



Мохамад Али, О.Ф. Михаль

ХНУРЕ, г. Харьков, Украина, fuzzy16@pisem.net

ЛОКАЛЬНО-ПАРАЛЛЕЛЬНАЯ ПРОЦЕДУРА ОБУЧЕНИЯ С ТЕКУЩЕЙ КОРРЕКТИРОВКОЙ РАНГОВ ИЗУЧЕННОСТИ И ПЕРЕСОРТИРОВКОЙ УЧЕБНОЙ ВЫБОРКИ

Рассмотрено применение локально-параллельной сортировки для организации алгоритма обучения с повторением и контролем полученных знаний. Предложен оптимальный вариант сегментной разбивки регистров представления данных. Описан алгоритм обучения, не предполагающий вспомогательных действий учащегося, помимо основного потока обучения. Область применения – разработка тренажеров для активизации словарного запаса при компьютеризированном обучении иностранным языкам.

ЛОКАЛЬНО-ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА, СОРТИРОВКА ДАННЫХ, КОМПЬЮТЕРНОЕ ОБУЧЕНИЕ

Введение

Текущий этап развития человеческой цивилизации характеризуется качественными изменениями в динамике лавинообразного нарастания накопления научной информации. До середины XX века нарастание шло в основном за счёт ресурса *человеческого интеллекта* (ЧИ) пропорционально росту численности исследователей. Рубеж тысячелетий явственно обозначил угрозу экологического кризиса, опасность исчерпания биологических ресурсов планеты и, как следствие, излом на кривой демографического роста и, в конечном счёте, - невозможность дальнейшего *экстенсивного* использования ресурса ЧИ [1]. Фактором *интенсификации* явилось расширенное введение *усилителей человеческого интеллекта* (УЧИ), какowymi на текущем этапе развития являются средства *вычислительной техники* (ВТ) [1]. ЧИ и УЧИ взаимно дополняют и взаимно влияют друг на друга. УЧИ изначально проявляются и развиваются как модели ЧИ, но когда они воспроизводят «лучшие образцы» и «наиболее эффективные аспекты функционирования» ЧИ, использование УЧИ проявляется как фактор реорганизации индивидуальной творческой человеческой деятельности [2]. По существу это является причиной и основанием расширенного использования средств ВТ в учебном процессе. Цель настоящей работы – демонстрация в указанном контексте одного из аспектов использования средств ВТ: применения *локально-параллельной* (ЛП) сортировки [3] для организации адаптивного алгоритма обучения с повторением и контролем полученных знаний.

1. Упорядоченность объектов

Сортировка объектов есть процесс организации их упорядоченности по некоторым признакам. В естественных (природных) и искусственных (технических) системах – сортировка есть всеобщее групповое массовое явление, результаты которого наблюдаются как крупномасштабное проявление упорядоченности *окружающего мира* (ОМ).

В мелком (ограниченном) масштабе времени и пространства сортировка предстаёт преимущественно как последовательность отдельных актов взаимодействия (сравнения) и изменения взаимного расположения ограниченного количества объектов. Это различие при разных масштабах рассмотрения качественно (т.е. в меру ограниченного человеческого понимания явлений ОМ) может быть трактовано как сохранение ближней и дальней упорядоченности – фундаментальное соотношение между близкодействием и дальнодействием. Система в укрупнённом пространственном и временном масштабе выглядит иначе, чем её малая подсистема, взятая в мелком (подробном) масштабе. Различие проявляется как в характере и общей направленности событий, так и в целях и решаемых задачах.

Подобные проявления (наличие ближнего и дальнего планов и, соответственно, ближнего и дальнего уровней упорядоченности) повидимому могут быть ретроспективно усмотрены так же в самом ЧИ. По существу – это естественно, поскольку ключевая функция ЧИ – отражение ОМ. Для этого ЧИ сам должен быть организован *подобно* (в некотором смысле) ОМ. Субъект постоянно получает информацию об ОМ и в большей или меньшей мере изменяет (переупорядочивает) отдельные элементы образа ОМ в своём сознании в соответствии со степенью новизны и актуальности поступающей информации. Поэтому, в частности в процессе обучения, при освоении нового учебного материала степень освоенности отдельных *элементов* (единиц) *изучаемого материала* (ЭИМ) должна быть соотнесена с уровнем текущей индивидуальной упорядоченности и с порядком (последовательностью) подачи нового материала (последующих ЭИМ). Как следствие, эффективный процесс обучения (освоения нового материала) целесообразно строить с учётом *индивидуального* (личностного) изменения упорядоченности ЭИМ по мере освоения. В идеале (сложно достижимом на практике) процесс обучения должен управляться самим

