Plant Growth Scheduling

Salima Jaoua and Dana Kalaaji
Supervised by Jonas Racine

EPFL

June 3, 2021

Overview

Recall: GrowthBotHub

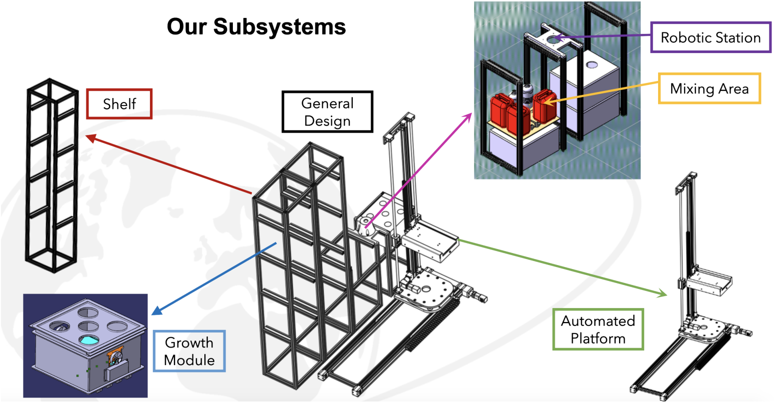
- Association at EPFL

Recall: GrowthBotHub

- Association at EPFL
- Fully automated farm

Recall: GrowthBotHub

- Association at EPFL
- Fully automated farm



Recall: The scheduling problem

- Each plant has needs that changes over time

Recall: The scheduling problem

- Each plant has needs that changes over time
- Each module contains certain nutrients

Recall: The scheduling problem

- Each plant has needs that changes over time
- Each module contains certain nutrients

$\longrightarrow$ A plant need to be moved between modules a certain points in time

Recall: The scheduling problem

- Each plant has needs that changes over time
- Each module contains certain nutrients

$\longrightarrow$ A plant need to be moved between modules a certain points in time

Max number of plants: What to plant? Where? When?

Recall: The scheduling problem

- Each plant has needs that changes over time
- Each module contains certain nutrients

$\longrightarrow$ A plant need to be moved between modules a certain points in time

Max number of plants: What to plant? Where? When?

- Our goal:

Recall: The scheduling problem

- Each plant has needs that changes over time
- Each module contains certain nutrients

$\longrightarrow$ A plant need to be moved between modules a certain points in time

Max number of plants: What to plant? Where? When?

- Our goal:
  - Improve and optimize last semester's algorithm

Recall: The scheduling problem

- Each plant has needs that changes over time
- Each module contains certain nutrients

$\longrightarrow$ A plant need to be moved between modules a certain points in time

Max number of plants: What to plant? Where? When?

