CE810 - Game Design 2

Artificial Intelligence

Joseph Walton-Rivers & Piers Williams

Monday, 21 May 2018

University of Essex

- Production Rule Agents

2

- Production Rule Agents
- Monte-Carlo Tree Search

- Production Rule Agents
- Monte-Carlo Tree Search
- Genetic Algorithms

2

- Production Rule Agents
- Monte-Carlo Tree Search
- Genetic Algorithms
- Neural Networks

2

Production Rule Agents

```java
@FunctionalInterface
public interface Rule {
  boolean couldFire(int playerID, GameState state);
  Action execute(int playerID, GameState state);
}
```
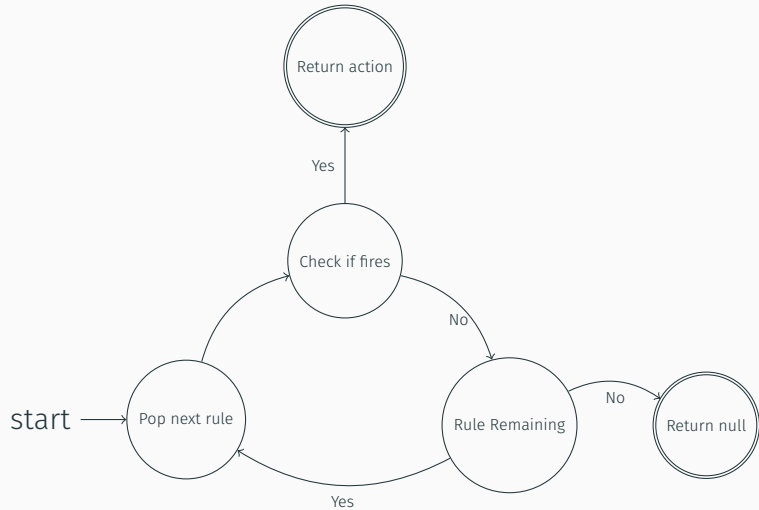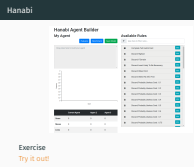
Diagram

4

Hanabi

Hanabi Agent Builder

My Agent

Available Rules

Exercise
Try it out!

CE810 GD2
└─Production Rule Agents

    └─Hanabi

Exercise
Try it out!

Do live demo - First Agent

1. Play If Certain
2. Tell Randomly
3. Discard Randomly

Second Agent

1. Play Probably Safe Card: 0.8
2. Tell Randomly
3. Discard Randomly

Place Discard at top of order and run. Order of rules important

```
public interface ProductionRule {

Map<UUID, Order> perform(
int playerId,
GameState state,
List<UUID> entities
);

}
```

CE810 GD2
└─Production Rule Agents

    └─Basic-TBS

2018-05-25

- Rules now apply to **0** or more Entities

7

- Rules now apply to **0** or more Entities
- **Remaining** entities considered by lower rules

- Rules now apply to **0** or more Entities
- **Remaining** entities considered by lower rules
- Orders generated by rules are **executed** by the rule

7

- Rules now apply to **0** or more Entities
- **Remaining** entities considered by lower rules
- Orders generated by rules are **executed** by the rule
  - **Simulating** their effects for other rules

7

- Rules now apply to **0** or more Entities
- **Remaining** entities considered by lower rules
- Orders generated by rules are **executed** by the rule
    - **Simulating** their effects for other rules
    - Allows **lower** rules to make **informed** decisions

7

- Rules now apply to **0** or more Entities
- **Remaining** entities considered by lower rules
- Orders generated by rules are **executed** by the rule
  - **Simulating** their effects for other rules
  - Allows **lower** rules to make **informed** decisions
  - Is why **non-determinism** is ill-advised

7

- Many rules are provided

- Many rules are provided
- Some complications:

8

- Many rules are provided
- Some complications:
  - Rules deal with **EntityTypes** and **Actions**

8

Built-Ins

- Many rules are provided
- Some complications:
  - Rules deal with **EntityTypes** and **Actions**
  - When I write the rules, you haven't **written** them yet

8

- Many rules are provided
- Some complications:
  - Rules deal with **EntityTypes** and **Actions**
  - When I write the rules, you haven't **written** them yet
  - Luckily they are **dynamically** built at **runtime**