субъектом, т.е. стать *самообучением*. Ключевыми (взаимосвязанными и взаимодополняющими) моментами в этом являются мотивация (самотивация) и адаптивная организация «обучающей среды». Последнее – есть совокупности средств и методов (алгоритмов обучения), ориентированных на организацию процесса подачи ЭИМ.

Показательным в этом отношении является использование методов ЛП сортировки малых наборов данных [3] при построении систем интерактивного взаимодействия ЧИ со средствами ВТ, рассматриваемое далее на примере компьютерной системы обучения.

2. Оптимальный выбор параметров

Алгоритмы ЛП обработки данных первоначально были разработаны для операций нечёткой логики [4]. Далее принципы ЛП обработки были распространены на сортировку данных [5]. Применительно к *малым наборам* данных вопросы сортировки рассмотрены в [3]. Там же отмечено, что ЛП сортировка максимально эффективна при обработке выборок в пределах разрядности процессора. Конкретизируем данный аспект.

На текущий момент реально распространены средства ВТ с процессорами с разрядностью 32 и 64 бит. Для ЛП представления информации регистр разбивается на непересекающиеся, плотно примыкающие друг к другу сегменты одинаковой длины N [4]. Произведение длины сегмента N на число целых сегментов S_{32} (соответственно S_{64}), поместившихся в регистр разрядностью 32 (соответственно 64) бита, не может превышать разрядность регистра: $N * S_{32} \leq 32$; $N * S_{64} \leq 64$. С учётом этого в таблице представлены предельные объёмы ЛП заполнения регистров. M – есть число разных возможных значений, которые могут быть записаны в сегменте длиной N , т.е. $M = 2^N$; k_{32} и k_{64} – коэффициенты полноты использования регистра – отношение числа возможных разных значений M к числу целых сегментов S_{32} или S_{64} , т.е. $k_{32} = M / S_{32}$, $k_{64} = M / S_{64}$.

Таблица

Пределы ЛП-заполнения регистров

N	M	S ₃₂	k ₃₂	S ₆₄	k ₆₄
2	4	16	0,25	32	0,125
3	8	10	0,8	21	0,375
4	16	8	2	16	1
5	32	6	5,33	12	2,66
6	64	5	12,8	10	6,4

Пример интерпретации: при $N = 5$ в каждом из сегментов может быть записано в битовой форме одно из 32 разных значений: от $0_{10} = 00000_2$, до $31_{10} = 11111_2$ (2 и 10 в индексе означают основание системы счисления). Как следует из

представленного, единственный вариант – 4-битный сегмент при 64-разрядном процессоре – является таким, что число возможных *разных* записей в сегменте равно числу сегментов, помещающихся в регистр. Это значит, что в одном регистре может быть охвачено в точности столько записей о разных ЭИМ, сколько имеется чисел соответствующей длины в двоичном представлении для указания номеров ЭИМ. Т.е. содержимое каждого сегмента может при этом быть интерпретировано как номер соответствующего ЭИМ. Это значит, что в 64-разрядном регистре в 16 4-битовых сегментах может быть записано в точности 16 *разных* чисел из набора от 0 до 15, которые далее могут быть отсортированы ЛП алгоритмом [3]. Таким образом, в смысле полноты использования наличных комбинаций, данный вариант является *оптимальным*. Остальные варианты, представленные в таблице – двух типов:

– вариантов заполнения *меньше*, чем сегментов в регистре (пример: $N = 3$, $M = 8$, $S_{32} = 10$, т.е. $M < S_{32}$), что соответствует заполнению регистра повторяющимися числами и неполному использованию потенциальных возможностей ЛП сортировки;

– вариантов заполнения *больше*, чем сегментов в регистре (пример: $N = 5$, $M = 32$, $S_{64} = 12$, т.е. $M > S_{64}$), что соответствует невозможности помещения полного набора *разных* чисел в один регистр, необходимости использования для этого нескольких регистров и, опять же, не максимально эффективному использованию возможностей ЛП сортировки.

