

РЕШЕНИЕ ЗАДАЧИ ПОИСКА ГРАФ-ПОДГРАФ ИЗОМОРФИЗМА ДЛЯ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ ОРГАНИЗАЦИИ

В работе представлен граф-аналитический подход к распределению ресурсов организации. Он основан на алгоритме поиска граф-подграф изоморфизма для взвешенных и помеченных графов и является развитием алгоритма поиска граф-подграф изоморфизма для взвешенных графов.

Ключевые слова: резервирование ресурсов, граф-подграф изоморфизм, взвешенные графы, помеченные графы, граф-аналитический подход.

ВВЕДЕНИЕ

Процесс разработки программного обеспечения при создании сложных систем осуществляется в кооперации участников, включающей в себя непосредственно сотрудников организации, закрепленных за определенными отделами, сторонних организаций-подрядчиков и партнеров, консультантов, обслуживающих организации. Эти участники, как правило, распределены географически (представительства в нескольких городах, работа с удаленными партнерами) и во времени, поскольку цикл разработки современного комплексного программного обеспечения достаточно длительный.

Перспективным направлением оптимизации работы подобных сложных организационных структур является создание распределенной компьютерной системы поддержки производственных процессов, выделения и распределения ресурсов организации для решения каждого из этапов производственного процесса, многие из которых выполняются повторно в силу итерационного характера процессов разработки программного обеспечения.

Разработка подобной системы распределения ресурсов организации позволит, прежде всего, сократить время и стоимость взаимодействия между участниками производственного процесса, совместно использовать аппаратные, информационные и программные ресурсы организации, а так же систематизировать и упорядочить управленческие действия, направленные на нахождение и выделение разнородных ресурсов в рамках организации для решения формируемых руководством организации задач.

В данной статье рассматривается граф-аналитический подход для поиска комплексного набора ресурсов организации, предоставляемых участникам кооперации для обеспечения необходимых производственных процессов. Этот подход является одним из перспективных направлений резервирования ресурсов различных типов [1, 2], уже доказавший свою эффективность при решении задач большой размерности [3].

1 ПОСТАНОВКА ЗАДАЧИ

Пусть в рамках организации необходимо осуществить один из подпроцессов, для которого необходим определенный набор человеческих, аппаратных и программных ресурсов.

Временная и пространственная распределенность участников производственного процесса, большое количество участников подразумевают неоднородную, структурированную организацию ресурсов. Задача состоит в поиске необходимого набора ресурсов организации, как человеческих, так и программно-аппаратных.

Представим организацию, как граф, вершинами которого являются имеющиеся в наличии ресурсы, а ребра графа отражают возможность совместного их использования.

Эта возможность может выражаться, например, в правах доступа к серверам или базам данных со стороны конкретного участника организации или права удаленного доступа к локальным ресурсам организации, наличие возможности прямого взаимодействия между участниками процесса и другие подобные отношения доступности и взаимосвязанности ресурсов.

В этом случае каждая вершина графа организации будет помечена и взвешена, где метка отражает тип ресурсов, описываемых вершиной графа (например, наличие установленного специфического программного обеспечения на компьютерной станции), а набор приписанных весов отражает объективные числовые характеристики ресурсов (объем оперативной памяти, размер дискового пространства, число вычислительных узлов и т. п.).

Введем также второй граф, который будет представлять описание ресурсов, необходимых для выполнения технологического процесса. Для описания ресурсов в этом графе будем использовать ту же нотацию, что и для графа ресурсов организации, т. е. вершины описывают отдельные типы необходимых ресурсов, метки вершин - тип ресурса, набор весов - числовые характеристики ресурсов, ребра между вершинами - потребность в совместном использовании различных ресурсов.

При таком однородном описании ресурсов организации и потребности в ресурсах со стороны участников производственного процесса становится возможной постановка задачи поиска необходимого набора ресурсов как поиск граф-подграф изоморфизма на взвешенных и помеченных графах.