8

- Many rules are provided
- Some complications:
  - Rules deal with **EntityTypes** and **Actions**
  - When I write the rules, you haven't **written** them yet
  - Luckily they are **dynamically** built at **runtime**
- I suppose we should list the included rules

8

Built-Ins

- Many rules are provided
- Some complications:
  - Rules deal with **EntityTypes** and **Actions**
  - When I write the rules, you haven't **written** them yet
  - Luckily they are **dynamically** built at **runtime**
- I suppose we should list the included rules
- And how to **use** them

8

Built-Ins

AttackMeleeRule

AttackRangedMostDamage-
dRule

Module

RandomRule

RunTowardsRule

UseActionOnEntity

AttackRangedClosestRule

EnsureEntityRule

NoopProductionRule

Filter

RunAwayRule

RunTowardsResource

UseActionOnResource

Some rules are similar - Will cover them together

- Two main types

CE810 GD2
└─Production Rule Agents

        └─Rule Types

2018-05-25

Not all rules can be done this way - What about EnsureEntity?

- Two main types
  - ProductionRule

Not all rules can be done this way - What about EnsureEntity?

- Two main types
  - ProductionRule
  - PerEntityRule *implements* ProductionRule

CE810 GD2
└─Production Rule Agents

  └─Rule Types

2018-05-25

Not all rules can be done this way - What about EnsureEntity?

- Two main types
  - ProductionRule
  - PerEntityRule *implements* ProductionRule
- Depends what you need to do in the rule

10

Not all rules can be done this way - What about EnsureEntity?

- Two main types
  - ProductionRule
  - PerEntityRule *implements* ProductionRule
- Depends what you need to do in the rule
- PerEntityRule is simpler

CE810 GD2
└─Production Rule Agents

    └─Rule Types

2018-05-25

Not all rules can be done this way - What about EnsureEntity?

- Two main types
  - ProductionRule
  - PerEntityRule *implements* ProductionRule
- Depends what you need to do in the rule
- PerEntityRule is simpler
  - Executes the orders **automatically**

10

CE810 GD2
2018-05-25 └─Production Rule Agents

└─Rule Types

Not all rules can be done this way - What about EnsureEntity?

- Two main types
  - ProductionRule
  - PerEntityRule *implements* ProductionRule
- Depends what you need to do in the rule
- PerEntityRule is simpler
  - Executes the orders **automatically**
  - What a **single** Entity does

10

2018-05-25

CE810 GD2
└─Production Rule Agents

└─Rule Types

Rule Types

- Two main types
  - ProductionRule
  - PerEntityRule implements ProductionRule
- Depends what you need to do in the rule
- PerEntityRule is simpler
  - Executes the orders **automatically**
  - What a **single** Entity does

Not all rules can be done this way - What about EnsureEntity?

- Two main types
  - ProductionRule
  - PerEntityRule *implements* ProductionRule
- Depends what you need to do in the rule
- PerEntityRule is simpler
  - Executes the orders **automatically**
  - What a **single** Entity does
  - Removes Entity from consideration for you

10

CE810 GD2
2018-05-25

└─Production Rule Agents

└─Rule Types

Not all rules can be done this way - What about EnsureEntity?

CE810 GD2
└─Production Rule Agents

        └─Attack Rules

2018-05-25

- PerEntityRule

- PerEntityRule

- AttackMeleeRule,
  AttackRangedClosestRule,AttackRangedMostDamagedRule

- PerEntityRule
- AttackMeleeRule, AttackRangedClosestRule,AttackRangedMostDamagedRule
- Melee rule will also **move** unit towards target

- PerEntityRule
- AttackMeleeRule, AttackRangedClosestRule,AttackRangedMostDamagedRule
- Melee rule will also **move** unit towards target
- Be careful:

11

- PerEntityRule

- AttackMeleeRule,
  AttackRangedClosestRule,AttackRangedMostDamagedRule

- Melee rule will also **move** unit towards target

- Be careful:

    - These rules do **not** check

- PerEntityRule
- AttackMeleeRule, AttackRangedClosestRule,AttackRangedMostDamagedRule
- Melee rule will also **move** unit towards target
- Be careful:
  - These rules do **not** check
  - They can issue an **invalid** order

EnsureEntity

Example

EnsureEntity[blue_town:blue_civilian:3]