- Our goal:
  - Improve and optimize last semester's algorithm
  - Re-scheduling

Recall : Modeling of the problem

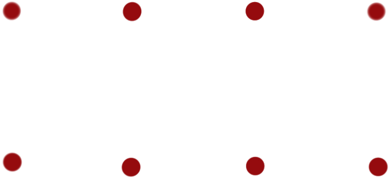
Multi-commodity flow

Recall : Modeling of the problem

Multi-commodity flow

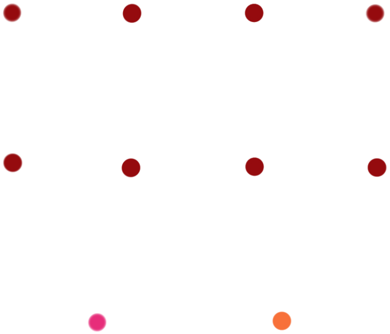
Recall : Modeling of the problem

Multi-commodity flow and Time-expanded graph



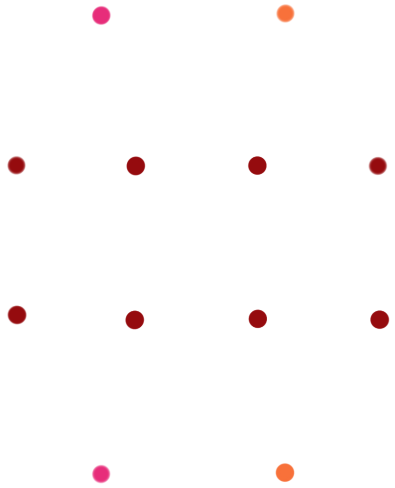
Recall : Modeling of the problem

Multi-commodity flow and Time-expanded graph



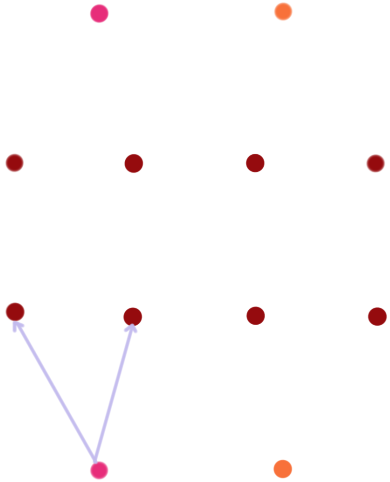
Recall : Modeling of the problem

Multi-commodity flow and Time-expanded graph



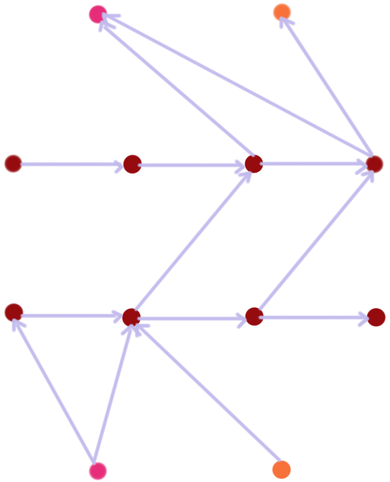
Recall : Modeling of the problem

Multi-commodity flow and Time-expanded graph



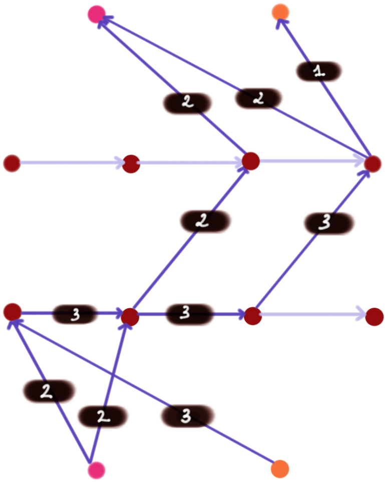
Recall : Modeling of the problem

Multi-commodity flow and Time-expanded graph



Recall : Modeling of the problem

Multi-commodity flow and Time-expanded graph



Recall : What we have seen last time

Last time: 1 node = 1 hole

Recall : What we have seen last time

Last time: 1 node = 1 hole

- Algorithm crashes after 12h on real-sized inputs

Recall : What we have seen last time

Last time: 1 node = 1 hole

- Algorithm crashes after 12h on real-sized inputs

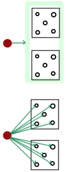
Our solution: reduced the size of the graph

Recall : What we have seen last time

Last time: 1 node = 1 hole

- Algorithm crashes after 12h on real-sized inputs

Our solution: reduced the size of the graph

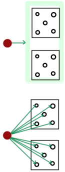


Recall : What we have seen last time

Last time: 1 node = 1 hole

- Algorithm crashes after 12h on real-sized inputs

Our solution: reduced the size of the graph



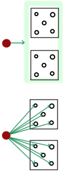
Tray edges	7200	720
Transfer edges	84600	846
Source edges	7220	722
Sink edges	7220	722
Total edges	106240	3010

Recall : What we have seen last time

Last time: 1 node = 1 hole

- Algorithm crashes after 12h on real-sized inputs

Our solution: reduced the size of the graph



Tray edges	7200	720
Transfer edges	84600	846
Source edges	7220	722
Sink edges	7220	722
Total edges	106240	3010

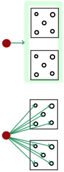
- Algorithm is able to run with real-size data

