

СИСТЕМЫ ИНФОРМАЦИОННОГО ПОИСКА ДЛЯ BIG DATA: МАШИННОЕ ОБУЧЕНИЕ РАНЖИРОВАНИЮ НА ОСНОВЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

С.В. Семенихин, Л.А. Денисова (ОмГТУ)

Рассмотрена проблема машинного обучения (МО) ранжированию при работе с Big Data в рамках концепции Industry 4.0. Представлен подход к МО ранжированию в системах информационного поиска на основе многокритериальной оптимизации на базе генетических алгоритмов, положенный в основу инструментария программного комплекса. Также рассмотрен альтернативный подход к машинному обучению, построенный на базе генетического алгоритма кооперативной коэволюции, позволяющий сократить время машинного обучения за счет использования параллельных вычислений. Предложенные методы обеспечивают эффективность автоматизации принятия аналитических решений о ранжировании альтернатив в порядке убывания релевантности. Представлены результаты сравнительного анализа предложенных методов на тестовых коллекциях.

Ключевые слова: машинное обучение, ранжирование, оптимизация, информационный поиск, кооперативная коэволюция, генетический алгоритм.

Введение

На современном этапе развития производственных информационных систем наблюдается тенденция значительного роста объемов данных, находящихся в разных источниках, генерируемых подключенными к Internet устройствами, датчиками и пользователями. Это приводит к тому, что вероятность существования нужной информации растет, а возможность ее нахождения пользователем уменьшается. Информационный поиск (ИП) является неотъемлемой составляющей аналитических и бизнес-процессов любых предприятий, независимо от отрасли. Современные информационно-поисковые системы (ИПС) работают с Big Data — данными больших объемов (Big Data понимается как составная часть концепции Industry 4.0 [1]), что делает актуальным разработку методов, направленных на эффективную обработку больших объемов данных с целью представления пользователю.

Частными случаями работы ИПС являются, например, поиск внутренних документов предприятия, аналитический поиск информации по отраслям, а также поиск информации в сети Internet. В современных ИПС число документов, семантически релевантных (от англ. *relevance* — актуальность) поисковому запросу, может превышать 1 млн. значений. Это существенно увеличивает время, необходимое для нахождения пользователем документа, удовлетворяющего его поисковые потребности, среди всех документов, выбранных ИПС. В связи с этим одной из важнейших задач, решаемых ИПС, является ранжирование документов, то есть их сортировка в поисковой выдаче. Эффективность ранжирования напрямую определяет время, которое пользователь ИПС затратит на поиск релевантной информации.

Одной из основных задач организации производства в соответствии с концепцией Industry 4.0 является развитие автоматизации обработки данных и принятия решений с целью предоставления больших объемов данных в виде, подготовленном для представления пользователю. В связи с этим, в современных системах информационного поиска возрастает актуальность машинного обучения (МО) ранжированию

[2]. На основе обучающих данных, набора запросов, каждый из которых связан со списком результатов поиска, отсортированным в соответствии с релевантностью, МО позволяет сформировать функцию ранжирования, которая может быть использована для сортировки результатов поиска в дальнейшей работе поисковой системы.

Применение методов машинного обучения позволяет в рамках концепции Industry 4.0 обеспечить эффективность принятия аналитических решений о ранжировании альтернативных результатов информационного поиска в условиях непрерывного притока объема данных. В связи с этим, актуальной является разработка новых методов машинного обучения модели ранжирования и построение ее программной реализации с возможностью интеграции в ИПС независимо от отрасли, в которой предприятие функционирует.

Хотя повышение точности в ходе машинного обучения является основной задачей, скорость выполнения МО при построении функции ранжирования также является важным вопросом. Из-за разнообразия запросов и документов, обрабатываемых ИПС, обучение включает все большие и большие наборы данных с большим числом факторов. Кроме того, в связи с быстрым увеличением объемов информации в сети Internet, построенные функции ранжирования должны периодически обновляться, то есть проходить повторное обучение ранжированию с целью адаптации к изменяющейся структуре документов.

В данной работе представлены два новых метода решения задачи машинного обучения ИПС ранжированию, позволяющих повысить эффективность информационного поиска.

Первый предлагаемый метод *LTR-PGA (Learning to rank — Pareto Genetic Algorithm)* создан на основе генетических алгоритмов (ГА) [3] с применением известного и эффективного подхода к многокритериальной оптимизации — метода Парето-оптимизации [4]. Генетический алгоритм представляет собой метод поиска оптимальных решений, основанный на копировании механизмов биологического отбора и гене-

тического наследования. Преимуществом ГА по сравнению с традиционными методами оптимизации является то, что поиск решения выполняется на основе анализа множества (популяции) точек. Специфическими для задачи построения множества Парето являются операции вычисления приспособленности индивида и селекции для сходимости к фронту Парето. В предлагаемом методе выполняется построение фронта Парето-оптимальных решений, не улучшаемых одновременно по всем рассматриваемым критериям эффективности поиска [5, 6]. В связи с тем, что выбор критериев качества ранжирования документов существенно влияет на результаты оптимизации, предварительно осуществлялся определение метрик качества ранжирования в задаче МО модели ранжирования в ИПС.

