

АВТОМАТИЗАЦІЯ ТА ІНТЕЛЕКТУАЛІЗАЦІЯ ПРИЛАДОБУДУВАННЯ

УДК 004.75, 004.724.2

TRAFFIC REGULATION AND REPUTATION HANDLING IN THE BITTORRENT PEER-TO-PEER NETWORKS

*Poryev G. PhD, Rudyk T. PhD, Sulima O. PhD, National Technical University of Ukraine
“KPI”, Kiev, Ukraine*

Various methods for traffic regulation and reputation handling in the distributive and client context of BitTorrent network are analyzed. The overview of the methods for the calculation of the user reputation in the private trackers and corresponding reputation-based access systems are performed

Introduction

Peer-to-peer networks were not nearly new at the beginning of modern century, and their concept was briefly outlined in the times of Internet very inception back in 1969. Although the contributors could not possibly have predicted the future scale of worldwide distribution of what was then a single link between just two mainframe computers, the idea of interconnected peer nodes was already there.

User interface terminals at the time were nowhere near to compare with host computers (mainframes), and were essentially lacking any computing and storage facilities whatsoever, hence the vision of peering networks remained dormant for long time since.

Only as the mainstream computers surged into the consumer market during 1970s and 1980s, the legacy of what we know today as “client-server architecture” was to be dominant for decades to come. It was assumed that should there be a network, it is naturally divided into servers (that provide access to resources) and clients (that make use of provided resources). The performance and capacity gap between server and client hardware and, which is more important, a difference between network interconnections was still too obvious.

At that time, peering was common practice when dealing with server software and network architecture. TCP/IP routing schemes was essentially peering to the point that the very word “peering” made it into the specific technical term on inter-networking routing, despite the fact that actual physical channels had (and still have) visible relevance to national backbones and traffic exchange points, making them more or less subordinate to each other. However, Usenet and e-mail servers were communicating with each other and there were no such thing as primary layer or central hub(s) through which all traffic should be passed – which is peering network.

Outside of Internet, attempts to build peering networks were also undertaken. One of the most successful of those attempts was FidoNet – amateur worldwide computer network, initially consisting of independent bulleting board systems (BBS), built on packet-switching principle over regular telephone lines using dialup modems. Unlike

Internet, FidoNet is not online-network and all user interaction could be and mostly done in offline state. Host software, however, is required to maintain online availability during the certain policy-defined hours each day.

Right upon emerging, the FidoNet was truly peering, in the sense that each originating node accessed its addressee directly by calling its address (phone numbers in this case). Later in 1990-x, however, FidoNet had also “suffered” from infrastructure growth, when the network had exploded into thousands of nodes worldwide. These times of FidoNet development were marked with strict hierarchical structure, roughly based on geography and various regulating authorities within the network. It is worth noting, that unlike Internet (IPv4 address space making up 2^{32} addresses, including non-routable and reserved), hierarchical address structure of FidoNet theoretically allowed address space of 2^{48} network nodes alone and 2^{64} connection points in total.

Despite all aforementioned advances and peeks into the future concept, truly peer-to-peer online networks as we understand them today were far from reach before the advent of third millennium.

The commercial grounds for real peer-to-peer networks have appeared not until permanent Internet connections (also called then “leased lines”) built on technologies such as ADSL or DOCSIS gained significant consumer market at homes and offices. In addition, not until average home and office computer hardware was closing to the average server hardware (often being built from the same parts indeed) was it plausible to build peer-to-peer networks with evenly distributed computing and storage resources [1].

It is widely believed, that commercial applications of the concept started to appear and gained much popularity in the beginning of XXI century.

An introduction to BitTorrent technology

One of the modern peer-to-peer network protocols, BitTorrent, was conceived in 2001 and to date remains responsible for largest part of consumer-generated Internet traffic, sometimes prompting Internet Service Providers (ISPs) to implement special, often unpopular, filtering measures and devices.

Unlike other popular peer-to-peer networks such as eDonkey2000 or Gnutella networks, BitTorrent does not constitute a single addressing or naming space. It is not even a network itself, because BitTorrent operates as multitude of independent content-tracking servers, called “trackers”. Each tracker maintains the list of published content entities, and for each entity, it maintains the list of peers associated with it. Most trackers do not communicate with each other, as eDonkey2000 servers do, unless they are sharing same content and are specially designed to exchange information among themselves.

