

**Ромм Яков Евсеевич** – ФГБОУ ВПО «Таганрогский государственный педагогический институт имени А.П. Чехова»; e-mail: romm@list.ru; 347926, г. Таганрог ул. Инициативная, 48; тел.: 88634601753, 88634601812, 88634601899; кафедра информатики; зав. кафедрой; д.т.н.; профессор.

**Соколов Иван Николаевич** – e-mail: jey-s@yandex.ru; 347932, г. Таганрог, ул. Тихона Руденко, 55; тел.: 89514952411; ведущий программист отдела контроля качества образования и инновационных технологий.

**Romm Yakov Evseevich** – Taganrog State Pedagogical Institute named after A.P. Chechov; e-mail: romm@list.ru; 48, Initsiativnaya street, Taganrog, 347926, Russia; phones: +78634601753, +78634601812, +78634601899; the department of computer science; head the department; dr. of eng. sc.; professor.

**Sokolov Ivan Nikolaevich** – e-mail: jey-s@yandex.ru; 55, Tixon Rudenko street, Taganrog, 347932, Russia; phone: +79514952411; leading programmer of department quality control of education and innovation technologies.

УДК 681.3

**Ю.О. Чернышев, Н.Н. Венцов, С.А. Мухтаров**

#### **АЛГОРИТМ СТАТИЧЕСКОЙ ОПТИМИЗАЦИИ ПЕРЕДАЧИ ДАННЫХ\***

*Сформулирована задача статической оптимизации доступа к данным. На основе механизмов индивидуальной и коллективной адаптации разработан алгоритм решения задачи. В используемой модификации метода роя частицы образуют социально-сетевую топологию типа "кольцо". Изменение скорости частицы пропорционально расстоянию между ней и наилучшим решением, найденным её соседями. Определены зависимости времени работы алгоритма от размерностей решаемых задач. Проведено сравнение решений получаемых предлагаемым и жадным алгоритмом. Установлено что на тестовых задачах размерностей  $5 \times 3, \dots, 15 \times 3$  и  $5 \times 5, \dots, 5 \times 15$  разработанный алгоритм находит решения, экономящие время передачи данных на 10–15 %, по сравнению с жадным алгоритмом.*

*Вычислительные ресурсы; индивидуальная и коллективная адаптация; интеллектуальные системы; оптимизация.*

**Yu.O. Chernyshev, N.N. Vencov, S.A. Mukhtarov**

#### **THE ALGORITHM OF STATIC OPTIMIZATION OF DATA TRANSFER**

*It is formulated the problem of static optimize data access. On the basis of the mechanisms of individual and collective adaptation of the developed algorithm of solving the problem. In the modified method of swarm particles form a social-network topology type "ring". Change the velocity of the particle is proportional to the distance between it and the best solution found its neighbors. The dependence of the time algorithm from the dimensions of solved tasks. Comparison of the solutions proposed and obtained a greedy algorithm. It is established that the test problems dimensions  $5 \times 3, \dots, 15 \times 3$  and  $5 \times 5, \dots, 5 \times 15$  developed algorithm finds time-saving solutions data on 10–15 %, compared with a greedy algorithm.*

*Computing resources; individual and collective adaptation; intelligent systems; optimization.*

**Введение.** Во многих случаях поиск информации сводится к обработке большого количества распределенных информационных массивов. Процесс выполнения распределенных запросов в значительной степени зависит от текущей загруженности сетевых ресурсов. Для оптимизации процесса передачи данных в динамике активно

\* Работа выполнена при финансовой поддержке РФФИ (проекты 12-01-00474, 13-01-00343).

ведутся разработки программируемых сетей. Например, сети класса GENI позволяют выделять в общей сети подсеть, состоящую из активного и пассивного сетевого оборудования [1]. Данная подсеть может конфигурироваться автономно, относительно общей сети. Управление процессом передачи данных также можно организовать на основе протокола Open Flow [2]. Он позволяет разделять трафик на «производственный» и «исследовательский». Управление «производственным» трафиком может осуществляться на основе классических, проверенных подходов, «исследовательский» может быть изолирован от «производственного» с целью апробации новых схем адресации, моделей безопасности, протоколов маршрутизации и т.д. Рассмотрим задачу оптимизации доступа к данным на примере. Пусть необходимо сформировать документ, состоящий из трех разделов ( $z=3$ ), хранящихся на трех различных БД (БД1, БД2, БД3). Приложение, компонующее запросы в единый документ, расположено удаленно относительно БД. Предположим также, что запросы независимы друг от друга, в том плане, что генерация одного запроса не зависит от результатов другого.

