

Problemområden i Starcraft II

...och AI-tekniker för att tackla problemen

Mattias Tiger
Fredrik Pröntare
Jonas Kvarnström



Introduktion och motivering

Ni ska inför er individuella uppgift **definiera ett problem** och **välja ut en eller flera AI-tekniker** som kan tänkas användas för att lösa detta problem.

- Det finns väldigt många *domänspecifika problem* i StarCraft II att lösa. Man behöver inte lösa *alla*, utan välja sådana som kan bidra till en bra agent.
- Det finns väldigt många AI-Tekniker. De kan ersätta varandra, komplettera varandra eller integreras för att lösa större problem.

För att stödja er ska vi nu ge er översiktliga **exempel på några domänspecifika problem** inom StarCraft II, och översiktliga **exempel på några AI-Tekniker**.

- Det är upp till er att **leta** ytterligare information och att **välja** vad ni vill göra.

Vad vi inte gör idag

Idag ska vi inte:

- Diskutera och definiera alla problem i detalj:
Ni ska själva utforska och skapa egna varianter av dessa generella problem
(vi ser stor variation mellan olika grupper – bra!)
- Diskutera och specificera alla lösningstekniker i detalj:
Vi ger tips, ni utforskar lösningar (enligt kursmålen),
och en förklaring på *mellan-nivå* förvirrar mer än den förklarar
 - När ni har läst på kan ni ta upp era frågor

StarCraft II - Domänspecifika problemområden

- **Placera byggnader / trupper**
 - (var får den/de plats? Vad är bästa platsen i närheten av X?)
- **Bestämma byggnadsordning**
 - (vilka byggnader/uppgraderingar krävs för enhet/byggnad X?)
- **Utforska kartan effektivt**
 - (upprepade avsökningar av intressanta områdena -- vem går vart, när?)
- **Välja strategi**
 - (bygga upp eco/industry, expansion, defensiva lägen, offensiva lägen.
Vad är motståndarens strategi?)
- **Välja taktik**
 - (truppförflyttningar, truppsammansättningar, offensiver, omställningar.
Vad kommer motståndaren att göra?)

StarCraft II - Domänspecifika problemområden

- **Hitta en säker väg**
 - (undvika fienden, ta minimalt med skada)
- **Skapa truppformationer**
 - (undvika Area of Effect där fienden når dig, nå fienden med egen eldgivning)
- **Skapa attackmönster**
 - (*Micro / micromanagement*: fokuserad eldgivning, *kiting*, utnyttja styrkor/svagheter)
- ...

Problemområden

Del 1



Problem: Byggnadsordning

Teknikexempel: Sökning (olika algoritmer), planering (för mer komplexa modeller)

Givet **prioriteringar, mål** och **nuvarande kunskap**:

Vad ska byggas och
i vilken ordning?

Exempel: Vilka byggnader
behövs för att kunna bygga
en Siege Tank?

Varning: Tech Tree är trivialt;
en lösning behöver t.ex. titta på
hur många enheter som behövs,
tiden det tar att bygga osv.



Problem: Utforska kartan effektivt

Teknikexempel: Sökning, Cost Field, Hidden Markov Models

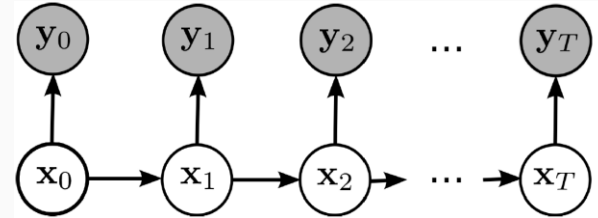
Givet vad ni observerar på kartan just nu och vad ni har observerat tidigare

- $P(\text{motståndar_bas} \mid \text{bas_plats}) = ?$
- $P(\text{motståndarenheter} \mid \text{position}) = ?$
- $P(\text{motståndarstridskraft} \mid \text{position}) = ?$
- ...