2 АЛГОРИТМ УСТАНОВЛЕНИЯ ГРАФ-ПОДГРАФ ИЗОМОРФИЗМА ДЛЯ ГРАФОВ ВЗВЕШЕННЫХ И ПОМЕЧЕННЫХ ПО ВЕРШИНАМ

Постановка задачи установления граф-подграф изоморфизма для помеченных графов. Пусть даны графы $G_1 = (V_1, E_1, L_1, R_1)$ и $G_2 = (V_2, E_2, L_2, R_2)$, где V – множество вершин графа, E – множество ребер графа, L – метки, приписанные вершинам графов и R – набор весов, приписанных вершинам графов. Граф G_1 изоморфен подграфу графа G_2 (обозначается, как $G_1 \cong S_2 \subseteq G_2$), если существует подстановка $\varphi: V_2 \rightarrow V_1$, такая, что для каждой пары вершин $v_i, v_j \in V_2$, если $(v_i, v_j) \in E_2$, то $(\varphi(v_i), \varphi(v_j)) \in E_1$ и для всех $l_i \in L_2$ выполняется $l_i \in L_2 = \varphi(l_i) \in L_1$ и для всех $r_i \in R_2$ выполняется $r_i \leq \varphi(r_i)$, где $\varphi(r_i) \in R_1$.

Алгоритм установления граф-подграф изоморфизма для взвешенных и помеченных графов является развитием и продолжением алгоритма установления изоморфности [4].

Алгоритм установления изоморфизма удобно описывать в терминах поиска в пространстве состояний. Каждое состояние s процесса совмещения вершин соответствует частичной подстановке $\varphi(s)$, которая содержит лишь часть вершин полной подстановки. Каждому состоянию также соответствуют подграфы $G_1(s)$ и $G_2(s)$, полученные из вершин графов G_1 и G_2 , вошедших в частичную подстановку $\varphi(s)$, и ребер, соединяющих эти вершины. В дальнейшем обозначим через $\varphi_1(s)$ и $\varphi_2(s)$ проекции подстановки $\varphi(s)$ на V_1 и V_2 .

Алгоритм состоит из предварительной и основной части. В предварительной части выполняются операции упорядочивания вершин графов и формируются граничные условия для однократно выполняемых, по ходу алгоритма, операций, призванные сократить область поиска основной, переборной части алгоритма.

2.1 Предварительная часть алгоритма

Основные действия, выполняемые в предварительной части алгоритма – сортировка вершин графов и формирование матрицы возможных совмещений. Идея предварительной перенумерации вершин графов при установлении их изоморфности с помощью back-tracking алгоритма была предложена в работе [5].

Матрица возможных совмещений $M_{i,j}$ – это таблица размером $|V_1| \times |V_2|$. Каждому элементу таблицы соответствует пара вершин исходных графов $V_{1,i}$ и $V_{2,j}$. Значения матрицы формируются следующим образом:

- $M_{i,j} = 0$, если на основании предварительных проверок вершины $V_{1,i}$ и $V_{2,j}$ совместить нельзя;
- $M_{i,j} = 1$, в противном случае.

Смысл матрицы возможных совмещений в том, чтобы выполнить однократно в рамках предварительной части алгоритма все проверки, не основанные на информации, полученной в процессе совмещения вершин, тем самым, ускорить обработку соответствующих ограничений, сведя ее к одной операции сравнения.