Обозначим через  $L_j$  ( $j=1, 2, \dots, z$ ) маршрут передачи результатов выполнения  $j$ -го запроса (т.е. запроса к  $j$ -ой БД). Сегменты сети пропускной способностью которых можно управлять, обозначим через  $u_i$  ( $i=1, 2, \dots, r$ ), где  $r$  – количество данных сегментов. Пусть  $L_1=\langle u_1, u_2 \rangle$ ,  $L_2=\langle u_4, u_3, u_2 \rangle$ ,  $L_3=\langle u_5, u_3, u_2 \rangle$ . Обозначим через  $\dot{u}_i$  пропускную способность сегмента  $u_i$ . Тогда через  $\dot{u}_{ij}$ , обозначим ширину полосы пропускания выделенную в сегменте  $u_i$  для передачи результатов  $j$ -го запроса. Предположим, что сегмент сети используется приложением, формирующим заключение монополюно, тогда:

$$\dot{u}_i = \sum_{j=1}^z \dot{u}_{ij}. \quad (1)$$

Скорость передачи результатов  $j$ -го запроса по маршруту  $L_j$  обозначим через  $L'_j$  тогда:

$$L'_j = \min(\dot{u}_{i,j}) : u_i \in L_j. \quad (2)$$

Формула (2) указывает на то, что скорость передачи данных  $L'_j$  по маршруту  $L_j$  определяется сегментом  $u_i$  маршрута  $L_j$  с наименьшей пропускной способностью.

Пусть  $D_j$  объем данных, которые необходимо передать в качестве результата  $j$ -го запроса. Тогда критерий оптимальности, выполнения  $j$ -го запроса, можно выразить формулой:

$$\frac{D_j}{L'_j} \rightarrow \min. \quad (3)$$

В соответствии с (3), критерием оптимальности поставленной оптимизационной задачи будет выражение:

$$\min \left( \frac{D_j}{L'_j} \right), j = 1, 2, \dots, z. \quad (4)$$

Решением оптимизационной задачи будет матрица  $U$  размерности  $r \times z$ , на элементы которой накладываются ограничения:

$$\forall i \in \{1, 2, \dots, r\} \dot{u}_i \geq \dot{u}_{i,j}; \quad (5)$$

$$u_i \in L_j \Leftrightarrow \dot{u}_{i,j} \neq -1. \quad (6)$$

На рис. 1 изображена структура информационного массива, описывающего распределение пропускных способностей каналов связи, для рассматриваемого примера, которая представляет матрицу  $U$ .

$\dot{U}_{1,1}$	-1	-1
$\dot{U}_{2,1}$	$\dot{U}_{2,2}$	$\dot{U}_{2,3}$
-1	$\dot{U}_{3,2}$	$\dot{U}_{3,3}$
-1	$\dot{U}_{4,2}$	-1
-1	-1	$\dot{U}_{5,3}$

Рис. 1. Матрица соответствующая ограничениям

**Предлагаемый подход.** В связи с большой областью поиска (порядка  $r \times z$ ) для решения задачи предлагается использовать подход основанный на коллективной адаптации – модификацию метода пчелиной колонии, называемую локально наилучшей *PSO*-метода (*lbest PSO*) [3–7, 9]. В указанной модификации метода частицы образуют социально-сетевую топологию типа “кольцо”. Для каждой частицы задаётся меньшее количество соседей, положение которых учитывается при расчете траектории движения особей роя в отличие от *gbest* метода [3]. В соответствии с концепцией *lbest PSO*, каждая частица пытается имитировать лучшего из своих соседей, смещаясь в его сторону. Социальная составляющая скорости, отражает обмен информацией между соседями частицы. Изменение скорости частицы пропорционально расстоянию между ней и наилучшим решением, найденным её соседями. Так как, при реализации стратегии *lbest PSO*-метода каждая частица ориентируется на лучшую позицию своих соседей, а не всего роя, в пространстве поиска ожидаемая скорость сходимости *lbest PSO* ниже, чем *gbest* метода. Применительно к данной задаче, низкая скорость сходимости может обеспечить более равномерное исследование области значений, что, в свою очередь, способствует уменьшению вероятности заикливания алгоритмов в локальных ямах.