Recall : What we have seen last time

Last time: 1 node = 1 hole

- Algorithm crashes after 12h on real-sized inputs

Our solution: reduced the size of the graph



Tray edges	7200	720
Transfer edges	84600	846
Source edges	7220	722
Sink edges	7220	722
Total edges	106240	3010

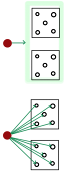
- Algorithm is able to run with real-size data
- Finds a 95% optimal solution in less than 2 minutes

Recall : What we have seen last time

Last time: 1 node = 1 hole

- Algorithm crashes after 12h on real-sized inputs

Our solution: reduced the size of the graph



Tray edges	7200	720
Transfer edges	84600	846
Source edges	7220	722
Sink edges	7220	722
Total edges	106240	3010

- Algorithm is able to run with real-size data
- Finds a 95% optimal solution in less than 2 minutes

What is left to do

Some of our goals changed :

What is left to do

Some of our goals changed :

- We want to obtain at least one plant everyday, with different species each day

$\rightarrow$ New diversity constraints

What is left to do

Some of our goals changed :

- We want to obtain at least one plant everyday, with different species each day

$\rightarrow$ New diversity constraints

We also need to take care of our original ones :

What is left to do

Some of our goals changed :

- We want to obtain at least one plant everyday, with different species each day

$\rightarrow$ New diversity constraints

We also need to take care of our original ones :

- Constraint about the plant's size

What is left to do

Some of our goals changed :

- We want to obtain at least one plant everyday, with different species each day

$\rightarrow$ New diversity constraints

We also need to take care of our original ones :

- Constraint about the plant's size

And a new goal has been added :

What is left to do

Some of our goals changed :

- We want to obtain at least one plant everyday, with different species each day

$\rightarrow$ New diversity constraints

We also need to take care of our original ones :

- Constraint about the plant's size

And a new goal has been added :

- Total number of growth modules ($\in [12, 16]$) unknown

What is left to do

Some of our goals changed :

- We want to obtain at least one plant everyday, with different species each day

$\rightarrow$ New diversity constraints

We also need to take care of our original ones :

- Constraint about the plant's size

And a new goal has been added :

- Total number of growth modules ($\in [12, 16]$) unknown

$\rightarrow$ Goal: find best combination of growth modules

Diversity

We keep the current diversity constraint

Diversity

We keep the current diversity constraint

- number of plants of each species is close to the average

Diversity

We keep the current diversity constraint

- number of plants of each species is close to the average

$$\text{maximize} \quad \sum_{p=1}^{P} \sum_{d=1}^{size(k_p)} \sum_{(i,j)\in\delta^+(s_p)} x_{ijk_{p_d}}$$

$$\text{subject to} \quad \sum_{(i,j)\in\delta^+(n)} x_{ijk} - \sum_{(i,j)\in\delta^-(n)} x_{ijk} = 0 \quad \forall n \in V - \{S, T\}$$

$$\forall (i,j) \in E \quad \sum_{p=1}^{N} \sum_{d=1}^{size(k_p)} x_{ijk_{p_d}} \leq 5$$

$$\forall p \in \{1....N\} \sum_{d=1}^{size(k_p)} \sum_{(i,j)\in\delta^+(s_p)} x_{ijk_{p_d}} \geq \frac{\sum_{p=1}^{N} \sum_{d=1}^{size(k_p)} \sum_{(i,j)\in\delta^+(s_p)} x_{ijk_{p_d}}}{N} - \alpha$$

$\longrightarrow$ ensures that we have a long term diversity

Harvest at least one plant each day

Divide the new constraint into two smaller ones

Harvest at least one plant each day

Divide the new constraint into two smaller ones

- Indicate if we harvest a plant on a given day

Harvest at least one plant each day

Divide the new constraint into two smaller ones

- Indicate if we harvest a plant on a given day
  Add a variable $(z_d)_{d \in \{1 \dots D\}}$ where D is the total number of days

Harvest at least one plant each day

Divide the new constraint into two smaller ones

- Indicate if we harvest a plant on a given day
  Add a variable $(z_d)_{d \in \{1 \dots D\}}$ where D is the total number of days

