Potential-Field-Based Unit Behavior Optimization for Balancing in StarCraft II

Jonas Schmitt
Friedrich-Alexander-Universität Erlangen-Nürnberg
jonas.schmitt@fau.de

Sabine Seufert
Friedrich-Alexander-Universität Erlangen-Nürnberg
sabine.seufert@fau.de

Christian Zoubek
Friedrich-Alexander-Universität Erlangen-Nürnberg
christian.zoubek@fau.de

Harald Köstler
Friedrich-Alexander-Universität Erlangen-Nürnberg
harald.koestler@fau.de

ABSTRACT

This article presents an evolutionary algorithm for optimizing the offensive behavior of opposing units in the real-time strategy game StarCraft® II. The goal for each group is to deal maximal damage to the opposing group while receiving a minimal amount of damage at the same time. The actions each unit performs are determined by accumulating a number of predefined potential fields. Dependent on the statistics of the involved units, the parameters of these fields then fully describe the behavior of each individual unit. Since this includes a huge number of possibilities, the set of optimal parameter values for both groups in an encounter is obtained by applying an evolutionary algorithm.

1. INTRODUCTION

With the work presented here, we aim to optimize micromanagement based on a simplified model of the StarCraft® II[1] game mechanics. To reduce the complexity of the simulation, the following assumptions are made:

- The area on which the units can move and interact is a rectangular field with neither obstacles nor height differences.
- No new units are produced during the simulation. Only units occurring in a predefined build order are considered.
- Structures are ignored.

To control the movement of single units, artificial potential fields are used. Each unit generates repulsive and attractive fields, whose characteristics depend on the units statistics. The movement of a single unit is then defined by the gradient of the accumulated fields applying to it. For each encounter, the parameters of the potential fields are optimized, such that the damage the units deal to their enemies is maximized, while the damage they receive themselves is minimized.

GECCO '15 July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07.
DOI: http://dx.doi.org/10.1145/2739482.2764643

2. FORWARD SIMULATION

Since there is no freely available API for controlling units in the game directly, an efficient forward simulation is required that approximates the real game mechanics.

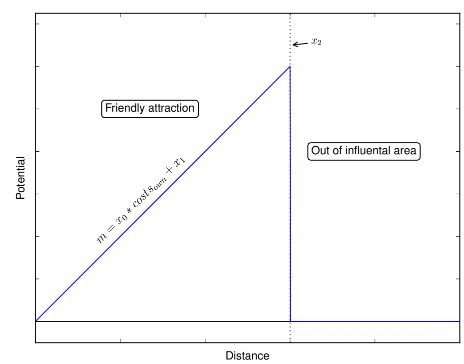

Figure 1: Function describing the attractive potential of friendly units.

The goal is to find a set of optimal parameters that describes a unit's movement strategy for a specific encounter. This is achieved by utilizing parametrized artificial potential fields. Each unit creates multiple potential fields around its position. The movement direction for every unit is then determined by accumulating the gradients of all potentials evaluated at the respective position. The intensity of repulsive fields decreases with distance, while the attractive fields' intensities increase. Three different potential fields are defined, modeled as linear functions (see figures 1, 2 and 3).

During the forward simulation, the following actions are performed in each time step:

1. *Attacking:* Each unit checks if any enemy unit is within its attack range. If multiple units can be targeted, a simple heuristic decides which unit to attack by favoring targets that can

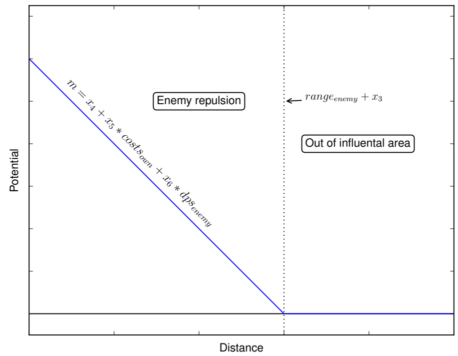

Figure 2: Function describing the repulsive potential of enemy units.

