Математическая Теория Игр и её Приложения, т.12, в.2, с. 110–121

УДК 519.83
ББК 22.176

МНОГОАГЕНТНЫЙ ПОИСК НА МНОЖЕСТВЕ: РАСПРЕДЕЛЕНИЕ УСИЛИЙ И ОЦЕНКА ЭФФЕКТИВНОСТИ

Илья А. Чернов*
Институт прикладных математических исследований
Карельского научного центра РАН
185910, Петрозаводск, ул. Пушкинская, 11
e-mail: chernov@krc.karelia.ru

В статье рассматривается модель поиска на множестве, в которой каждая точка является локацией для поиска ценных объектов. Агенты, ведущие поиск с разной эффективностью, распределяются по множеству, выбирая локации исходя из априорной перспективности каждой. Выбравшие одну и ту же локацию конкурируют между собой. Показано, что в свободных предположениях распределение агентов совпадает с априорной перспективностью. Это позволяет оценить удельный поток результатов, который постоянен на множестве. С целью более точного предсказания этого потока, зависящего от производительности отдельных агентов, предлагается политика заявок, успешное выполнение которых сулит награду. Рассмотрены случаи различных распределений неточности оценки агентом своей производительности и оценено отличие заявленной производительности от средней.

Ключевые слова: добровольные вычисления, вычислительные сети, предсказание производительности, распределенный поиск.

Поступила в редакцию: 03.10.19 *После доработки*: 17.01.20 *Принята к публикации*: 20.05.20

---

* Поддержано грантом РФФИ 18-07-00628

1. Введение

Многие прикладные задачи поиска по существу двухэтапны: сначала выбирается область, в которой затем ведется перебор элементов. Во многих задачах пространство поиска может быть разложено на несколько местоположений, каждое из которых является локацией для поиска, причем локации могут иметь различную априорную перспективность. Примерами являются виртуальный скрининг, моделирование в материаловедении, сканирование неба, проверка грубой силой (brute force) различных гипотез в теории чисел и т.д. Виртуальный скрининг [1] — оценка вычислительными методами характеристик (энергии связи и др.) соединения сложных биохимических молекул (белка и т.н. лиганда) для целей медицины, фармакологии и биологии. Существуют базы данных лигандов, содержащие большое число молекул, которые можно упорядочить в виде множества в многомерном пространстве химических характеристик. Для данного белка комплексу требований может удовлетворять множество лигандов, и возникает задача их эффективного поиска. Разные подмножества этой области могут иметь различную перспективность, оцениваемую, например, по истории поиска или исходя из свойств соединений.

Многие физико-химические процессы в материаловедении описываются моделями, зависящими от нескольких — в пределах десяти — параметров, характеризующих материал. Аппроксимация измеренных кривых модельными дает оценку качества набора параметров, и встает задача идентификации модели по данным эксперимента на подмножестве многомерного пространства. Здесь опять различные подобласти различаются оценкой ожидаемого качества находок, и возникает задача эффективного перебора.

Удобной иллюстрацией является поиск клада на архипелаге: агент выбирает остров, на котором затем осуществляет поиск. Возникает задача выбора наиболее перспективной локации для поиска, осложняющаяся взаимодействием с другими агентами. Например, закономерен выбор самого перспективного острова, однако если его выберут другие, в том числе более эффективные агенты, другой остров выглядит предпочтительнее.

Проекты добровольных вычислений [2] занимают важную нишу в

области научных вычислений. Достаточно упомянуть такие популярные проекты, как SETI@home, положивший начало парадигме добровольных вычислений, Folding@home, Collatz Conjecture, Climate Prediction — полное количество насчитывает несколько десятков, а охват научных областей весьма широк. Разработано несколько программных систем для организации таких вычислительных сетей, наиболее популярной из них является BOINC[1] — на сайте проекта опубликован актуальный список активных проектов на базе системы. Помимо публичных проектов, научные группы развертывают локальные вычислительные сети, задействуя ресурсы одной или нескольких организаций. При этом удается достичь высоких показателей пиковой и средней производительности при предельно низких капитальных затратах.

