

Another well known development of cellular automata used to model warfighting is Map Aware Non-uniform Automata (MANA), see Lauren *et al.* (2001), developed by the New Zealand Defence Technology Agency. MANA is a sophisticated model which incorporates a wide range of features not available in other similar products. In comparison to ISAAC and MANA, the BactoWars project, see Millikan *et al.* (1996), is a developmental build of software for the purpose of modelling warfighting. BactoWars was developed by the Australian Defence Science and Technology Organisation (DSTO), Land Operations Division. Without loss of generality, we use BactoWars as a simulation tool for our agent-based models.

Three inputs are used in order to instantiate simulations in BactoWars. First, the nature of the battlespace is defined as a two dimensional grid of squares. This grid is interpreted as a two dimensional topological map by associating each of the grid squares to information about the nature of the terrain in the corresponding map coordinate. The map we use is based on a generic instantiation of an operation held in a littoral environment, see Australian Army (2002). Second, the entities populating the grid are defined on a generic 52-calibre self-propelled 155mm artillery, see Jane's (2002). These entities are associated with concepts such as firepower, red or blue force membership and the status of alive or dead. Entities are initially placed over the grid according to particular scenario requirements. Third, the rules each entity follows are defined. These rules involve the characteristics of movement and life and death. The rules assigned to these entities are designed to reflect the behaviour of the 155mm artillery in the field.

Two alternative Command-and-Control architectures for the entities in our simulations are considered. These two models are based on the *Ant* and *Borg* models as described in Sands *et al.* (2003). These models represent concepts for distributed and collective control respectively in a networked enabled environment. This study uses agent-based modelling to simulate the operational components of these models. Furthermore, this study investigates an application of two different kinds of optimisation algorithms embedded within the behaviours of the agents in the two corresponding models. Using these models, the differences between a user-optimised approach and a systems-optimised approach is discussed.

2. Experimental Method

2.1 Environment and Scenario

Simulations are run in a littoral environment, see Concept for Manoeuvre Operations in the Littoral Environment (MOLE), Australian Army (2002), modelled for agent-based simulations. This environment is divided into a 1000×1000 playing grid. Each of the square regions of the grid represent a 500m^2 area and are labelled with either *minor*, *nominal*, *substantial*, or *impassable* terrain restrictions. The grid size of 500m^2 is chosen for convenience. Hence, ten grid coordinates represents a distance of five kilometers. This size is sufficiently small enough to model lethal area information but is large enough to be tractable when the simulation is run. The four edges of the 1000×1000 grid are marked impassable to prohibit movement outside the playing grid. The map used in this study can be found in Wheeler and White (2004).

Traversal difficulty scores are used during movement of entities, see Table 1. Grid squares with high traversal difficulty scores require more effort to traverse than those with a low score. Impassable grids may not be entered at all. The values listed in Table 1 are representative of actual recorded speeds of 155mm artillery platforms and are scaled for the purpose of simulation. Note that entities in the simulations are positioned at the centre of the grid they occupy and may only move horizontally, vertically and diagonally. The actual movement of entities is explained in greater detail in subsequent sections.

Table 1. *Characteristics assigned to grid coordinates*

Terrain Restriction	Traversal Difficulty Score
Minor	210 units
Nominal	266 units
Substantial	580 units
Impassable	Infinite

The scenario is played out in discrete time $t \in \mathbb{N}$. Each step in time t represents an interval of six minutes in real-time. Hence, ten iterations of the algorithm together correspond to one hour elapsing in real-time. This representation of six minutes is the largest possible interval that is meaningful when considering, for example: movement speeds, rate of fire, and the time it takes to align the gun to a target.

2.2 Entities

Entities are modelled on a generic 155mm 52-calibre self-propelled artillery system, see Jane's (2002). Each entity is assigned a number of characteristics as explained below and summarized in Table 3.