- Goal: $z_d = 0 \ \forall d \in \{1 \dots D\}$

Harvest at least one plant each day

Divide the new constraint into two smaller ones

- Indicate if we harvest a plant on a given day
  Add a variable $(z_d)_{d \in \{1 \dots D\}}$ where D is the total number of days

- Goal: $z_d = 0 \ \forall d \in \{1 \dots D\}$

  ⚠may be impossible

Harvest at least one plant each day

Divide the new constraint into two smaller ones

- Indicate if we harvest a plant on a given day
  Add a variable $(z_d)_{d\in\{1\ldots D\}}$ where D is the total number of days

- Goal: $z_d = 0 \ \forall d \in \{1 \ldots D\}$

  ⚠may be impossible

  $\longrightarrow$ penalize the objective function each day where we don't harvest a plant.

Harvest different species each day

- Need to diversify the outcomes each day $\rightarrow$ add a variable for every edge:

Harvest different species each day

- Need to diversify the outcomes each day $\rightarrow$ add a variable for every edge:

  Indicator of flow

Harvest different species each day

- Need to diversify the outcomes each day → add a variable for every edge:

  Indicator of flow

$$y_{ij} = 1 \text{ if } x_{ijk} > 0 \quad \forall (i,j) \in E \quad \forall k$$

Harvest different species each day

- Need to diversify the outcomes each day $\rightarrow$ add a variable for every edge:

  Indicator of flow

$$y_{ij} = 1 \text{ if } x_{ijk} > 0 \quad \forall (i,j) \in E \quad \forall k$$

- Limit number of days in a where we can harvest a specie

Harvest different species each day

- Need to diversify the outcomes each day $\rightarrow$ add a variable for every edge:

  Indicator of flow

$$y_{ij} = 1 \text{ if } x_{ijk} > 0 \quad \forall (i,j) \in E \quad \forall k$$

- Limit number of days in a where we can harvest a specie

  $\rightarrow$ Once we harvest a plant of type p, we cannot harvest another one for the next 3 days

Harvest different species each day

- Need to diversify the outcomes each day $\rightarrow$ add a variable for every edge:

  Indicator of flow

$$y_{ij} = 1 \text{ if } x_{ijk} > 0 \quad \forall (i,j) \in E \quad \forall k$$

- Limit number of days in a where we can harvest a specie

  $\rightarrow$ Once we harvest a plant of type p, we cannot harvest another one for the next 3 days

$$\forall d \quad \forall p = 1 \ldots N \quad \sum_{d}^{d+3} y_{t_{p_d}} \leq 1$$

Diversity-Analysis

At least a plant per day constraint:

Diversity-Analysis

At least a plant per day constraint:

- More even distribution over time
- Some loss of plant but negligible

Diversity-Analysis

At least a plant per day constraint:

- More even distribution over time
- Some loss of plant but negligible
- Does not affect the running time

Diversity-Analysis

At least a plant per day constraint:

- More even distribution over time
- Some loss of plant but negligible
- Does not affect the running time

Different species each day constraint:

Diversity-Analysis

At least a plant per day constraint:

- More even distribution over time
- Some loss of plant but negligible
- Does not affect the running time

Different species each day constraint:

- Really slows down the code: from 2min to +3h!
- If we limit the harvesting of a same species of plants to at least every...

Diversity-Analysis

At least a plant per day constraint:

- More even distribution over time
- Some loss of plant but negligible
- Does not affect the running time

Different species each day constraint:

- Really slows down the code: from 2min to +3h!
- If we limit the harvesting of a same species of plants to at least every...
  - ...2 days $\rightarrow$ we obtain $\frac{2}{3}$ of the total number of plants

Diversity-Analysis

At least a plant per day constraint:

- More even distribution over time
- Some loss of plant but negligible
- Does not affect the running time

Different species each day constraint:

- Really slows down the code: from 2min to +3h!
- If we limit the harvesting of a same species of plants to at least every...
  - ...2 days → we obtain $\frac{2}{3}$ of the total number of plants
  - ...3 days → we obtain 0 plants