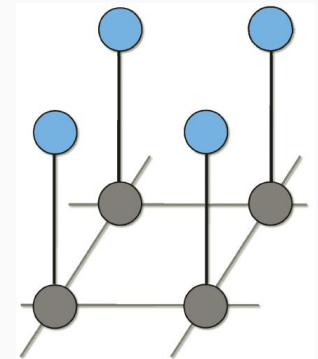
“när utforskades plats x sist”

vs

“hur troligt är det att det finns något nytt att observera på plats x ”.



Exempel på temporal HMM. Från <http://iacs-courses.seas.harvard.edu/courses/am207/blog/lecture-18.html>



Exempel på spatial HMM (på en grid).

Problem: Välja strategi, taktik

Teknikexempel: Supervised learning med Bayesian Networks
(supervised \approx generalisering från exempel på korrekta svar)

Exempel:

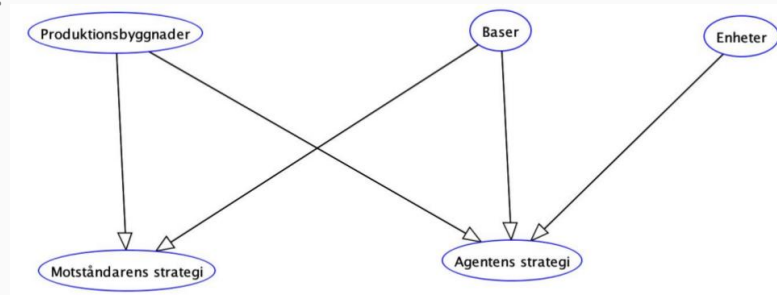
- strategi \in {Expansiv, Defensiv, Offensiv}
- tillstånd \in {#baser, #byggnader av typ x, #försvarande enheter av typ y, ...}
- observationer: iakttagelser av motståndarens tillstånd (vad är intressant?)

$P(\text{egen_strategi} \mid \text{observationer, eget_tillstånd}) = ?$

$P(\text{motståndare_strategi} \mid \text{observationer}) = ?$

Välj/anta strategi med högst sannolikhet.

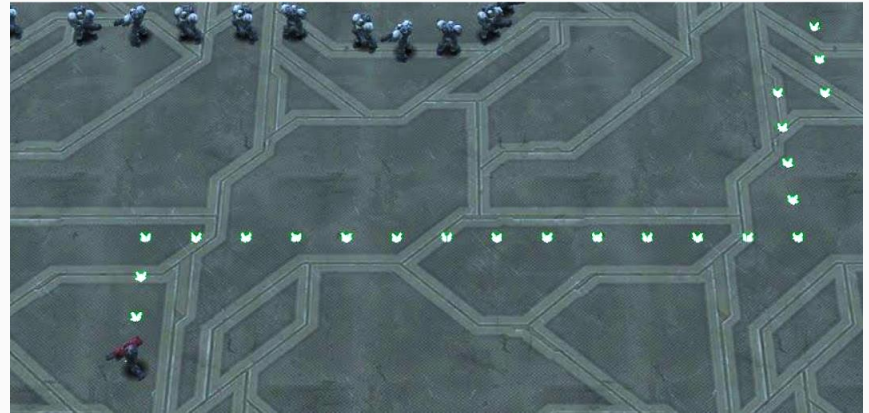
Kan använda replays för att träna (vissa risker)



Problem: Hitta en säker väg

Teknikexempel: A*/Dijkstra-sökning, potential fields, flow fields

- A* med bågkostnad proportionell mot risken att bli skadad i ett steg (t. ex. att vara inom skotthåll)
- Potential fields som ger en kostnad / utility att vara i en viss punkt (t. ex. hur farligt det är)
- Flow fields som attraherar till ett mål och repellerar från fienden (girig sökning)



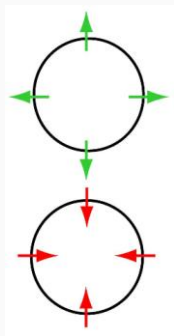
Problem: Skapa truppformationer, attackmönster

Teknikexempel: Flocking/Swarming genom Flow fields / Potential fields

Olika beteenden är bra för olika taktiker

- Utdela maximal skada/tid
- Undvika skada (range, melee, AoE)
- Marschering (undvik att bli upptäckt)

Attraktorer och repellerare



Man kan även använda mer generella vektorfält...

