На протяжении многих лет соревнования по искусственному интеллекту StarCraft побуждали многих людей и группы внедрять различные боты, способные играть в полноценные игры StarCraft 1v1. Структура большинства современных ботов возникает в результате попыток разложить игру на иерархию более мелких подзадач, таких как стратегия более высокого уровня, тактика, управление боевыми единицами, анализ местности и сбор разведданных. Боты различаются по сложности, и, хотя многие из них основаны на правилах, наиболее эффективные боты в настоящее время используют более сложные методы искусственного интеллекта, такие как поиск/планирование в реальном времени, предварительно обученные контроллеры нейронных сетей и онлайн-обучение во время соревновательных игр. В этом разделе мы предоставляем обзор подборки ботов и обсуждаем некоторые подходы к искусственному интеллекту, которые они реализуют. Мы упоминаем только тех ботов, которые были активны в одном из соревнований 2017 года, недавно были обновлены и используют некоторые более сложные методы искусственного интеллекта.

• CherryPi: CherryPi - бот-зерг TorchCraft [12], разработанный исследовательской группой Facebook по искусственному интеллекту. Он реализован как набор разнородных модулей, которые могут быть добавлены, удалены или настроены с помощью командной строки. Такая конструкция позволяет легко заменять отдельные модули на обучающие модули или проводить узкие эксперименты, используя только их подмножество. Модули взаимодействуют путем обмена двумя типами элементов данных через архитектуру blackboard: парами ключ-значение и так называемыми объектами UPC (единица измерения, позиция, команда), которые имеют достаточно общее значение, чтобы быть свободно интерпретированными другими модулями. В общем случае UPC представляет собой вероятностное распределение подразделений, вероятностное распределение позиций и единую команду. Заказы на сборку представлены в виде набора приоритетных запросов, помещенных в очередь, которая заполняет требования и применяет оптимизации (например, выделяет ресурсы для строительства точно в срок). Решения о сражении или бегстве принимаются путем объединения юнитов в группы и запуска боевой симуляции. Боевое моделирование CherryPi работает аналогично пакету моделирования SparCraft (см. UAlbertaBot) с наивной политикой “атаковать ближайшую цель” для поведения противника. Поиск путей с учетом угроз используется для поиска наименее опасных маршрутов в безопасное место. Выбор стратегии высокого уровня в нескольких играх основан на алгоритме UCB1, выбирающем из фиксированного списка стратегий против каждой расы. Бот использует TorchCraft для связи с BWAPI по протоколу TCP, что позволяет ему работать на другом хосте, чем StarCraft.

• cpac: Бот-зерг, созданный командой из 13 человек из Китая, cpac сочетает жестко закодированные правила с многослойной сетью персептронов для производства единиц. Сеть обучается на парах состояние-действие, извлеченных из большого набора данных BroodWar games 8. Ядро cpac bot основано на ботах UAlbertaBot и Steamhammer (см. ниже).

• ForceBot: ForceBot - это бот зергов, написанный на агентном языке программирования GOALan, разработанном поверх BWAPI для программирования когнитивных агентов.9 Поскольку язык ЦЕЛЕЙ предназначен для реализации многоагентных систем, все подразделения ForceBot имеют своего собственного соответствующего агента с конкретными убеждениями и целями. Каждый агент более или менее следует шаблону искусственного интеллекта, основанному на правилах.

• Железо: 10 Iron bot выиграл конкурс AIIDE 2016 года и представляет собой децентрализованную мультиагентную систему, где каждое устройство управляется высокоавтономным индивидуальным агентом, способным переключаться между 25 вариантами поведения. Все его подразделения преследуют одну простую цель: добраться до главной вражеской базы и уничтожить ее. Он часто кажется ботом-преследователем, что связано с тем, что его юниты ведут себя в основном индивидуально. Существуют также так называемые “экспертные” агенты, которые автономно рекомендуют, как следует расходовать ресурсы и какие подразделения следует обучать, основываясь на эвристике.

• KillAll: KillAll - это бот зергов, основанный на боте Overkill от Sijia Xu, и большая часть его функциональности основана на правилах. Однако его производственный модуль использует Q-learning для выбора типов единиц для производства на основе текущей ситуации.