Diversity-Analysis

At least a plant per day constraint:

- More even distribution over time
- Some loss of plant but negligible
- Does not affect the running time

Different species each day constraint:

- Really slows down the code: from 2min to +3h!
- If we limit the harvesting of a same species of plants to at least every...
  - ...2 days $\rightarrow$ we obtain $\frac{2}{3}$ of the total number of plants
  - ...3 days $\rightarrow$ we obtain 0 plants
- More cons than pros $\rightarrow$ dropped by the association

Different species each day: alternative solution

Each growth module can carry up to 5 plants

Different species each day: alternative solution

Each growth module can carry up to 5 plants

$\rightarrow$ Constraint: each module can contain at most 2 types of plant from the same kind

Different species each day: alternative solution

Each growth module can carry up to 5 plants

$\rightarrow$ Constraint: each module can contain at most 2 types of plant from the same kind

- only changes the capacity per commodity of an edge

Different species each day: alternative solution

Each growth module can carry up to 5 plants

$\rightarrow$ Constraint: each module can contain at most 2 types of plant from the same kind

- only changes the capacity per commodity of an edge

Analysis of this constraint

Different species each day: alternative solution

Each growth module can carry up to 5 plants

$\rightarrow$ Constraint: each module can contain at most 2 types of plant from the same kind

- only changes the capacity per commodity of an edge

Analysis of this constraint

- Does not reduce the total number of plants

Different species each day: alternative solution

Each growth module can carry up to 5 plants

$\rightarrow$ Constraint: each module can contain at most 2 types of plant from the same kind

- only changes the capacity per commodity of an edge

Analysis of this constraint

- Does not reduce the total number of plants
- Running time is the same

Different species each day: alternative solution

Each growth module can carry up to 5 plants

$\rightarrow$ Constraint: each module can contain at most 2 types of plant from the same kind

- only changes the capacity per commodity of an edge

Analysis of this constraint

- Does not reduce the total number of plants
- Running time is the same
- Constraint not respected to perfection

Different species each day: alternative solution

Each growth module can carry up to 5 plants

$\rightarrow$ Constraint: each module can contain at most 2 types of plant from the same kind

- only changes the capacity per commodity of an edge

Analysis of this constraint

- Does not reduce the total number of plants
- Running time is the same
- Constraint not respected to perfection

$\rightarrow$ Compromise between constraint

Different species each day: alternative solution

Each growth module can carry up to 5 plants

$\rightarrow$ Constraint: each module can contain at most 2 types of plant from the same kind

- only changes the capacity per commodity of an edge

Analysis of this constraint

- Does not reduce the total number of plants
- Running time is the same
- Constraint not respected to perfection

$\rightarrow$ Compromise between constraint perfectly respected or good output and running time

Plant's size: architecture's precision

Graph's architecture changed

Plant's size: architecture's precision

Graph's architecture changed $\longrightarrow$ need to update last semester's plants' size constraint

$$1 \text{ node} = 1 \text{ type of module}$$

Plant's size: architecture's precision

Graph's architecture changed $\longrightarrow$ need to update last semester's plants' size constraint

1 node $=$ 1 type of module

- Can't control the size plants each growth module

Plant's size: architecture's precision

Graph's architecture changed $\longrightarrow$ need to update last semester's plants' size constraint

1 node = 1 type of module

- Can't control the size plants each growth module
- We need to have control over the arrangement of a growth module

Changing the architecture

Change the algorithm: one node = one growth module

Changing the architecture

Change the algorithm: one node = one growth module

- Control over what happens inside a module

Changing the architecture

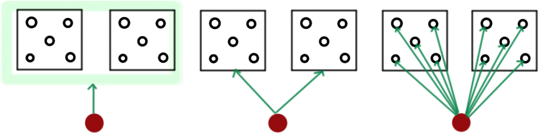
Change the algorithm: one node = one growth module

- Control over what happens inside a module
- Knowledge on what plants are inside what module