Вычислительные сети из компьютеров общего назначения, соединенные коммуникационными сетями, называют грид-системами (Desktop Grid). Задачи, которые могут эффективно решаться, обычно состоят из большого числа сравнительно простых независимых между собой вычислительных заданий. В большинстве случаев это задачи перебора или поиска. Отметим, что BOINC предполагает использование процессоров во время простоя. По этой причине реальная производительность узла зависит не столько от его аппаратного обеспечения, сколько от режима работы и способна существенно меняться в течение суток.

Вычислительные узлы обычно получают задания от сервера, что приводит к задаче диспетчеризации. С учетом гетерогенности [3] сети, то есть переменной мощности узлов и их численности и большого разнообразия вычислительных устройств и платформ, это задача сложна, а в ряде случаев NP-полна [4], причем даже в предположении идентичности узлов. Различные узлы могут выбирать предпочтительную для них область поиска исходя из собственной оценки своей производительности. Предполагается, что они заинтересованы в обнаружении возможно большего количества ценных результатов, за которые получают вознаграждение в той или иной форме. Сервер, то есть организатор вычислительного проекта, в свою очередь, заинтересован, помимо максимальной производительности, также в

[1] https://boinc.berkeley.edu/

оценке реальной производительности и в оценке времени завершения проекта. Для этого недостаточно стимулировать участников участвовать в проекте в качестве вычислителей, но желательно получать от них достоверную информацию о мгновенной вычислительной производительности. Здесь мы говорим об обмене информацией с узлами, а не об оценке доступности узлов [5] или поиске шаблонов поведения [2].

Теоретико-игровой подход выглядит естественно в данном контексте, однако применяется реже, чем это было бы целесообразно. В [6] игровая схема используется для снижения нагрузки на сервер, а статья [7] посвящена игровой оптимизации задачи поиска.

В этой статье рассматривается две игровые модели. В одной игроки — это узлы сети, характеризуемые своей эффективностью. Они ведут поиск на множестве, на котором задана функция перспективности — оценка ожидаемой ценности результатов. Каждая точка множества — локация для поиска. Конкуренция между двумя и более агентами, ведущими поиск в одной локации, приводит к разделению результатов и снижению выигрыша каждого. Показано, что распределение агентов по множеству в широких предположениях совпадает с распределением перспективности, то есть удельный эффективный поток результатов постоянен на множестве.

Вторая задача посвящена игре между сервером и вычислительными узлами. Вознаграждение за ценные результаты и за проделанную работу выплачивается не напрямую, а по двухэтапной схеме, аналогичной аукционному (контрактному) бриджу [8][2]. Игрок оценивает свою производительность за заданный период времени (например, сутки) и сообщает серверу объем работы, которую предполагается проделать, или ценность результатов, которые ожидается найти, а также локацию, в которой будет вестись поиск. При выполнении заявки вознаграждение существенно выше за обещанную работу и ниже — за перевыполненную. При невыполнении заявки проделанная работа оплачивается, но по еще более низкому тарифу. При подходящем подборе тарифных ставок сервер получает более точную оцен-

[2]В карточной игре бридж выигрыш пары игроков в текущей партии (сдаче) зависит не только от числа набранных очков (взяток), а еще от предварительно сделанной заявки.

ку совокупной производительности сети и, следовательно, времени окончания расчета.

2. Модель

Рассмотрим дискретное множество $\{T_i\}$, $i = 1, \dots I$. Его элементы — это локации для поиска, каждое содержит много элементов для перебора. Распределение полезных находок известно: f_i. Игроки, которых много (N), выбирают T_i, в котором будут вести поиск. У каждого игрока скорость поиска составляет v, при этом пока все идентичны. Распределение игроков по множеству обозначим x_i, так что Nx_i игроков заняли элемент i.