Due to the absence of overhead related to maintaining global naming or addressing space, BitTorrent networks are quite faster in comparison with eDonkey2000 or Gnutella in terms of download and upload speed and length of download queues. BitTorrent clients are most likely to consume their bandwidth to exhaustion, despite

the fact that BitTorrent does not imply sophisticated load-balancing algorithms for upload, reward scores and so on [2].

Typical content lifecycle in BitTorrent could be described as the following. Preparation – content publisher prepares torrent file, which describes the number, names and size of files and the control checksums of each slice of binary stream made up from content files. Publication – publisher uploads torrent file in such a way that tracker became aware of its existence, not necessarily knowing all the details specified in the torrent file. Distribution – publisher distributes torrent file among clients who wish to download its content. It is usually done through web-based forums, either public or private or via other means. It is worth noting that publication and distribution is not the same process, although in most cases they are done simultaneously in the scope of one server. For example, uploading torrent file as file attach to the message on forum automatically registers torrent contents in the tracker. Initial seeding – publisher running BitTorrent-compliant client starts accepting incoming requests for content. Leeching – other clients proceed to download published torrent file, requesting tracker for the address of initial seeder and requesting initial seeder for content. Downloading – clients actively downloading content file will enable already downloaded slices to be shared among other clients, effectively speeding up the transfer for them. Secondary seeding – clients that completed the download, engage in seeding it by themselves. End of interest – all involved clients finishes and became seeders, and no downloading clients are left in the swarm. Fadeout – seeders stop seeding one by one, and eventually there are neither seeders nor downloading clients associated with this torrent.

Once the content entity is fully downloaded (the transition between stage 6 and 7), the BitTorrent client must ascertain the data integrity of it. In this part BitTorrent specification seems to be slightly under-developed in comparison with its counterparts of eDonkey2000 and Gnutella networks. While the latter does use sophisticated tree-hashing algorithms designed to minimize traffic overhead, BitTorrent simply calculates hashing stream from binary stream with variable-sized chunks. If an error is detected, the whole chunk needs to be re-downloaded.

Analysis of Load-Balancing technique

Most peer-to-peer network will eventually encounter the phenomenon called “leeching”. The network client involving in leeching will only download content and not share it among others. Although such behavior is necessary for some time just after initial publication of the content (since some time is required to download at least one complete shareable piece of data), leeching beyond necessary period and for long time is considered bad, because it forces excess resource usage on other clients interested in the same content [3].

Peer-to-peer networks often employ various sophisticated algorithms to discourage leeching.

One of prominent example is the credit reward system found on popular eDonkey2000 clients. Such clients maintain a “performance record” for each incoming client, who expressed interest in published content.

Typically, incoming clients are arranged into queue in order of time of their appearance. The foremost client in queue is served by the content piece and then re-scheduled at the end of queue, therefore advancing other queue members.

However, incoming client can advance queue member by more than single step in the queue, taking into account its contribution (in case the sharing client is not completed seeder, of course). That is, the more content pieces were provided by the incoming client, the faster it progresses in the queue. This effectively places “bad” leechers to the end of queue and slows their advance.

Unfortunately, no such reward system is currently employed by the majority of the BitTorrent clients. There are number of reasons for it, including the aforementioned difference in distribution speed (BitTorrent content usually distributes faster than comparable eDonkey2000 counterpart due to small size of swarm). However, similar scheme are designed in so called “private trackers”.

As BitTorrent is developing technology, new protocol extensions are constantly added to improve the overall efficiency of content sharing. These include, for example, so-called “Fast Peer Extensions” to allow new peers bootstrap into swarm more rapidly. Although it is uncertain whether the performance itself is nearly topping its potential for the current BitTorrent development stage, it is beyond the scope of this paper.

Public vs. Private Trackers

Roughly, trackers can be called “public” or “private”. Public tracker, such as famous Sweden-based ThePirateBay usually does not require invitation or registration to be able to download its advertised content, therefore do not maintain download and upload rating records of its users.

In contrary, private trackers, such as Torrents.Ru or many others running TorrentPier software, do implement some restrictions against anonymous access. This is possible using so-called private keys – special passwords attached to the announce URL of tracker, designed so that the tracker could ascertain the user identity of every announce or update request coming from BitTorrent clients.

