УДК 519.682.2

В. Вальковський[1], докт. фіз.-мат. наук; Ю. Рак[2], докт.техн. наук; М. Кухта[1]

[1]*Національний університет "Львівська політехніка"*

[2]*Тернопільський державний технічний університет імені Івана Пулюя*

ОПТИМІЗАЦІЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ ПРИ ВИПУСКУ ПОЛІГРАФІЧНОЇ ПРОДУКЦІЇ ШЛЯХОМ РОЗПАРАЛЕЛЮВАННЯ

Досліджуються граничні можливості прискорення випуску друкованої продукції шляхом розпаралелювання технологічних процесів. Аналізуються відповідні теоретичні результати із сфери інформатики. Здійснено адекватне перенесення цих результатів в галузь поліграфії. Викладення ілюструється на прикладах.

Постановка задачі.

Існує широко відома і, без сумніву, правильна думка, що більшість цікавих результатів у науці і техніці досягаються на межі суміжних чи навіть далеких між собою напрямків. Стаття, що пропонується, теж знаходиться на межі двох напрямків досліджень. Один із них – паралельна обробка інформації – відноситься до сфери теорії програмування й обчислювальної техніки. Інший – технологія випуску друкованої продукції – стосується поліграфії. Матеріал даної статті демонструє подвійне "проникнення" ідей і результатів із одного напрямку в інший. Абстрактні математичні результати першого напрямку використовуються при проектуванні оптимальних технологічних ліній оперативного друку. Зворотній рух полягає в тому, що абстрактні формальні моделі мають конкретну інтерпретацію. Це може служити їх цілеспрямованому розвитку і наближенню до практичних потреб.

Якщо говорити більш конкретно, то стаття присвячена застосуванню методів розпаралелювання обробки інформації при паралельній реалізації процесів друку. В якості базису будуть використані джерела [1,2]. Перша монографія розглядає проблеми максимального розпаралелювання обчислень. В другій пропонуються деякі підходи до формалізації, аналізу і синтезу технологічних ліній оперативної поліграфії. В ній же зроблені перші кроки до проектування паралельних технологічних ліній. Розпаралелювання процесів друку зменшує час випуску як одного видання або іншого елементу продукції, так і всього тиражу. Подібним проблемам формалізації і аналізу процесів друку присвячено багато інших праць.

Метою, поставленою авторами цієї статті, є дослідження граничних можливостей при розпаралелюванні процесів друку, тобто, спроба розв'язати задачу мінімізації часу підготовки та випуску поліграфічних видань у відповідності з деякою гіпотетичною складною технологією, яка включає умовні переходи і цикли. Пояснимо це на прикладі схеми S, зображеної на рис. 1. Блок b задає умову, при якій обробка йде по одному із двох альтернативних технологічних ланцюгів $\{b_1, b_2\}$. Умовою може бути наявність необхідної сировини, фінансове забезпечення, дозвіл якої-небудь уповноваженої комісії і т.д. Блок e, в залежності від значення x, скеровує обробку на блок f або на нове проходження циклу. В найпростішій інтерпретації кожне нове проходження може подавати, наприклад, нову "порцію" тиражу. Символи x, y і z, через які блоки схеми пов'язані залежностями, інтерпретуються як одиниці сировини, напівфабрикати, фінанси, інформація про умови видання і тощо.