Второй рассматриваемый метод — это *LTR-CGA* (*Learning to rank — Coevolutionary Genetic Algorithm*), реализуемый как параллельный генетический алгоритм (ПГА), основанный на алгоритме кооперативной коэволюции (ККЭ) [7]. Метод *LTR-CGA* предназначен для непосредственной оптимизации дискретных метрик качества ранжирования. Кроме того, метод *LTR-CGA* позволяет сократить время исполнения процедуры МО в сравнении с традиционными генетическими алгоритмами.

Применяемый для метода *LTR-CGA* алгоритм кооперативной коэволюции представляет собой особый вид ГА, где приспособленность индивида определяется в зависимости от того, насколько хорошо он сотрудничает с особями из других популяций. Алгоритмы ККЭ построены на принципе «разделяй и властвуй» — стратегии декомпозиции сложной проблемы на подзадачи и объединения их решений для того, чтобы сформировать полное окончательное решение.

При выполнении алгоритмов кооперативной коэволюции для решения задачи ранжирования несколько ГА развиваются параллельно. Таким образом, алгоритм может быть естественным образом распараллелен, что позволяет добиться существенного повышения скорости машинного обучения. В данной работе исследуется возможность применения параллельного алгоритма ККЭ в контексте МО ранжированию. Для разработки алгоритмического, программного обеспечения ИПС и подсистемы машинного обучения ранжированию создан программный комплекс на языках C++ и Python.

Тестовые коллекции и метрики качества ранжирования

Для выполнения машинного обучения ранжированию необходим набор документов и запросов с соответствующими оценками релевантности, определенными экспертами. Эти данные используются для построения ранжирующей функции, а также для оценки качества ранжирования поисковой выдачи. В проведенных исследованиях использована тестовая коллекция *LETOR 4.0*, выпущенная в 2009 г.

Microsoft Research Asia [8]. Коллекция *LETOR 4.0* [8] использует набор Web-страниц *Gov2* и два набора запросов *MQ2007* и *MQ2008* из подколлекций *TREC 2007* и *TREC 2008* соответственно. Набор *MQ2007* содержит около 1700 запросов с 69623 ссылками, набор *MQ2008* — около 800 запросов с 15211 ссылками. Каждый набор данных был разбит на пять частей, чтобы провести пятикратную перекрестную проверку. Для каждого из исходных наборов три части используются для обучения, одна часть для проверки, а оставшаяся часть — для тестирования.

Факторы релевантности. В тестовой коллекции *LETOR 4.0* для каждого документа предоставляется: шесть факторов релевантности SRF_{1-6} на основе гиперссылок (классический ссылочный признак *PageRank*, число внешних ссылок и внутренних ссылок, число слешей в URL-адресе документа, длина URL-адреса и число дочерних страниц). Также предоставляется 40 факторов SRF_{7-46} на основе содержимого документа, состоящих из 20 классических функций (таких, как длина документа и частота термов) и 20 высокоуровневых факторов (таких как результаты вероятностной модели *BM25* и алгоритмов *LMIR*).

Метрики качества ранжирования. Для оценки разработанного метода *LTR-CGA* из списка общепринятых метрик [9] качества ранжирования документов поисковыми системами были выбраны две основные — $MAP@n$ и $NDCG@n$. Рассмотрим эти метрики качества подробнее.

1) Средняя точность ранжирования MAP . Сначала определяется точность ранжирования на уровне n первых документов:

$$P@n = \frac{r}{(r+b)}, \quad (1)$$

где r — число найденных релевантных документов, b — число документов, ошибочно принятых за релевантные. На основе (1) рассчитывается средняя точность ранжирования MAP (*Mean Average Precision*):

$$MAP = \frac{\sum_n (P@n \cdot rel(n))}{r}, \quad (2)$$

где $rel(n)$ — бинарная функция, принимающая значение 1, если n -ый документ релевантен запросу (0, если иначе).

2) Дискретная метрика DCG (*Discounted Cumulative Gain*):

$$DCG@n = \sum_{p=1}^n \frac{2^{grade(p)} - 1}{\log_2(2+p)} \quad (3)$$

Здесь $grade(p)$ — средняя оценка релевантности, выставленная экспертами документу, расположенному на позиции p в списке результатов; $grade \in [0,3]$; $1/\log_2(2+p)$ — коэффициент, зависящий от позиций документа (первые документы имеют больший вес).