- *Force Membership.* Entities belongs to either the red force (traditionally the enemy force) or the blue force (traditionally the friendly force). Civilian and allied forces are not considered.
- *Movement Counter.* During each time step, entities' movement counters are reduced according to the terrain traversed, see Table 1 above. Entities may not enter a grid that requires more movement units than the entities possess. Movement in an impassable region, such as the edge of the grid, is prohibited. Entities' movement counters are replenished at the start of each new time step.
- *Operational Time of Birth.* Entities are not all placed upon the playing grid at $t = 0$. Instead, entities operational time of birth counters record the time at which those entities are placed on the playing grid. This allows the red and blue forces to receive reinforcements throughout the simulation.
- *Operational Status.* Entities are initially considered to be operational. Entities that are hit by enemy fire are marked as destroyed and removed from the playing grid in the subsequent time step.
- *Position.* The playing grid is referenced by the tuple (x, y) . Both the abscissa and ordinate are defined over the discrete points $1 \dots 1000$. Note that the boundaries $x = 1$, $x = 1000$, $y = 1$, and $y = 1000$ are defined above to be impassable. Hence, entities may not be initialized in these regions.
- *Target Priority List.* A target priority list defines an exhaustive set of acceptable targets for the entities belonging to the blue force. Each acceptable target is ranked, on the natural numbers, according to priority. This ranking increases linearly in proportion to priority. Targets not on the target priority list are not targeted.

- *Target Status.* Target status records which, if any, of the acceptable targets from an entity's target priority list is the current target allocated to that entity. More than one entity may be allocated the same target.
- *Weapons Blast Template.* Shells only strike targets that are allocated by valid target status parameters. Collateral damage is not considered.
- *Weapons Range.* Entities' weapons have a maximum range of 78 grids radius. Firing upon targets beyond this range is prohibited.
- *Weapons Kill Probability.* Shells striking a target destroy that target according to a binomial distribution $p = 0.05$. Hence, if there are n shells striking a target, that target is destroyed with probability $P(\text{destroyed}) = 1 - (1 - p)^n$, where n denotes the number of shells striking the target. This study does not take into account the ability of the weapons platforms to converge their firepower on an enemy by successively correcting their aim. That is, the probability of killing a target with the next shell fired remains constant and is independent of the number of shells previously fired at that target. Hence, the act of firing upon a target is best interpreted as a desire to suppress the enemy by shelling the grid it resides in rather than a necessity to destroy the enemy with precise fire.

Table 2. Example probability of destroying a target

Weapons Rate of Fire	Probability of Kill
1 shell	0.05
2 shells	0.0975
⋮	⋮
10 shells	0.401263
11 shells	0.4312
12 shells	0.45964
⋮	⋮
100 shells	$1 - (0.95)^{100} < 1.0$

- *Weapons Rate of Fire.* Entities may fire more than once each time step. However, a weapons rate of fire of two shells per time step is analogous to modelling assets travelling in pairs and a weapons rate of fire of three shells per time step is analogous to modelling assets travelling in groups of three. We consider assets to travel as individuals with a rate of fire of 12 rounds per time step. Entities fire upon the targets allocated by their respective target status parameters. Hence, entities engage only one enemy target in any single time step irrespective of the potential number of times that they are permitted to fire in that time step. Note that irrespective of the number of shells landing on a target, it is impossible to guarantee that target is destroyed. Instead, if we require that targets be destroyed with at least 45% confidence, for example, we calculate the number of shells required and group that number of platforms together to form the capability brick which defines an entity. In Table 2, it is easy to see that to achieve at least a 45% kill probability, given a single round kill probability of 5%, we require: 12 assets with a rate of fire of 1 round per turn, 6 assets with a rate of fire of 2 rounds per turn, 4 assets with a rate of fire of 3 rounds per turn, 3 assets with a rate of fire of 4 rounds per turn, 2 assets with a rate of fire of 6 round per turn, or 1 asset with a rate of fire of 12 rounds per turn.