В программе реализованы следующие предварительные проверки:

1. $M_{i,j} = 0$, если $|V_{1,i}| < |V_{2,j}|$, где $|V_{X,Y}|$ – степень вершины Y графа X ;
2. $M_{i,j} = 0$, если $|V_{1,i}^{in}| < |V_{2,j}^{in}|$, где $|V_{X,Y}^{in}|$ – число входящих ребер вершины Y графа X ;
3. $M_{i,j} = 0$, если $|V_{1,i}^{out}| < |V_{2,j}^{out}|$, где $|V_{X,Y}^{out}|$ – число исходящих ребер вершины Y графа X ;
4. $M_{i,j} = 0$, если $W_{1,i}^{vertex} < W_{2,j}^{vertex}$, где $W_{X,Y}^{vertex}$ – число вершин в первых 4-х волнах волнового разложения подграфа окружения вершины Y графа X ;
5. $M_{i,j} = 0$, если

$$\sum_{m=1}^n W_{1,i,m}^{vertex} < \sum_{m=1}^n W_{2,j,m}^{vertex}, n = 1..4, |W_{2,j}^{vertex}|, m = 1..4, \text{ где}$$

$W_{X,Y,m}^{vertex}$ – число вершин в m -ой волне волнового разложения графа X , начиная с вершины Y ;

6. $M_{i,j} = 0$, если $W_{1,i}^{ribes} < W_{2,j}^{ribes}$, где $W_{X,Y}^{ribes}$ – число ребер в первых 4-х волнах волнового разложения подграфа окружения вершины Y графа X ;
7. $M_{i,j} = 0$, если

$$\sum_{m=1}^n W_{1,i,m}^{ribes} < \sum_{m=1}^n W_{2,j,m}^{ribes}, n = 1..4, |W_{2,j}^{ribes}|, m = 1..4, \text{ где } W_{X,Y,m}^{ribes}$$

– число ребер в m -ой волне волнового разложения графа X , начиная с вершины Y .

8. $M_{i,j} = 0$, если $L_{1,i} \neq L_{2,j}$, где $L_{X,Y}$ – метка вершины Y графа X .
9. $M_{i,j} = 0$, если $R_{1,i} \leq R_{2,j}$, где $R_{X,Y}$ – вес вершины Y графа X .

Возможно использование и других критериев для оценки возможности совмещения вершин графов. Разработка таких критериев основана на волновом разложении графов, начиная с заданной вершины [6]. По мере

распространения волны получают подграфы окружения каждой вершины. Сравнивая параметры соответствующих подграфов окружения вершин графов, которые предполагается совмещать, делается вывод о потенциальной возможности или принципиальной невозможности такого совмещения. В приведенных критериях для этого использовались количество вершин и ребер в подграфах окружения сравниваемых вершин для всех этапов распространения волны.

Сортировка вершин графов производится с целью ускорения нахождения изоморфной подстановки, в случае, если такая подстановка существует. В переборной части алгоритма переставляются только вершины большего графа, в то время, как порядок вершин меньшего графа не меняется. Порядок следования вершин меньшего графа определяется в предварительной части алгоритма.

Пусть $T_{2,i}$ – количество ребер инцидентных верши-

нам с меньшими номерами и $P_{2,i} = \sum_{j=1}^{|V_1|} M_{j,i}$ – суммарное количество вариантов совмещения вершины i графа G_2 с вершинами графа G_1 . Тогда порядок сортировки вершин графа G_2 следующий:

$$V_{2,i} = V_{2,k}, \text{ где } T_{2,k} = \min_{j=i+1}^{|V_2|} (T_{2,j}).$$

Если $T_{2,i} = T_{2,j}$, то $V_{2,i} = V_{2,k}$, где $P_{2,k} = \min(P_{2,i}, P_{2,j})$.

Т. е. вершины графа G_2 сортируются в порядке убывания количества связей с вершинами, имеющими меньшие номера или в порядке убывания количества вариантов совмещения вершин, если количество связей одинаково. Такой порядок следования вершин обусловлен тем, что чем больше связей, с уже совмещенными, имеет вершина, тем жестче будет ограничивающее условие, включающее эту вершину, и, соответственно, меньше общее количество совмещений, которые необходимо перебрать.

2.2 Основная часть алгоритма

Основная часть алгоритма представляет собой последовательное наложение вершин с возвратом, описывать которое удобно в терминах метода поиска в пространстве состояний.