На основе (3) нормализованная версия $NDCG$ запишется в виде:

$$NDCG@n = DCG@n / Z, \quad (4)$$

где Z — фактор нормализации, который равен максимально возможному значению $DCG@n$ для данного запроса (то есть равен DCG идеального ранжирования).

3) Метрика $Bpref$ используется в случаях, если информация о релевантности известна только для части документов. Данный критерий вычисляется по формуле:

$$Bpref = \frac{1}{r} \sum_n (1 - NonRelBefore(n) / r). \quad (5)$$

Здесь r — число релевантных документов; n — число известных релевантных документов; $NonRelBefore(n)$ — число известных нерелевантных документов, ранжированных выше, чем n (при вычислении учитываются только первые n оцененные нерелевантные документы из прогона).

Рассмотренные метрики качества используются для построения критериев оптимальности при оценке качества ранжирования.

Метод машинного обучения LTR-PGA

Постановка задачи. На основе композиции факторов релевантности SRF , построена функция ранжирования RF :

$$RF(q, d) = \sum_{i=1}^n W_i SRF_i,$$

где SRF_i — i -ый фактор релевантности для документа d и запроса q , W_i — весовой коэффициент i -го факто-

ра релевантности, n — число факторов релевантности в системе ранжирования.

Исследуем ИПС, в которой задействованы $N=3$ частных критериев оптимальности, представленных метриками MAP , $NDCG$ и $Bpref$ (в соответствии с выражениями (2), (4), (5)), образующими векторный критерий оптимальности:

$$C(W) = \{C_1, C_2, \dots, C_N\} = \{C_{MAP}, C_{NDCG}, C_{Bpref}\}, i = \overline{1, N}$$

и $C(W) \in \{C\}$.

Здесь $W \in \{W\}$ — вектор варьируемых параметров (весовых коэффициентов) для простых ранжирующих функций SRF ; $\{W\}$, $\{C\}$ — пространства параметров и векторных критериев, соответственно.

В рассматриваемом случае ставится задача максимизации метрик качества информационного поиска. Следовательно, во введенных обозначениях задача многокритериальной оптимизации запишется в виде:

$$\max_{W \in \{W\}} C(W) = C(W^*) = C^*,$$

где W^* , C^* — решения задачи.

Представление решения. Алгоритм машинного обучения $LTR-PGA$ разработан на основе известного генетического алгоритма $NSGA-II$ (*Non-Dominated Sorting Genetic Algorithm-II*) [10]. Программная реализация алгоритма, выполняет многокритериальную оптимизацию с помощью построения множества Парето — эффективных решений.

Алгоритм разработанного метода машинного обучения $LTR-PGA$ представлен на рис. 1. На вход $LTR-PGA$ передается специально сформированный массив данных, содержащий экспертные значения релевантности для пар документ-запрос из тестовой коллекции $LETOR$, которая также содержит соответствующие значения факторов релевантности SRF .

Построение фронта Парето выполнялось для двух основных выбранных критериев, в качестве которых были выбраны полнота (*recall*) и точность (*precision*), то есть $C_1 = C_{MAP}$ и $C_2 = C_{NDCG}$ соответственно. Для выбора из множества полученных Парето-оптимальных решений единственного решения в качестве дополнительного критерия использована метрика $Bpref$: $C_3 = C_{Bpref}$. В работе [6] было продемонстрирована эффективность использования дополнительных критериев для сужения множества Парето-оптимальных решений после окончания работы основного алгоритма.

Для выполнения многокритериальной Парето оптимизации на основе генетических алгоритмов, весовые коэффициенты ранжирующей функции W_i , $i = 1, \dots, n$ (n — число весовых коэффициентов) должны быть предварительно закодированы. Выбор метода кодирования является важным этапом ГА в связи с его значительным влиянием на быстродействие алгоритма и точность результата. В данной работе используется метод вещественного кодирова-

