
Hacking Dota Underlords With Integer Programming Model

A Preprint

Alexander A. Ponomarenko, Dmitry S. Sirontkin*
Laboratory of Algorithms and Technologies for Network Analysis
National Research University Higher School of Economics
Nizhny Novgorod, Russia
aponomarenko@hse.ru, dsirotkin@hse.ru

8 июня 2020 г.

Abstract

В данной работе мы демонстрируем, как задача оптимального выбора состава команды в популярной компьютерной игре Dota UnderLords может быть сведена к задаче целочисленного линейного программирования. Мы приводим модель и её решения. Также мы доказываем, что данная задача относится к классу NP-hard.

Keywords Целочисленное программирование · NP-hard

1 Описание Dota Underlords

По ходу игры восемь игроков составляют команду из «героев» – существ, способных сражаться друг с другом на игровой карте. У каждого из героев есть базовые параметры - здоровье, урон, скорость атаки и прочие, а также особая способность, которая определяет его роль в игре. Каждый герой принадлежит к двум или более «альянсам» - классам, в которые входят несколько героев. так, например, герой Enchantress принадлежит одновременно к альянсу «друиды» и к альянсу «хищники». При наборе нескольких героев из одного альянса (для каждого альянса это число индивидуально) игрок получает бонус, который выражается в усилении всех героев из альянса, усилении всех своих героев или ослаблении всех героев соперника. Последнее может быть интерпретировано как относительное усиление своих героев и поэтому на протяжении работы будут рассматриваться только первые два случая. Следует отметить, что для одного альянса может быть несколько бонусов, которые открываются разным количеством героев соответствующего альянса, при этом они могут быть разного типа.

Также в ходе игры можно усилить своих героев до более высоких уровней или путём покупки внутриигровых предметов. В рамках данной работы эти аспекты учитываться не будут.

Таким образом, сила команды игрока определяется как:

1. Силой выбранных героев
2. Бонусами от альянсов в которых они состоят

Как оказывается, данную задачу можно представить как задачу комбинаторной оптимизации. В рамках данной работы мы показываем её NP-полноту, приводим её формулировку как задачу ЛП и решаем её симплекс-методом для частного случая Dota Underlords.

*Use footnote for providing further information about author (webpage, alternative address)—not for acknowledging funding agencies.

2 Перевод задачи в язык LP

2.1 Простейшая постановка задачи

Формализуем задачу следующим образом:

Будем считать, что всего у нас в есть n героев на выбор. Будем считать, что сила некоторого i -го героя определяется за некоторую неотрицательную величину s_i . За x_i будем обозначать принадлежность героя выбранной команде. Условимся, что когда $x_i = 1$, если i -й герой принадлежит набранной команде и $x_i = 0$ в противном случае. Тогда условие того, что в команде не более, чем m героев можно записать в виде $\sum_{i=1}^n x_i \leq m$. Тогда в простейшей форме данную задачу можно сформулировать следующим образом:

$$\begin{aligned} \max \quad & \sum_{i=1}^n x_i s_i \\ \text{s.t.} \quad & \sum_{i=1}^n x_i \leq m \\ & x_i \in \{0, 1\} - \text{управляющая переменная} \\ & n, m, s_i - \text{константы} \end{aligned} \tag{1}$$

В данной постановке задача решается элементарно – достаточно взять n элементов с наибольшими весами

2.2 Постановка задачи с альянсами

Как было упомянуто, в «Dota Underlords» каждый герой принадлежит к двум или более «альянсам» - классам, в которые входят несколько героев. Когда в команде присутствует несколько героев из одного альянса (для каждого альянса это число индивидуально) игрок получает бонус, который выражается в усилении всех героев из альянса, усилении всех своих героев или ослаблении всех героев соперника. Последнее может быть интерпретировано как относительное усиление своих героев и поэтому на протяжении работы будут рассматриваться только первые два случая. Следует отметить, что для одного альянса может быть несколько бонусов, которые открываются разным количеством героев соответствующего альянса, при этом они могут быть разного типа.