- Producer, Product, Quantity

Example

EnsureEntity[blue_town:blue_civilian:3]

- Producer, Product, Quantity
- Simple:

12

Example

EnsureEntity[blue_town:blue_civilian:3]

- Producer, Product, Quantity
- Simple:
    - Counts how many Product we have

Example

EnsureEntity[blue_town:blue_civilian:3]

- Producer, Product, Quantity
- Simple:
    - Counts how many Product we have
    - Finds Producer for each missing Product

12

EnsureEntity

Example

EnsureEntity[blue_town:blue_civilian:3]

- Producer, Product, Quantity
- Simple:
    - Counts how many Product we have
    - Finds Producer for each missing Product
    - Issues order for Producer to build Product

EnsureEntity

Example

EnsureEntity[blue_town:blue_civilian:3]

- Producer, Product, Quantity
- Simple:
  - Counts how many Product we have
  - Finds Producer for each missing Product
  - Issues order for Producer to build Product
- Supports abstract types:

EnsureEntity

Example

EnsureEntity[blue_town:blue_civilian:3]

- Producer, Product, Quantity
- Simple:
  - Counts how many Product we have
  - Finds Producer for each missing Product
  - Issues order for Producer to build Product
- Supports abstract types:
- EnsureEntity[abstract_civilian:abstract_town:1]

Example

EnsureEntity[blue_town:blue_civilian:3]

- Producer, Product, Quantity
- Simple:
    - Counts how many Product we have
    - Finds Producer for each missing Product
    - Issues order for Producer to build Product
- Supports abstract types:
- EnsureEntity[abstract_civilian:abstract_town:1]
- Uses BuildOrder.

2018-05-25

CE810 GD2
└─Production Rule Agents

└─EnsureEntity

Run Rules

Example

RunTowards[0.0]
RunAway[0.5]
RunTowardsResource[gold]

- Causes Entity to travel

13

Example

RunTowards[0.0]

RunAway[0.5]

RunTowardsResource[gold]

- Causes Entity to travel

- **To** or **from** something

13

Example

UseActionOnResource[BuildOnResource[gold_mine:gold]:gold]

- Causes Entity to use an Action on either:

Example

UseActionOnResource[BuildOnResource[gold_mine:gold]:gold]

- Causes Entity to use an Action on either:
  - Another Entity

Example

UseActionOnResource[BuildOnResource[gold_mine:gold]:gold]

- Causes Entity to use an Action on either:
    - Another Entity
    - A resource

Example

UseActionOnResource[BuildOnResource[gold_mine:gold]:gold]

- Causes Entity to use an Action on either:
  - Another Entity
  - A resource
- Great for building mines

Example

UseActionOnResource[BuildOnResource[gold_mine:gold]:gold]

- Causes Entity to use an Action on either:
  - Another Entity
  - A resource
- Great for building mines
- Great for supporting custom actions.

14

Example

UseActionOnResource[BuildOnResource[gold_mine:gold]:gold]

- Causes Entity to use an Action on either:
  - Another Entity
  - A resource
- Great for building mines
- Great for supporting custom actions.
- Attack actions could have been written with this

Example

UseActionOnResource[BuildOnResource[gold_mine:gold]:gold]

- Causes Entity to use an Action on either:
  - Another Entity
  - A resource
- Great for building mines
- Great for supporting custom actions.
- Attack actions could have been written with this
- But they predate this

14

Example

Filter[AttackMelee:abstract_knight]

- PerEntityRule

Example

Filter[AttackMelee:abstract_knight]

- PerEntityRule
- PerEntityRule usually consults **every** Entity

Example

Filter[AttackMelee:abstract_knight]

- PerEntityRule
- PerEntityRule usually consults **every** Entity
- This doesn't **usually** make sense

Filter Rules

Example

Filter[AttackMelee:abstract_knight]

- PerEntityRule
- PerEntityRule usually consults **every** Entity
- This doesn't **usually** make sense
- Rather than write **conditions** on other rules

Example

Filter[AttackMelee:abstract_knight]

- PerEntityRule

- PerEntityRule usually consults **every** Entity

- This doesn't **usually** make sense

- Rather than write **conditions** on other rules

- **Wrap** rules with this condition

15

Example

Filter[AttackMelee:abstract_knight]