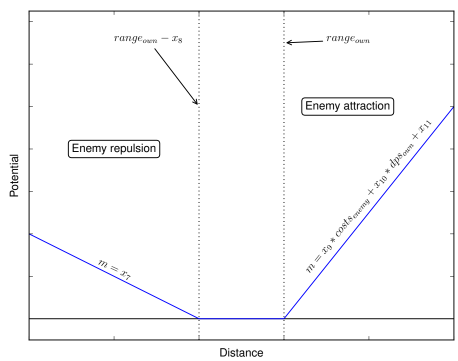

Figure 3: Function describing the attractive potential of enemy units.

be defeated. Additionally, units are prioritized by the amount of applicable damage.

2. *Moving:* The distance each unit can move is determined by its movement speed. The position of a unit at the next time step is then computed with the following equation:

$$\vec{p_{i+1}} = \vec{p}_i + \frac{\sum \vec{F}}{|\sum \vec{F}|} \times s \quad (1)$$

where $\vec{p}_i$ is the position at time step i, $\sum \vec{F}$ the sum of all forces and s the movement range.

The forward simulation finishes when either all units of a player have been defeated or the simulation's duration exceeds a certain limit.

3. OPTIMIZATION

The movement strategy of a specific unit is determined by twelve parameters $x_0, \ldots, x_{11}$ as it can be seen in figure 1, 2 and 3. They describe the weights of the different potential fields and thus influence the direction a unit moves during the current time step.

Initially, the parameters are set to reasonable values which define the initial strategy for a player. In the first iteration, a number of distinct strategies for both players is obtained by optimizing the parameters against the opponent's initial strategy. During each further iteration, the optimization is performed against the opponent's best strategies from the last iteration. To solve the subproblem of finding the optimal set of values for both players with regards to their combat efficiency against a number of opposing strategies, a single-objective genetic algorithm is applied. Therefore, the fitness value of an individual x is computed with the following equation:

$$\text{Fitness}(x) = \frac{1}{n} \sum_{i=1}^{n} (\text{damage}_i(x) + \text{health}_i(x)) \quad (2)$$

n is the number of encounters used for fitness evaluation, $\text{damage}_i(x)$ is the total damage the units corresponding to x have dealt to their enemies and $\text{health}_i(x)$ the sum of their remaining health and shield values at the end of encounter i.

4. EXPERIMENTS AND RESULTS

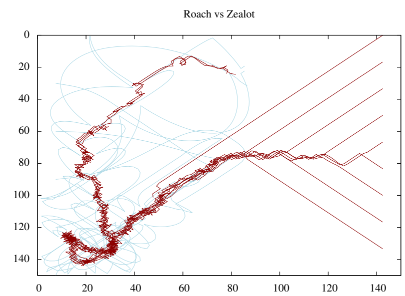

Figure 4: Tracked paths of the encounter between Roaches in light blue and Zealots in dark red.

To evaluate the movement strategies obtained by the optimization, the path of each individual unit is tracked, representing its tactical behavior. As an example the tracked paths of an encounter between Roaches and Zealots are shown in figure 4.

Zealots are melee units with a very small attack range and therefore are easily outranged by their enemies. To maximize their chances, all Zealots accumulate at the beginning and then try to reach the enemy all at once. In consequence, they can attack their opponents with their whole strength which is more effective than trying to reach them as single individuals. This also makes it more difficult to eliminate them one by one before they are able to approach the enemy close enough for an attack. In contrast, the Roaches' movement paths show a widely spread pattern, as seen in figure 4. As the Roaches are ranged units, their best strategy in this particular encounter is to stay away from the group of Zealots, while slowly decimating them. Their pattern shows an even more intelligent extension of this behavior as they split up, surround their opponents and successfully lure single individuals away from the group.

By means of the outcome of the presented optimization certain aspects of the game can be examined. Based on the optimal combat behavior it can be verified if the statistics specific to the two participating unit groups are balanced in the respective encounter. Hence, it can be revealed if some units have an advantage or disadvantage in a direct combat against certain other units. Balancing in real-time strategy games is generally difficult and is until now commonly achieved by involving human test players. The described approach enables the verification of this property without the need of a fully developed game engine. Consequently, it could allow a shortening of the time consuming testing phases. Furthermore, visualized movement patterns provide important indications for human players for an optimal micromanagement in certain combat situations. While the accuracy of these observations is limited by the simplification made in the forward simulation, the optimization itself is independent of these restrictions.