Вершины графа G_2 остаются нетронутыми и каждой из них ставится в соответствие одна из вершин графа G_1 . При этом проверяется допустимость такого совмещения. Если удастся найти соответствие всем вершинам графа G_2 , при этом выполнено условие изоморфизма, то найденное состояние возвращается как искомая подстановка.

Пусть $T_{1,i}$ – количество связей вершины i графа G_1 с вершинами $V_{1,j} \in \Phi_1(s)$, а $T_{2,i}$ – количество связей вершины i графа G_2 с вершинами $V_{2,j} \in \Phi_2(s)$.

Начальному состоянию $\Phi(s)_0 = 0$ соответствует состояние, при котором не совмещено еще ни одной пары вершин.

Для получения i -го состояния для вершины $V_{2,i}$ ищется соответствие среди вершин $V_{1,j}$, таких что:

1. $M_{i,j} = 1$, т. е. вершины совместимы на основании предварительных проверок.
2. $T_{1,i} \geq T_{2,j}$
3. Для $k = 1..i$, если $(v_i, v_k) \in E_1$, то $(\Phi(v_i), \Phi(v_k)) \in E_2$.

Если выполнены все условия, из которых третье является прямым следствием определения граф-подграф изоморфизма, то соответствующая пара вершин входит в частичную подстановку и формируется новое состояние $\Phi(s)_i$.

Перебор состояний производится методом поиска в глубину.

3 ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК АЛГОРИТМА

В зависимости от решаемых задач, организационной структуры и географической распределенности различные организации на различных этапах своего существования могут иметь совершенно разные соответствующие им графы, отражающие набор имеющихся в наличии ресурсов и возможностей их совместного использования. Однако очевидно, что чем меньше организация, тем больше плотность реберного заполнения соответствующего графа, поскольку между меньшим числом участников, в условиях ограниченных возможностей коммуникации, существует большее число связей. В то же время с ростом масштабов организации, плотность реберного заполнения соответствующего графа будет падать, поскольку участники производственного процесса с ростом масштабов деятельности не могут взаимодействовать все со всеми и иметь потребности в использовании всех ресурсов организации.

Подобная ситуация наблюдается и для графа задачи, которую необходимо решить в рамках организации. Для небольших задач характерно интенсивное использование ограниченного набора ресурсов и, как следствие, наличие большего числа связей в графе задачи. Вместе с ростом масштабов задачи, в ее решение вовлекается все больше ресурсов, которые естественным образом структурируются и образуют менее плотный граф задачи в терминах этой работы.

Поэтому для исследования характеристик алгоритма использовались случайно сгенерированные графы с заданной плотностью реберного заполнения. Такие графы могут представлять в грубом приближении описанные выше предоставляемые организацией ресурсы и потребности в них со стороны решаемых задач.

Для исследования формировались организации произвольной структуры, содержащие ресурсы различных типов (метки, приписываемые вершинам графов) и объема (веса, приписываемые вершинам графа). Для задан-

ного набора ресурсов формировалась сеть, отражающая возможность их совместного использования с заданной плотностью реберного заполнения.

Вершинам графов приписывались метки трех типов, соответствующие возможности использования CPU, RAM и HDD, а так же условный доступный объем соответствующих ресурсов скорректированный в одном масштабе от 200 до 3000 условных единиц соответствующих ресурсов.

Задачи генерировались случайным образом из графа сети, путем удаления части вершин и ребер графа сети и уменьшения приписанных весов. Параметрами являлись величины:

- k_{nv} – коэффициент уменьшения числа вершин для графа задачи;
- k_r – коэффициент числа ребер графа;
- k_w – коэффициент уменьшения весов графа.

Метки, приписанные вершинам графов, сохранялись.

Для оценки производительности алгоритма в случае когда необходимые для задачи ресурсы не могут быть выделены в рамках организации (граф задачи не является строгим подграфом графа организации) в графе задачи заменялось случайное количество ребер, таким образом что он переставал быть строгим подграфом организации, но общее число вершин и ребер графа задачи сохранялось таким же как и для случая строгого граф-подграф изоморфизма.