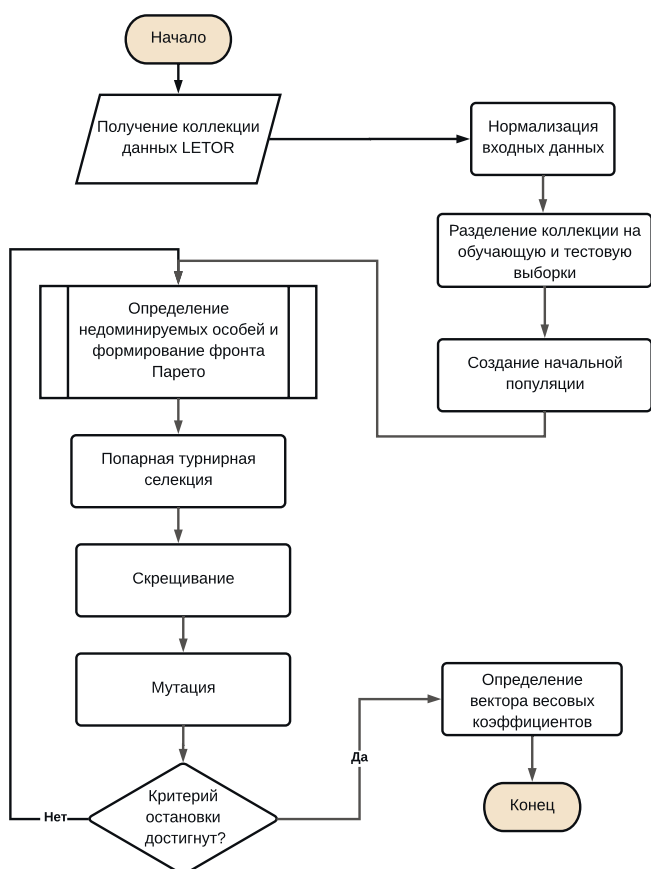


Рис. 1. Алгоритм метода $LTR-PGA$

Хромосома

0.916	0.325	0.08	0.743	0.112	0.259
W_1	W_2	W_3	W_4	W_5	W_6

Рис. 2. Пример хромосомы при вещественном кодировании весовых коэффициентов ранжирующей функции

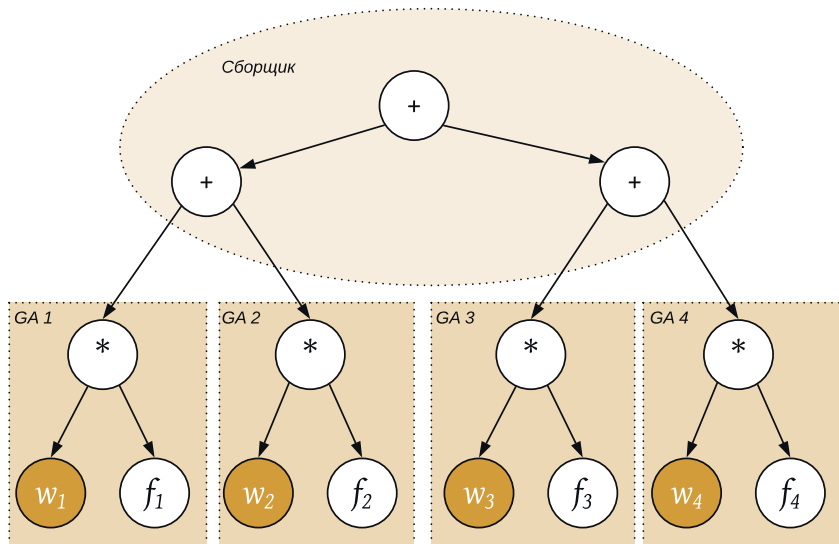


Рис. 3. Декомпозиция и сборка в методе LTR-CGA

ния, при котором отдельный ген представляет один из весовых коэффициентов ранжирующей функции в виде вещественного числа, а их совокупность составляет хромосому (закодированный вектор весовых коэффициентов). На рис. 2 показан пример формирования хромосомы особи с помощью вещественного кодирования шести весовых коэффициентов $W_1 - W_6$. Начальные значения весовых коэффициентов выбирались случайно из диапазона $W_i \in [0; 1], i = \overline{1...n}$ [2].

Применяемый для ранжирования метод недоминируемой сортировки *NDS* (*Non-Dominated Sorting algorithm*) [10], требует использования турнирного отбора особей в ходе выполнения ГА. Ранг особи $i = \overline{1...n}$ (i -го весового коэффициента) в его текущем состоянии W_i обозначим r_i . В соответствии с методом *NDS* для вычисления рангов особей реализован следующий алгоритм.

1. Среди всех особей популяции (агентов), представленных хромосомами закодированных весовых коэффициентов, выбираются недоминируемые. Им присваивается ранг, равный единице, и они исключаются из дальнейшего рассмотрения.

2. Данная процедура повторяется для оставшихся агентов до исчерпания популяции. Ранг особи легко преобразуется в функцию приспособленности, которая используется при селекции, вычисляемую по формуле:

$$f(W_i) = \frac{1}{1+r_i}, i = \overline{1...|S|}.$$

3. На последней итерации, среди особей финального поколения выявляется фронт Парето-опти-

мальных решений. Из точек на фронте с помощью привлечения дополнительного критерия V_{pref} выбирается оптимальная точка, а соответствующий ей вектор варьируемых коэффициентов W^* является решением оптимизационной задачи. Полученные весовые коэффициенты далее используются в ранжирующей функции.

Метод машинного обучения LTR-CGA

Второй предложенный метод машинного обучения *LTR-CGA* адаптирует алгоритм ККЭ к проблеме МО ранжированию.

Постановка задачи. Применение генетических алгоритмов для МО ранжированию делает возможным максимизацию дискретных метрик качества, таких как *NDCG* [10]. В таком случае основной задачей становится определение вектора весовых коэффициентов W функции ранжирования f , при котором показатель качества ранжирования *NDCG* достигает максимальных значений.

