concrete Problem Instance...



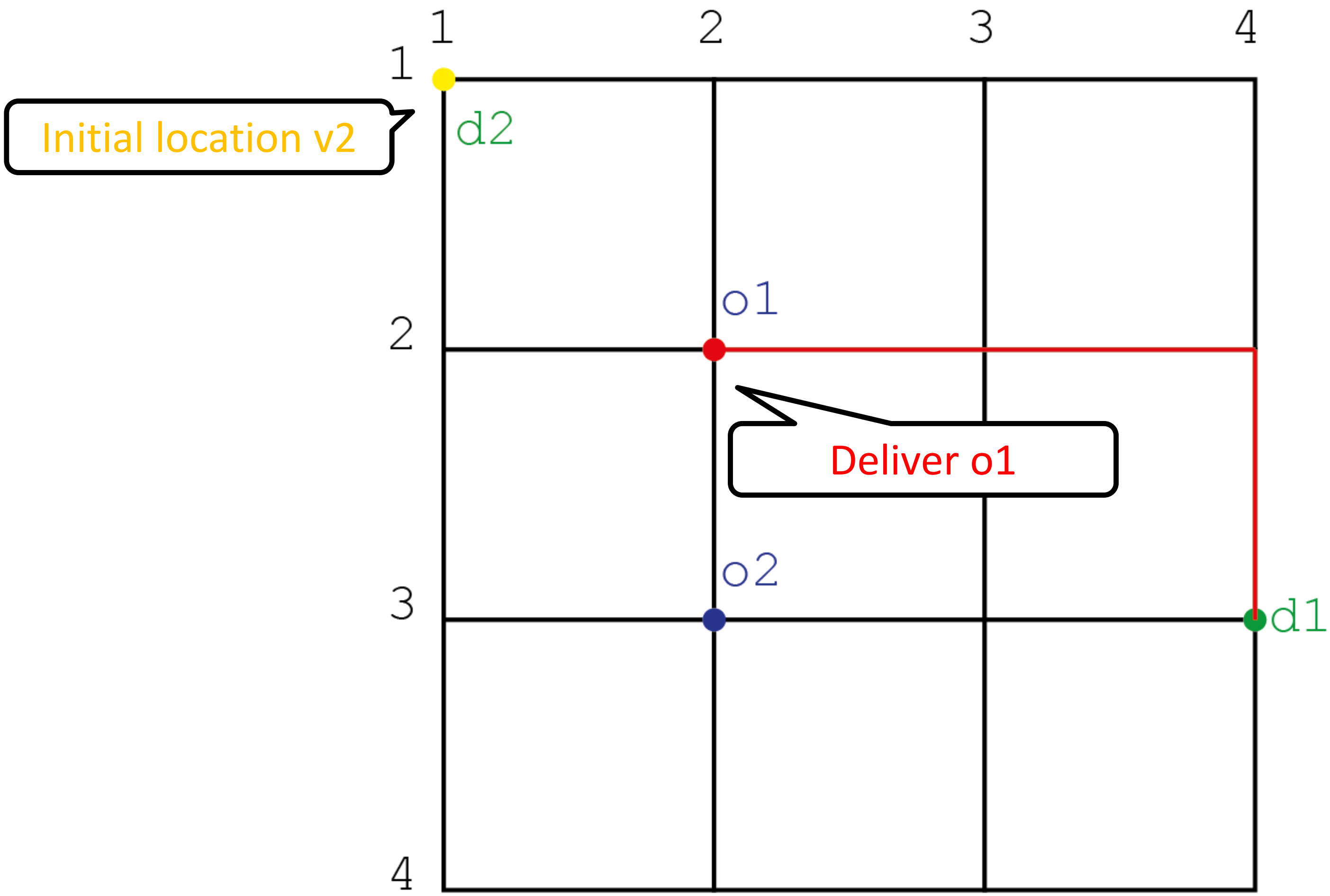
A Concrete Delivery Plan

```
*** Schedule for Day 1 ***

< Vehicle v1 >

Time       Loc.       Load     Action
09:00     (4,3)        0kg     Pick up order o1 from depot d1
09:05     (4,3)      0.2kg     Drive 3km to the intersection of 2nd avenue and 2nd street.
09:07     (2,2)      0.2kg     Deliver order o1.
09:12     (2,2)        0kg     Drive 2km to the intersection of 1st avenue and 1st street.
09:13     (1,1)        0kg     Park at depot d2.


*** Schedule for Day 3 ***

< Vehicle v2 >

Time       Loc.       Load     Action
06:00     (1,1)        0kg     Pick up order o2 from depot d2
06:05     (1,1)       25kg     Drive 3km to the intersection of 2nd avenue and 3rd street.
06:08     (2,3)       25kg     Deliver order o2.
06:13     (2,3)        0kg     Drive 2km to the intersection of 4th avenue and 3rd street.
06:15     (4,3)        0kg     Park at depot d1.
```