- PerEntityRule
- PerEntityRule usually consults **every** Entity
- This doesn't **usually** make sense
- Rather than write **conditions** on other rules
- **Wrap** rules with this condition
- Takes 1 or more type as an array

- These are largely designed for civ style games

- These are largely designed for civ style games
- Great news if you have made one

16

- These are largely designed for civ style games
- Great news if you have made one
- Less than great news if your game is radically different

Custom Rules

- These are largely designed for civ style games
- Great news if you have made one
- Less than great news if your game is radically different
- You'll need to **provide** new rules

16

- These are largely designed for civ style games
- Great news if you have made one
- Less than great news if your game is radically different
- You'll need to **provide** new rules
  - Same way you did for Actions

- These are largely designed for civ style games
- Great news if you have made one
- Less than great news if your game is radically different
- You'll need to **provide** new rules
  - Same way you did for Actions
  - Dynamically scanned at runtime

"RangedRush":"PRA[EnsureBase,EnsureWorker,En-
sureArchery,BuildGoldMine,BuildWoodMine,EnsureEntity[ab-
stract_civilian:farm:3],TravelToGold,TravelToWood,En-
sureArcher,ArcherAttack,ArcherChase,WorkerEvade]",

17

- That was a lot in one line

- That was a lot in one line
- **Sorry**, but that is how it is

18

- That was a lot in one line
- **Sorry**, but that is how it is
- These do tend to be quite in-flexible.

18

- That was a lot in one line
- **Sorry**, but that is how it is
- These do tend to be quite in-flexible.
- More on that later

18

Genetic Algorithms

Genetic Algorithms

- Powerful algorithms for variety of uses

19

Genetic Algorithms

- Powerful algorithms for variety of uses
- Can even **play** games

19

Genetic Algorithms

- Powerful algorithms for variety of uses
- Can even **play** games
- Apologies to those that have encountered GA's

19

Genetic Algorithms

- Powerful algorithms for variety of uses
- Can even **play** games
- Apologies to those that have encountered GA's
- Even more apologies to those that have encountered RHEA's

19

20

Genetic Algorithms

- You **all** have some experience using these

21

- You **all** have some experience using these
- Asteroids assignment had one built in

21

- You **all** have some experience using these
- Asteroids assignment had one built in
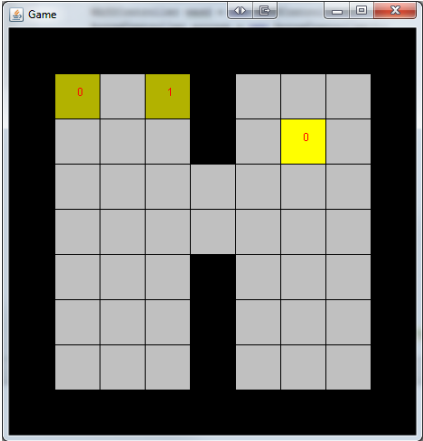- But how to play a game with a GA?
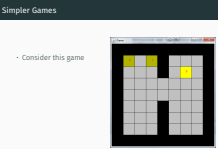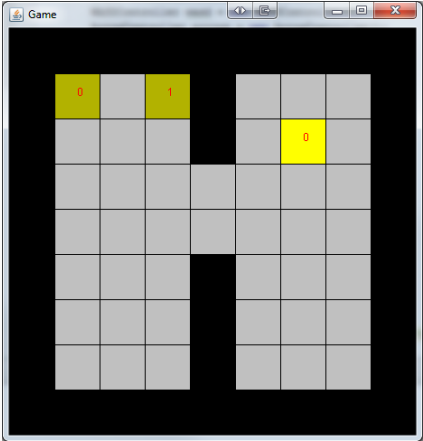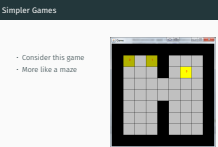
21