Игрок, перебирая объекты в локации T_i со скоростью v, имеет следующую вероятность обнаружения полезного объекта в единицу времени:

$$F_i = f_i \cdot \frac{v}{v \cdot Nx_i} = \frac{f_i}{Nx_i}.$$

Если эта величина, как функция i, не постоянна, то игроки выберут локацию i с большим ее значением. Соответствующее x_i увеличится, а F_i уменьшится. Поэтому равновесное распределение $F_i = \text{const}$: распределение игроков совпадает с распределением перспективности локаций.

С учетом дискретности числа узлов, x_i на практике не может быть слишком малым. Эту сложность нетрудно обойти, исключив из T_i локации с близкой к нулю перспективностью f_i. Критерий малости, например, дает $Nx_i < 1$. То есть исключаются те локации, поиск в которых неперспективен. К ним можно вернуться позднее, когда основное количество ценностей уже извлечено.

Теперь предположим, что игроки имеют разные скорости v_j, $j \le J$, а также есть много игроков с разными скоростями: N_j. Долю игроков, имеющих скорость поиска объектов v_j и выбравших локацию i, обозначим x_{il}. Тогда условием равновесия является постоянство

$$f_i \cdot \frac{v_j}{\sum\limits_{l=1}^{J} v_l N_l x_{il}} = \text{const}_j$$

Это равновесие по Нэшу, поскольку ни одному игроку не выгодно отклоняться от стратегии.

Теперь рассмотрим еще один фактор — природу, которая может как-то выбирать натуральное $k \le K$ с вероятностями q_k, определяя $f_i = f_{ik}$ из некоторого множества. Предположим, что ранг матрицы f_{ik} — полный. Тогда игрок угадывает распределение f_{is}, выбирая одно из s с вероятностью p_s. Получается разбиение игроков по группам тех, кто поставил на данное распределение. В данной локации i работают, в среднем, $p_s f_{is}$ игроков, поставивших на распределение с номером s. Их производительность, с учетом различных скоростей разных игроков, выражается формулой

$$\sum_{l=1}^{J} v_l N_l \sum_{s=1}^{K} p_s f_{is},$$

вклад в средний выигрыш некоторого игрока j исхода с номером k в данной локации i равен

$$q_k f_{ik} \cdot \frac{v_j}{\sum\limits_{l=1}^{J} v_l N_l \sum\limits_{s=1}^{K} p_s f_{is}} = q_k \frac{f_{ik}}{\sum\limits_{s=1}^{K} p_s f_{is}} \cdot \frac{v_j}{\sum\limits_{l=1}^{J} v_l N_l},$$

а в среднем по исходам получится

$$\frac{\sum\limits_{k=1}^{K} q_k f_{ik}}{\sum\limits_{s=1}^{K} p_s f_{is}} \cdot \frac{v_j}{\sum\limits_{l=1}^{J} v_l N_l}.$$

Эта величина должна быть постоянной по i в равновесии, так как иначе игроки будут чаще выбирать более выгодную локацию. Второй множитель (приведенная скорость игрока) от i не зависит; поэтому получается, с учетом нормировки распределений p_s, q_k, I равенств

$$\sum_{k=1}^{K} q_k f_{ik} = \sum_{s=1}^{K} p_s f_{is},$$

которые можно переписать в виде системы линейных уравнений

$$\sum_{k=1}^{K} (p_k - q_k) f_{ik} = 0 \quad \forall i = 1, \dots I.$$

Поскольку, что ранг матрицы f_{ik} полный, эта система имеет единственное решение $p_k = q_k$, то есть распределение игроков по исходам совпадает с распределением исходов, известным природе.