Каждое решение представим как частицу, движущуюся в  $r \times z$ -мерном пространстве. Координаты частицы можно описать  $r$ -мерным вектором (кортежем)  $X = \langle x_{1,1}, x_{1,2}, \dots, x_{1,j}, \dots, x_{1,z}; x_{2,1}, x_{2,2}, \dots, x_{2,j}, \dots, x_{2,z}; x_{i,1}, x_{i,2}, \dots, x_{i,j}, \dots, x_{i,z}; x_{r,1}, x_{r,2}, \dots, x_{r,j}, \dots, x_{r,z} \rangle$ . Значение каждого элемента  $x_{ij}$  кортежа  $X$  соответствует элементу  $u_{ij}$  таблицы  $U$ .

С помощью кортежа  $X_k(t) = \langle x_{k,1,1}(t), x_{k,1,2}(t), \dots, x_{k,1,j}(t), \dots, x_{k,1,z}(t); x_{k,2,1}(t), x_{k,2,2}(t), \dots, x_{k,2,j}(t), \dots, x_{k,2,z}(t); x_{k,i,1}(t), x_{k,i,2}(t), \dots, x_{k,i,j}(t), \dots, x_{k,i,z}(t); x_{k,r,1}(t), x_{k,r,2}(t), \dots, x_{k,r,j}(t), \dots, x_{k,r,z}(t) \rangle$ , обозначим позицию  $k$ -ой частицы в пространстве поиска решений в момент времени  $t$ . Факт вычисления функционала (4), на основе кортежа  $X_k(t) = \langle x_{k,1,1}(t), x_{k,1,2}(t), \dots, x_{k,1,j}(t), \dots, x_{k,1,z}(t); x_{k,2,1}(t), x_{k,2,2}(t), \dots, x_{k,2,j}(t), \dots, x_{k,2,z}(t); x_{k,i,1}(t), x_{k,i,2}(t), \dots, x_{k,i,j}(t), \dots, x_{k,i,z}(t); x_{k,r,1}(t), x_{k,r,2}(t), \dots, x_{k,r,j}(t), \dots, x_{k,r,z}(t) \rangle$ , соответствующего матрице  $U$ , обозначим как:

$$F[X_k(t)] \rightarrow \min. \quad (7)$$

Процесс поиска решений, данным методом, начинается с генерации частиц. Начальное состояние частицы  $k$ , в нулевой момент времени, описывается кортежем  $X_k(0) = \langle x_{k,1,1}(0), x_{k,1,2}(0), \dots, x_{k,1,j}(0), \dots, x_{k,1,z}(0); x_{k,2,1}(0), x_{k,2,2}(0), \dots, x_{k,2,j}(0), \dots, x_{k,2,z}(0); x_{k,i,1}(0), x_{k,i,2}(0), \dots, x_{k,i,j}(0), \dots, x_{k,i,z}(0); x_{k,r,1}(0), x_{k,r,2}(0), \dots, x_{k,r,j}(0), \dots, x_{k,r,z}(0) \rangle$ , и определяется, в простейшем случае, следующим образом  $X_k(0) = U(\xi, A)$ . Функция  $U(\xi, A)$  для каждого измерения создаваемой особи, генерирует случайное число из диапазона  $[\xi_j, a_j]$ .