Private trackers often employ rating system, where rating is a value calculated using various formulas including overall download and overall upload amount of a particular user. Users with low rating are restricted from further downloading or they are potential candidates to be banned from tracker. Users with high rating have certain privileges such as ability to download more torrents simultaneously, priority to access and search across tracker, etc.

Hence in order to encourage content sharing and discourage leeching, tracker server must somehow be made aware of how much some particular BitTorrent client did download and upload to others. This is currently made by issuing special HTTP request (“tracker updates”) to the tracker. Such requests usually contain user identity,

content identity (hash), client activity state, amount of downloaded and uploaded data and other relevant information [4].

Proposed speed-up based on logical distance measurement

As either public or private trackers became popular, commonly encountered overloading problems may arise. Although trackers itself do not store any shared content and the storage of torrent-files themselves require comparatively low resources, the “tracking” itself takes much up the processor speed and memory consumption. This was the reason for many popular public trackers to separate tracking services from forum and torrent file storages to a dedicated server or server clusters.

However efficient this solution might be, we believe that the expansive approach is not the only nor it is optimal. As peer-to-peer technology develops rapidly, the traffic its implementations generate became more and more noticeable in overall Internet traffic, as mentioned above. Modern end-user connection technologies such as ADSL, DOCSIS and end-user optical fiber etc, made high-speed Internet connections available to virtually every technically experienced customer.

Despite this fact, the network latency still plays important role in peer-to-peer applications.

It is usually up to the vision of tracker software authors, how to report seeds and peers available to new clients. Every tracker implement its own balancing mechanism, some tend to shift balance to non-completed peers about to become seeds, others tend to report seeds more than ordinary peers.

Complex methods involving calculations regarding which parts are distributed across swarm more frequently than others, are currently not implementable, as BitTorrent protocol does not allow specific piece information to be sent in regular tracker update request.

However, what was left obvious is the load balance based on logical proximity of network nodes. Although it is commonly encountered phenomenon whereas a network packet designated to neighboring building may travel slower than the packet designated to another continent, the understanding of the relative logical position of network nodes may help packet to travel faster. It is widely used practice to build national Internet Traffic Exchange Points (IXes). Countries such as Ukraine, have single exchange point (UA-IX), whereas the geographically large country such as United States of America, have seven exchange points.

Implementation of exchange points generally allow involved members to peer internet traffic to each other on mutually free-of-charge agreements thus implicitly providing customers with higher traffic speeds with resources linked under the same exchange point.

Consider the single shared content over BitTorrent network swarm, to which the newly interested client connects and requests. The tracker, which is generally unaware of the logical proximity of new client to the existing peers in swarm, reports them either randomly or based on some internal optimization algorithm. Client then proceeds to request each received peer for shared content, and, naturally, might ex-

perience faster responses if some of the remote party happened to be located under the same internet exchange point, or even linked to the same ISP.

Conclusion and recommendation

Implementation of “logical topology”-based algorithms for peer selection in either public or private trackers could potentially speed up the content distribution in BitTorrent swarms as well as with any other similar peer-to-peer technology, where clients are obliged to inquire many peer clients periodically.

Social engineering means to encourage content downloaders may also help distribute shared content more efficiently, for example, in the systems where the number of peers and their actual network proximity depends on the user rating or otherwise calculated contribution value.

This research was conducted as a part of state-funded program under the government contract Ф25/624-2007.

References

1. Stephanos Androutsellis-Theotokis, Diomidis Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies // ACM Computing Surveys, 2004. – 36(4). – P. 335–371.
2. Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. Technical Report UW-CSE-01-06-02, University of Washington, Department of Computer Science and Engineering, July 2001.
3. Poryev G.V. The Application of the Peer-to-Peer Network Technologies // Proceedings of Scientific Workshop of Donetsk National Technical University. Issue #12(118) “Computing Technology and Automation”. – DNTU, Donetsk (Ukraine), 2007. – p. 150.
4. Poryev G.V. Data Integrity Control in the Distributed Networks // Western-European Magazine on Advanced Technologies. Issue №4/2(22). – KNURE, Kharkiv (Ukraine), 2006. – P.32-35.