3. Политика контрактов

Введем дополнительного игрока — сервер, или организатор вычислений. Он заинтересован в оценке времени завершения проекта, то есть — эффективной мгновенной производительности. Предположим, что каждый узел знает свою производительность v и сообщает ее серверу. Эффективная производительность зависит от загрузки вычислительного устройства и от других волатильных факторов. Поэтому, с целью заинтересовать пользователя сообщать актуальную информацию, предлагаем схему, аналогичную контрактному бриджу. Узел заявляет объем работы или количество ценностей, которые предполагает обнаружить (контракт), а также локацию, в которой будет вести поиск. Если через заданный интервал времени (например, сутки) он выполнил контракт, он получает некоторую дополнительную премию gt, линейно зависящую от размера контракта t (а не от реально выполненной работы). Пока не вводим премии за превышение контракта и штрафы за его невыполнение.

Ясно, что при наличествующей информации узлы будут заявлять именно актуальную v, гарантирующую в среднем максимальный выигрыш, так как недооценка снизит выигрыш напрямую, а переоценка — за счет участившихся провалов.

Интересно учесть неопределенность оценки v. Пусть производительность узла является случайной величиной с абсолютно непрерывной функцией распределения $F(x)$, $f(x) = F'(x)$, $F(0) = 0$. Тогда средний выигрыш игрока имеет вид

$$E = \int_t^\infty gt dF(x) = gt\Big(1 - F(t)\Big), \quad \frac{dE}{dt} = g\Big(1 - F(t)\Big) - gtf(t).$$

Это величина премии, умноженная на вероятность превышения скорости поиска, указанной в контракте. Отсюда получается простое необходимое условие максимума:

$$1 - F(t) = tf(t).$$

Рассмотрим три примера: равномерное, нормальное и экспоненциальное распределения. Для них условие является также и достаточным. Для последнего, $F(x) = 1 - \exp(-\lambda x)$, получаем удобную оценку $t^* = 1/\lambda$, то есть оптимальный выбор игроков в точности совпадает с математическим ожиданием распределения их производительности.

Рассмотрим равномерное распределение на отрезке $[2a, 2b]$, причем число $2a$ (нижний предел производительности) серверу известно. Условие максимума принимает вид $t^* = b$, позволяя вычислить среднюю производительность, равную $a + b$. Отметим, что узел занижает свою производительность, если минимальная производительность больше нуля. Отметим также, что премия в форме $g \cdot (t - 2a)$ сразу приводит к оптимальному $t^* = a + b$, совпадающему со средним — в этом случае премия выплачивается за дополнительную работу относительно минимальной $2a$, которая всегда выполняется.

Пусть теперь распределение нормальное, со средним a и дисперсией σ^2. Условие оптимальности сводится к уравнению

$$\Psi(z) = 1 - \Phi(z) - (z + b)\varphi(z) = 0, \tag{3.1}$$

где Φ — функция распределения стандартного нормального закона, $\varphi = \Phi'$, $z = (t - a)/\sigma$, $b = a/\sigma$.

Понятно, что при малых (относительно a) σ решение z^* будет близко к среднему. В самом деле, функция $\Psi(0) < 0$ при $b > \sqrt{\pi/2} \approx 1.25$. При достаточно больших по модулю отрицательных z функция $\Psi(z) > 0$, поэтому имеется решение $z^* < 0$. Оценим его асимптотику при $b \to \infty$ (что при фиксированном $a > 0$ эквивалентно $\sigma \to 0$). Примем $z^* = -\sqrt{2 \ln \left(b/\sqrt{2\pi}\right)}$. Тогда $\varphi(z^*) = 1/b$ и

$$\Psi(z^*) = -\Phi(z^*) - \frac{z^*}{b} \to 0 \text{ при } b \to +\infty.$$

Теперь видно, что решение z^* имеет асимптотику $-\sqrt{2 \ln b}$, то есть $-\sqrt{2 \ln \sigma^{-1}}$, а разность $t^* - a \to 0$ при $\sigma \to 0$.