Simpler Games

- Consider this game

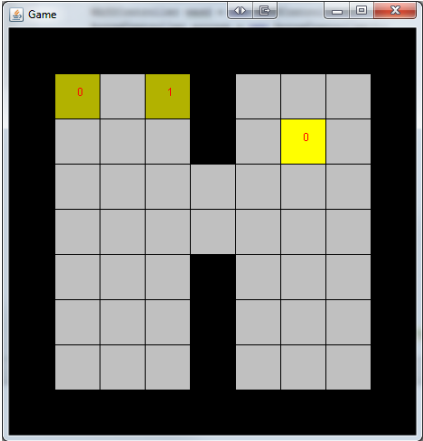

CE810 GD2
└─Genetic Algorithms
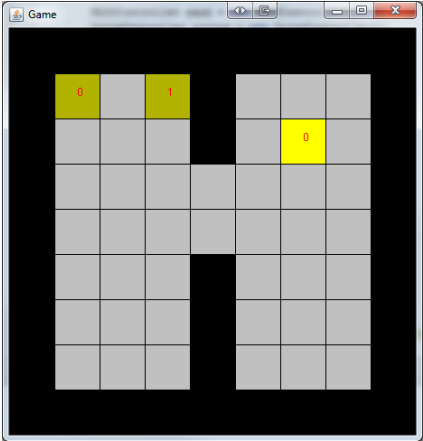
2018-05-25

└─Simpler Games

Simpler Games

- Consider this game

Simpler Games

- Consider this game
- More like a maze

Simpler Games

- Consider this game
- More like a maze
- Get each Agent to the goal

22

Simpler Games

- Consider this game
- More like a maze
- Get each Agent to the goal
- 5 possible moves

CE810 GD2
└─Genetic Algorithms

└─Simpler Games

2018-05-25

- Consider this game
- More like a maze
- Get each Agent to the goal
- 5 possible moves
- Controller returns a single move per turn

Simpler Games

- Consider this game
- More like a maze
- Get each Agent to the goal
- 5 possible moves
- Controller returns a single move per turn

2018-05-25

CE810 GD2
└─Genetic Algorithms

└─Simpler Games

- Given a time budget

23
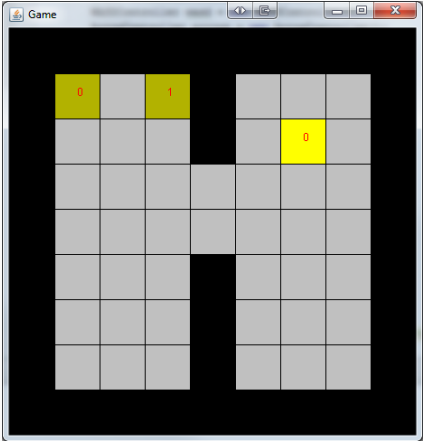
CE810 GD2
2018-05-25
└─Genetic Algorithms

    └─GA for Simple Game

- Given a time budget
- Spend it **evolving** "plans"

- Given a time budget
- Spend it **evolving** "plans"
- Plan is a **sequence** of possible actions

· Given a time budget

· Spend it **evolving** "plans"

· Plan is a **sequence** of possible actions

· Simulate the plan and evaluate resultant state for score

23

- Given a time budget
- Spend it **evolving** "plans"
- Plan is a **sequence** of possible actions
- Simulate the plan and evaluate resultant state for score
- Sound like a possible GA?

23

- Given a time budget

- Spend it **evolving** "plans"

- Plan is a **sequence** of possible actions

- Simulate the plan and evaluate resultant state for score

- Sound like a possible GA?

- This is called a Rolling Horizon Evolutionary Algorithm

· Length of the sequence increases parameters for GA

- Length of the sequence increases parameters for GA
- RHEA is quite **jerky** in games

Boosting the horizon

- Length of the sequence increases parameters for GA
- RHEA is quite **jerky** in games
- Macro Actions can sometimes solve this

24

- Length of the sequence increases parameters for GA
- RHEA is quite **jerky** in games
- Macro Actions can sometimes solve this
- Locking the agent to consider each move *N* times

24

- Length of the sequence increases parameters for GA
- RHEA is quite **jerky** in games
- Macro Actions can sometimes solve this
- Locking the agent to consider each move N times
- Means you can think for N turns

24

- Length of the sequence increases parameters for GA
- RHEA is quite **jerky** in games
- Macro Actions can sometimes solve this
- Locking the agent to consider each move N times
- Means you can think for N turns
- Works great in real-time engines like PTSP