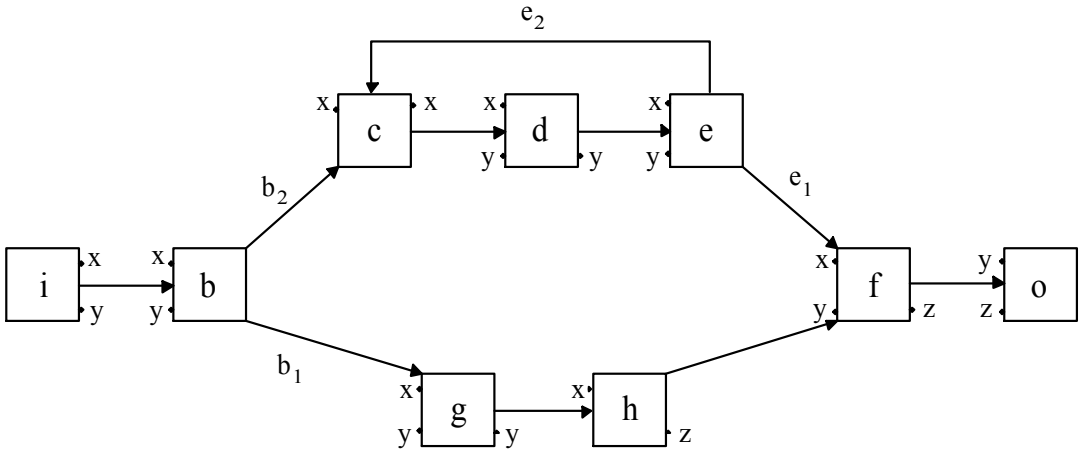

Рис. 1. Послідовна схема

Основні теоретичні результати.

Опишемо основні необхідні для подальшого викладу елементи формалізму і теоретичні результати.

Базою формалізму є так званий інформаційний базис, який включає:

1) $M = \{x, y, z, ...\}$ – множину комірок пам'яті, тобто субстанцій, через які здійснюється зв'язок між блоками.

2) $F = \{b, c, d, ...\}$ – множину операторів (блоків), кожний з яких здійснює деяке елементарне перетворення.

3) Для кожного $b \in F$ визначені множини $in(b) \subseteq M, out(b) \subseteq M$ вхідних і вихідних комірок пам'яті, а також множина $\sum b = \{b_1, b_2, ...\}$ альтернатив (стрілок), які ілюструють одночасно як спрацювання блоку *b*, так і скерування на подальше продовження обробки в напрямку однієї зі стрілок b_i.

Для схеми *S*, яка розглядається на рис. 1, базисом є наступний набір: $M = \{x, y, z\}$; $F = \{i, b, c, d, e, f, g, h, o\}$, де *i* і *o* – вхідний і вихідний оператори; $in(i) = Ø$ (Ø – пуста множина), $in(c) = in(h) = \{x\}$, $in(b) = in(d) = in(e) = in(f) = in(g) = \{x, y\}$, $in(o) = \{y, z\}$; $in(i) = \{x, y\}$; $out(b) = out(e) = out(o) = Ø$; $out(c) = \{x\}$, $out(d) = out(g) = \{y\}$, $out(f) = out(h) = \{z\}$; $\sum b = \{b_1, b_2\}$; $\sum e = \{e_1, e_2\}$. $\sum o = Ø$. Інші оператори мають по одній вихідній стрілці, тобто $\sum i = \{i_1\}$, $\sum c = \{c_1\}$, $\sum d = \{d_1\}$, $\sum f = \{f_1\}$, $\sum g = \{g_1\}$, $\sum h = \{h_1\}$. Щоби не захаращувати рисунок, стрілки на ньому цими символами не позначені.

Наступне важливе поняття – обчислювальний процес *X* над інформаційним базисом (*M, F*).

Якщо відійти від громіздкого суворого означення, можна сказати, що обчислювальний процес (в нашому контексті – процес випуску одиниці поліграфічної продукції) є послідовністю спрацьовування операторів (блоків) з одночасною видачею однієї з альтернатив. Для даного прикладу процесами будуть наступні послідовності:

$$X = i_1 b_1 g_1 h_1 f_1 o_1;$$

$$Y = i_1 b_2 c_1 d_1 e_1 f_1 o_1;$$

$$Y = i_1 b_2 c_1 d_1 e_2 c_1 d_1 f_1 o_1;$$

$$\vdots$$

$$Y_k = i_1 b_2 c_1 d_1 e_2 c_1 d_1 f_1 o_1.$$

У позначеннях з метою "канонічності" використовується символ вихідної стрілки o_1, хоч на рисунку її немає.