A Concrete Delivery Plan

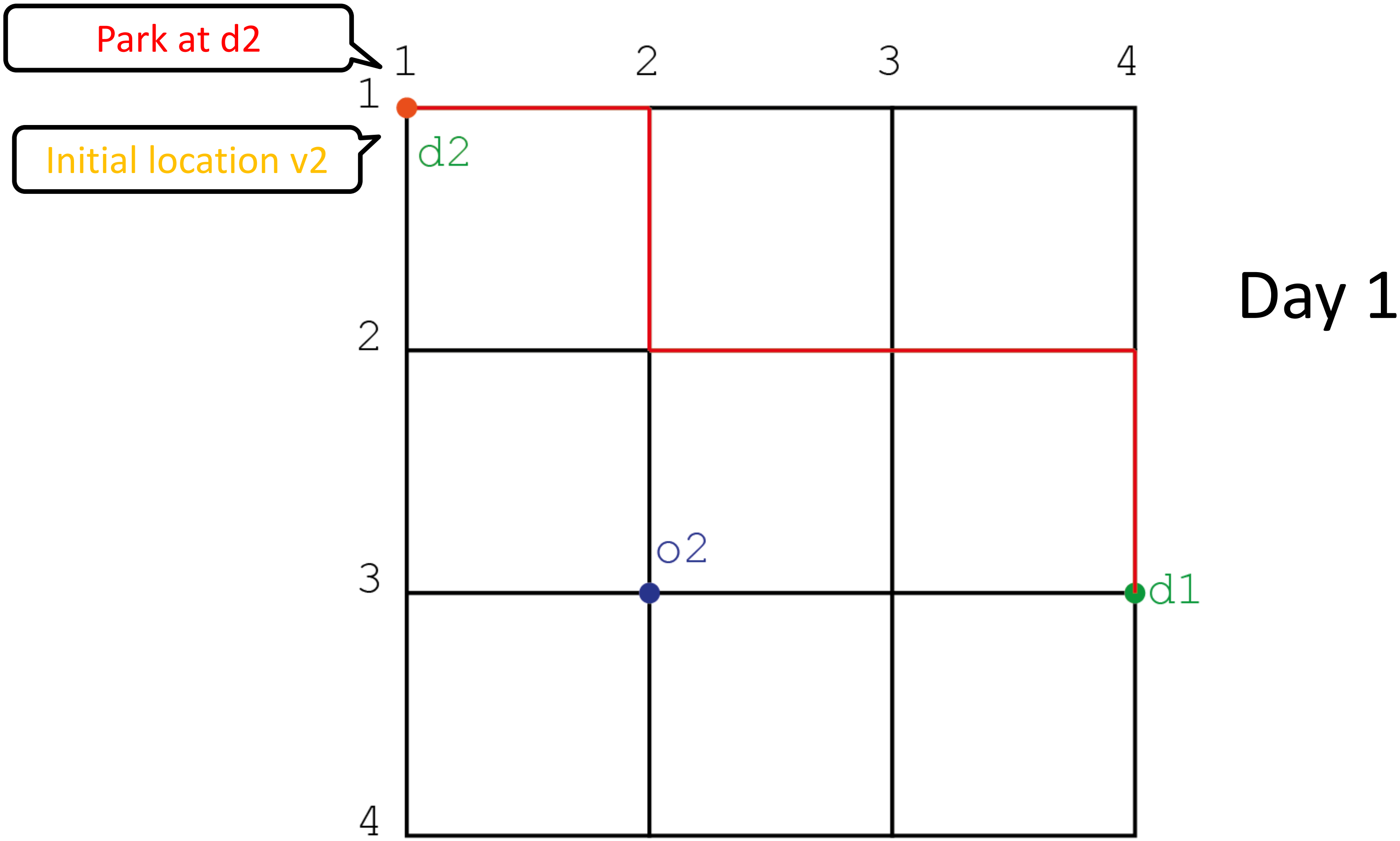


A Concrete Delivery Plan

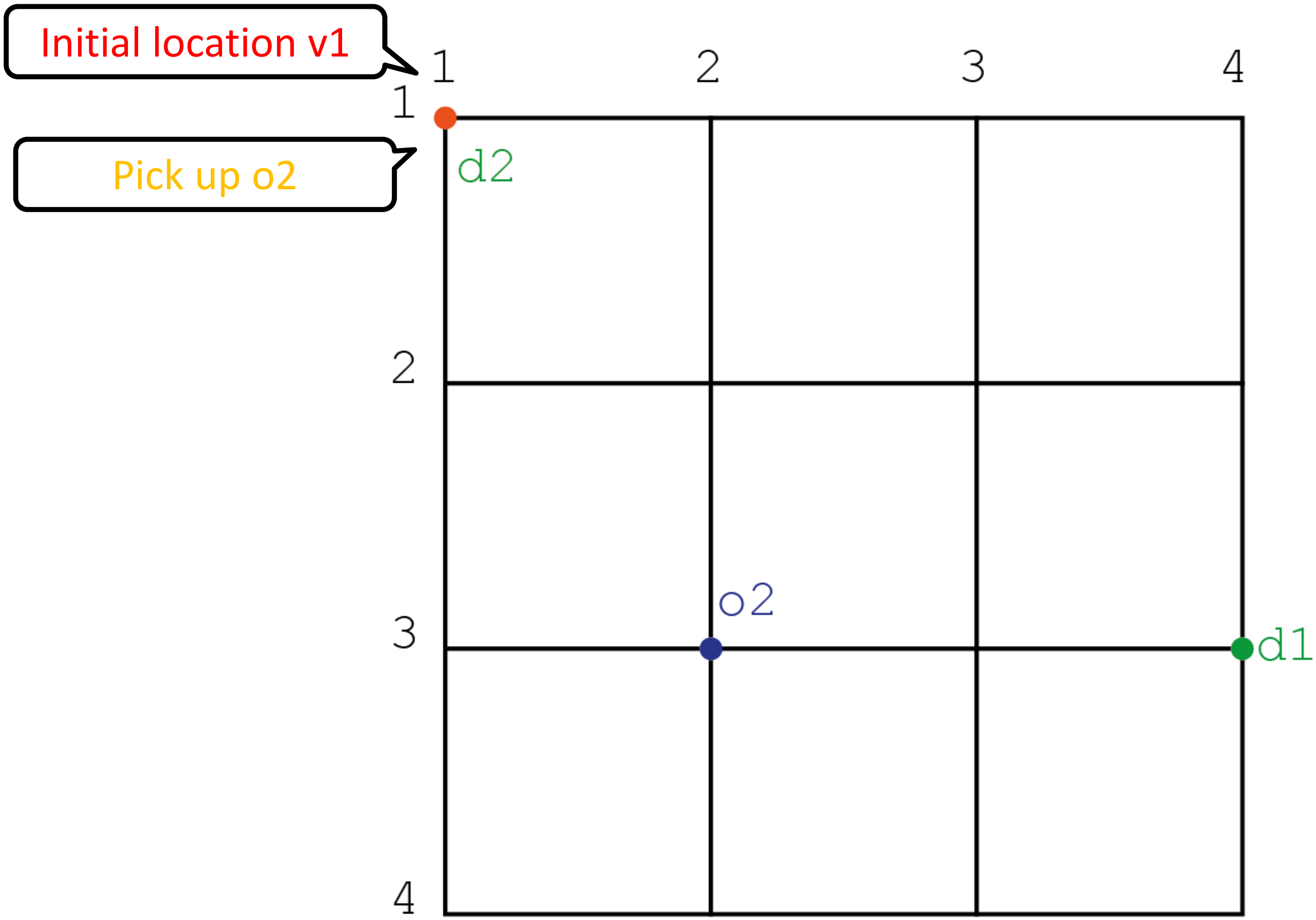


Day 1

A Concrete Delivery Plan

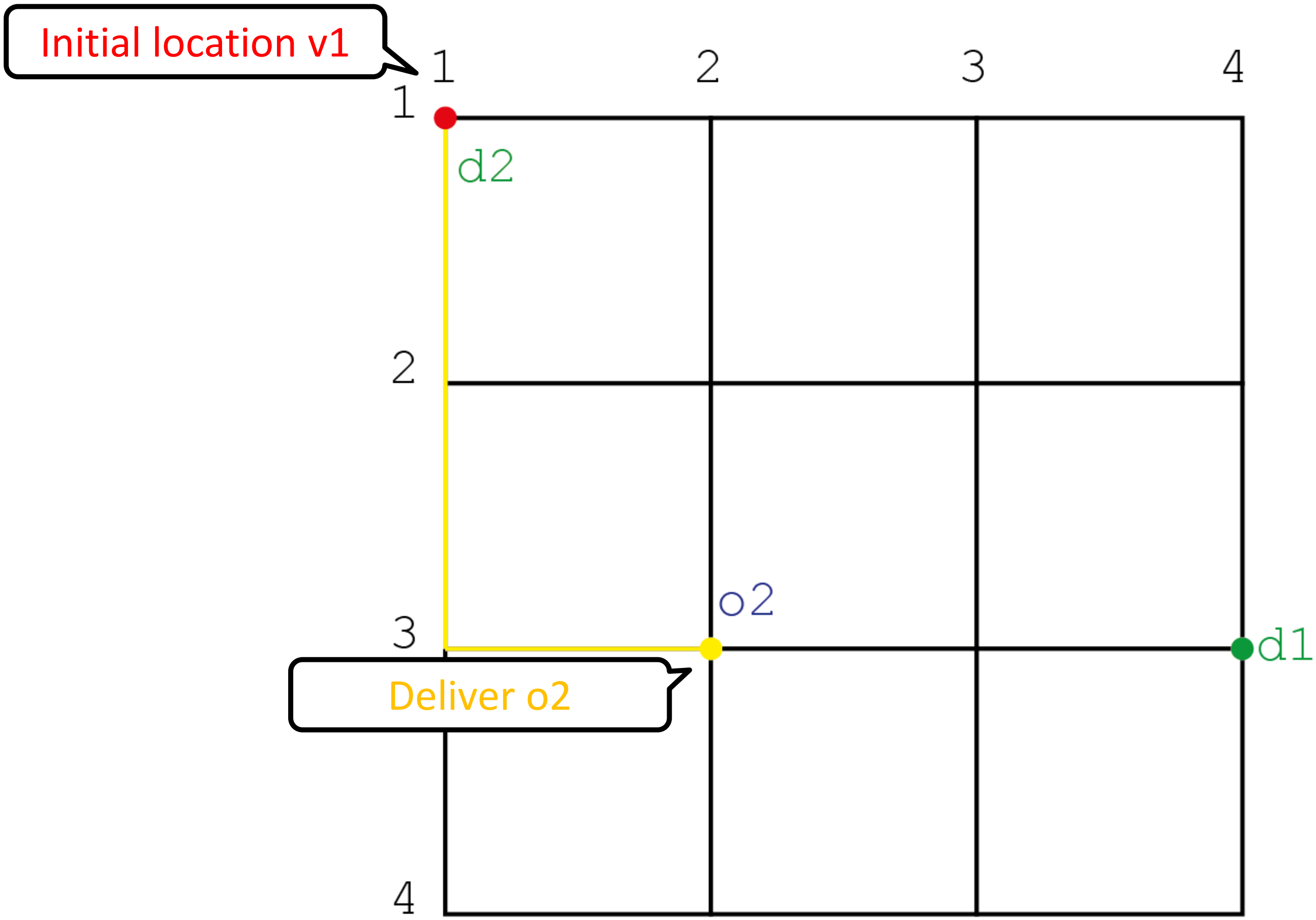


A Concrete Delivery Plan



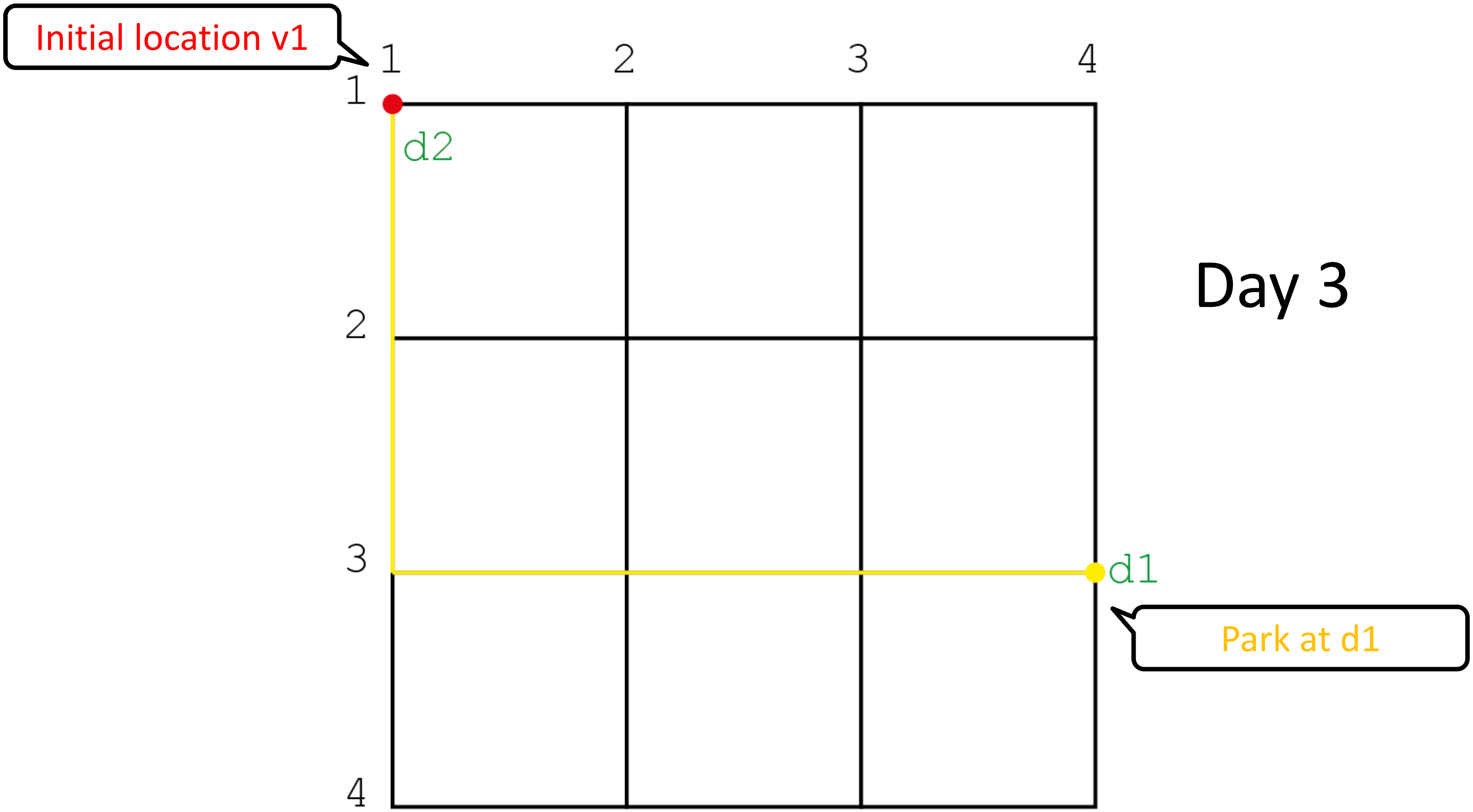
Day 3

A Concrete Delivery Plan



Day 3

A Concrete Delivery Plan



Day 3

A Concrete Delivery Plan

Or also:

```
plan(
    [
     schedule(v1,1,[o1,d2]),
     schedule(v2,1,[]),
     schedule(v1,3,[]),