<p>Порев Г.В., Рудик Т.А., Сулима О.В. Регулировка траффика и обработка рейтинга в одноранговых сетях BitTorrent Методы регулировки траффика и обработки пользовательской репутации в сетях с одноранговой коммуникацией BitTorrent. Рассмотрены различные методы регулировки траффика и обработки пользовательской репутации в контексте сетей BitTorrent. Выполнен обзор методов вычисления пользовательской репутации и способов разграничения качества сервиса на их основе.</p>	<p>Порев Г.В., Рудик Т.А., Сулима О.В. Регулювання трафіку та обробка рейтингу в однорангових мережах BitTorrent Розглянуті різні методи регулювання трафіку та обробки користувацької репутації в контексті мереж BitTorrent. Виконано огляд методів обчислення користувацької репутації та способів розділення якості сервісу на їх основі.</p>
---	--

Надійшла до редакції
27 листопада 2008 року

УДК 536.2:536.42:532.66

ВЫБОР МЕТОДОВ ОБРАБОТКИ СОЧЕТАНИЯ ПОВЕРХНОСТЕЙ
ДЕТАЛИ С ИСПОЛЬЗОВАНИЕМ ПОДСИСТЕМЫ ИНТЕЛЛЕКТУАЛЬНОГО

АНАЛИЗА КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКИХ ДАННЫХ

Лапа М.В., Севастопольский национальный университет ядерной энергии и промышленности, г.Севастополь, Украина

Статья посвящена исследованию возможности автоматизированного синтеза различных схем обработки сочетания (групп) поверхностей детали

Введение

Многообразие средств и методов обработки поверхностей и детали в целом, наличие различного оборудования и оснастки является причиной многовариантности задачи проектирования технологического процесса изготовления детали и необходимости выбора наиболее рационального варианта ТП обработки детали, отвечающего условиям конкретной производственной системы. Задача многовариантного проектирования технологических процессов (ТП) заключается в том, чтобы на основе схем обработки отдельных поверхностей детали построить граф допустимых вариантов структуры технологического процесса обработки детали в целом, а затем выбрать вариант, имеющий наименьшую себестоимость [1]. В работах [1,2] для постановки и решения такой задачи использовалась модель ТП в виде графа, вершинами которого являются состояния поверхности, дугами – технологические переходы. Однако задача построения такого графа программными средствами решена не была, также поиск оптимального варианта ТП также ограничен был возможностями вычислительной техники. Существующие САПР технологических процессов используют диалоговый режим для проектирования схем обработки как отдельных поверхностей, так и отдельных частей ТП (групп поверхностей). При этом используются методы проектирования с использованием одного или нескольких ТП-аналогов (путем заимствования фрагментов ТП), с использованием дерева конструкторско-технологических элементов (САПР Вертикаль), библиотеки пользователя (архив наиболее часто используемых технологом операций, переходов, конструкторско-технологических элементов или их сочетаний, фрагментов ТП). В [3,5] рассмотрен алгоритм автоматизированного синтеза различных альтернативных схем обработки отдельных поверхностей детали.

Исследуем возможности автоматизированного синтеза различных схем обработки сочетания (групп) поверхностей детали.

Цель статьи

Синтез схем обработки группы поверхностей детали на основе вариантов обработки отдельных поверхностей. Разработка алгоритма автоматизированного построения графа допустимых вариантов обработки сочетания поверхностей с использованием подсистемы интеллектуального анализа данных.

Постановка задачи

Для построения вариантов ТП обработки сочетания поверхностей детали

(СПД), состоящей из K поверхностей, предлагается построить ориентированный граф, как объединение подграфов $G_i(U_{N1-N2})$, содержащих возможные схемы обработки для каждой отдельной поверхности [3,5]. Порядок обработки поверхностей Π_i задается технологом в диалоговом режиме:

$$\Pi = \{ \Pi_1, \Pi_2, \dots, \Pi_k \}.$$

В качестве сочетания поверхностей могут быть: поверхности со стороны правого торца, поверхности со стороны левого торца, поверхности основной конструкторской базы и др.

Тогда граф $GT(TR, TE)$, содержащий все возможные варианты обработки сочетания поверхностей детали определяется как

$$GT(TR, ER) = \bigcup_{i=1}^k G_i(U_{N1-N2}) \cup GS(US, ES), \quad (1)$$

где $G_i(U_{N1-N2})$ – подграфы вариантов обработки отдельных поверхностей детали [3,4,5], $GS(US, ES)$ – часть графа $GT(TR, TE)$, содержащая связи между подграфами для отдельных поверхностей $G_i(U_{N1-N2})$ [6]. Для формирования подграфов и связей между ними используем подсистему интеллектуального анализа интервальных данных о качестве и шероховатости поверхностей.