Схема S (рис. 1) визначається безвідносно якої-небудь графової структури як функція на префіксах, тобто початкових відрізках своїх же процесів. Значенням цієї функції є множина операторів, дозволених для виконання після реалізації префікса-аргумента. Позначивши як ε пустий префікс, схему S можна описати так:

$S(\varepsilon)=\{i\}$, $S(i_1)=\{b\}$, $S(i_1b_1)=\{g\}$, $S(i_1b_1g_1)=\{h\}$, $S(i_1b_1g_1h_1)=\{f\}$, $S(i_1b_1g_1h_1f_1)=\{o\}$, $S(i_1b_1g_1h_1f_1o_1)=\text{Ø}$.

Це значення схеми, яке відповідає проходженню шляху x.

Проходженню по верхній частині схеми S відповідають наступні її значення:

$S(i_1b_2)=\{c\}$, $S(i_1b_2c_1)=\{d\}$, $S(i_1b_2c_1d_1)=\{e\}$, $S(i_1b_2c_1d_1e_1)=\{f\}$, $S(i_1b_2c_1d_1e_1f_1)=\{o\}$, $S(Y)=\text{Ø}$ – проходження схеми без зациклювання.

$S(i_1b_2c_1d_1e_2c_1d_1)=\{e\}$ – це ж саме з зациклюванням на k ітерацій. Послідовний характер схеми S відбивається в тому, що результатом S на будь-якому аргументі є один і тільки один оператор, окрім, звичайно, префіксів, які співпадають з самими процесами, на яких значення S рівне Ø.

Переходимо до проблем розпаралелювання технологічних ліній в сенсі введеного формалізму. В роботах [1,2] розрізняються "економне максимальне розпаралелювання" і "абсолютно максимальне розпаралелювання". Пояснимо різницю між ними знову ж таки на нашому прикладі схеми S_1 рис.1. Нехай умова b пов'язана, наприклад, з виборами в народні депутати. По лінії b_1 йде технологічний ланцюг, який відповідає випуску друкованої продукції до і після виборів для одного, більш "авторитетного" з точки зору соціальних досліджень, кандидата в депутати. Наприклад, він бажав би, щоб після його обрання терміново розповсюдили 100 000 примірників його книги, і цей тираж він авансом оплачує. На противагу йому виступає менш "авторитетний" депутат, який регулює випуск своєї друкованої продукції не тільки за результатами виборів, але й за попитом на неї. Це відображає верхня частина схеми рис. 1. На обох суб`єктів надходять замовлення, сукупність яких виконавець може сформулювати по-різному.

1. Забезпечити в короткі терміни випуск замовленої продукції, не звертаючи уваги на затрати.

2. Випустити за мінімальний час замовлену продукцію, не витрачаючи виділених коштів дарма.

Що залишається робити виробнику продукції?

Всі ці підходи формалізовані й обгрунтовані у вигляді математичних теорем.

Теорема 1. Існує алгоритм розпаралелювання будь-якої схеми S, який перетворює цю схему в максимально паралельну схему S_1, яка еквівалентна схемі S. При цьому всі спрацювання операторів схеми S_1 не є зайвими або збитковими. Це означає, що будь-яке проходження схеми S_1 "на паралельний манер" містить ті і тільки ті оператори, які в будь-якому випадку виконуються схемою S при проходженні "послідовним чином".

Теорема 2. Існує алгоритм розпаралелювання будь-якої схеми S, який перетворює цю схему в максимально паралельну схему S_2, яка еквівалентна схемі S. При цьому схема S_2 допускає спрацювання зайвих операторів, які не використовуються далі. Зате за рахунок цих додаткових витрат, алгоритм, який пропонується даною теоремою, є абсолютно максимально розпаралелюючим. Жодний інший алгоритм у жодному іншому існуючому чи розробленому в майбутньому формалізмі без інформації про особливості інтерпретації не може згенерувати більш швидкого обчислення, навіть у динамічному режимі.

Застосування в галузі поліграфії.