Исследования проводились на компьютере Athlon 64 x2 5400+, использовалось только одно ядро процессора (однопоточная реализация представленных алгоритмов). Каждое значение на графике получено путем усреднения времени вычислений на 1000 случайно сгенерированных парах графов.

Некоторые характерные результаты исследований представлены на рис. 1, 2.

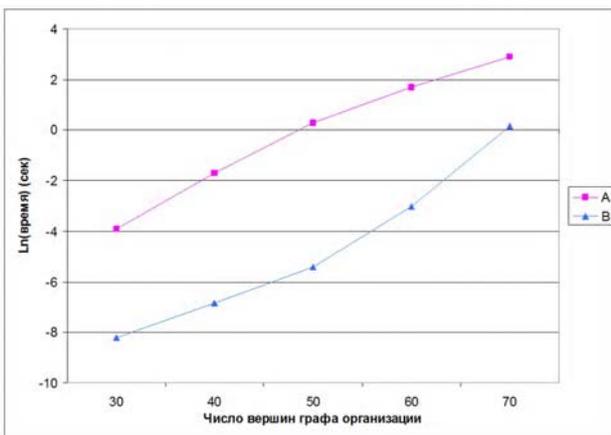


Рис. 1. Среднее время нахождения размещения задачи в сети при параметрах: $k_{nv} = 0,2$, $k_r = 0,5$, $k_w = 0,2$, при плотности реберного заполнения характеристических графов сети 50 % (A – граф задачи не является подграфом графа организации; B – граф задачи является подграфом графа организации)

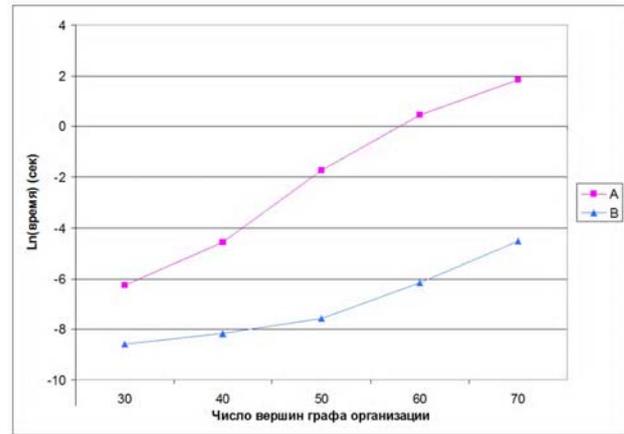


Рис. 2. Среднее время нахождения размещения задачи в сети при параметрах: $k_{nv} = 0,1$, $k_r = 0,5$, $k_w = 0,2$, при плотности реберного заполнения характеристических графов сети 30 % (A – граф задачи не является подграфом графа организации; B – граф задачи является подграфом графа организации)

Как видно из рисунков, алгоритм имеет в общем случае не полиномиальную вычислительную сложность, но при уменьшении числа вершин в графах или плотности реберного заполнения, большую роль играют предварительные проверки, уменьшающие количество вариантов совмещения вершин в переборной части, поэтому алгоритм имеет высокую производительность на многих значимых для практических задач конфигурациях графов, характеризующихся низкой плотностью реберного заполнения.

4 ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ АЛГОРИТМА

В изложении задачи поиска граф-подграф изоморфизма применительно к области поиска ресурсов организации для решения комплексных задач имеется ряд факторов, каждый из которых ведет к повышению производительности алгоритма применительно к ожидаемым практическим конфигурациям графов самой организации и задачи, для которой ищутся ресурсы. К таким факторам относятся:

1. Метки, приписываемые вершинам. При решении задачи поиска граф-подграф изоморфизма для поиска ресурсов в рамках организации каждый из типов ресурсов имеет свою метку, приписываемую всем вершинам графов. Приписанные метки являются дополнительными условиями, ограничивающими рекурсию при переборе возможности совмещения вершин графов, тем самым, сужая область поиска возможных решений и повышая производительность алгоритма. Чем больше различных типов ресурсов используется в рамках организации, тем сильнее влияние ограничивающего условия, основанного на метках. В реальных условиях ресурсы организации гораздо более разнообразны и для их классификации понадобится гораздо большая номенклатура меток, чем упрощенная модель, содержащая три вида меток, рассмотренная в качестве примера в данной работе.