• Krasi0bot: Krasi0bot участвует в соревнованиях каждый год с 2010 года и все еще активно развивается. По словам автора, изначально он начинался как бот, основанный на правилах, и в настоящее время в некоторой степени использует генетические алгоритмы, нейронные сети и потенциальные поля. Поскольку бот не является открытым исходным кодом, эти данные не могут быть проверены. Krasi0bot играет за расу землян и известен своими сильными оборонительными возможностями и широким разнообразием реализованных стратегий.

• LetaBot: 11 LetaBot выиграли турниры SSCAIT 2014, 2015 и 2016 годов. Он использует поиск по дереву Монте-Карло (MCTS) для планирования перемещения групп юнитов по карте. Аналогичный подход ранее использовался автором Nova bot Альберто Уриарте [13]. Он использует совместный поиск путей для сбора ресурсов и интеллектуальный анализ текста для извлечения заказов на сборку из статей Liquipedia.

• McRave: Все решения бота McRave основаны на текущем составе вражеских юнитов – нет жестко запрограммированных технических вариантов. Бот также создает модель противника и использует ее для выбора заказов на построение.

• MegaBot:12 Для каждой игры MegaBot [14] выбирает один из трех подходов, каждый из которых реализован как отдельный бот (Skynet, Xelnaga или NUSBot). Алгоритм выбора моделируется как многорукий бандит. В начале игры выбирается алгоритм, использующий стратегию эпсилон-жадности. После игры определяется награда (+1, 0, -1 за победу, ничью и проигрыш соответственно), а значение выбранного алгоритма обновляется с помощью инкрементной версии взвешенного по времени экспоненциального среднего (правило обновления Q-learning).

• PurpleWave:13 Принятие решений ботом PurpleWave в основном основано на иерархических сетях задач. Для микроменеджмента он использует гибридный подход команды / мультиагента и кластеризацию ближайших соседей. Затем бот моделирует результаты сражений и предлагает тактику для отрядов, минимизируя тактические подходы каждой стороны (например, “атаковать”, “убегать” или “сражаться с рабочими”). В конце концов, каждое подразделение принимает тактическое предложение к сведению, но ведет себя независимо. Устройства выбирают примерно из двух десятков простых, многократно используемых вариантов поведения без состояния. Эвристика бота включает в себя использование потенциальных полей для перемещения юнитов. Стратегии выбираются на основе результатов предыдущих игр, расы, карты и количества стартовых позиций. В нем есть график выбора стратегии, например, порядок открытия сборки в сочетании с переходами в середине игры и составами в конце игры.

• StarcraftGP: StarcraftGP - это первая программа StarCraft meta-bota, которая автономно создает программу, которая автономно играет в StarCraft [15]. В настоящее время StarcraftGP v0.1 использует (линейное) генетическое программирование и может напрямую писать код на C++. Его первые творения: Сальса и Текила, были первыми ботами, не написанными непосредственно человеком, для участия в международных конкурсах. Steamhammer:14 Бот зергов Steamhammer, разработанный Джеем Скоттом, и его версия для случайных гонок Randomhammer основаны на UAlbertaBot (см. ниже), использующем сложную боевую симуляцию для прогнозирования исхода сражений. Боты также используют иерархическое реактивное управление для юнитов. Для производства протоссов и терран Randomhammer использует поиск по ветвям и границам, в то время как Zerg в настоящее время основан на правилах.

• tscmoo: 15 tscmoo выиграли соревнования AIIDE 2015 и CIG 2016. Бот не использует внешних библиотек: у него есть свой собственный код моделирования боя для прогнозирования исхода сражений, он не использует BWTA16 для анализа местности и даже имеет свой собственный поиск путей с учетом угроз для отдельных юнитов. Бот является одним из самых стратегически разнообразных и выбирает среди своих многочисленных стратегий, основываясь на их успехе в предыдущих играх. Последние версии бота экспериментировали с рекуррентными нейронными сетями для высокоуровневой стратегии и принятия решений о порядке сборки.