Данную ситуацию мы предлагаем моделировать с помощью введения 3-х индексного тензора $e_{ijk} \in \mathbb{R}$ означающего бонус герою i , за альянс j , в котором присутствуют не менее k героев альянса j . Другими словами, e_{ijk} это k -й бонус альянса j для героя i .

С помощью тензора e_{ijk} можно поддерживать два типа альянсов – те которые дают бонусы своим членам и те, которые дают бонусы всем героям игрока. При этом, альянсы рассмотренных видов отличатся только тем, что в альянсах, дающих бонус своим членам, величина e_{ijk} равна нулю тогда и только тогда, когда i -й герой не принадлежит j -му альянсу. Заметим, что тензор e_{ijk} будет сильно разрежен, поскольку альянсы от которых бонусы идут всем героям не много. Контролировать вхождение бонуса e_{ijk} в общую силу команды предлагается с помощью управляющей бинарной переменной I_{ijk} . Так мы можем записать целевую функцию как следующую сумму $\sum_{i=1}^n x_i s_i + \sum_{i=1}^n \sum_{j=1}^t I_{ijk}$. Связь переменных x_i и I_{ijk} задаётся неравенствами $\forall i, j, k : \sum_{i'=1}^n a_{i'j} x_{i'} - k \geq M(I_{ijk} - 1)$. Здесь надо вставить пару предложений про эти неравенства, в чём логика этой связи. Они не дают бинарной переменной I_{ijk} принимать значение 1, если в решение входит меньше чем k героев входящих в альянс j . Когда решение содержит героев из альянса j меньше чем k , левая часть этого неравенства отрицательная, поэтому чтобы неравенства соблюдались правая часть должна быть ещё меньше. Такое возможно только, когда бинарная I_{ijk} равно нулю. В этом случае правая часть равна $-M$, где константа заведомо большая, чем k , то есть больше, чем максимальный размер альянса q . Разумно требовать, чтобы бонус для героя i мог быть активирован ($I_{ijk} = 1$), только когда герой i входит в решение. Это задаётся неравенствами $\forall i, j, k : I_{ijk} \leq x_i$. Также мы хотим, чтобы бонус e_{ijk} был активирован, только для если герой i принадлежит альянсу j . Для этого мы в модель включили неравенства $\forall i, j, k : I_{ijk} \leq a_{ij}$.

Таким образом, после введения в модель альянсов, она выглядит следующим образом.

$$\begin{aligned}
 & \text{Целевая функция:} \\
 & \max \sum_{i=1}^n x_i s_i + \sum_{i=1}^n \sum_{j=1}^t \sum_{k=1}^q e_{ijk} I_{ijk} \\
 & \text{Ограничения на входные данные} \\
 & \forall j : \sum_{i=1}^n a_{ij} \leq q \\
 & \text{Ограничения на управляющие переменные} \\
 & \forall i, j, k : \sum_{i'=1}^n a_{i'j} x_{i'} - k \geq M(I_{ijk} - 1) \\
 & \sum_{i=1}^n x_i \leq m \\
 & \forall i, j, k : I_{ijk} \leq x_i \tag{2} \\
 & \text{Управляющие переменные:} \\
 & I_{ijk} \in \{0, 1\}, 1 - \text{если для героя } i, \text{ активирован } k\text{-й бонус } j\text{-го альянса,} \\
 & x_i \in \{0, 1\}, 1 - \text{если герой } i - \text{ входит в решение} \\
 & \text{Константы:} \\
 & n \in \mathbb{N} - \text{число героев,} \\
 & m \in \mathbb{N} - \text{максимальный размер команды} \\
 & t \in \mathbb{N} - \text{общее количество альянсов} \\
 & q \in \mathbb{N} - \text{максимальный размер одного альянса,} \\
 & s_i \in \mathbb{R} - \text{сила героя } i, \\
 & e_{ijk} \in \mathbb{R} - \text{бонус для героя } i, \text{ если активирован } k\text{-й бонус } j\text{-го альянса} \\
 & a_{ij} \in \{0, 1\}, 1 - \text{если герой } i \text{ входит в альянс } j
 \end{aligned}$$