Figure A. "Primary Formations"

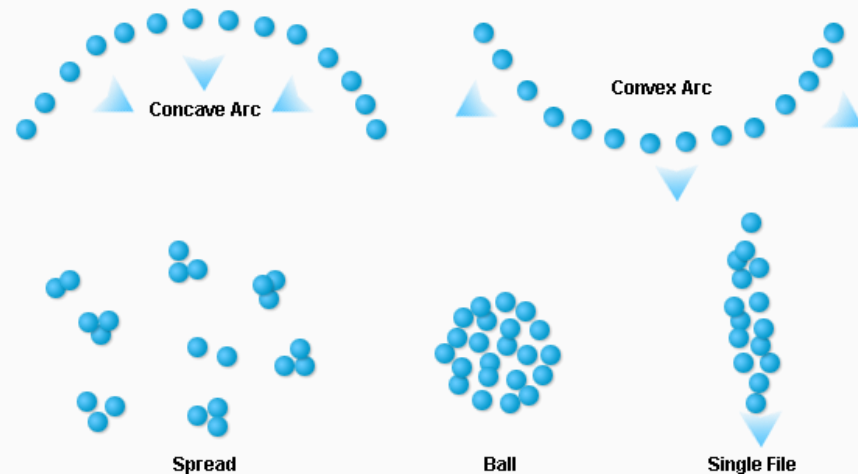


Figure From: <http://www.teamliquid.net/forum/sc2-strategy/187892-positioning-formations-and-tactics>

Problem: Attackmönster (1/2)

Teknikexempel: Reinforcement Learning med Q-learning

(För varje beslut får man belöning/"straff"; vilken strategi blir bäst i längden?)

Enheterna är oberoende av varandra.

- Ursprungligt beteende



- Undviker att dö
- Fokuserad eldgivning (lägst HP)



Problem: Attackmönster (2/2)

Teknik: Reinforcement Learning med Q-learning

1) Välj ut intressanta features

- Exempel: egen hälsa, motståndarhälsa, relativt avstånd, stridskraft...

2) Diskretisera features

- Exempel: egen hälsa $\in \{0, 10, 20, 30, \dots, 90, 100\}$ (procent)

3) Välj utility att maximera

- Exempel: $100 * \#egna_trupper + \#egen_hälsa - 100 * \#motståndartrupper - \#motståndarhälsa$

4) Konstruera representativa scenarier och träna fram Q-tabellen: $\mathbf{P}(\text{action} \mid \text{state})$

Allmänna AI-områden och specifika AI-tekniker / algoritmer i Del 1

- **Search and Planning**
 - **Problem:** Graph traversal, Path finding, Task planning
 - Tekniker: bäst-först-sökning, bredden först, djupet först, Dijkstra, A*
- **Control with Scalar Fields and Vector Fields**
 - **Problem:** Path finding, Motion, Flocking/Swarming, Building placement, Army positioning
 - Tekniker: Scalar/Cost/Potential field, Vector/Flow field
- **Supervised Learning**
 - **Problem:** Opponent modeling, Build order prediction, Plan prediction
 - Tekniker: Bayesian networks, Gaussian Mixture Model, Hidden Markov Model, Neural networks
- **Reinforcement Learning**
 - **Problem:** Micro
 - Tekniker: Q-learning

Problemområden

Del 2



- **Tilldelningsproblemet**

(e.g. Assignment Problem och Task Allocation)

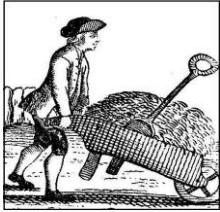
- **Grupperingsproblemet**

(e.g. Coalition Structure Generation)

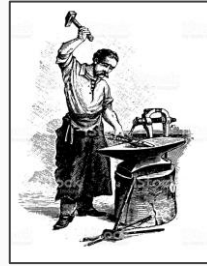
- **Placera byggnader / trupper**

(e.g. Automated Decomposition)

