



ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.75

РАСПРЕДЕЛЕННАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА ОПТИМИЗАЦИИ ЗАКУПОК

В.В. МУРОМЦЕВ**Ю.А. КАЛАШНИКОВА**

Белгородский государственный национальный исследовательский университет

*e-mail:**muromtsev@bsu.edu.ru*

Рассматривается система, которая может быть использована лицами, принимающими решения, связанные с выбором поставщиков при организации закупок товаров. Система ориентирована на параллельную обработку данных в распределенной вычислительной среде.

Ключевые слова: поддержка принятия решений, планирование поставок, линейное программирование, параллельные вычисления.

Одной из основных задач, решаемой на многих экономических объектах, является задача планирования поставок товаров. При этом требуется принятие решения о том, у какого поставщика закупать те или иные товары. Обычно решение о закупке принимают менеджеры на основе собственного опыта. Однако, при большом ассортименте товаров и возможных поставщиков, принятие качественных решений эмпирическим путем становится невозможным. Даже в простейшем случае, когда требуется учитывать только дискретные системы скидок, предоставляемые поставщиками, задача оптимизации закупок товаров сводится к поиску экстремума многомерной мультимодальной дискретной функции [1,2] и не может быть эффективно решена на персональном компьютере. Выходом из данной ситуации может стать разработка параллельных алгоритмов оптимизации закупок и их реализация на вычислительном кластере, созданном на базе локальной вычислительной сети имеющейся практически на любом экономическом объекте.

Пусть требуемые товары предлагают b поставщиков и для каждого поставщика P_j , $j = 1, \dots, b$ задан вектор $C_j = (c_{j,1}, \dots, c_{j,S_j})$, где $c_{j,m}$ – m -ый порог скидки, то есть если у j -го поставщика купить товаров на сумму $c_{j,m}$, то он предоставит некоторую скидку. Информация о скидках определяет аналогичным вектором. S_j – число порогов скидки у j -го поставщика. Пусть также информация о предложениях товаров представлена дву-



дольным графом $G = (V, E)$, где множество вершин представляет собой объединение множеств товаров и поставщиков $V = \{T_1, T_2, \dots, T_n\} \cup \{\Pi_1, \Pi_2, \dots, \Pi_b\}$, а множество ребер E формируется следующим образом: $(T_i, \Pi_j) \in E; i = 1, \dots, n; j = 1, \dots, b$, если j -ый поставщик предлагает i -ый товар. Каждая дуга графа помечается парой $(z_{T_i, \Pi_j}, \Pi_{T_i, \Pi_j})$, где z_{T_i, Π_j} – количество i -го товара, которое может поставить j -ый поставщик, Π_{T_i, Π_j} – цена единицы i -го товара у j -го поставщика.

Задачу минимизации затрат на закупку товаров можно решить путем перебора всех композиций (c_1, c_2, \dots, c_b) составляющих множество $C = \left\{ (c_1, \dots, c_b) \mid 0 < \sum_{j=1}^b c_j \leq S; c_j \in \{0\} \cup \{c_{j,1}, \dots, c_{j,s_j}\} \right\}, j = 1, \dots, b$, где S – сумма, требуемая на закупку товаров, если не учитывать информацию о скидках, предоставляемых поставщиками. Значение S можно найти, решив задачу 1, представляющую собой следующую задачу линейной оптимизации:

$$\sum_{i=1}^n \sum_{j:(T_i, \Pi_j) \in E} x_{T_i, \Pi_j} \cdot \Pi_{T_i, \Pi_j} \rightarrow \min ;$$

$$x_{T_i, \Pi_j} \geq 0; i, j : (T_i, \Pi_j) \in E; \quad x_{T_i, \Pi_j} \leq z_{T_i, \Pi_j}; i, j : (T_i, \Pi_j) \in E; \quad \sum_{j:(T_i, \Pi_j) \in E} x_{T_i, \Pi_j} = K(T_i); i = 1, \dots, n,$$

где $K(T_i)$ – требуемое количество i -го товара; x_{T_i, Π_j} – количество i -го товара, заказанное у j -го поставщика.

Значение S равно найденному в ходе решения данной задачи минимуму.

После нахождения множества C для каждой композиции $(c_1, c_2, \dots, c_b) \in C$ решается задача 2, получаемая путем добавления к задаче 1 следующих ограничений:

$$\sum_{j:(T_j, \Pi_i) \in E} x_{T_j, \Pi_i} \cdot \Pi_{T_j, \Pi_i} \geq c_i; i = 1, \dots, b.$$

После решения пересчитывается стоимость заказа с учетом скидок, предоставляемых поставщиками. Заказ, имеющий наименьшую стоимость, запоминается.

Рассмотренный алгоритм минимизации затрат на закупку товаров сводится к многократному решению задачи линейного программирования. Задача линейного программирования решается для каждой композиции множества C .