Позиция  $k$ -ой частицы, в пространстве поиска решений, изменяется добавлением скорости  $V_k(t) = X_k(0) = \langle v_{k,1,1}(0), v_{k,1,2}(0), \dots, v_{k,1,j}(0), \dots, v_{k,1,z}(0); v_{k,2,1}(0), v_{k,2,2}(0), \dots, v_{k,2,j}(0), \dots, v_{k,2,z}(0); v_{k,i,1}(0), v_{k,i,2}(0), \dots, v_{k,i,j}(0), \dots, v_{k,i,z}(0); v_{k,r,1}(0), v_{k,r,2}(0), \dots, v_{k,r,j}(0), \dots, v_{k,r,z}(0) \rangle$ , к текущей позиции:

$$X_k(t+1) = X_k(t) + V_k(t+1). \quad (8)$$

В разновидности *lbest PSO*-метода каждая частица, тяготеет к лучшему решению полученному её соседями, поэтому скорость  $k$ -ой частицы, в  $ij$ -ом измерении, определяется по формуле:

$$v_{k,i,j}(t+1) = v_{k,i,j}(t) + c_1 r_1(t) [y_{k,i,j}(t) - x_{k,i,j}(t)] + c_2 r_2(t) [y_{k,j}^*(t) - x_{k,i,j}(t)], \quad (9)$$

где  $v_{k,i,j}(t)$  – скорость  $k$ -ой частицы в  $ij$ -ом измерении в момент времени  $t$ ;  $x_{kij}(t)$  – координаты частицы  $k$  в измерении  $ij$ ;  $c_1$  и  $c_2$  – положительные константы ускорения, варьирующие когнитивную и социальную компоненты скорости частицы;  $r_1$  и  $r_2$  – случайные переменные, принимающие значения 0 или 1;  $y_{kij}(t)$  – координата наилучшей достигнутой позиции частицы  $k$  в  $ij$ -м измерении.  $y_{kj}^*(t)$  – координата наилучшей достигнутой соседней частицы  $k$  в  $j$ -м измерении.

В случае, если  $(x_{i,j}(t) \neq -1)$  и  $(x_{i,j}(t+1) < \xi_j)$ , то  $x_{i,j}(t+1) := \xi_j$  и, если  $x_{i,j}(t+1) > a_j$ , то  $x_{i,j}(t+1) := a_j$ .

Кортеж  $Y_k = \langle x_{1,1}, x_{1,2}, \dots, x_{1,j}, \dots, x_{1,z}; x_{2,1}, x_{2,2}, \dots, x_{2,j}, \dots, x_{2,z}; x_{i,1}, x_{i,2}, \dots, x_{i,j}, \dots, x_{i,z}; x_{r,1}, x_{r,2}, \dots, x_{r,j}, \dots, x_{r,z} \rangle$  отображает наилучшую позицию частицы  $k$ , которую она посещала, начиная с первой итерации. В момент времени  $t+1$  следующая оптимальная позиция частицы  $k$  рассчитывается по формуле:

$$\begin{cases} Y_k = Y_k, & \text{если } F(X_k(t+1)) \leq F(Y_k), \\ Y_k = X_k(t+1), & \text{если } F(X_k(t+1)) > F(Y_k). \end{cases} \quad (10)$$

В формуле (10) выражение  $F(X_k(t+1))$  – описывает значение целевой функции вычисленной на основе кортежа  $X_k$ , полученного на  $t+1$  – ой итерации, а  $F(Y_k)$  – на основе кортежа  $Y_k$ .

**Алгоритм решения.** Работа алгоритма сводится итерационному выполнению следующих шагов:

1. «Генерация особей» – создание особей, образующих пчелиный рой. Для каждой особи, на основе ее координат, вычисляется значение целевой функции. Скорости всех сгенерированных особей равны нулю. После того как особи сгенерированы производится их группировка, т.е. определение для каждой особи роя множества соседних особей.

2. «Расчет статистики» – процесс нахождения для каждой частицы величины  $y^*(t)$  – координаты наилучшей достигнутой позиции её соседей найденной за  $t$  итераций и величины  $y_i(t)$  – координаты наилучшей достигнутой позиции частицы найденной за  $t$  итераций. Под наилучшей позицией понимается такое положение частицы в пространстве, при котором значение целевой функции минимально. Таким образом величина  $y^*(t)$  определяется для множества соседних по отношению к текущей частиц, а величина  $y_i(t)$  – для каждой частицы в отдельности.