• UAlbertaBot:17 UAlbertaBot участвовал во всех крупных соревнованиях по искусственному интеллекту StarCraft с 2010 года и выиграл конкурс AIIDE 2013 года. UAlbertaBot использует динамическую систему поиска заказов на сборку на основе эвристического поиска (BOSS) для планирования всех своих заказов на сборку в режиме реального времени, а также систему моделирования боевых действий StarCraft под названием SparCraft для оценки результатов внутриигровых сражений. Бот использует результаты предыдущих игр против конкретных противников, чтобы выбрать стратегию для реализации в начале каждой игры, причем каждая стратегия определяется во внешнем файле конфигурации JSON. Его разработка была сосредоточена на простоте использования и модификации, и как таковая стала основой для более чем 10 других ботов в текущих соревнованиях, включая LetaBot, Overkill и Steamhammer. В 2017 году UAlbertaBot стал CommandCenter18, первым ботом, способным играть как в BroodWar, так и в StarCraft 2.

ZZZKbot: 19 ZZZKBot, бот-зерг, разработанный Крисом Коксом, стал победителем соревнований AIIDE и CIG 2017 года. Его общая стратегия реализует 4 простые стратегии 1base rush: 4-pool, Speedlings, Hydralisks и Mutalisks. Если первоначальный бросок не завершает игру, бот переключается либо на муталисков, либо на Стражей для поздней игры, одновременно исследуя обновления для всех своих юнитов. Бот записывает информацию о выигрыше / проигрыше для каждого соперника и использует эту информацию для выбора наилучшей комбинации параметров стратегии для будущих игр на основе правил. Большинство правил ботов для управления подразделениями и микроменеджмента - это простые правила поведения, основанные на приоритизации экспертных знаний.

За последние несколько лет мы можем наблюдать, что ИИ-боты StarCraft действительно становятся сильнее в целом. В соревнованиях AIIDE и CIG несколько ботов предыдущих лет намеренно оставляются на следующий год, чтобы служить ориентиром для прогресса, и мы каждый раз видим, что эти боты-эталоны со временем становятся хуже. Кроме того, опытные игроки и энтузиасты наблюдают за повторами и отмечают, как, по их мнению, боты стали лучше или хуже с течением времени. Наиболее примечательно, что многие из этих опытных игроков считают, что боты постепенно адаптируют более "стандартный" стиль игры, чем более ранние боты, которые традиционно использовали одну стратегию, такую как бросок, но не более того. Более современные боты разработали тактику в середине и даже в конце игры, которая не была замечена в более ранних спешащих ботах. В целом, боты, похоже, становятся лучше в составлении армии, выборе порядка сборки, размещении зданий и общей стратегии игры. В то время как самые сильные боты в настоящее время играют на любительском уровне, опытные игроки отмечают, что они все еще кажутся слабыми в нескольких ключевых областях. Самое главное, что боты по-прежнему кажутся довольно слабыми в динамической адаптации своих стратегий во время матча в ответ на полученную информацию о своем противнике. Большинство ботов используют набор из нескольких стратегий, которые они выбирают в начале матча и выполняют до конца игры, и лишь несколько ботов пытаются кардинально изменить ситуацию, если противник делает что-то неожиданное. Это означает, что боты по-прежнему довольно уязвимы для игроков-людей, которым легче менять стратегии и тактику по ходу игры.

Современные боты также кажутся довольно уязвимыми для человеческой способности быстро идентифицировать паттерны и поведение ботов и быстро использовать это во время матча. Например, один игрок-человек во время матча "человек против машины" отметил, что один бот-юнит будет преследовать его зерглинга, когда приблизится к юнитам ботов, и продолжил управлять зерглингом рядом с армией ботов, а затем повел бота в погоню за дикими гусями по всей карте. Все это время бот, возможно, рассуждал о том, что его армия может выиграть битву с единственным подразделением зерглингов, не понимая при этом, что человек просто выигрывал время, пока его армия не будет готова к финальной атаке. Это также иллюстрирует одну из самых больших проблем во всем искусственном интеллекте: понимание долгосрочных последствий действий, которые приводят к отсроченному вознаграждению. Опытный человек способен быстро понять, что его эксплуатируют таким образом, и что это будет иметь негативные последствия в будущем, и способен остановить такое поведение. Это долгосрочное видение, которое настолько интуитивно понятно людям, остается проблемой для современного RTS AI.