При планировании реальных поставок мощность множества C может оказаться большой, что приведет к неприемлемому увеличению времени работы алгоритма. Можно предложить два подхода решения данной проблемы. Первый заключается в поисках стратегий сокращения мощности множества C . В этом случае следует привлекать эксперта. Разработка каких-либо формальных процедур сокращения мощности C вряд ли целесообразна. Второй подход заключается в распараллеливании алгоритма. При этом следует ориентироваться на аппаратные средства, доступные большинству потенциальных пользователей. Оба подхода могут сочетаться.

Наиболее рациональным решением является параллельная реализация алгоритма с помощью аппаратуры локальной вычислительной сети. Такое решение обусловлено тем, что локальная вычислительная сеть есть практически на всех экономических объектах.



В сети выделяется головной компьютер, на котором выполняется управляющая программа, и np подчиненных компьютеров, на которых выполняется параллельная часть приложения, состоящая из np процессов, распределенных по одному на каждый из подчиненных компьютеров.

Алгоритм управляющей программы представлен укрупненной блок-схемой на рис. 1. Вначале управляющая программа осуществляет ввод информации о предложениях товаров и информации о требуемых товарах. Далее исходные данные обрабатываются. В результате этой обработки формирует граф $G = (V, E)$ и вектор K , в котором каждая компонента равна требуемому количеству соответствующего товара. На следующем шаге решается задача 1 и тем самым находится значение S . Зная S , рассчитывается число композиций $nc = |C|$ и формируется начальная композиция (первая композиция множества C , композиции могут формироваться в любом порядке).

После таких действий управляющая программа инициализирует параллельную часть приложения с np процессами и переходит в цикл передачи данных этим процессам (первый цикл на рис.1). В зависимости от числа процессов и композиций может возникнуть два случая: 1) число процессов меньше числу композиций ($np < nc$); 2) число процессов больше или равно числу композиций ($np \geq nc$).

В первом случае все задачи линейной оптимизации, которые требуется решить для определения оптимального заказа, распределяются по процессам и загрузка новых данных не потребует. Во втором случае по процессам распределяется только часть задач линейной оптимизации. Оставшиеся задачи распределяются по процессам по мере их освобождения. Это требует загрузки новых данных.

Для уменьшения времени, затрачиваемого на обмен данными между процессами, исходные данные разбиты на две группы. Первая группа включает граф G и вектор K . Вторая – текущую композицию C_i и номер i . Данные первой группы не зависят от номера текущей композиции, поэтому посылка этих данных каждому процессу осуществляется только один раз.

После того, как процессы получили данные, они решают соответствующие задачи линейной оптимизации (задача 2) и рассчитывают стоимости заказа с учетом скидок, а управляющая программа переходит в цикл ожидания завершения этих расчетов (второй цикл на рис.1). Если некоторый процесс завершил расчет, то он пересылает управляющей программе номер композиции, соответствующей решенной задаче линейной оптимизации и стоимость заказа. Алгоритм процесса представлен на рис.2.

Управляющая программа анализирует данные, полученные от завершеного процесса, и находит номер композиции (i_{min}), для которой достигается глобальный минимум (min , см.рис.1). Если проанализированы не все композиции, т.е. решены не все задачи линейной оптимизации соответствующие композициям, то освободившемуся процессу пересылается текущая композиция и ее номер. После этого формируется следующая композиция (см. рис.1).

После того как проанализированы все композиции, управляющая программа формирует информацию о заказе, соответствующего композиции с номером i_{min} . Данный заказ является оптимальным.