2. Плотность реберного заполнения графа организации и графа задачи. В проведенном исследовании допускалось, что плотность реберного заполнения графов может быть сравнима с 50 % (т. е. наихудший вариант, с наибольшей плотностью реберного заполнения, сверх которой задачу выгоднее решать на графах, дополнительных к исходным, которые будут иметь меньшую плотность реберного заполнения и, соответственно, большую производительность алгоритма). В реальных организациях с множеством узлов и различных типов ресурсов высокая плотность реберного заполнения не достижима, как в следствие ограниченного доступа к ресурсам различных членов кооперации, так и в виду разнородности ресурсов и невозможности совместного использования ресурсов совершенно различных типов.

3. Размер графа задачи. Во многих случаях для решения задачи необходимо использовать лишь часть, имеющихся в распоряжении организации ресурсов (как человеческих, так и аппаратно-программных), часть ресурсов никак не могут использоваться совместно ввиду их несовместимости, и т. д. Это все факторы приводящие к тому, что во многих случаях размер графа задачи многократно меньше размера графа сети, что так же является фактором приводящим к повышению реальной производительности алгоритма.

В рамках современных организаций производительность алгоритма поиска ресурсов порядка сотен вершин графа вполне достаточна и позволяет решать реальные задачи поиска и управления ресурсами в современных организациях, решающих комплексные задачи как разработки сложного программного обеспечения, так и других областей деятельности.

ВЫВОДЫ

В работе представлен граф-аналитический подход к решению задачи поиска и резервирования ресурсов

организации для решения комплексных задач, основанный на поиске граф-подграф изоморфизма для графов, взвешенных и помеченных по вершинам.

В работе представлен точный алгоритм решения задачи поиска граф-подграф изоморфизма для графов, взвешенных и помеченных по вершинам, описана структура алгоритма и приведены результаты оценки его производительности на тестовых графах, показывающие возможность его практического применения для решения реальных задач поиска и управления ресурсами организации.

СПИСОК ЛИТЕРАТУРЫ

1. *Haijun, Zhang*. Research on co-reservation in the manufacturing grid system / Haijun Zhang, Yefa Hu, Zude Zhou // The International Journal of Advanced Manufacturing Technology. – 2010. – Volume 47, Issue 5–8. – P. 699–717.
2. *Christoph, Langguth*. Optimizing resource allocation for scientific workflows using advance reservations / Christoph Langguth, Heiko Schuldt // Proceeding SSDBM'10 Proceedings of the 22nd international conference on Scientific and statistical database management. – Springer-Verlag. – 2010. – P. 434–451.
3. *Ильяшенко, М. Б.* Алгоритм нахождения графподграф изоморфизма для взвешенных графов и его применение / Ильяшенко М. Б. // Радиоэлектроника, информатика, управление. – 2007. – № 1. – С. 62–68.
4. *Ильяшенко, М. Б.* Разработка и исследование параллельного алгоритма проверки граф-подграф изоморфизма / Ильяшенко М. Б. // Радиоэлектроника, информатика, управление. – 2006. – № 1. – С. 63–69.
5. *Пинчук, В. П.* Распознавание изоморфности графов: ПНВ-алгоритм / В. П. Пинчук // Складні системи і процеси. – 2002. – № 1. – С. 4–11.
6. *Пинчук, В. П.* Основанная на волновом разложении система инвариантов для простых графов и алгоритм распознавания изоморфности / В. П. Пинчук // Киев, 1995. – Деп. в ГНТБ Украины 10.05.95, N 1002 – Ук 95.