Из числа неоптимальных вариантов, наиболее близким к оптимальному является $N = 3$, $M = 8$, при котором $k_{32} = 0,8$. Его целесообразно избрать при использовании 32-разрядных средств ВТ.

3. Алгоритмизация процесса обучения

Применение средств ВТ в обучении целесообразно и максимально эффективно преимущественно при изучении таких дисциплин, в которых обязательным является полное и безошибочное запоминание ЭИМ, содержащих определённую фиксированную информацию. Показательной в этом отношении является задача освоения нового *лингвистического материала* (ЛМ). Для её решения требуется плотный текущий контроль и многократное повторение, что хорошо автоматизируется средствами ВТ. Напротив, если компьютеризирован некоторый описательный учебный курс (последовательно упорядоченное обзорное изложение материала), то в методологическом плане это не слишком отличается от использования традиционного печатного учебника. Разумеется, возможен учёт времени изучения определённых фрагментов, активизация внимания учащегося, (например

мультимедийными эффектами), принудительное введение дисциплины (дозирование подачи ЭИМ, чередование активных фаз и перерывов) и т.д. Но всё это является скорее компьютеризированными организационными мерами, чем собственно компьютерным обучением. Ключевое различие состоит в наличии и качестве обратной связи (степени адаптивности, интеллектуальности), индивидуализирующей (персонализирующей) процесс подачи ЭИМ.

Интересны обучающие программы – тренажёры по освоению ЛМ, реализующие процедуры с повторением и текущим контролем, типа иллюстрируемой рис. 1.



Рис. 1. Обучение с пошаговой корректировкой ранга вопросов и пересортировкой

Тренируемой областью может быть, в частности, активизация словарного запаса при изучении иностранных языков. Тогда ЭИМ есть пары изучаемых слов. Большинство словарных тренажёров работают по фиксированной схеме:

- выбор из меню изучаемой смысловой темы;
- фиксированный набор слов из этой темы;
- циклическая подача слов (в фиксированном порядке или со случайным выбором), запрос и ввод ответа (альтернативный (типа рис. 2 а) или текстовый (типа рис. 2 б)), контроль правильности;

– предложение удалить элемент из списка (пометить слово как изученное), если учащийся считает, что он его уже выучил;

– по завершении требуемого количества циклов – оценка работы учащегося в баллах или в виде процента правильных ответов по всей теме, или по каждому элементу (слову) в отдельности.

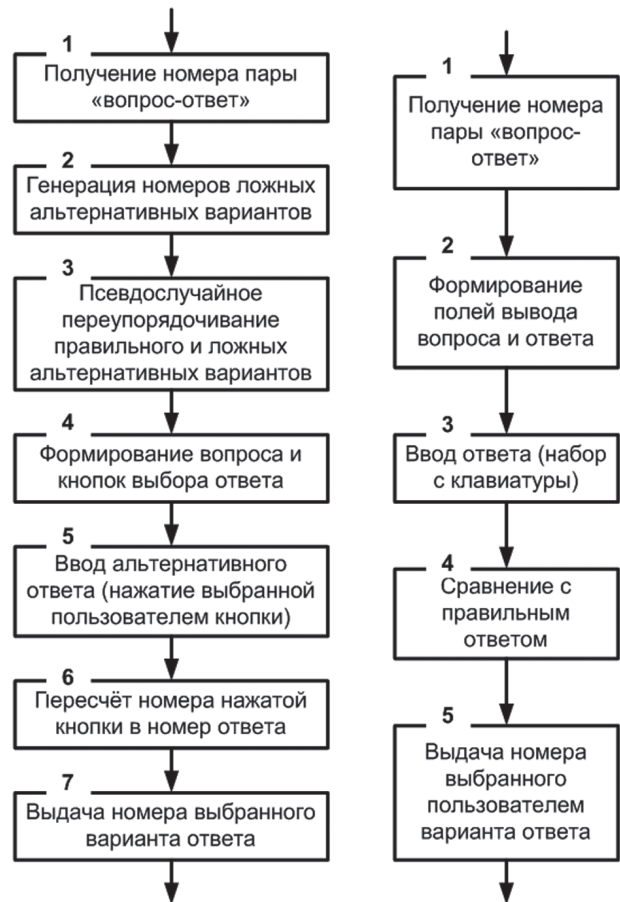


Рис. 2. Предъявление очередного вопроса с выбором альтернативного варианта (а) и с непосредственным текстовым вводом ответа (б)