Рассмотрим предлагаемый подход на примере двух поверхностей: Π_1 – наружная цилиндрическая поверхность, Π_2 – плоская поверхность. Для каждого типа поверхностей построим модель в виде графа, вершинами которого являются методы обработки поверхности, дуга связывает две вершины R_n и R_k если метод обработки k может быть применен после метода обработки n . Знания о возможности последовательного применения методов обработки поверхности могут быть введены в диалоговом режиме в процессе интервьюирования эксперта-технолога либо автоматизировано с помощью алгоритма, рассмотренного в [3-6]. В таблице 1 приведены данные о точности и качестве при обработке наружных цилиндрических поверхностей различными методами согласно [8,9,10], используя их построим граф $G1(R1, E1)$ возможности последовательного применения методов обработки для наружных цилиндрических поверхностей, матрица смежности графа приведена в таблице 2.

Таблица 1 – Точность и качество наружных цилиндрических поверхностей

	Обработка	IT_{min}	IT_{max}	Ra_{min}	Ra_{max}
R_1	обтачивание черновое	14	12	50	6,3
R_2	обтачивание получистовое, однократ.	13	11	25	1,6
R_3	обтачивание чистовое	10	8	6,3	0,4
R_4	обтачивание тонкое	9	6	1,6	0,2
R_5	шлифование предварительное	9	8	6,3	0,4
R_6	шлифование чистовое	7	6	3,2	0,2
R_7	шлифование тонкое	6	5	1,6	0,1
R_8	притирка, суперфиниширование	5	4	0,8	0,1
R_9	алмазное выглаживание	10	5	0,8	0,05

Таблица 2 – Матрица смежности графа G1(R1,E1)

	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈	R ₉
R ₁	0	1	0	0	0	0	0	0	0
R ₂	0	0	1	0	0	0	0	0	0
R ₃	0	0	0	1	1	0	0	0	1
R ₄	0	0	0	0	0	1	1	1	1
R ₅	0	0	0	0	0	1	0	0	1
R ₆	0	0	0	0	0	0	1	0	1
R ₇	0	0	0	0	0	0	0	1	1
R ₈	0	0	0	0	0	0	0	0	1
R ₉	0	0	0	0	0	0	0	0	0

Пусть значения качества ИТ и шероховатости R_a поверхности Π₁ заготовки (14, 6.3), требуемые значения ИТ, R_a поверхности детали (9,1.6). Зная начальные и конечные значения качества и шероховатости поверхности, выделим подграф G₁(U_{1-N1-N2}) графа G1(R1,E1), содержащий все возможные схемы обработки поверхности Π₁ (рис.1). Каждый альтернативный путь на подграфе - это схема обработки поверхности, позволяющая получить поверхность с заданными значениями качества ИТ и шероховатости R_a.

На рис.1 в скобках для каждой вершины графа указаны подмножества пар значений качества и шероховатости P_i. Подмножества P_i формируются с помощью рекурсивного алгоритма, используя данные таблицы 1 в зависимости от количества методов обработки, которыми можно достичь данные значения качества и шероховатости поверхности (таблица 3, фрагмент).

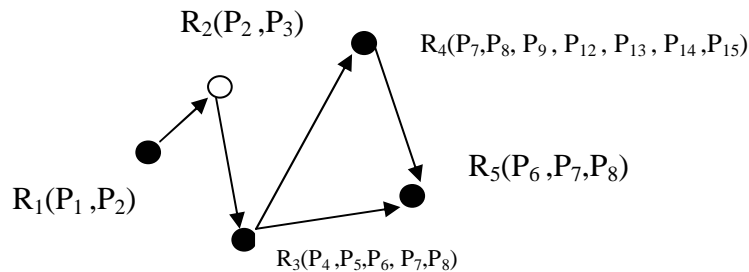


Рис.1 Подграф G₁ (U₁₋₅)

Таблица 3 – Подмножества значений (ИТ, R_a) и методы обработки поверхности

Подмножество значений (ИТ, R _a)	Двоичный булев вектор (набор) \tilde{R}^n	Наборы значений (ИТ, R _a)	Альтернативные методы обработки (или)
P ₁	100000000	(14;50), (13;50), (12;50), (14;25), (14;6,3)	Обтачивание черновое
P ₂	110000000	(13;25), (12;25), (13;6,3), (12;6,3)	Обтачивание черновое, ИЛИ обтачивание полуступенчатое однократное
...		...	
P ₁₉	000000010	(4;0,8), (4;0,4), (4;0,2), (4;0,1)	Притирка, суперфиниширование

Каждому подмножеству P_i ставим в соответствие конъюнкцию вида $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5 \wedge R_6 \wedge R_7 \wedge R_8 \wedge R_9$ и т.д.