24

Boosting the horizon

CE810 GD2
2018-05-25 └─Genetic Algorithms

└─Boosting the horizon

- Length of the sequence increases parameters for GA
- RHEA is quite **jerky** in games
- Macro Actions can sometimes solve this
- Locking the agent to consider each move *N* times
- Means you can think for *N* turns
- Works great in real-time engines like PTSP
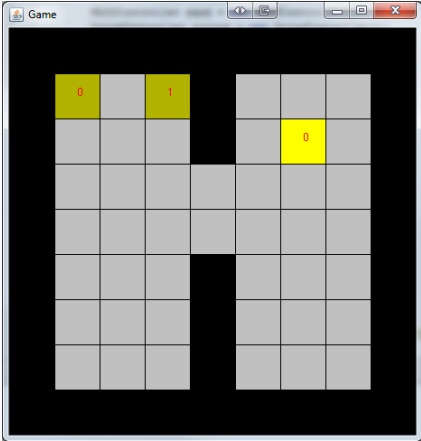- Getting *N* wrong means poor performance

- Length of the sequence increases parameters for GA

- RHEA is quite **jerky** in games

- Macro Actions can sometimes solve this

- Locking the agent to consider each move N times

- Means you can think for N turns

- Works great in real-time engines like PTSP

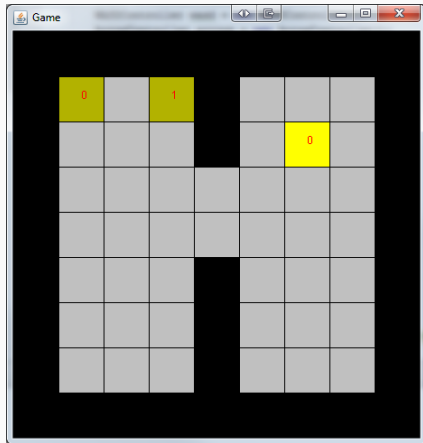- Getting N wrong means poor performance

- What if *N* is 2?
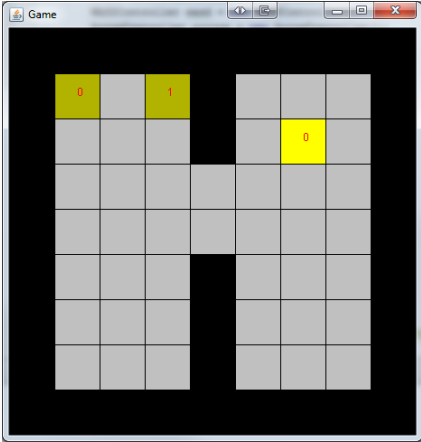
25

- What if *N* is 2?
- Can we reach the Goal?

- What if *N* is 2?
- Can we reach the Goal?
- MacroActionGA is poor at discrete boards
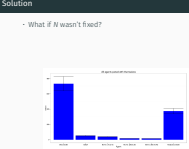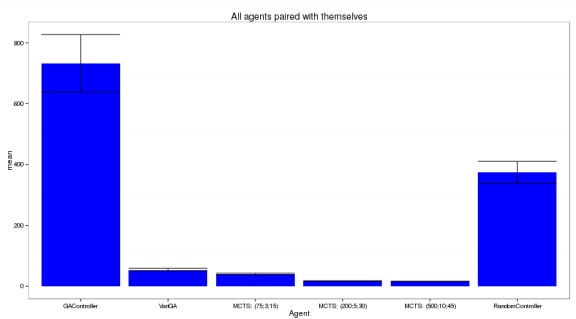
25

- What if N wasn't fixed?

All agents paired with themselves

26

- What if *N* wasn't fixed?
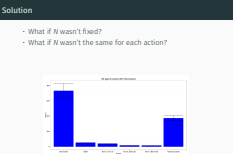- What if *N* wasn't the same for each action?

All agents paired with themselves
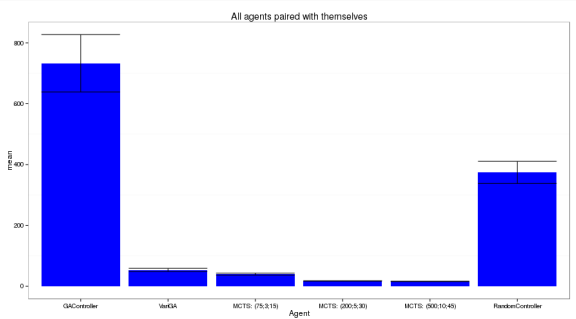
Solution