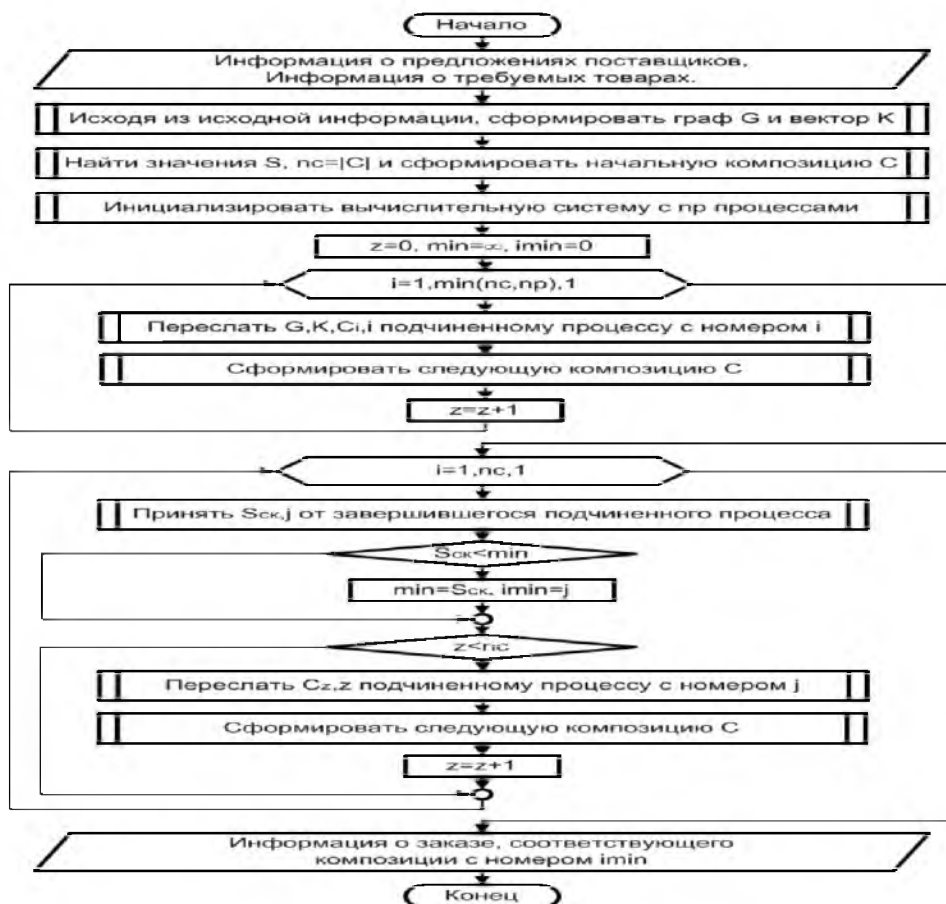


Рис. 1. Алгоритм управляющей программы

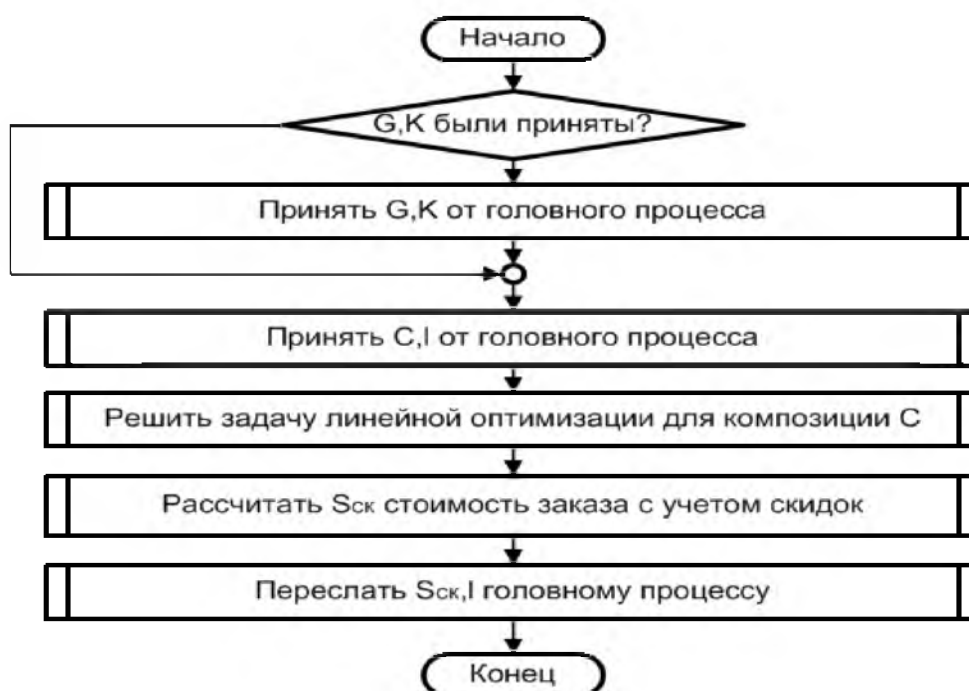


Рис. 2. Алгоритм процесса



Рассмотренный алгоритм позволяет формировать оптимальный заказ с учетом дискретных скидок, предоставляемых поставщиками. Для реализации алгоритма используется распределенная вычислительная система, функционирующая на базе локальной вычислительной сети. Такое решение доступно практически для любых экономических объектов. Использование алгоритма позволяет повысить качество принимаемых решений о закупке товаров.

Работа выполнена при финансовой поддержке ФЦП "Научные и научно-педагогические кадры инновационной России" на 2009-2013 годы. ГК № 14.740.11.0591 от 05.10.2010.

Литература

1. Муромцев В.В. Организация конкурсных закупок // Научные ведомости. Белгородский государственный университет. Сер. «Информатика и прикладная математика». – 2006. – №1(21). – Вып.2. – с.56-65.

2. Муромцев В.В., Слободюк А.А. Об использовании генетических алгоритмов и распараллеливания вычислений для оптимизации заказов предприятия // Научные ведомости БелГУ. Сер. История, политология, экономика, информатика. – 2009. – Вып.11/1, № 9 (64), с.160 – 164

DISTRIBUTED COMPUTING SYSTEM OPTIMIZATION OF PROCUREMENT

V.V.MUROMTSEV
J.A.KALASHNIKOVA

*Belgorod National
Research University*

e-mail:
muromtsev@bsu.edu.ru

We consider a system that can be used by decision-makers associated with the choice of suppliers for the procurement of goods. The system is focused on parallel processing in a distributed computing environment.

Key words: decision support, supply planning, linear programming, parallel computing.