Tildelingsproblemet



Task 1



Task 2



Task 3

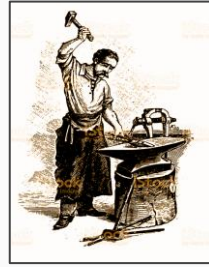


Agents with different skills and abilities

Tildelingsproblemet



Task 1



Task 2



Task 3



Agents with different skills and abilities

Tilldelningsproblemet

Teknik: The Hungarian method

StarCraft-exempel: Tilldela enheter till “jobb”, exempelvis:

- $A = \{SCV_1, SCV_2, SCV_3, Marine_1, Marine_2\}$
- $B = \{\text{bygga, samla mineraler, samla gas, scouta, anfalla}\}$
- **Constraint:** Alla enheter måste ges exakt ett arbete var.
- **Kostnadsfunktion:** Avståndet mellan agent $a \in A$ och uppgift $b \in B$.

Exempel på lösning:

$\{(SCV_1, \text{bygga}), (SCV_2, \text{mineraler}), (SCV_3, \text{gas}), (Marine_1, \text{scout}), (Marine_2, \text{anfalla})\}$

Tilldelningsproblemet

Kan, i StarCraft, också användas till:

- **Micro**: Tilldela mål i strid, fokusera eldkraft.
- **Produktion**: Bestämna vilken byggnad som skall producera vad.
- **“Strategi”**: Tilldela abstrakta uppgifter till grupper av enheter.

... och mycket mer!



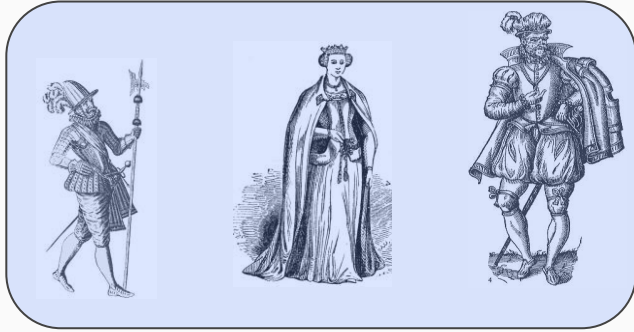
Grupperingsproblemet



Agents with different skills and abilities

Grupperingsproblemet

Coalition Blue



Värde: 3

Coalition Red



Värde: 1

Coalition Green



Värde: 5

**Totalt värde:
 $3 + 1 + 5 = 9$**

Grupperingsproblemet

Teknik: Coalition structure generation

StarCraft-exempel: Skapa effektiva koalitioner av enheter som sedan får koordinera sig själva.

- $A = \{Marauder_1, Marauder_2, Medivac_1, Medivac_2, Marine_1, Marine_2, Marine_3, Marine_4, Marine_5, Tank_1\}$
- **Constraint:** Varje enhet måste vara medlem i exakt ett team.
- **Utility-funktion:** Exempelvis baserad på förhållandet mellan antalet agent-typer och storleken på koalitionen.



Placera byggnader / trupper

Problem: Decomposition (Dela upp världen i olika områden)

Tekniker: flood fill, graph traversal

Byggnadsplacering:

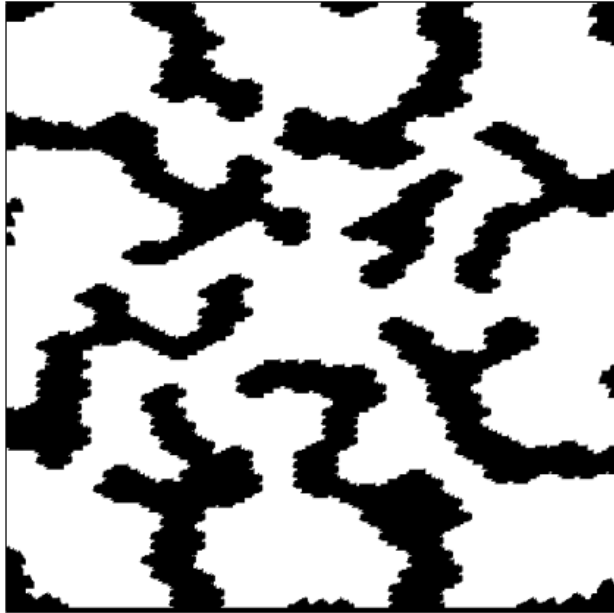
- Placera byggnader på ett sådant sätt så att era enheter kan röra sig fritt i er bas.
- Bygga murar för att göra det svårare för fienden att ta sig in i er bas.
- Decentraliserad produktion och flera baser.