3 Доказательство NP-полноты задачи Dota Underlords с альянсами

Покажем, что данная задача в указанной в предыдущем разделе формулировке является NP-полной. Рассмотрим её частный случай, удовлетворяющий следующим ограничениям

1. Веса элементов равны нулю
2. Никакой элемент не принадлежит более чем одному классу
3. Каждый элемент принадлежит хотя бы одному классу

Очевидно, в рамках данной задачи брать только часть элементов из класса смысла не имеет, поскольку это не увеличит суммарный вес. Таким образом, мы либо берём весь класс, либо не берём. Это можно проинтерпретировать как задачу о рюкзаке, где «предметом», который можно положить в рюкзак, является класс, его вес — это количество элементов в нём, «стоимость» — суммарный бонус класса, а «емкость рюкзака» — общее количество элементов, которые необходимо взять.

Данная задача является классической NP-полной задачей и сводится к задаче «Dota Underlords».

4 Практическое применение для реальной задачи Dota Underlords

Мы применяем данную модель для анализа реальной задачи Dota Underlords. Отметим, что полученные результаты не стоит считать некоторой объективной оценкой качества команды героев. Причина состоит в неизбежном упрощении сил героев и влияния, которое оказывают альянсы. Каждый герой в Underlords обладает некоторой способностью, которая активируется при заполнении шкалы маны и обладает некоторым временем перезарядки. Способности и бонусы альянсов также весьма разнообразны по своему влиянию на игру - они могут наносить урон, лечить союзников, мешать врагам пользоваться

своими способностями и прочее. К счастью, в игре есть система из пяти «ярусов», устроенная так, что герои внутри яруса примерно равны по силе.

В рамках упрощённой модели мы принимаем следующее.

- 1) Силы всех героев первого яруса равны 1, второго 1.5, третьего - 2, четвертого - 2.5, пятого - 3
- 2) Все альянсы дают один и тот же мультипликативный бонус 1.1. Если у альянса больше одного уровня влияния на героев (например альянс воинов даёт своим героям последовательно +10, +15 и + 25 к броне), то второй уровень даёт мультипликативный бонус 1.2, третий - 1.3.

Особым случаем является альянс Scrappy, который даёт бонус на своём первом уровне одному своему случайно выбранному члену. Мы считаем этот бонус равномерно распределённым между всеми членами альянса в рамках общей модели и считаем как и прочие бонусы первого уровня. Бонус второго уровня мы в этом альянсе считаем так же, как в других.

5 Частные разрешимые случаи

Как и многие подобные задачи, задача Dota Underlords становится полиномиально разрешимой при определённых ограничениях.

Рассмотрим простейший нетривиальный случай

1. Каждый герой принадлежит ровно двум альянсам
2. Каждый альянс содержит ровно двух героев
3. Сила каждого героя равна a
4. Бонус от каждого альянса на каждого героя одинаков и равен $\frac{b}{2}$

Таким образом, суммарный бонус от сбора альянса в данной модели равен b .

Теорема 1. Задача Dota Underlords в указанной формулировке решается за $O(nb)$, если b - натуральное и NP-полна, если числа a и b - несравнимы.