3. «Расчет скоростей» – процесс поиска для каждой особи роя по формуле, новой скорости в каждом из пространств.

4. «Перемещение особей» описывает процесс расчета для каждой особи роя по формуле новых координат.

5. «Расчет ЦФ» соответствует процессу вычисления целевой функции для каждой особи роя на основе новых координат.

Критерием остановки алгоритма является выполнение заданного числа итераций.

На рис. 2 приведена зависимость времени работы предлагаемого алгоритма от размерности задачи, связанной с описанием количества  $z$  экземпляров оборудования, задействованного при выполнении запросов.

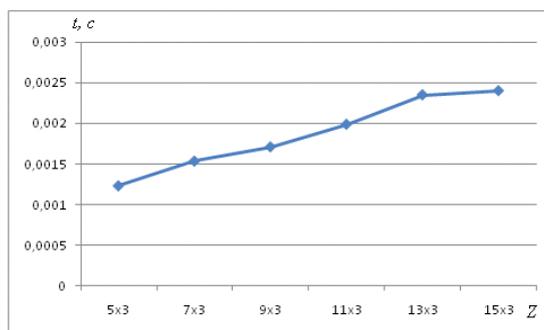


Рис. 2. Зависимость времени работы предлагаемого алгоритма от размерности, описывающей количество сегментов сети

Из данных, представленных на рис. 2, видно, что время решения линейно зависит от размерности задачи  $z$ . Кроме того, из рис. 2 следует, что время поиска решения, предлагаемым алгоритмом, на задачах размерности  $5 \times 3, \dots, 15 \times 3$ , изменяется от  $1,2 \cdot 10^{-3}$  до  $2,5 \cdot 10^{-3}$  секунд.

На рис. 3 представлена зависимость времени работы предлагаемого алгоритма от количества оптимизируемых запросов  $r$ .

На основании данных представленных на рис. 3 можно предположить, что время решения линейно зависит от размерности задачи  $r$ . Из рис. 3 следует, что время поиска решения, предлагаемым алгоритмом на задачах размерности  $5 \times 5, \dots, 5 \times 15$ , изменяется от  $1,5 \cdot 10^{-3}$  до  $3,5 \cdot 10^{-3}$  секунд.

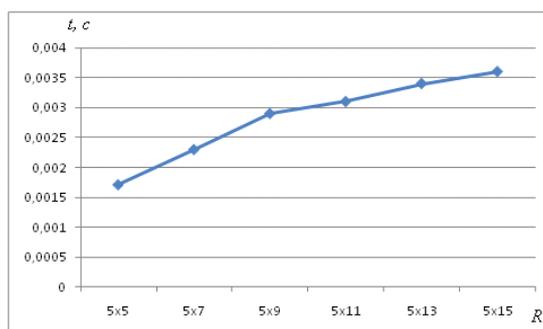


Рис. 3. Зависимость времени работы предлагаемого алгоритма от количества оптимизируемых запросов

Сравнение результатов работы предложенного и жадного алгоритмов позволяет заключить:

- ♦ при изменении размерностей решаемых задач от  $5 \times 3$  до  $15 \times 3$  предлагаемый алгоритм обеспечивает получение решений, экономящих время передачи данных на 10–15 %;
- ♦ при изменении размерностей решаемых задач от  $5 \times 5$  до  $5 \times 15$  предлагаемый алгоритм обеспечивает получение решений, экономящих время передачи данных на 10–15 %.

**Выводы.** В случаях, когда жадный алгоритм способен определять рациональные решения, предлагаемый алгоритм обеспечивает получение решений, экономящих время передачи данных на 10–15 %. Время работы предложенного алгоритма зависит линейно от каждой из двух размерностей задачи оптимизации  $z$  и  $r$  ( $z$ -размерность экземпляров оборудования, задействованного при выполнении запросов, и  $r$ -размерность количества оптимизируемых запросов). Линейная ВСА подтверждена экспериментально на задачах размерности  $5 \times 3 \dots 15 \times 3$  по оси  $z$  и  $5 \times 5, \dots, 5 \times 15$  по оси  $r$ .