Таким образом, на основе выражений (3) и (4) для оптимизации (максимизации) метрики *NDCG*, целевая функция запишется в следующем виде:

$$f(W) = \left(\sum_{p=1}^n \frac{2^{\text{grade}(p)} - 1}{\log_2(2+p)} \right) Z^{-1} \rightarrow \max. \quad (6)$$

Представление решения. Предлагаемый метод машинного обучения *LTR-CGA* использует древовидные структуры данных для представления решений таким образом, что особи представлены в виде поддеревьев. Деревья не только имеют достаточную выразительную силу для представления нелинейных функций, но также имеют преимущество, заключающееся в простоте извлечения, внедрения и интерпретации данных.

Деревья формируются таким образом, что все их внутренние узлы представляют собой основные тематические операторы: сложение, вычитание и умножение. В то же время, листья дерева представлены факторами релевантности и соответствующими весовыми коэффициентами. В рассматриваемой задаче, весовые коэффициенты являются рациональными числами на интервале $[0; 1]$.

Глубина h дерева, определяющего полное решение, характеризуется числом (F_N) факторов релевантности используемых в функции ранжирования, а также числом (W_N) соответствующих весовых коэффициентов. Опытным путем установлено, что дерево должно быть достаточно глубоким, чтобы число конечных узлов было больше, чем $F_N + W_N$, то есть $h = \lceil \log_2(F_N + W_N) \rceil + 1$. Так как в рассматриваемой системе $F_N = 42$ и $W_N = 42$, то $h = \lceil \log_2(42 + 42) \rceil + 1 = 7 + 1 = 8$. Таким образом, дерево имеет глубину $h=8$ и $2^{8-1} = 128 \geq (42 + 42)$ листовых узлов.

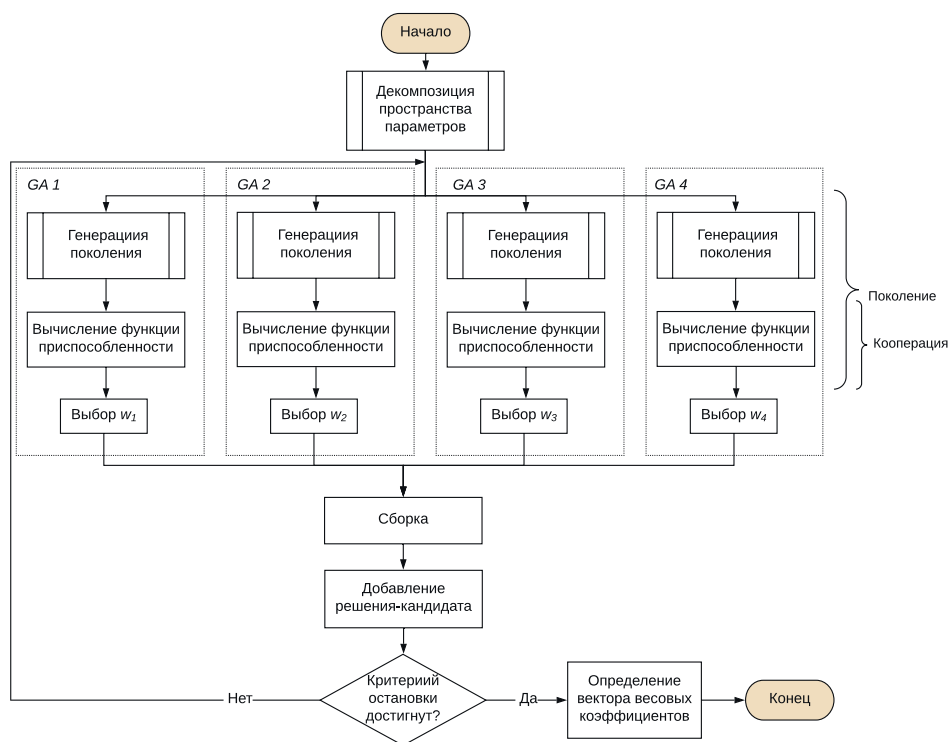


Рис. 4. Параллельные генетические алгоритмы в методе LTR-CGA

Этап декомпозиции. Для выполнения алгоритмов кооперативной коэволюции, производится декомпозиция общего пространства параметров на отдельные подпространства параметров, относящиеся к одной подпроблеме. Корректность подобного разделения обусловлена тем, что пространство поиска является декартовым произведением параметров.

Так как данный метод относится к классу стохастических методов оптимизации, начальные решения выбираются случайно на всем пространстве поиска. После этого, каждое дерево разбивается на N поддеревьев, каждое из которых формирует при этом независимую популяцию. Для каждой из этих популяций выделяется отдельный генетический алгоритм. На рис. 3 показана структура декомпозиции и сборка функции ранжирования в виде дерева для случая $F_N = 4$ и $W_N = 4$.