Получим 19 элементарных конъюнкций – функций алгебры логики (ФАЛ). Каждой конъюнкции поставим в соответствие двоичный номер, где R_n соответствует значению 1 (данный квалитет и шероховатость могут быть получены методом R_n), $\neg R_n$ соответствует значению 0 (данный квалитет и шероховатость не могут быть получены методом R_n), табл. 3. Количество единиц в двоичном номере («вес») соответствует количеству альтернативных методов обработки, которыми можно получить точность и шероховатость наружной цилиндрической поверхности, соответствующую данному подмножеству значений квалитета и шероховатости $P_1 - P_{19}$ (табл.3). В нашем примере подмножество значений квалитета и шероховатости поверхности Π_1 заготовки P_1 представлено функцией алгебры логики

$$R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5 \wedge R_6 \wedge R_7 \wedge R_8 \wedge R_9, \quad (2)$$

(булев набор 100000000, формируется подсистемой интеллектуального анализа данных, рис. 2, см. табл.5), подмножество значений квалитета и шероховатости поверхности детали P_7 представлена функцией алгебры логики $\neg R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5 \wedge R_6 \wedge R_7 \wedge R_8 \wedge R_9$, (булев набор 001110000, формируется подсистемой интеллектуального анализа данных, рис.2).

	1	2	3	4	5	6
1	0	1	0	1	1	0
2	0	0	0	0	1	0
3	0	0	0	0	0	1
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0

Рис. 2. Формирование ФАЛ подсистемой интеллектуального анализа данных

Выделим подграф $G_1 (U_{1N_1-N_2})$ графа $G_1(R_1,E_1)$ (рис.1) с множеством вершин $U_{n_1-n_5} \subset R_1$. Множество $U_{1N_1-N_2}$ включает все вершины исходного графа с номерами от N_1 до N_2 , где N_1 - номер первой переменной без знака отрицания в ФАЛ (2) для подобласти заготовки или номер первой переменной, равной «1» в соответствующем двоичном булевом наборе (для нашего примера $N_1=1$), N_2 – номер последней переменной без знака отрицания в ФАЛ для подобласти детали или номер последней переменной, равной «1» в соответствующем двоичном булевом наборе (для нашего примера $N_2=5$). На рис.1 приведен подграф G_1

(U_{1-5}). Матриця смежності для виділеного підграфа $G_1(U_{1-5})$ виділена в табл.2 сірим кольором.

Пусть Π_2 плоская поверхность, которую необходимо обработать после наружной цилиндрической поверхности Π_1 . Шероховатость и качество поверхности заготовки (25;14), детали (3,2;11). Булев набор для подмножества значений качества и шероховатости заготовки 000100000000000, формируется подсистемой интеллектуального анализа данных. Булев набор для подмножества значений качества и шероховатости детали 010010010000100, формируется подсистемой интеллектуального анализа данных. Аналогично выделяется подграф $G_2(U_{2_{N1-N2}})$ из графа методов обработки плоских поверхностей $G_2(R_2,E_2)$. Матрица смежності для підграфа $G_2(U_{2_{4-13}})$ варіантів обробки плоской поверхности Π_2 приведена в таблице 4 (подмножество $U_{2_{4-13}}$ содержит вершины графа $G_2(R_2,E_2)$ с 4 по 13).