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Global Environment for Network Innovations. Web site <http://geni.net>.
2. Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. March 14, 2008.
3. Kennedy J. The Particle Swarm: Social Adaptation of Knowledge // In Proceedings of the IEEE International Conference on Evolutionary Computation. – 1997. – P. 303-308.
4. Курейчик В.М. Биоинспирированный поиск с использованием сценарного подхода // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 7-12.

5. *Cong J., Fang J., Xie M. and Zhang Y.* MARS—A multilevel full-chip gridless routing system // IEEE Trans Comput.-Aided Design Integr. Syst. – 2005. – Vol. 24, № 3. – P. 382-394.
6. *Лебедев Б.К., Лебедев В.Б.* Поисквые процедуры канальной трассировки, базирующиеся на моделировании адаптивного поведения роя частиц в пространстве решений с упорядоченным лингвистическим шкалированием // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 15-22.
7. *Dorigo M. and Stützle T.* Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
8. *Walter Banks, Gordon Hayward.* Fuzzy logic in embedded microcomputers and control systems. Published by Byte Craft Limited, Waterloo Ontario Canada, 2001.
9. *Курейчик В.М.* Особенности построения систем поддержки принятия решений // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 92-98.
10. *Чернышев Ю.О., Венцов Н.Н., Крупенин А.В., Щербаков А.В., Мухтаров С.А.* Адаптивная оптимизация запросов в современных системах управления базами данных. Монография. Краснодар: ФВАС, 2011.
11. *Курейчик В.М., Кажаров А.А.* Использование роевого интеллекта в решении NP-трудных задач // Известия ЮФУ. Технические науки. – 2010. – № 7 (120). – С. 30-36.

Статью рекомендовал к опубликованию д.т.н., профессор Б.К. Лебедев.

**Чернышев Юрий Олегович** – Донской государственный технический университет; e-mail: myvnn@list.ru; 344000, г. Ростов-на-Дону, пл. Гагарина, 1; тел.: 88632738582, кафедра автоматизации производственных процессов; профессор.

**Венцов Николай Николаевич** – e-mail: vencov@list.ru; кафедра информационных технологий; доцент.

**Мухтаров Сергей Артурович** – Филиал Военной академии связи (г. Краснодар); e-mail: samucht@list.ru; 350035, г. Краснодар, ул. Красина, 4; соискатель.

**Chernyshev Yury Olegovich** – Don State Technical University; e-mail: myvnn@list.ru; 1, Gagarin Square, Rostov-on-Don, 344000, Russia; phone: +78632738582; the department of automation of productions; professor.

**Vencov Nikolay Nikolaevich** – e-mail: vencov@list.ru; the department of information technologies; associate professor.

**Mukhtarov Sergey Arturovich** – Branch of Military academy of communication (Krasnodar); e-mail: samucht@list.ru; 4, Krasina street, Krasnodar, 350035, Russia; competitor.

УДК 004.912

**Р.Ю. Вишняков, Ю.М. Вишняков**

#### **«ВЫЧИСЛИТЕЛЬНОЕ» ПРЕДСТАВЛЕНИЕ СМЫСЛА ТЕКСТОВЫХ ФРАГМЕНТОВ НА ОСНОВЕ ОБРАТНОЙ ПОЛЬСКОЙ ЗАПИСИ**

*Предлагаемая работа развивает интерпретационную модель представления смыслов предложений текстов научно-технического стиля и посвящена конструированию «вычислительного» представления функционала смысловыразительности текстовых фрагментов на основе идей обратной польской записи. В работе разрабатывается процедура представления функционала смысловыразительности текстового фрагмента в нотации обратной польской записи, формулируются правила ее получения и вычисления. Определяются понятия операции контекстного уточнения смысла, контекстной связки, дерева контекстной связки и конструируются их представление в нотации обратной польской записи. Рассматриваются свойства и особенности операции контекстного уточнения смысла и исследуются особенности контекстной связки. Все рассмотренные понятия и соотношения разбираются и иллюстрируются на конкретных примерах.*

*Контекст; контекстная связка; операция контекстного уточнения смысла; дерево контекстной связки; обратная польская запись.*