Глубина поддеревьев для генетических алгоритмов $GA1$, $GA2$, $GA3$ и $GA4$ ограничивает пределы пространства параметров особей. Глубина сборщика ($AsmD$) определяется выражением $AsmD = \lceil \log_2 P_N \rceil$, где P_N — число ядер процессора вычислительной машины. Предположим, что глубина деревьев решений $h = 8$. Пусть число ядер процессора $P_N = 8$. Тогда глубина дерева для сборщика $AsmD = \lceil \log_2 8 \rceil = 3$, следовательно можно вычислить глубину дерева для особей $IndD = 8 - 3 = 5$.

Этап эволюции. Эволюция в методе LTR-CGA выполняется итеративно, а число итераций предопределено установленным в начале параметром ГА. Все популяции эволюционируют параллельно, каждая популяция имеет свои независимые поколения осо-

бей, а также победителей (особей, имеющих наибольшие значения функции приспособленности в своем поколении).

На рис. 4 приведен алгоритм предлагаемого метода LTR-CGA. Эволюция происходит параллельно в каждой из четырех независимых популяций с помощью генетических алгоритмов $GA1$, $GA2$, $GA3$ и $GA4$ соответственно. На каждой итерации происходит формирование новых особей на основе предыдущего поколения. Далее в каждом из генетических алгоритмов для всех особей производится вычисление значений функции приспособленности, представленной соответствующей подпроблемой.

Далее, на следующем шаге, на основе вычисленных значений функции приспособленности в каждой из популяций выбирается лучшая особь (победитель), после чего параллельная работа алгоритма приостанавливается. При помощи процесса сборки, объединяются победители каждой из параллельных популяций, формируя при этом общее решение-кандидат. Если критерий остановки (ограничение на число итераций) не достигнут, начинается новая итерация и четыре популяции продолжают эволюцию в новом поколении параллельно.

Этап сборки и вычисление функции приспособленности. В предложенном подходе сборка производится в двух случаях: при вычислении функции приспособленности (в конце работы параллельных генетических алгоритмов $GA1$, $GA2$, $GA3$, $GA4$, как показано на рис. 4) для определения победителей, а также при генерации решений-кандидатов. Процесс сборки является процессом, обратным декомпозиции. На рис. 4 показано объединение четырех особей в общее решение-кандидат с помощью сборщика. За счет подобного подхода к вычислению функции приспособленности, в алгоритмах кооперативной коэволюции приспособленность особи определяется тем, как хорошо она взаимодействует с особями из остальных популяций.

Результаты исследований и их обсуждение

При выполнении исследования проведены две серии экспериментов с использованием коллекций тестовых данных LETOR 4.0 для оценки точности и скорости работы метода LTR-CGA.

Точность ранжирования. В первой серии экспериментов оценивалось качество ранжирования, достигаемое с помощью алгоритма LTR-CGA,

Таблица. Результаты сравнения методов машинного обучения ранжированию по метрикам качества ранжирования на тестовой коллекции LETOR

Название алгоритма	Метрика качества ранжирования			
	Точность	<i>Bpref</i>	Полнота	<i>NDCG</i>
<i>LTR-PGA</i>	0,401	0,388	0,545	0,418
<i>LTR-CGA</i>	0,392	0,383	0,531	0,395
<i>rankSVM</i>	0,335	0,356	0,487	0,305
<i>listNet</i>	0,342	0,371	0,463	0,341
<i>AdaRank</i>	0,32	0,273	0,346	0,41
<i>RankBoost</i>	0,294	0,248	0,271	0,263
<i>FRank</i>	0,387	0,391	0,412	0,326

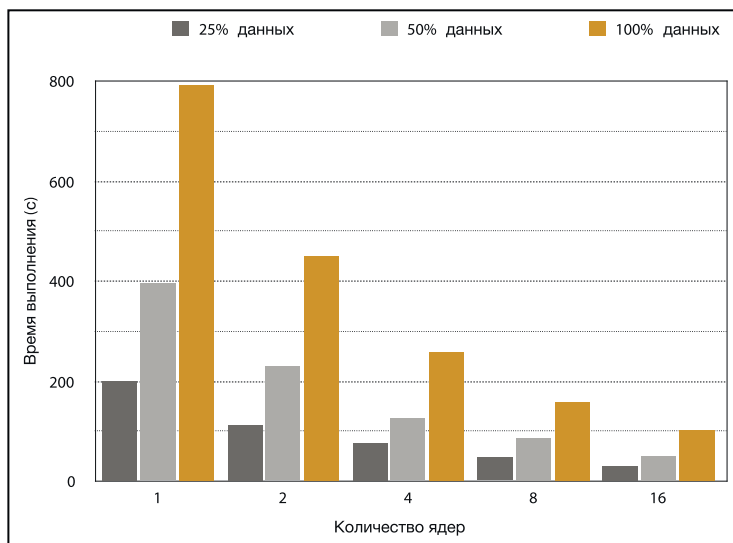


Рис. 5. Зависимость времени выполнения машинного обучения метода *LTR-CGA* от числа ядер процессора для разных объемов обучающей выборки

в сравнении с алгоритмами МО *AdaRank*, *FRank*, *RankBoost*, *RankSVM* и *ListNet* на тестовых коллекциях *MQ2007* и *MQ2008*.