Не оспаривая в целом эффективности этой схемы, отметим имеющееся в ней ограничение – п. 4: чтобы «отделаться» от надоедливого известного слова (освоенного ЭИМ), учащийся должен отвлечься (выйти на время из потока обучения), оценить («максимально объективно» и «независимо») степень изученности и принять волевое решение. Эти действия качественно отличаются от собственно освоения ЭИМ (процесса запоминания), что вовсе не способствует повышению эффективности. Между тем, освоенные ЭИМ могут быть удалены из предъявляемой последовательности (убраны совсем, либо перенесены в выборку для контрольного повторения – в зависимости от методики) автоматически, на основании текущего (происходящего непосредственно в сеансе обучения) диалога с учащимся. Для этого должны быть сделаны (обеспечены алгоритмически) следующие четыре вещи.

1. В процессе предъявления обучающей выборки ЭИМ после каждого предъявления и получения ответа должен быть пересчитан индивидуальный *ранг изученности* ЭИМ — показатель частоты, с которой этот элемент (слово) должен впредь выбираться и предъявляться учащемуся.

2. После каждого изменения *ранга* элементы в выборке должны быть пересортированы в порядке убывания частоты предъявления. Самые «низкочастотные» (уже хорошо освоенные) ЭИМ должны оказаться «в хвосте», из которого выборка и предъявление маловероятны. «Высокочастотные» (плохо изученные, подлежащие частому предъявлению) ЭИМ должны оказаться «в голове» выборки.

3. Выбор и предъявление очередного ЭИМ осуществляется с использованием генератора случайных чисел с отчётливо нелинейным монотонным профилем закона распределения: максимум «в голове» выборки, минимум — «в хвосте».

4. Блокируется возможность выборки и предъявления ЭИМ, который предъявлялся непосредственно перед этим. Подобное дискомфортное двукратное предъявление — негативно влияет на мотивацию.

Блок-схема, представленная на рис. 1, укрупнённо иллюстрирует описанную обучающую процедуру. На рис. 2 а, б показаны два варианта алгоритмической реализации блока (14) (рис. 1). Нумерация блоков — по столбцам. В алгоритме рис. 1 реализован цикл с предъявлением ЭИМ (4), сортировкой по степени изученности (13) и блокировкой повторений (7). Выход из цикла (9) выполняется инициативно самим учащимся (пользователем программы), либо программой по достижении порогового уровня изученности материала. Формирование очередного вопроса (14) и, соответственно, предъявление ЭИМ (4) и ввод ответа (10), могут осуществляться двумя способами: выбором альтернативного варианта (рис. 2 а) и непосредственным текстовым вводом ответа (рис. 2 б). Вариант на рис. 2 б, по сравнению с вариантом на рис. 2 а, короче на два блока. В методологическом плане данные способы являются взаимно дополнительными, поскольку активизируют разные фазы процесса освоения ЭИМ учащимся. Поэтому в реальном тренажёре должны присутствовать оба режима.

4. Структура представления данных

Как отмечалось, оптимальным является вариант ЛП представления с 4-битовыми сегментами на 64-разрядном процессоре, когда полный набор чисел от 0_{10} до 15_{10} может быть помещён в одном регистре и эффективно отсортирован ЛП алгоритмом [3]. Приведённое соображение целесообразно в качестве исходного при определении структуры организации данных для хранения информации в процессе обучения.