Стаття надійшла до редакції 05.11.2013.

Після доробки 25.03.2014.

Ильяшенко М. Б.

Канд. техн. наук, доцент, Запорізький національний технічний університет, Україна

ВИРШЕННЯ ЗАДАЧІ ПОШУКУ ГРАФ-ПОДГРАФ ІЗОМОРФІЗМУ ДЛЯ РОЗПОДІЛЕННЯ РЕСУРСІВ ОРГАНІЗАЦІЇ

У роботі запропоновано граф-аналітичний підхід для розподілу ресурсів організації. Він базується на алгоритмі пошуку граф-підграф ізоморфізму для зважених та помічених графів і є розвитком алгоритму пошуку граф-підграф ізоморфізму для зважених графів.

Ключові слова: резервування ресурсів, граф-підграф ізоморфізм, зважені графи, помічені графи, граф-аналітичний підхід.

Ilyashenko M.

Doctor of Philisiphy, Associate Professor, Zaporizhzhya National Technical University, Ukraine

GRAPH-SUBGRAPH ISOMORPHISM PROBLEM SOLVING FOR ORGANIZATION RESOURCES DISTRIBUTION

The paper presents graph-analytical approach for organizations resources distribution. It based on graph-subgraph isomorphism algorithm for weighted and labeled graphs and can be considered as development of graph-subgraph isomorphism algorithm for weighted graphs proposed before.

Paper describes requirements and specifics of human and technical resources reservation in modern distributed organizations, that can have rather complicated structure, taking into account relations between available resources, and specifics of requirements in resources provided by complicated tasks that need to be solved by organizations. All types of resources considered as weighted and labeled graphs.

Next presented advanced version of graph-subgraph isomorphism algorithm enhanced to work with graphs both weighted and labeled by vertexes. Provided full set of preliminary conditions aim to narrow main combinatorial part of algorithm, where branch and bound method used to find final substitution.

Finally presented numerical results of algorithm benchmarking on randomly generated sets of graphs represented most severe possible conditions for realistic graphs organization in possible practical applications and grounded developed algorithm productivity being enough for solving real scale resources distribution problems on existed hardware.

Keywords: resources reservation, graph-subgraph isomorphism, weighted graphs, labeled graphs, graph-analytical approach.

REFERENCES

1. Haijun Zhang, Yefa Hu, Zude Zhou Research on co-reservation in the manufacturing grid system, *The International Journal of Advanced Manufacturing Technology*, 2010, Volume 47, Issue 5–8, pp. 699–717.
2. Christoph Langguth, Heiko Schuldt Optimizing resource allocation for scientific workflows using advance reservations, *Proceeding SSDBM'10 Proceedings of the 22nd international conference on Scientific and statistical database management*, Springer-Verlag, 2010, pp. 434–451.
3. Il'yashenko M. B. Algoritm naxozhdeniya grafpodgraf izomorfizma dlya vzveshenny'x grafov i ego primenenie, *Radio Electronics, Computer Science, Control*, 2007, No. 1, pp. 62–68.
4. Il'yashenko M. B. Razrabotka i issledovanie parallel'nogo algoritma proverki graf-podgraf izomorfizma / M.B. Il'yashenko, *Radio Electronics, Computer Science, Control*, 2006, No. 1, pp. 63–69.
5. Pinchuk V. P. Raspoznavanie izomorfnosti grafov: PNV-algoritm, *Skladni sistemi i procesi*, 2002, No. 1, pp. 4–11.
6. Pinchuk V. P. Osnovannaya na volnovom razlozhenii sistema invariantov dlya prosty'x grafov i algoritm raspoznavaniya izomorfnosti. Kiev, 1995, Dep. v GNTB Ukrainy' 10.05.95, N 1002 – Uk 95.