В таблице показан результат сравнения методов по метрикам *точности*, *Bpref*, *полноты* и *NDCG*. Оба алгоритма — *LTR-CGA* и *LTR-PGA* показывают более высокие значения метрик качества ранжирования, чем аналоги, что позволяет сделать вывод об их эффективности. Следует отметить, что в данной серии экспериментов метод *LTR-PGA* превзошел метод *LTR-CGA*. Это может быть вызвано тем, что метод *LTR-PGA* за счет применения метода Парето-оптимизации учитывает дополнительные метрики качества информационного поиска, такие как *NDCG*, *MAP* и *Bpref*, что позволяет более эффективно выполнять машинное обучение.

Время выполнения алгоритма. Для демонстрации повышения скорости выполнения МО за счет параллельной эволюции из тестовой коллекции *MQ2008* были последовательно извлечены 25%, 50% и 100% данных, на основе которых были подготовлены три обучающие выборки, содержащие 3803, 7606 и 15211 документов, соответственно. Затем, ал-

горитм *LTR-CGA* выполнялся для каждого из этих наборов с разным числом процессоров (1, 2, 4, 8, и 16).

Рис. 5 иллюстрирует, как время выполнения алгоритма *LTR-CGA* зависит от числа ядер процессора. Видно, что параллельная эволюция приводит к значительному уменьшению временных затрат на выполнение генетических алгоритмов, то есть и к приросту скорости машинного обучения при использовании метода *LTR-CGA*. По сравнению с выполнением алгоритма при использовании одного процессора средний прирост скорости составил 173%, 299%, 486% и 736% для 2, 4, 8, и 16 ядер процессора соответственно.

Следует отметить, что в идеальной ситуации с увеличением размера набора данных кривая повышения скорости выполнения алгоритма должна приблизиться к линейному росту. Однако программная реализация метода *LTR-CGA* испытывает трудности в достижении подобных результатов по следующим причинам. Во-первых, как показано на рис. 4, операции алгоритма *LTR-CGA* не всегда выполняются параллельно. После каждого поколения параллельное выполнение алгоритма временно приостанавливается для выполнения процедуры сборки окончательного решения. Во-вторых, ГА может потратить разное время на эволюцию в каждой из параллельных популяций, но при этом сборка может начаться только после того, как итерация завершена во всех алгоритмов. Таким образом, время, которое алгоритм *LTR-CGA* тратит на поколение, не меньше наибольшей длительности времени, затраченного на эволюцию среди всех параллельных генетических алгоритмов.

Заключение

Результаты исследований подтвердили, что предложенные методы эффективны для задачи машинного обучения ранжированию ИПС. Предложенный метод МО ранжированию *LTR-PGA* для многокритериальной оптимизации дискретных метрик качества информационного поиска позволяет повысить качество ранжирования документов поисковой системой в сравнении с существующими аналогами [12]. Кроме того, показана эффективность метода *LTR-CGA*, применяющего механизмы параллельного выполнения кооперативной эволюции генетических алгоритмов, позволяющего повысить скорость выполнения информационного поиска без потерь в точности ранжирования.

Проведенные испытания на тестовых коллекциях продемонстрировали значительное повышение скорости выполнения машинного обучения ранжированию алгоритмом *LTR-CGA* по сравнению с эталонными алгоритмами за счет использования параллельных вычислений. Это позволяет надеяться, что применение данного метода обеспечит новый уровень в работе с Big Data применительно к машинному обучению ранжированию.

Умение ставить разумные вопросы уже есть важный и необходимый признак ума и проницательности.

Иммануил Кант

Следует отметить, что эффективность предлагаемых методов проявляется в полной мере для информационно-поисковых систем, не только имеющих дело с большим объемом обучающей выборки, но также работающих в условиях необходимости частого проведения повторного обучения. В свою очередь, повторное переобучение, выполненное с высокой скоростью, позволит системе ранжирования адаптироваться к изменяющейся структуре документов, а также бороться с искусственным повышением ранга нерелевантных документов.