     schedule(v2,3,[o2,d1])
    ]
)
```

Functionality

- Implement the following predicates:

  Auxiliary functionality (2pt)

  - `driving_duration(+VID,+From,+To,-Distance)`
  - `earning(+OID,+Day,-Value)`
  - `load(+OIDs,-Weight)`
  - `update_inventory(+Inventory,?OID,?NewInventory)`

Functionality

- Implement the following predicates:

  Core functionality (16pt)

  - `is_valid(+P)`
  - `profit(+P,-Profit)`
  - `find_optimal(-P)`
  - `find_heuristically(-P)`
  - `pretty_print(+P)`

Functionality

- Implement the following predicates:

  Extended Functionality (up to 3pt)

  - `is_valid(?S)`
  - `is_optimal(?S)`

  ...

- Required for a grade > 18/20.

Non-Functional Requirements

- Your program must work on the lab computers (E 1.4.)

- Comment your source code
  ➔ Prolog commenting conventions

- Work modular
  ➔ Prolog modules

- Find a careful balance between:
  – Declarative Style
  – Efficiency

Reporting Requirements

- Briefly explain your solution approach
- Clearly specify the strengths & weaknesses of your implementation:
  - What functionality? Works for all instances?
  - Non-functional requirements?
- Report results:
  - Optimal plan for small
  - Profit of the heuristic solution obtained for all
- Experimental results must be reproducable in under 3 min!

Deadline

- Deadline 1st term: 13th of January

- Deliverables:
  - Commented Source Code
  - Report
  - User Manual

- Project Defenses: 23, 24 and 25th of January