Таким образом, узлы с малым разбросом производительности будут немного занижать ее. Критическое значение $b = \sqrt{\pi/2}$ гарантирует $t^* = a$. Это значение, а тем более — более низкие, уже непрактично, так как значительна вероятность отрицательных значений производительности. Однако формально при этом $z^* > 0$, то есть узлы

с таким большим разбросом будут завышать свою производительность, но незначительно: $\Psi(0) > 0$ при таких b, $\Psi(1) < 0$, то есть $t^* - a < \sigma$.

Можно оценить максимальное отклонение более точно. Дифференцируя равенство $\Psi(z) = 0$ по σ, считая a постоянным, получаем

$$\big(2\varphi(z) + (z+b)\varphi'(z)\big)\frac{dz}{d\sigma} = \frac{a}{\sigma^2}\varphi(z).$$

Учитывая определение z, переходим к переменной t и потребуем, чтобы $dt/d\sigma = 0$, а потом вернемся к переменной z:

$$\big(2\varphi(z) + (z+b)\varphi'(z)\big)z = -b\varphi(z).$$

Учтем соотношение $\varphi'(z)/\varphi(z) = -z$:

$$b = z\frac{2-z^2}{z^2-1}. \tag{3.2}$$

Подставим это значение в уравнение (3.1):

$$1 - \Phi(z) = \frac{z\varphi(z)}{z^2-1}. \tag{3.3}$$

Наконец, выразим

$$T = \frac{a-t}{a} = -\frac{z}{b} = -\frac{z^2-1}{2-z^2}. \tag{3.4}$$

Это максимальное относительное занижение заявленной производительности относительно средней. Критическое $z = z^*$ легко определяется численно из (3.3): $z^* \approx -0.84$. Соответствующее $b^* \approx 3.69$ из (3.2), а (3.4) дает максимальное относительное занижение

$$T^* = \frac{a-t^*}{a} = \frac{1-z^2}{2-z^2} \approx 22.75\%.$$

В проекте RakeSearch [9], по крайней мере, некоторые узлы (наиболее эффективные) имеют[3] близкое к нормальному распределение производительности с $a \approx 12$ и $3\sigma \approx 5$. При этом $b = a/\sigma \approx 7.2$,

[3]Личное сообщение Н. Никитиной

$z \approx -1.367$ — решение (3.1), что соответствует недооценке своей средней производительности $a - t = z\sigma \approx 2.278$ единиц, то есть относительное занижение $T \approx 18.98\%$.

Возможны примеры, когда функция не имеет максимума, например при одностороннем распределении Коши с плотностью

$$f = \frac{2}{\pi(1+t^2)}, \quad t \geq 0.$$

В самом деле, функция $G(t) = 1 - F(t) - tf(t)$ монотонно убывает, так как

$$G' = -2f(t) - tf'(t) = -\frac{4}{\pi(1+t^2)} + \frac{4t^2}{\pi(1+t^2)^2} = \frac{4t^2 - 4(1+t^2)}{\pi(1+t^2)^2} < 0.$$

Предел $G(t)$ при $t \to +\infty$ равен нулю, поэтому $G(t) > 0$ при $t > 0$. При таком распределении целесообразно введение штрафов за невыполнение работы. Линейный штраф $-p(t-x)$ при невыполнении контракта приводит к условию

$$g\big(1 - F(t)\big) - gtf(t) - pF(t) = 0,$$

которое всегда выполняется хотя бы в одной точке $t > 0$, так как левая часть непрерывна, положительна при $t = 0$ и отрицательна при достаточно больших t.

В случае мультимодального распределения локальных максимумов может быть больше одного и из них следует выбрать наибольший. Ситуация принципиально не отличается от дискретного распределения, при котором имеется несколько исходов и следует выбрать максимальный.