Список литературы

1. Liu T.Y. Learning to Rank for Information Retrieval // Journal Foundations and Trends in Information Retrieval. 2009. March. Vol. 3, issue 3. P. 225-331.
2. Холланд Дж. Генетические алгоритмы // В мире науки. 1992. № 9. С. 32-40.
3. Noghin V. Reducing the Pareto Set Based on Set-point Information / Scientific and Technical Information Processing 38 (6), 2011. pp. 435-439.
4. Семенихин С.В., Денисова Л.А. Автоматизация информационного поиска на базе многокритериальной оптимизации и генетических алгоритмов, Динамика систем, механизмов и машин, 2014. № 3. С. 224-227.
5. Денисова Л.А. Многокритериальная оптимизация на основе генетических алгоритмов при синтезе систем управления. Омск: Изд. ОмГТУ, 2014. 172 с.
6. Collins R. and Jeerson D. Selection in massively parallel genetic algorithms // Proceedings of the Fourth Intl. Conf. on Genetic Algorithms, ICGA-91. Morgan Kaufmann, 1991. P. 249-256.
7. Liu Tie-Yan, Xu Jun, Qin Tao, Xiong Wenyong, and Li Hang. LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval // SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (LR4IR 2007), 2007. P. 3-10.
8. Grineva M., Grinev M., Lizorkin D., Boldakov A., Turdakov D., Sysoev A., Kiyko A. // WWW '11 Proceedings of the 20th international conference companion on World wide web, 2011. P. 213-216.
9. Семенихин С.В., Денисова Л.А. Метод машинного обучения ранжированию на основе модифицированного генетического алгоритма для метрики NDCG // Омский научный вестник. 2016. № 3 (147). С. 100-104.
10. Burges C., Shaked T., Erin R. Learning to Rank using Gradient Descent // Proceeding ICML '05 Proceedings of the 22nd international conference on Machine learning. 2005. P. 89-96.

Семенихин Святослав Витальевич — аспирант, Денисова Людмила Альбертовна — д-р техн. наук, проф. Омского государственного технического университета. E-mail: svsemenihin@gmail.com, denisova@asoju.com Контактный телефон (381-2) 65-20-84.

ПРЕИМУЩЕСТВА INDUSTRY 4.0 В СТАНКОСТРОЕНИИ

DMG MORI

Компания DMG MORI — родоначальник внедрения цифровых технологий в станкостроение понимает, что станки, оснащенные системами автоматизации, будут отличительной чертой «цифрового завода» будущего и, как следствие, неотъемлемой частью Industry 4.0. Системы ЧПУ должны быть интегрированы в производственную среду. В этой ситуации операторы должны только контролировать и отслеживать операции, пока станки самостоятельно выполняют задания в соответствии с заданной последовательностью. В связи с этим компания DMG MORI AKTIENGESELLSCHAFT объединила свои усилия с производителем систем автоматизации — компанией HEITEC AG, в результате чего было основано совместное предприятие DMG MORI HEITEC.

Теперь на станкостроительных заводах появились специалисты в области решений по автоматизации, которые отвечают за проект заказчика от начала и до конца. Все время после того, как заказчик сделал запрос, специалисты по автоматизации создают специализированное решение под требования заказчика со своей командой экспертов и координируют все последующие шаги, являясь единым контактным лицом для заказчика. Комплексная поддержка, оказываемая в ходе процесса приобретения решения по автоматизации, охватывает не только подбор нужного оборудования и его ввод в эксплуатацию, но и вопросы, относящиеся к 24-месячному гарантийному и сервисному обслуживанию.

С целью снижения затрат на этапах проектирования индивидуальных и серийных решений и одновременного повышения производительности компания DMG MORI HEITEC GmbH поставила перед собой цель разработки «цифровых двойников» — виртуальных моде-



лей станков, установок, роботов и маршрутов движения материалов. С помощью цифрового инжиниринга производства время выполнения проекта может быть сокращено примерно на 20%, а ввод оборудования в эксплуатацию на объекте может быть сокращен еще больше — до 80%. Будущие дополнения или изменения можно спланировать и протестировать с помощью цифрового двойника, работающего параллельно с оборудованием в режиме реального времени.

На сегодняшний день линейка продукции компании включает такие давно зарекомендовавшие себя системы автоматизации, как:

- система PH 150 для загрузки-выгрузки паллет. В число станков, интегрируемых с данной системой, входит 5-осевой фрезерный станок DMU 50, выпускаемый в России на заводе в Ульяновске (рисунки);

- круговой накопитель RS 3 и система загрузки-выгрузки деталей WH;

- порталный загрузчик GX в новом дизайне VERTICO. Свое название новый дизайн VERTICO от DMG MORI получил благодаря вертикально расположенным перфорированным стенкам ограждения. Перфорированные листы с отверстиями треугольной формы используются в защитном ограждении всех систем автоматизации и подходят как для моделей премиум-дизайна, так и для моделей дизайна Stealth.

В будущем благодаря модульному принципу комплектации компания DMG MORI HEITEC будет поставлять заказчикам интегрированные решения по автоматизации.

[Http://www.dmgmori.com](http://www.dmgmori.com)