Доказательство. Данную задачу удобно переформулировать на языке теории графов. Обозначим героев за вершины графа, а отношение "находятся в одном альянсе за ребро. Тогда в данном графе степени всех вершин равны 2 и он разбивается на несколько (возможно один) циклов, несвязных между собой. Т.к. число элементов в искомом наборе фиксировано, то итоговая целевая сумма зависит от числа рёбер, порождаемых в данном графе множеством взятых вершин-героев. Таким образом, если мы берём из некоторого цикла множество вершин мощности k , которое не совпадает с множеством всех вершин цикла, то мы можем получить самое большее бонус $(k - 1)b$ - взяв вершины, порождающие путь в данном цикле. В случае, если мы берём весь цикл, то мы получаем бонус kb . таким образом, для достижения наибольшего бонуса необходимо, чтобы все взятые вершины порождали в графе некоторое множество циклов (возможно ноль) и не более одного пути. При этом, случай, когда пути вообще нет лучше случая, когда он есть. В случае, если нам надо взять всего n элементов и мы смогли взять n вершин, порождающих исключительно циклы, мы получаем итоговую сумму $n(a + b)$. Если же присутствует путь, то мы получаем сумму $na + (n - 1)b$. Отметим, что фактически необходимо проверить возможность построения набора, порождающих исключительно циклы, т.к. построить набор, порождающий циклы и путь — элементарно, достаточно добавлять циклы в набор в произвольном порядке, пока не окажется, что мы хотим добавить цикл размера большего, чем оставшееся количество вершин. Тогда вместо этого добавляем произвольный путь из этого цикла.

Данная задача легко сводится к задаче о рюкзаке с целыми весами и решается методом динамического программирования. Действительно, размеры циклов в этой задаче соответствуют весам и объёму элементов, общее их количество в итоговом решении - объёму рюкзака. Таким образом, необходимо лишь проверить, что в решении данного экземпляра задачи о рюкзаке рюкзак заполнится полностью.

Сложность алгоритма в данной задаче равна $O(nb)$ при целом b . Задача тогда решается методом динамического программирования.

В случае произвольного b данная задача представляет собой специфический случай задачи о сумме. Задачей о сумме называется задача в которой требуется определить, есть ли в данном множестве чисел подмножество с некоторой суммой s (в каноническом случае - с суммой 0). Известно, что он - NP-полна. Таким образом, задача Dota Underlords сводится к ней и значит NP-сложна. \square

6 Доказательство NP-трудности задачи Dota Underlords

Покажем что задача Dota Underlords NP-трудна

Рассмотрим её частный случай - пусть все альянсы имеют размер равный двум, и сила всех героев одинакова. Рассмотрим частный случай задачи Dota Underlords со следующими ограничениями:

1. Силы всех героев одинаковы ($\forall i, j, x_i = x_j$)
2. Альянсы могут давать бонусы исключительно героям в них состоящим ($\forall i, j, k, a_{ij} = 0 \implies e_{ijk} = 0$)
3. Все альянсы имеют одинаковый размер, равный двум ($\forall j, \sum_i a_{ij} = 2$)
4. Все альянсы дают бонус исключительно при наличии в них обоих героев ($\forall i, j, e_{ij1} = 0$)
5. Бонусы ото всех альянсов равны ($\forall i, j, i', j', a_{ij} = 1, a_{i'j'} = 1 \implies e_{ij2} = e_{i'j'2}$)

Тогда данные можно представить в виде графа $G(V, E)$, где множество вершин V соответствует героям, а множество рёбер E - активным альянсам. Задача поиска оптимальной команды размера k таким образом сводится к поиску порождённого графа G на k вершинах с максимальной плотностью. Под плотностью в данной формулировке может понимается величина $\frac{G'(E)}{G'(V)}$. Действительно, при данных ограничениях общая сила команды линейно зависит от количества активных альянсов, что соответствует $G'(E)$. Т.к. k неизменно, то с ростом плотности графа G' растёт итоговая сила команды.

В работе ?? было показано, что задача поиска порождённого подграфа с максимальной плотностью и фиксированным размером в графе является NP-полной. Как было показано, она является частным случаем задачи Dota Underlords и к ней сводится. Отсюда следует её NP-сложность.

Список литературы