Changing the architecture

Change the algorithm: one node = one growth module

- Control over what happens inside a module
- Knowledge on what plants are inside what module

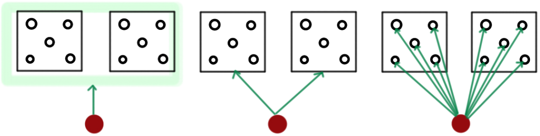


Analysis:

Changing the architecture

Change the algorithm: one node = one growth module

- Control over what happens inside a module
- Knowledge on what plants are inside what module



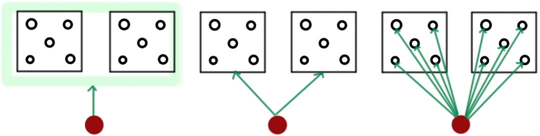
Analysis:

- Does it run on real-sized inputs ?

Changing the architecture

Change the algorithm: one node = one growth module

- Control over what happens inside a module
- Knowledge on what plants are inside what module



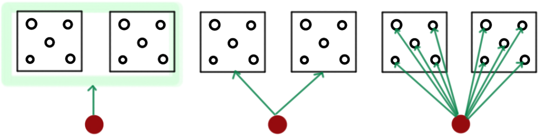
Analysis:

- Does it run on real-sized inputs ? Yes !!
- Running time: from less than 2 min to more than 3h30

Changing the architecture

Change the algorithm: one node = one growth module

- Control over what happens inside a module
- Knowledge on what plants are inside what module



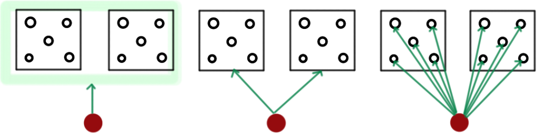
Analysis:

- Does it run on real-sized inputs ? Yes !!
- Running time: from less than 2 min to more than 3h30
- Still good for its purpose

Changing the architecture

Change the algorithm: one node = one growth module

- Control over what happens inside a module
- Knowledge on what plants are inside what module



Analysis:

- Does it run on real-sized inputs ? Yes !!
- Running time: from less than 2 min to more than 3h30
- Still good for its purpose → run the code once for the next 6 months

Plant's size

- Best case: we know the size of each specie over time

Plant's size

- Best case: we know the size of each specie over time
  $\rightarrow$ Precise algorithm

Plant's size

- Best case: we know the size of each specie over time
  $\rightarrow$ Precise algorithm
- Problem: no experiment done by GrowBotHub

Plant's size

- Best case: we know the size of each specie over time
  $\rightarrow$ Precise algorithm
- Problem: no experiment done by GrowBotHub
  $\rightarrow$ No data on the plant's size !

Plant's size

- Best case: we know the size of each specie over time
  $\rightarrow$ Precise algorithm

- Problem: no experiment done by GrowBotHub
  $\rightarrow$ No data on the plant's size !
  $\rightarrow$ Can't control the size of plants in a growth module if we have no data

Plant's size

- Best case: we know the size of each specie over time
  $\rightarrow$ Precise algorithm

- Problem: no experiment done by GrowBotHub
  $\rightarrow$ No data on the plant's size !
  $\rightarrow$ Can't control the size of plants in a growth module if we have no data

We still need to find some way to implement this constraint

Plant's size

Determine biggest plants as they will cause problem over time

Plant's size

Determine biggest plants as they will cause problem over time
How ?

Plant's size

Determine biggest plants as they will cause problem over time
How ?
$\longrightarrow$ 3 out of our 6 plants will not cause problem :

Plant's size

Determine biggest plants as they will cause problem over time

How ?

$\longrightarrow$ 3 out of our 6 plants will not cause problem :

- Radish and Endive : small.

Plant's size

Determine biggest plants as they will cause problem over time
How ?
$\longrightarrow$ 3 out of our 6 plants will not cause problem :

- Radish and Endive : small.
- Strawberries : not in the same growth module.

Plant's size

Determine biggest plants as they will cause problem over time

How ?

$\longrightarrow$ 3 out of our 6 plants will not cause problem :

- Radish and Endive : small.
- Strawberries : not in the same growth module.