Таблица 4 – Матрица смежності для підграфа $G_2(U_{2_{4-13}})$

	R ₄	R ₅	R ₆	R ₇	R ₈	R ₉	R ₁₀	R ₁₁	R ₁₂	R ₁₃
R ₄	0	0	0	1	0	0	0	0	0	0
R ₅	0	0	0	0	1	0	0	0	0	1
R ₆	0	0	0	0	0	1	0	1	0	0
R ₇	0	0	0	0	0	0	0	0	0	0
R ₈	0	0	0	0	0	0	0	0	0	1
R ₉	0	0	0	0	0	0	0	1	0	0
R ₁₀	0	0	0	0	0	0	0	0	0	0
R ₁₁	0	0	0	0	0	0	0	0	0	1
R ₁₂	0	0	0	0	0	0	0	0	0	0
R ₁₃	0	0	0	0	0	0	0	0	0	0

Построим объединенную матрицу смежності для вариантов обработки сочетания поверхностей Π_1, Π_2 (рис.3). Множество вершин графа $TR = U_{1-5} \cup U_{2_{4-13}}$. Нижняя левая часть матрицы смежності объединенного графа заполняется нулями, (строки 6-15, столбцы 1-5 в нашем примере), рис.3. В верхней правой части необходимо проставить связи между двумя подграфами следующим образом. Построим $GS(US,ES)$ - часть графа $GT(TR,TE)$, содержащую связи между подграфами для отдельных поверхностей $G_i(U_{i_{N1-N2}})$ (формула 1), используя ФАЛ, сгенерированные подсистемой интеллектуального анализа данных (ПИАД). В булевом наборе 001110000, сгенерированном ПИАД для подобласти детали поверхности Π_1 , см.табл.3, выделим часть, соответствующую вершинам подграфа $G_1(U_{1-5})$: 00111, в булевом наборе для подобласти заготовки поверхности Π_2 (000100000000000) выделим часть, соответствующую подграфу $G_2(U_{2_{4-13}})$ - 1000000000. Из вершин подграфа $G_1(U_{1-5})$, соответствующих значениям «1» в булевом поднаборе для подобласти детали поверхности Π_1 (т.е. из вершин 3,4,5 в нашем примере), необходимо провести связывающие дуги в вершины подграфа $G_2(U_{2_{4-13}})$, соответствующие значениям «1» в булевом поднаборе для подобласти заготовки поверхности Π_2 1000000000, и заполнить «1»

в соответствующих позициях матрицы смежности объединенного графа. На рис.3 представлен результат автоматизированного формирования матрицы смежности для графа вариантов обработки рассмотренных двух поверхностей – наружной цилиндрической и плоской.

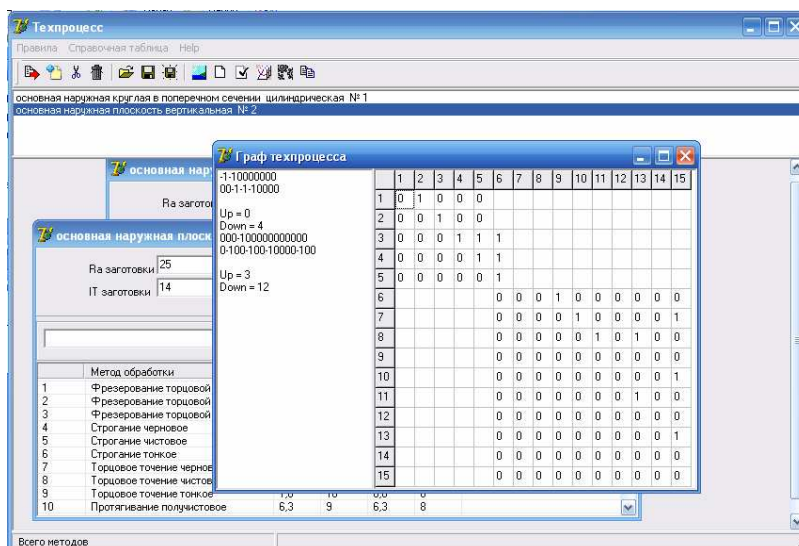


Рис.3. Автоматизированное построение матрицы смежности для графа вариантов обработки сочетания поверхностей

На рис.4 представлен результат объединения двух подграфов – граф, содержащий все альтернативные маршруты обработки двух заданных поверхностей.