Викладена нами інтерпретація технологічних структур в термінах формалізму і дві наведені неформально інтерпретовані теореми дозволяють здійснити просування поліграфічних задумів, сформульованих у роботі [2], на найбільш абстрактному математичному рівні. Саме це розв'язує сформульовану задачу: дослідити граничні можливості прискорення випуску поліграфічної продукції.

Конкретизуємо викладені вище математичні результати з теореми 1. Алгоритм, який підтримує цю теорему, є найбільш "економним" і дозволяє, як правило, зобразити вихідну схему у вигляді графа. В нашому випадку цей алгоритм заснований на наступному принципі необхідності: в кожний момент часу призначати на оброблення ті блоки, котрі:

- так чи інакше виконаються при будь-якому протіканні процесу;
- для яких до цього моменту вже готові всі вхідні дані (читай, вихідні матеріали).

Використовуючи відповідний алгоритм, наведений в [1], можемо отримати паралельну технологічну схему, яка задовільняє наступні умови:

а) вона забезпечує випуск замовленої друкованої продукції за мінімальний можливий термін;

б) при її реалізації вся виготовлена, а також проміжна продукція (напівфабрикати) "не пропадає дарма", тобто використовується далі в кінцевому продукті;

в) весь кінцевий продукт має попит.

Щодо схеми S досліджуваного прикладу згадуваний алгоритм буде мати результатом схему S_1, яка зображена у вигляді графа на рис. 2.

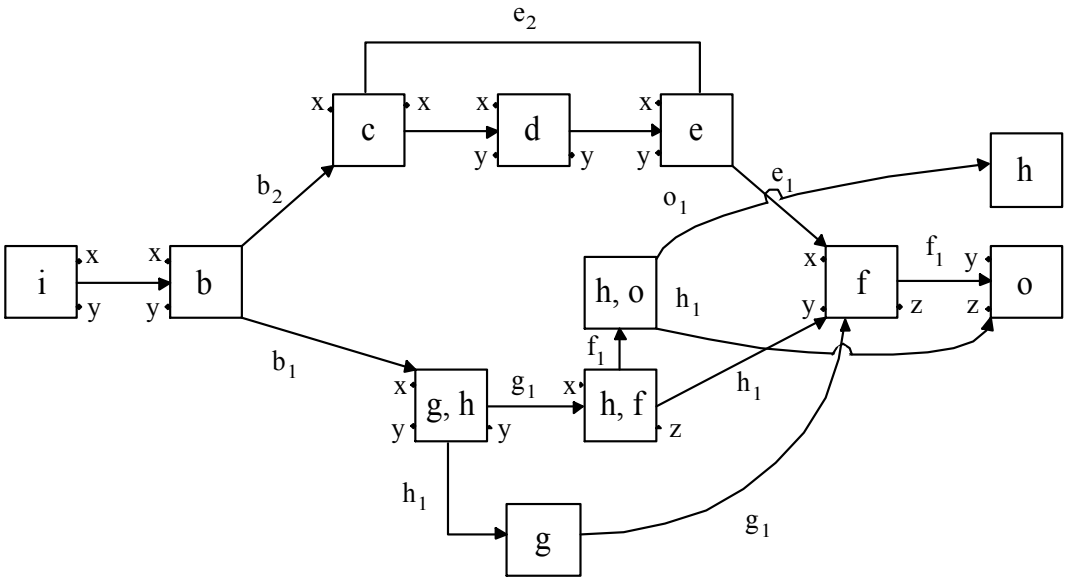

Рис. 2. Паралельна схема (варіант 1)

На відміну від рис. 1, деякі із блоків рис. 2 включають не один, а два оператори. Це означає, що при досягненні блоку b, включені в нього оператори можуть бути запущені в довільному порядку. Наприклад, проходячи по стрілці b_1, бачимо, що оператори g і h можуть бути виконані незалежно один від одного. На це вказує блок, який містить g і h. Якщо після цього виконується оператор g, який виробляє необхідну для оператора f змінну y, то попадаємо в блок, який містить оператор h, який ще не виконаний, і додатково новий оператор f. Нехай після цього виконається оператор f. Тоді з'явиться можливість одночасного виконання h і o. Враховуючи те, що o завершує технологічну лінію, стає очевидною безкорисливість оператора h у цьому процесі. Таким чином, процедура розпаралелювання водночас здійснює оптимізацію технологічної лінії.