Рассмотрим пример дискретного распределения. Пусть у узла есть два возможных режима работы в проекте добровольных вычислений: с низкой производительностью h и с высокой H. Это правдоподобное предположение, так как часто рабочий или домашний компьютер отдельного человека либо загружен своей работой, либо простаивает. Предположим также, что серверу известны величины h и H (они определяются аппаратными характеристиками устройства), но не известны вероятности p, q этих исходов. Очевидно, что узел, делая заявку на контракт, выбирает между h и H, получая средние

выигрыши, соответственно, gh и gHq. Контракт позволяет оценить q относительно порога h/H. Это может быть полезно, например, в распространенном случае, когда одно ядро двухъядерного процессора все время работает на проект, а второе подключается только во время простоя. Тогда $H = 2h$ и схема контрактов позволяет сравнить частоту простоев с $1/2$.

4. Заключение

Удалось показать, применяя методы теории игр, что распределение игроков по пространству поиска в достаточно широких предположениях совпадает с априорной перспективностью локаций. Также предложена схема контрактов (аналогичная контрактному бриджу), позволяющая предсказать производительность узла за интервал времени. Рассмотрены различные распределения неточности оценки узлом своей производительности и показано, как скорректировать заявку узла для более точной оценки.

СПИСОК ЛИТЕРАТУРЫ

1. Ивашко Е.Е., Никитина Н.Н., Möller S. *Высокопроизводительный виртуальный скрининг в Enterprise Desktop Grid на базе BOINC* // Труды Международной суперкомпьютерной конференции «Научный сервис в сети Интернет». 2015. С. 58–60.

2. Durrani, N., Shamsi, J. *Volunteer computing: requirements, challenges, and solutions* // Journal of Network and Computer Applications. 2014. V. 39 P. 369–380.

3. Anderson, D., Reed, K. *Celebrating diversity in volunteer computing* // System Sciences. 42nd Hawaii International Conference. 2009. P. 1–8.

4. Xhafa, F., Abraham, A. *Computational models and heuristic methods for grid scheduling problems* // Future Generation Computer Systems. 2010. V. 26. N 4. P. 608–621.

5. Kianpisheh, S., Kargahi, M., Charkari, N.M. *Resource availability prediction in distributed systems: An approach for modeling non-stationary transition probabilities* // IEEE Transactions on Parallel and Distributed Systems. 2018. v. 28. N 8. P. 2357–2372.

6. Mazalov V.V., Nikitina N.N., Ivashko E.E. *Hierarchical Two-Level Game Model for Tasks Scheduling in a Desktop Grid* // Applied Problems in Theory of Probabilities and Mathematical Statistics Related to Modeling of Information Systems. 2014. P. 641–645.

7. Nikitina N.N., Ivashko E.E., Tchernykh A. *Congestion Game Scheduling for Virtual Drug Screening Optimization* // Journal of Computer-Aided Molecular Design. 2018. V. 32. N 2. P. 363–374.

8. Frank I., Basin D. *Search in games with incomplete information: a case study using Bridge card play* // Artificial Intelligence. 1998. V. 100. P. 87–123.

9. Manzyuk M., Nikitina N., Vatutin E. *Start-up and the results of the volunteer computing project RakeSearch* // Russian Supercomputer Days. 2019. In press.

MULTI-AGENT SEARCH IN A SET: DISTRIBUTION OF EFFORT AND ESTIMATING THE EFFICIENCY

Ilya A. Chernov, IAMR KRC RAS, Cand.Sc.
(chernov@krc.karelia.ru).

Abstract: We consider a search model on the set, in which each point is a search location for valuable objects. Searchers of different efficiencies are distributed across a multitude, choosing locations based on the apriori idea of how promising each location is. Those who choose the same location compete with each other. We show that in quite free assumptions the distribution of agents coincides with that of the prospects. This allows us to estimate the specific flow of results, which is constant on the set. In order to more accurately predict this flow, which depends on the performance of individual agents, we offer a policy of bids: a reward for winning one. Cases of various distributions of uncertainty in the agent assessment of their own productivity are considered and the difference between the declared productivity and the average one is estimated.

Keywords: volunteer computing, desktop grid, throughput prediction, distributed search.