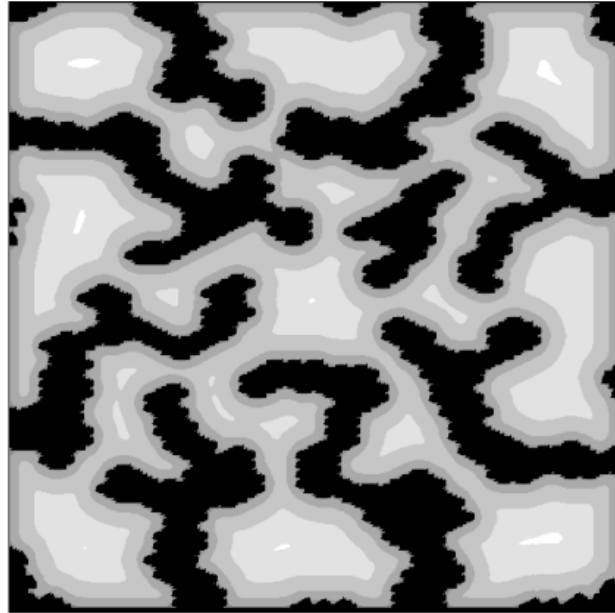


Placera byggnader / trupper

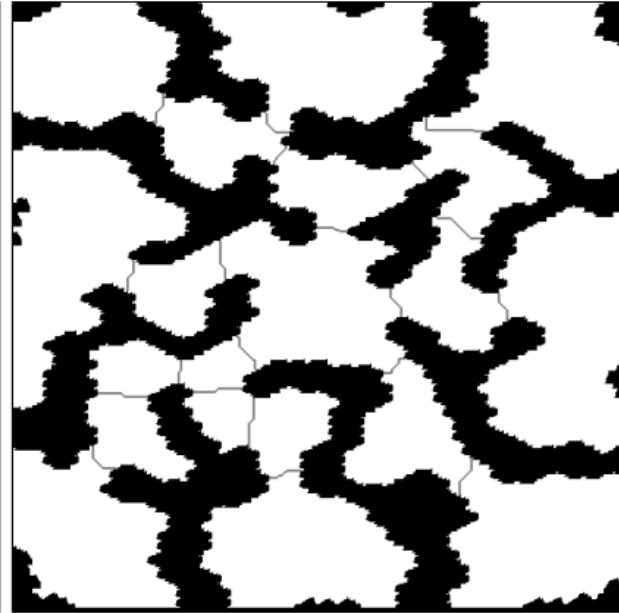
StarCraft-exempel: Identifiera choke-points för placering av försvar



Accessible terrain



Flood fill



Choke areas

Tips och Sammanfattning



Tips

Tänk på:

- ... att det finns en **litteraturlista** med relevanta referenser på kursens **wiki-sida**.
- ... att ni inte måste välja något av de problem som vi exemplifierade, eller lösa dem med de tekniker vi tog upp.
- ... vad er agent får för problem om ni saknar en bra lösning.

Tänk på:

- ... att en lösningsteknik kräver *indata*
 - Sökmeter kräver en graf. Vad innehåller den? Var kommer den ifrån?
 - Potential fields kräver attraktorer / repellerare. Var kommer de ifrån?
- ...och ger *utdata*, eller *styr* något
 - Potential fields ger ett vektorfält. Hur använder man detta?

Sammanfattning

Det finns många intressanta AI-problem att lösa och många tekniker och algoritmer att experimentera med!

Fundera på vad ni vill göra! Komplettera varandra!