$\Longrightarrow$ Lettuce, Fennel, Cabbage : ⚠Big plants

Plant's size

Determine biggest plants as they will cause problem over time

How ?

$\longrightarrow$ 3 out of our 6 plants will not cause problem :

- Radish and Endive : small.
- Strawberries : not in the same growth module.

$\Longrightarrow$ Lettuce, Fennel, Cabbage : ⚠Big plants

$\longrightarrow$ they will cause problem in the growth modules of type 2

Plant's size

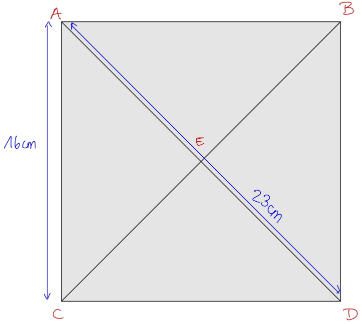


Figure: A growth module with its dimensions.

Plant's size

Add a constraint to avoid having more than two big plants in one growth module.

Plant's size

Add a constraint to avoid having more than two big plants in one growth module.

- Plants too big only once reached modules of type 2

Plant's size

Add a constraint to avoid having more than two big plants in one growth module.

- Plants too big only once reached modules of type 2
- Strawberries : ✔

Plant's size

Add a constraint to avoid having more than two big plants in one growth module.

- Plants too big only once reached modules of type 2
- Strawberries : ✔

Analysis.

Plant's size

Add a constraint to avoid having more than two big plants in one growth module.

- Plants too big only once reached modules of type 2
- Strawberries : ✔

Analysis.

- Number of plants remains the same
- Running time

New goal: Optimal number of modules

$\longrightarrow$ Total number of growth modules ($\in [12, 16]$) unknown
General idea:

New goal: Optimal number of modules

$\longrightarrow$ Total number of growth modules ($\in [12, 16]$) unknown
General idea:

- run algorithm and output best combination

New goal: Optimal number of modules

$\longrightarrow$ Total number of growth modules ($\in [12, 16]$) unknown
General idea:

- run algorithm and output best combination
- give this as inputs to the scheduling algorithm

New goal: Optimal number of modules

$\longrightarrow$ Total number of growth modules ($\in [12, 16]$) unknown
General idea:

- run algorithm and output best combination
- give this as inputs to the scheduling algorithm

New goal: Optimal number of modules

$\longrightarrow$ Total number of growth modules ($\in [12, 16]$) unknown
General idea:

- run algorithm and output best combination
- give this as inputs to the scheduling algorithm

Implementation:

- run scheduling algo on all combination of modules $\rightarrow$ output best one
- Brute force $\rightarrow$ each iteration needs to run fast

New goal: Optimal number of modules

$\longrightarrow$ Total number of growth modules ($\in [12, 16]$) unknown
General idea:

- run algorithm and output best combination
- give this as inputs to the scheduling algorithm

Implementation:

- run scheduling algo on all combination of modules $\rightarrow$ output best one
- Brute force $\rightarrow$ each iteration needs to run fast
- use old algo version (1 node = 1 type of module)

New goal: Optimal number of modules

$\longrightarrow$ Total number of growth modules ($\in [12, 16]$) unknown

General idea:

- run algorithm and output best combination
- give this as inputs to the scheduling algorithm

Implementation:

- run scheduling algo on all combination of modules $\rightarrow$ output best one
- Brute force $\rightarrow$ each iteration needs to run fast
- use old algo version (1 node = 1 type of module)
- each iteration takes a few minutes instead of a few hours!