Паралельне виконання операторів *g* і *h* може бути передбачено парою операторів fork і goin, які задають розпаралелювання і злиття процесів. Відповідний граф технологічної лінії зображений на рис. 3.

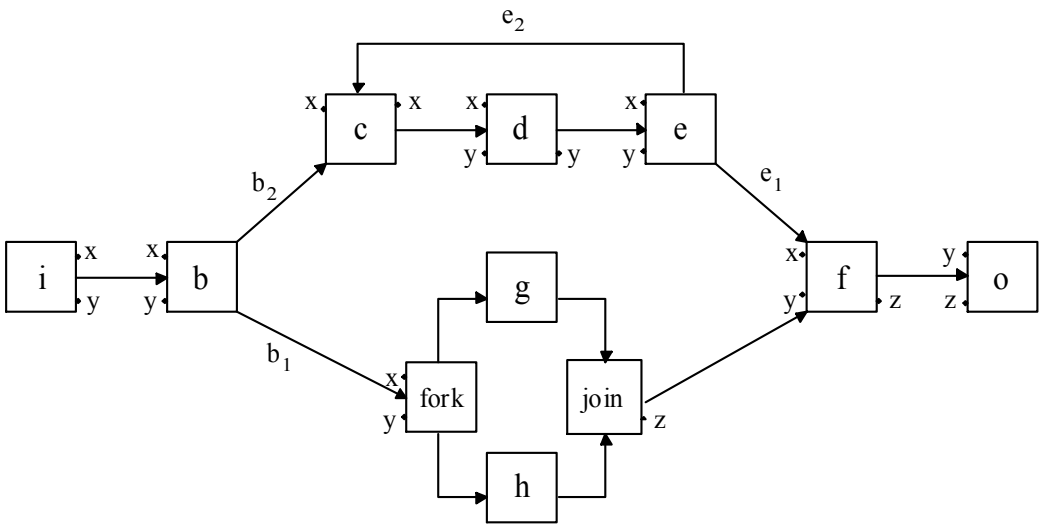

Рис. 3. Паралельна схема (варіант 2)

На противагу першому алгоритму, алгоритм, який відповідає теоремі 2, є найбільш неекономним. Він заснований на принципі можливості: в кожний момент часу слід назначати на оброблення ті блоки, для яких:

- не виключена можливість їх використання при виробленні кінцевого продукту;
- до цього моменту вже готові всі вхідні дані (вхідні компоненти).

За аналогією з алгоритмом із теореми 1, алгоритм теореми 2 дає паралельну схему S_2, яка задовільняє пункт "а", але не задовільняє пункти "б" і "в". Це відбувається з відомої причині. Так, алгоритм теореми 2 в інтерпретації, наведеній на початку статті, передбачає розпочати випуск продукції по обох технологічних лініях, не чекаючи "рішення комісії *b*". Очевидно, що після виконання один із ланцюгів виявиться "зайвим", і все, що було на ньому напрацьовано, пропаде. Оскільки другий алгоритм обмежений меншою кількістю умов, він дає результат швидше, ніж перший. Але при цьому, як бачимо, є великі додаткові витрати.

Зображення результатної технологічної лінії для другого алгоритму досить проблемне. На рис. 4 результат поданий у вигляді "початку" дерева, яке демонструє можливі шляхи реалізації алгоритму. Прокоментуємо роботу схеми S_2, виходячи з рис. 4. Кожний блок рис. 4 містить запис типу: *S* (реалізований відрізок процесу) = {список можливих після цієї реалізації}.

На початку роботи, тобто при пустому процесі ε, може включитися тільки оператор *i*. Вихід із нього тільки по стрілці i_1.