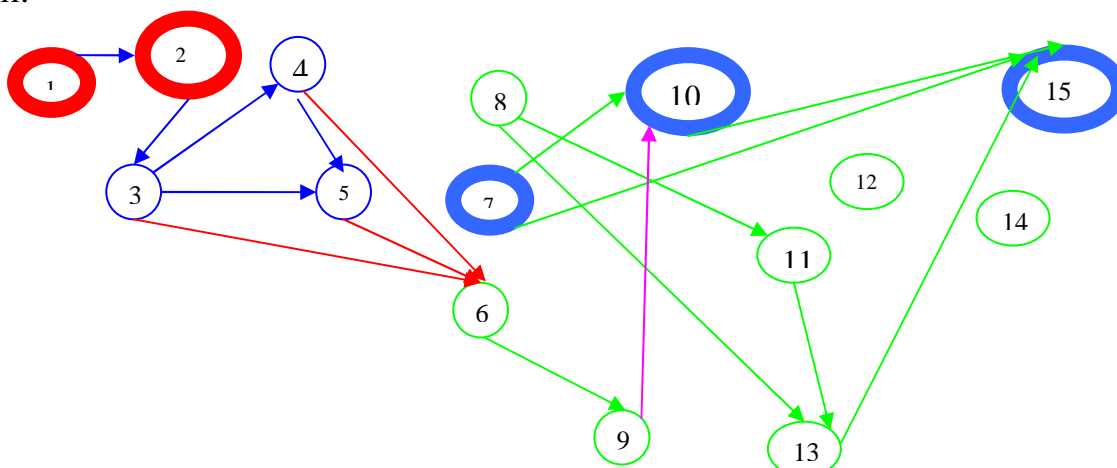


Рис.4. Маршруты обработки сочетания двух поверхностей P_1 и P_2

Используя алгоритм поиска «DFS» на полученном графе, получим все возможные варианты обработки сочетания поверхностей P_1 и P_2 с заданными значениями качества, шероховатости поверхностей заготовки и детали.

Выводы

Разработанная графовая модель позволяет автоматизировать синтез различ-

ных вариантов обработки сочетания поверхностей на уровне переходов. Предложенный алгоритм и программный модуль позволяет автоматизированно формировать матрицу смежности графа. Алгоритм поиска альтернативных методов обработки на полученном графе с использованием рекурсивного алгоритма «в глубину» позволяет сократить время поиска и требования к вычислительным ресурсам. В дальнейшем целесообразно исследовать возможности поиска наиболее оптимального варианта обработки группы поверхностей с помощью алгоритма Дейкстры и полученного графа.

Литература

1. Цветков В.Д. Система автоматизации проектирования технологических процессов. – М.: Машиностроение, 1972. – 240 с.
2. Глоба Л.С. Інтелектуальна комп'ютерна технологія конструкторсько-технологічної підготовки механо-оброблюючого виробництва: Дис.д-ра.техн.наук: 05.12.13. - К., 1996. – 304 с.
3. Лапа М.В. Формирование нечеткого исходного множества технологических альтернатив для выбора методов обработки поверхностей // Открытые информационные и компьютерные технологии: Сб. науч. тр. Вып. №31. – Харьков: Нац. аэрокосмич. ун-т, 2006. – С. 215–220.
4. Лапа М.В. Інтелектуальні технології обробки та ефективного використання техніко-економічної інформації підприємства для створення нечіткої бази конструкторсько-технологічних знань // Тезиси доповідей XIV міжнарод. науч.-техн. конф. «Автоматика-2007».-Севастополь:СНУЯЕтаП. – 2007. – С.76–78.
5. Лапа М.В. Создание базы знаний машиностроительного предприятия с использованием подсистемы интеллектуального анализа конструкторско-технологических данных // Вестник Харк. нац. ун-та. Сер. "Математическое моделирование. Информационные технологии. Автоматизированные системы управления". – 2007. – № 780. – С.151–162.
6. Бардачев Ю.Н., Соколова Н.А., Ходаков В.Е. Основы дискретной математики. –Херсон: Издательство ХГТУ, 2000. – 356 с.
7. Руденко П.О. Проектування технологічних процесів у машинобудуванні. –К.: Вища школа, 1993. – 413 с.
8. Косилова А.Г., Мещеряков Р.К. Справочник технолога машиностроителя. –М.: Машиностроение, 1985. – Т. 1 – 655 с.

Лапа М.В. Вибір методів обробки поверхонь деталі з використанням підсистеми інтелектуального аналізу конструкторсько-технологічних даних
Стаття присвячена дослідженню можливості автоматизованого синтезу альтернативних схем обробки груп поверхонь деталі.

Lapa M.V. Choise of detail surfaces processing methods with use of the engineering data intellectual analysis subsystem
Article is devoted to research of an opportunity of the automated synthesis of various plans of detail surfaces combinations processing.

*Надійшла до редакції
21 березня 2008 року*