- What if *N* wasn't fixed?
- What if *N* wasn't the same for each action?
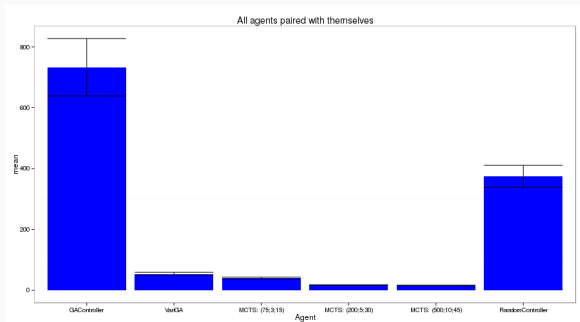
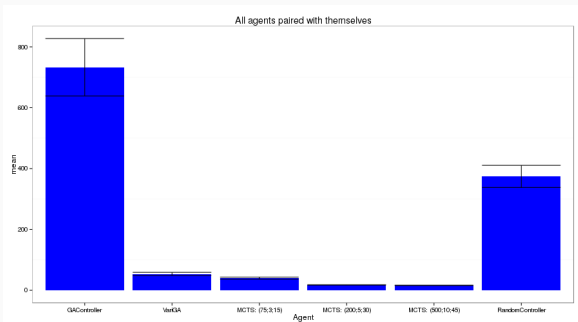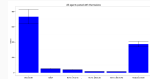CE810 GD2
└─Genetic Algorithms

2018-05-25

└─Solution

- What if *N* wasn't fixed?
- What if *N* wasn't the same for each action?
- Learn the values for *N* as well as the actions

- What if *N* wasn't fixed?
- What if *N* wasn't the same for each action?
- Learn the values for *N* as well as the actions
- Include them in the GA

All agents paired with themselves

- This is great for simple games

27

More complex games

- This is great for simple games
- More complex games are too large

27

- This is great for simple games
- More complex games are too large
- Action space in our game is huge!

- This is great for simple games
- More complex games are too large
- Action space in our game is huge!
- Macro actions don't make sense either!

27

More complex games

- This is great for simple games
- More complex games are too large
- Action space in our game is huge!
- Macro actions don't make sense either!
- Need something **higher** level

CE810 GD2
└─Genetic Algorithms

2018-05-25

└─Strategy Search

- Instead of choosing actions

28

- Instead of choosing actions
- Choose between **strategies**

- Instead of choosing actions
- Choose between **strategies**
- But where will we get those?

28

```
Map<EntityType, List<ProductionRuleAgent>> rules;
```

- Choose between PRA's for each EntityType

```
Map<EntityType, List<ProductionRuleAgent>> rules;
```

· Choose between PRA's for each **EntityType**
· Why not per Entity?

29

"MedievalGA":"VLMAGA[1000:10:EandM:abstract_civilian,abstract_town,abstract_knightery,abstract_archery,abstract_knight,abstract_archer:noopRule/ResourceBuilder/BuildBase/BuildMilitary,noopRule/BuildWorker,noopRule/BuildKnights,noopRule/BuildArchers,KnightAttack,ArcherAttack:noop:RangedRush]"

- Will do automatic single actions first

"MedievalGA":"VLMAGA[1000:10:EandM:abstract_civilian,abstract_town,abstract_knightery,abstract_archery,abstract_knight,abstract_archer:noopRule/ResourceBuilder/BuildBase/BuildMilitary,noopRule/BuildWorker,noopRule/BuildKnights,noopRule/BuildArchers,KnightAttack,ArcherAttack:noop:RangedRush]"

- Will do automatic single actions first
- Then will follow its learned policy

30

"MedievalGA":"VLMAGA[1000:10:EandM:abstract_civilian,abstract_town,abstract_knightery,abstract_archery,abstract_knight,abstract_archer:noopRule/ResourceBuilder/BuildBase/BuildMilitary,noopRule/BuildWorker,noopRule/BuildKnights,noopRule/BuildArchers,KnightAttack,ArcherAttack:noop:RangedRush]"

- Will do automatic single actions first
- Then will follow its learned policy
- Then will use the fallback agent for the rest

30

Controller	ops/min	Error
random	512.840	13.846
noop	509.314	5.730
RangedRush	273.362	2.345
MixedRush	284.875	7.453
MedievalGA	3.538	2.286

- An "op" is building a full game and playing it to the end
- MedievalGA is the VLMAGA
- I wish that were an error for MedievalGA