Представляется удобным сгруппировать материал в списки по 16 ЭИМ и обрабатывать учебный процесс списками. В варианте тренажёра по изучению иностранных слов — это таблица с 3 полями: номер строки (от 0 до 15), слово на иностранном языке, перевод. Для реализации самого процесса обучения согласно алгоритму (рис. 1) требуется два рабочих регистра: *регистр рангов* (РР) для хранения числовых значений, соответствующих рангам изученности (освоенности) ЭИМ (слов); и *регистр адресации* (РА), в котором в соответствующих сегментах содержится адресация ЭИМ. Требуются также вспомогательные числа — маски для работы с сегментами регистров РР и РА, как описанные в [3]. Регистр РР изначально заполнен нулями; в регистре РА — хранятся числа от 0_{10} до 15_{10} — адреса слов в фиксированной 16-строчной таблице.

В процессе работы (в ходе процесса освоения материала, в диалоге с учащимся) ранги изученности отдельных ЭИМ изменяются, РР сортируется в соответствии с текущими значениями рангов, элементы в РА переставляются вслед за сортировкой элементов РР. При этом всякий раз при прерывании процедуры обучения (инициативно — учащимся, или при достижении пороговых значений — программой) имеется «текущая картина результатов»: хорошо освоенный материал «в хвосте» регистра и материал «требующий доучивания» — «в голове» регистра. Всё это — с адресацией конкретных ЭИМ — изучаемых пар слов.

5. Программная реализация

Процедура обучения (рис. 1 и 2) по теперешним представлениям не слишком сложна в реализации. Тем не менее, подобный подход не был реализован в традиционном обучающем программном обеспечении (с самого начала, 20–25 лет назад, на заре развития компьютерных обучающих систем) в силу ограниченности ресурсов тогдашних средств ВТ. Пересортировка списка предъявляемых слов после каждого ответа учащегося на прежнем (старом) аппаратном обеспечении приводила бы к задержкам, совершенно не комфортным для учащегося. Процедура становится практически реализуемой (приемлемой по времени) только на теперешнем высокопроизводительном оборудовании, с применением низкоресурсных ЛП методов обработки [2, 3].

Идея, реализованная в блоке (14) (рис. 1), в целом соответствует описанному выше: генератор случайных чисел с существенно нелинейным монотонным профилем закона распределения. Нелинейность профиля обеспечивает селекцию и преимущественное предъявление учащемуся наименее изученного им материала. Отдельного обсуждения требует согласованная ЛП сортировка пары регистров — блок (13), рис. 1. Детализация

дана на рис. 3. Блоки (7) и (13), рис. 3 – однотипные и детализированы блок-схемой рис. 4.

Имеется пара регистров: РР, в сегментах которого содержатся числовые значения, соответствующие рангам изученности (освоенности) ЭИМ (слов); и РА, в котором в соответствующих сегментах содержится адресация ЭИМ. Суть согласованной ЛП сортировки, как отмечалось, состоит в том, что непосредственно сортируются (переупорядочиваются в соответствии с текущими значениями) только сегменты РР, а соответствующие им сегменты РА тасуются (переставляются) вместе с ними и вслед за ними. Таким образом, всякий раз имеется отсортированный набор адресов ЭИМ (слов).

