AI algorithms and new approaches to wargame simulation

S Lucek, NSC
S Collander-Brown, Dstl
31 ISMOR
Overview

- **Mission Planner**: combat decision-making (AI) toolset
  - Supports Dstl high intensity warfighting simulations, to reduce/eliminate:
    - Complex pre-scripting
    - Human-in-the-loop

- **Stochastic Optimisation AI**
  - Genetic Programming, Simulated Annealing (novel approach)
  - Generic algorithms & architecture – Plug and Play
    - Simple application to different problems

- **Formulate the problem**: Military-like syntax
  - AI algorithms efficiently generate plans for tactical problems
    - Resemble human-like decision making

- **META model - AI generates plans** against a reduced problem set
  - Representing essential elements of the full problem.
  - Resulting solution evaluated against the full problem set
    - SimBrig assessing brigade level land engagements
  - Overcomes some of the limitations of AI techniques used
Stochastic Optimisation

- Family of techniques for solving any generalised problem
- Complex problem
  - Finds good but not guaranteed best solution
  - Explores whole solution space not just locally good solution
- Explore solution space in controlled way
  - Based on fitness measure
  - Definition of “good” can vary with user requirement
- Wide range of problems
  - Timetabling/Scheduling (travelling salesmen)
  - Game solutions (chess/soccer bots)
Genetic Algorithms

- Entity – abstract representation of a candidate solution to a problem
  - Typically bit stream
  - Decoded to a solution
  - Solution evaluated to obtain fitness measure
- Population of Entities
  - Initialise randomly
  - Evolve in generations, mutation, parent crossover & selection effects mimic survival of fittest
- End with a ‘best’ solution
- Genetic Programming
  - Ensures efficient decoding
Fitness measure

- Core of all Stochastic Optimisation algorithms
- Good measure of fitness
  - Allows algorithm to correctly apply selection pressure
  - Ensures fittest elements of population are evolved
- Each entity represents order set
  - Run through game & assess results
    - Losses
    - Achievements (positions held or denied from the enemy, or enemy losses or neutralisation)
    - Risk (enemy proximity, own units mutually supporting)
    - Efficiency (minimum resource consumption)
- SO algorithms notorious for finding loopholes
Simulated Annealing

- Well understood & efficient optimisation technique
- Candidate solution
  - Randomly perturbed for new solution
  - Probability new solution accepted:
    \[ e^{-\frac{F_C - F_N}{T}} \]
  - T: “Temperature”
  - F: Fitness measure – Well understood/constrained
- Annealing Schedule
  - Initially large T – explore solution space
  - As progress – reduce T to “polish” good solution
SA: Solution Perturbation

- Problem in application to gaming problems
  - How to perturb candidate solution for new solution
  - Efficient algorithm should consider T schedule
    - Large perturbations for high T
    - Small perturbations for low T
  - Node/Input tree of GP solution easy to do this
- Perturbation
  - Node: select node in tree replace with randomly generated node tree
  - Input: change input value(s)
- High T
  - Favour Node perturbation
  - Favour Nodes with many descendants
  - Change multiple Inputs
Stochastic Optimisation: limitations

- Slow – consider many solutions
- Solutions are problem specific:
  - Only optimises based on fitness criteria
    - Excellent at novel solutions tailored to detail of problem
      - Not doctrinally correct solution found in Staff Officers Handbook
    - Exploits loopholes
      - In fitness criteria
      - In evaluation model
        - Good test for model
Mission Planner: Iterative

- Generate/evolve best solution
- Next iteration, use last iteration best solution &
  - Change side
  - Change problem scenario
  - Change AI algorithm control parameters
- Solution can be evolved against multiple scenarios
  - Must be good against each
- Solution can be seeded
  - Library of solutions
Mission Planner: Generic

- AI algorithms Generic Solution
  - Nodes and inputs have no meaning
  - No concept of problem applied to
  - Simply randomly changed

- Decoder
  - Only problem specific part
  - Translates generic node tree to order set
  - Runs evaluation model to get fitness score

- Architecture: plug and play decoders
  - Proof of Concept
  - SimBrig
  - META
  - Iterations can use different decoder
  - Generate end solution using different decoder
Military Syntax

- Military Synch Matrix
- Nodes correspond to:
  - Areas
  - Timelines
  - Orders
  - Seize, Hold etc.
  - Linked to Areas & Timelines
- Units naturally co-operate in time and space
- Efficiently generate "human-like" orders
META

- Model for EngagementT Analysis
- AI algorithms requirements for plan evaluation model
  - Fast
  - Robust
    - No logic loopholes
    - Evaluate nonsense order sets
      - AI considers bad solutions
      - Need good measure of fitness

- META Simple
  - Representing ONLY the essential elements of the full problem
  - Simple and Robust algorithms
  - Quick and cheap (3 months, 2 man team)
META

- Aggregated land model
  - Brigade level (SimBrig)
  - Algorithms at an aggregated level
    - Quickest execution speed
    - Expose the fewest loop holes in algorithm logic and application
  - Combat:
    - Lanchester-like
    - BAMS-like 2-D cross matrix of attrition rates by unit type/size
      - Handles units of differing sizes/capabilities
  - Movement, combat, detection, artillery models dependent on
    - Terrain, unit types, postures, suppression
  - Arc, node movement network (SimBrig)
    - Zones of control
    - Internode visibility
Results

- MP Demo
- META Demo
Conclusions

- “Simplicity is the ultimate sophistication”
  - Leonardo da Vinci

- AI Algorithms
  - Generic architecture – Plug and Play
    - Simple application to different problems

- Formulate the solution: Military-like syntax
  - AI algorithms efficiently generate plans for tactical problems
    - resemble human-like decision making

- Meta model - AI generates plans against a reduced problem set
  - Representing essential elements of the full problem.
  - Simple, Fast, Robust
  - Overcomes limitations of AI techniques
Dstl Conclusions

- Mission planner will allow
  - Greater range of potential solution space to be examined
  - More reactive Red allowing more robust testing of plans
  - Rapid generation of variations, examining changes in
    - Force Structure
    - Constraints
  - Improved testing of complex models