Поскільки *i* виробляє змінні *x*, *y*, то після його спрацювання, крім неминуче працюючого оператора *b*, з'явиться можливість спрацювання операторів *c*, *g*, *h*. Всі вони "живляться" інформацією з оператора *i*. Для інших операторів потрібна інша вхідна інформація, яка ще не готова. Тому вони не можуть бути запущені, оскільки акт їх запуску в цей момент не може бути використаний ні при якому протіканні процесу. Не може бути запущений навіть оператор *f*, хоча він реалізується при будь-якому проходженні схеми. Для нього ще не вироблений жоден комплект вхідних даних, який йому підходить.

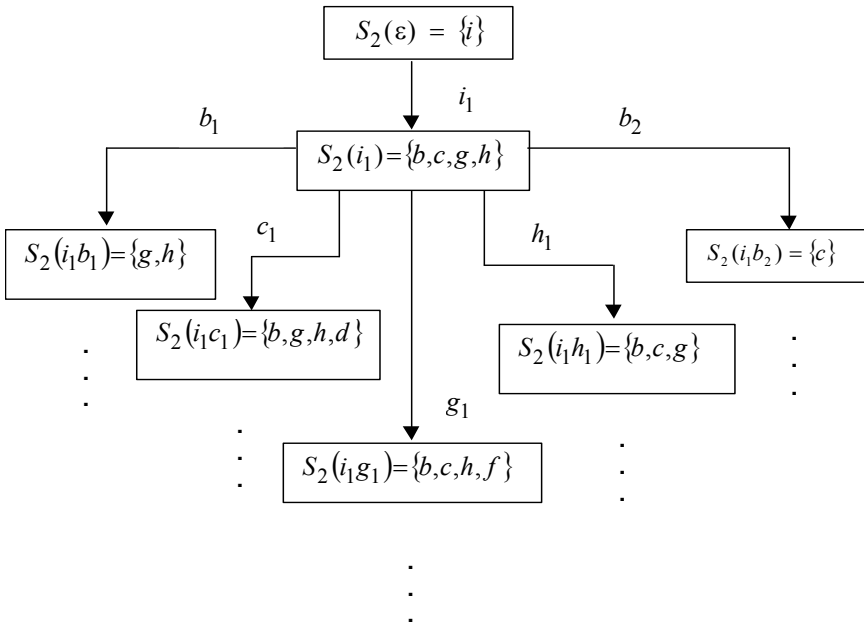

Рис. 4. Дерево реалізації при динамічному розпаралелюванні

Далі процес може піти за п`ятьма альтернативами. Прокоментуємо їх за порядком.

Якщо виконався оператор b і видав альтернативу b_1, то зрозуміло, що виключається можливість виконання c, а залишається можливість виконання g і h. Інші можливості не з`являються. При альтернативі b_2 виключається g, h. Інших можливостей також не виникає. При реалізації c залишається можливість виконання для b, g, h. Крім цього, виникає можливість виконання d, оскільки оператор виробив для нього недостатню вхідну змінну x. При проходженні g_1 в списку кандидатів на реалізацію, крім b, c, h, що залишилися, з`являється f. Оператор g виробив необхідну для нього в цій гілці змінну y. Реалізація h не приносить нічого нового.

Висновки

Таким чином, практичне використання алгоритму 2, якщо результатна схема не може бути проілюстрована у вигляді графу, а слідом, відповідна їй технологічна конструкція не може бути “змонтована”, тоді можна скористатися “послідовним монтажем”, але відповідним чином організованим на ньому процесом випуску друкованої продукції.

The limiting possibilities of acceleration of printed output receiving by the means of technological processes parallelization are investigated. The corresponding theoretical result from the informatics are analised. The adequate transference of these results to the polugraphy area is fulfilled. The presentation is illustrated by the examples.

Література

1. Вальковский В.А. Распараллеливание алгоритмов и программ: структурный подход.– М.: Радио и связь, 1989. – 179 с.
2. Рак Ю.П. Малі друкарські системи прогнозування, аналіз, синтез. – К.: Наукова думка, 1999. – 256 с.

Одержано 20.11.2004 р.