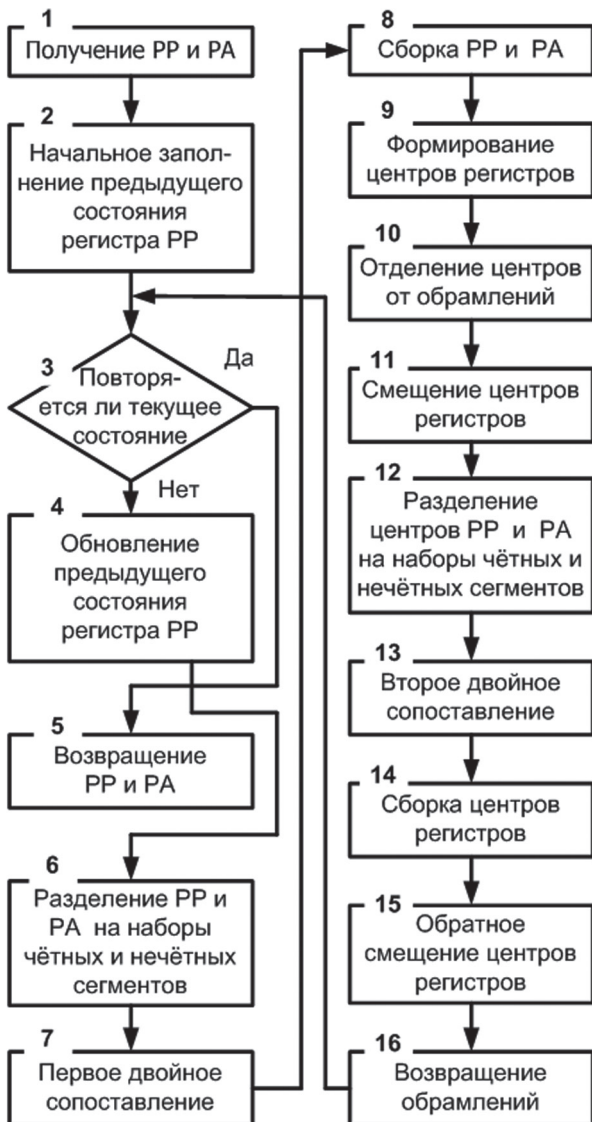


Рис. 3. Локально-параллельная сортировка сегментов пары регистров

Имеется особенность в реализации самого ЛП алгоритма сортировки регистра РР. Как отмечалось в [5], логика алгоритма ЛП-сортировки единообразно и корректно работает применительно к средней части регистра. Здесь всегда без проблем

реализуется разбивка на чётные и нечётные наборы сегментов, с применением масок – констант с чередующейся раскладкой нулевых и единичных сегментов, вида:

... (000...0) (111...1) (000...0) (111...1) (000...0) ...;
 ... (111...1) (000...0) (111...1) (000...0) (111...1) ...,

а также сравнением и селекцией большего и меньшего значений (содержимых сравниваемых сегментов) ЛП алгоритмами нечётких операций *min* и *max* [1]. Единообразие нарушается на краях регистра, в связи с чем в [5] в качестве масок использовались специально подобранные константы.

В рассматриваемом варианте применён иной, более единообразный подход. Раздельно обрабатываются центр регистра (всё кроме старшего и младшего сегментов) и обрамление (крайние сегменты). Центр вырезается (отделяется от обрамления (9), (10), рис. 3) и смещается на один сегмент в сторону младших разрядов (11). Таким образом, возникает ситуация, единообразно обрабатываемая при любых ЛП конфигурациях. После обработки (12) - (14), центр возвращается в прежнее положение (сдвигается на один сегмент в сторону старших разрядов (15)) и объединяется с обрамлением (16). Применённый подход единообразен относительно параметров ЛП процедуры: размера и чётного или нечётного количества сегментов.

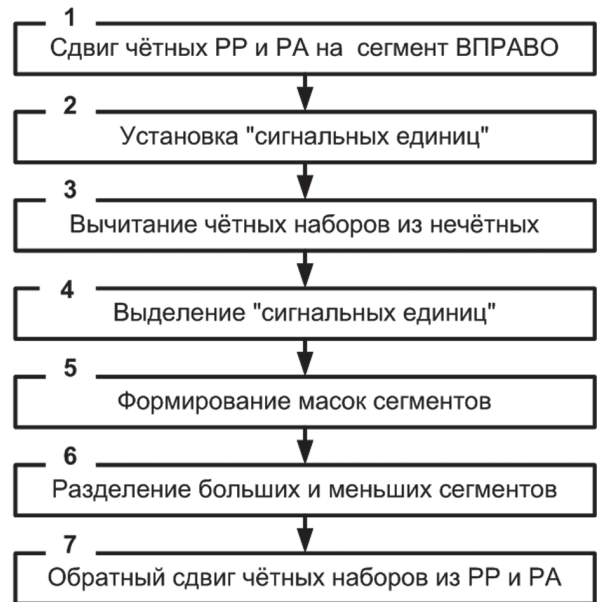


Рис. 4. Согласованное сопоставление регистров

Процедура сопоставления рис. 4 применена дважды ((7) и (13), рис. 3). Она построена на ЛП операциях *min* и *max*, с применением «сигнальных единиц» и формированием сегментных масок, как это было сделано в [1] для нечётких теоретико-множественных операция пересечения и объединения. Указанная экономия кода оказалась возможной за счёт применения описанного подхода с формированием центра и обрамления регистра.