Conclusion

	Last semester	Now
Size of the graph	(14412 , 212'490)	(2892, 38'280)
Number of modules	8	16
Running time	MemoryError	5h30
Constraint Diversity	✖	✔
Plant's size	✔	✔
Number of plants	MemoryError	206

Table: Comparison of the scheduling algorithm before and after our project

$$\text{maximize} \quad \sum_{p=1}^{P} \sum_{d=1}^{size(k_p)} \sum_{(i,j)\in\delta^+(s_p)} x_{ijk_{p_d}} - \sum_{d=1}^{D} z_p - v \sum_{(i,j)\in E} y_{ij}$$

$$\text{subject to} \quad \sum_{(i,j)\in\delta^+(n)} x_{ijk} - \sum_{(i,j)\in\delta^-(n)} x_{ijk} = 0 \quad \forall n \in V - \{S, T\}$$

$$\forall (i,j) \in E \quad \sum_{p=1}^{N} \sum_{d=1}^{size(k_p)} x_{ijk_{p_d}} \leq 5$$

$$\forall p \in \{1 \ldots . N\} \quad \sum_{d=1}^{size(k_p)} \sum_{(i,j)\in\delta^+(s_p)} x_{ijk_{p_d}} \geq \frac{\sum_{p=1}^{N} \sum_{d=1}^{size(k_p)} \sum_{(i,j)\in\delta^+(s_p)} x_{ijk_{p_d}}}{N} - \alpha \quad \alpha \geq 0$$

$$z_d \geq 0$$

$$z_d \geq 1 - \sum_{p=1}^{N} x_{t p_d}$$

$$\forall (i,j) \in E \quad \forall p \quad \forall d \quad x_{ijk_{p_d}} \leq 5 y_{ij}$$

$$\forall i, j \quad y_{ij} \in \{0, 1\}$$

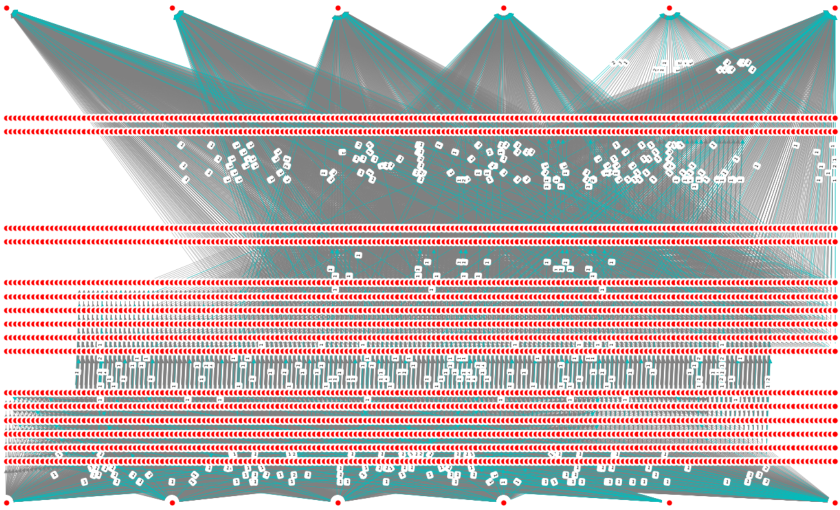
$$\forall d \quad \forall p = 1 \ldots . N \quad \sum_{d}^{d+3} y_{t p_d} \leq 1$$

$$x_{ijk_{p_d}} \leq 2$$

The inputs

- 4 types of growth module
  - 6 trays of module 1 ("seedling")
  - 6 trays of module 2 ("vegetative")
  - 2 trays of module 3 ("flowering")
  - 2 trays of module 4 ("development")
- Total number of days: 180 (6month)
- 6 types of plants
  - Lettuce: 30 days in module 1, 25 days in module 2
  - Endive: 20 days in module 1, 40 days in module 2
  - Cabbage: 20 days in module 1, 30 days in module 2
  - Fennel: 20 days in module 1, 30 days in module 2
  - Raddish: 15 days in module 1, 15 days in module 2
  - Strawberry: 49 days in module 1, 21 days in module 2, 28 days in module 3, 19 days in module 4

Graph of the optimal solution of the scheduling problem



Results

- Total number of lettuces:
- Total number of endives:
- Total number of cabbages:
- Total number of fennels:
- Total number of raddishes:
- Total number of strawberries:

$\longrightarrow$ Total number of plants: 168

Snippet of the instruction file

Snippet of the content of a module on a given day