Модель реализована в демонстрационном варианте. Ввиду перспективности последующего сетевого использования тренажёра в качестве языка реализации выбран JavaScript. Благодаря этому, демонстрационная версия может быть преобразована в WEB-приложение, основанное на PHP, без дополнительных постановочных алгоритмических решений. Для простоты демонстрации и снижения сложности отладки модель реализована применительно к 32-разрядной технике. Как отмечалось, для $N = 4$ этот вариант близок к оптимальному. Алгоритмических сложностей при переходе на 64 разряда (и далее по мере развития перспективных средств ВТ) – не предвидится, в частности, ввиду применённого подхода с разделением центра и обрамления.

Выводы

Применительно к задаче освоение лингвистического материала сформулирован принцип оптимальности при локально-параллельной организации данных. Рассмотрены оптимальная 64-разрядная и близкая к оптимальной 32-разрядная структуры представления данных. Предложенный вариант реализации алгоритма обучения с использованием локально-параллельной пересортировки выборки по мере изучения, не предполагает вспомогательных действий учащегося, помимо основного потока обучения. В рамках алгоритмизации процесса локально-параллельной сортировки сформулирован подход с разделением центральной части и обрамления регистра. Программно реализован 32-разрядный демонстрационный вариант на языке JavaScript, с ориентацией на последующее сетевое применение.

Список литературы: 1. Михаль О.Ф. Глобальный системный контекст развития ЭВМ / О.Ф. Михаль // Информатика, математическое моделирование, экономика: Сборник научных статей. Т. 1 – Смоленск: Смоленский филиал АНО ВПО ЦС РФ «Российский университет кооперации», 2012. – С. 38-47. 2. Михаль О.Ф. Информационный аспект организации индивидуальной творческой человеческой деятельности / О.Ф. Михаль // Информатика, математическое моделирование, экономика: Сборник на-

учных статей. Т. 2. – Смоленск: Смоленский филиал АНО ВПО ЦС РФ «Российский университет кооперации», 2013. – С. 81-88. 3. Мохамад Али, Михаль О.Ф. Малые наборы данных. Локально-параллельная сортировка / Мохамад Али, О.Ф. Михаль // Информатика, математическое моделирование, экономика: Сб. научных статей. Т. 1. – Смоленск: Смоленский филиал АНО ВПО ЦС РФ «Российский университет кооперации», 2012. – С. 64-67. 4. Михаль О.Ф., Руденко О.Г. Принципы организации систем нечеткого регулирования на однородных локально-параллельных алгоритмах / О.Ф. Михаль, О.Г. Руденко // Управляющие системы и машины. – 2001. – № 3. – С. 3-10. 5. Михаль О.Ф. Локально-параллельные алгоритмы определения степени включения и степени равенства нечетких множеств / О.Ф. Михаль // Проблемы бионики. Всеукраинский межведомственный научно-технический сборник. – Харьков: ХНУРЭ, 2001. – Вып. 55. – С. 80-90.

Поступила в редколлегию 04.04.2013

УДК 681.513

Локально-паралельна процедура навчання з поточним корегуванням рангів вивченості та пересортуванням навчальної вибірки / Мохамад Алі, О.П. Міхаль // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 2 (81). – С. 75-80.

Для вирішення задачі освоєння лінгвістичного матеріалу сформульовано принцип оптимальності в організації алгоритму навчання з локально-паралельним сортуванням даних. Запропоновано варіант алгоритму з вилученням будь-яких допоміжних дій учня, крім основного навчального потоку. Реалізовано 32-бітову демонстраційну версію на мові JavaScript з орієнтацією на наступне мережеве застосування.

Табл. 1. Іл. 4. Бібліогр.: 5 найм.

UDK 681.513

Local-parallel procedure of the tutoring with current adjusting of rank of results and sequencing of the samples / Mohamad Ali, O.Ph. Mikhal // Bionics of Intelligence: Sci. Mag. – 2013. – № 2 (81). – P. 75-80.

The principle is defined of optimal local-parallel organization of tutoring algorithm for learning the linguistical material. Variant of algorithm is offered, without any auxiliary actions of student, aside from the mainstream of the education. 32-bit demo version is made on JavaScript, with orientation to the future network usage.

Tabl. 1. Fig. 4. Ref.: 5 items.