

Алгоритмы решения некоторых вариантов  
задачи выполнения заказов клиентов с  
использованием графических вычислений

Борисовский П.А.

Омский филиал Института математики  
им. С.Л. Соболева СО РАН

Исследование выполнено за счет гранта Российского научного  
фонда № 22-71-10015, <https://rscf.ru/en/project/22-71-10015/>

# Задача выполнения заказов с различными способами производства

$S = \{1, 2, \dots, n\}$  - продукты;

$U = \{1, 2, \dots, m\}$  - устройства;

$I = \{1, 2, \dots, k\}$  - операции;

$D_s$  - спрос на продукт  $s \in S$ .

Для каждой операции  $i$  задано:

$u_i$  - требуемое устройство (только одно)

$s_i$  - производимый продукт (только один)

$r_i$  - производительность, тонн/час.

$s_{ij}$  - время переналадки (если операции  $i$  и  $j$  выполняются на одном устройстве).

Критерий: Минимизация  $C_{max}$ .

# Пример из химического производства

Производство каждого продукта включает пять стадий, главной из которых является **экструзия**:

1. **Подача сырья.**
2. **Временное хранение сырья.**
3. **Экструзия (выдавливание).**
4. **Временное хранение продукта.**
5. **Отгрузка продукта.**

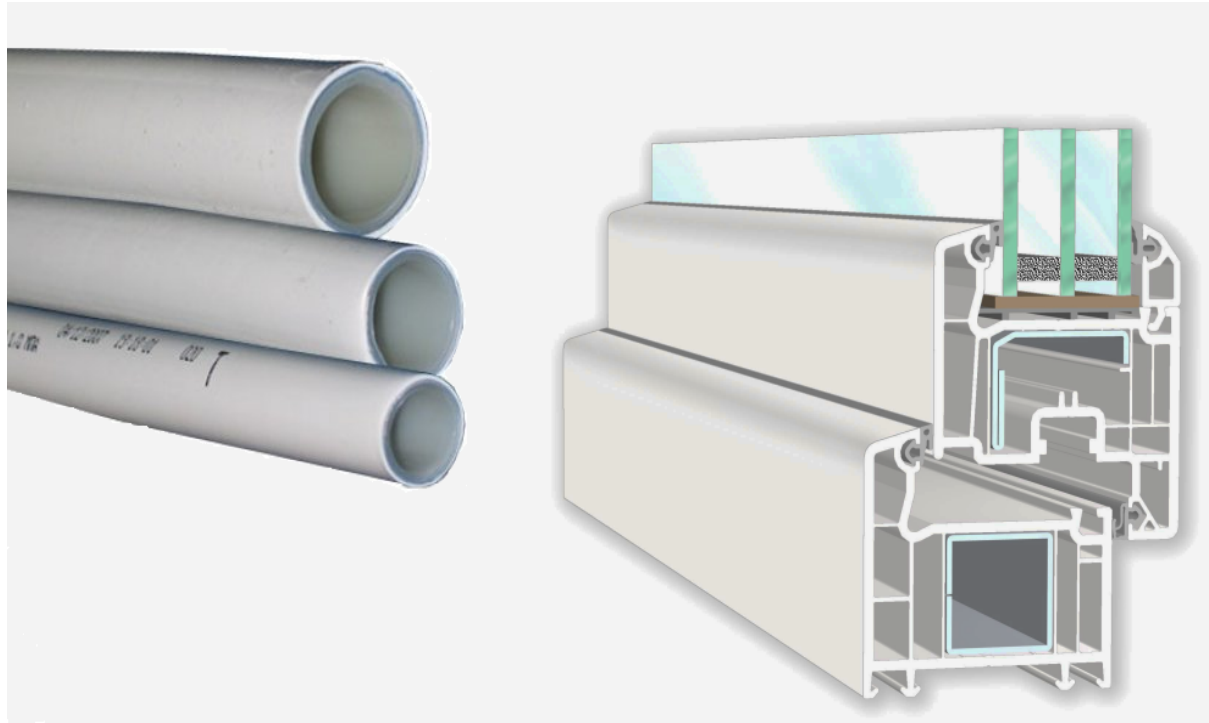


---

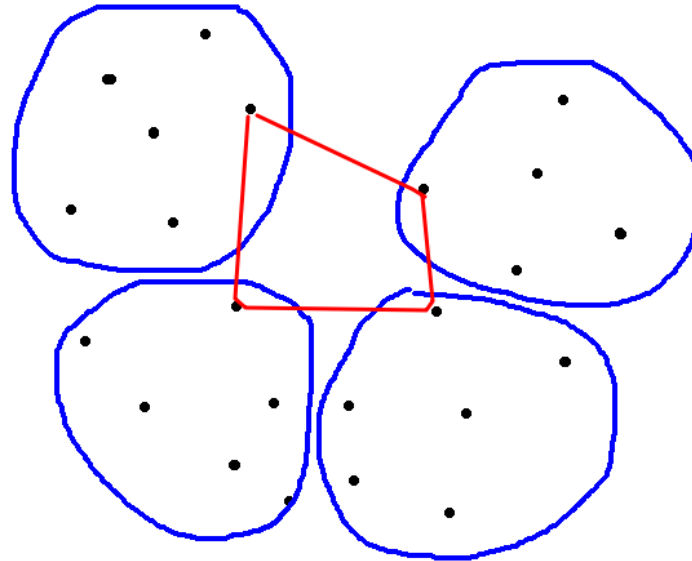
Shaik M.A., Floudas C.A., Kallrath J., and H.J. Pitz. Production Scheduling of a Large-Scale Industrial Continuous Plant: Short-Term and Medium-Term Scheduling // Computers and Chemical Engineering, 33, 670–686 (2009).



Иллюстрация с сайта <https://www.coperion.com/en/products-services/plants-systems/compounding-plants/>



# Частный случай: обобщенная задача коммивояжера

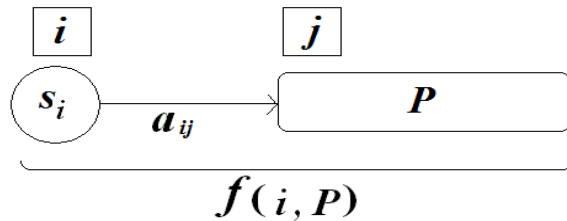


---

P.C. Pop et.al. A comprehensive survey on the generalized traveling salesman problem  
// European Journal of Operational Research, 314, 819–835 (2024).

Вариант 1: каждый продукт должен быть произведен полностью за одну операцию

Алгоритм DP для задачи коммивояжера



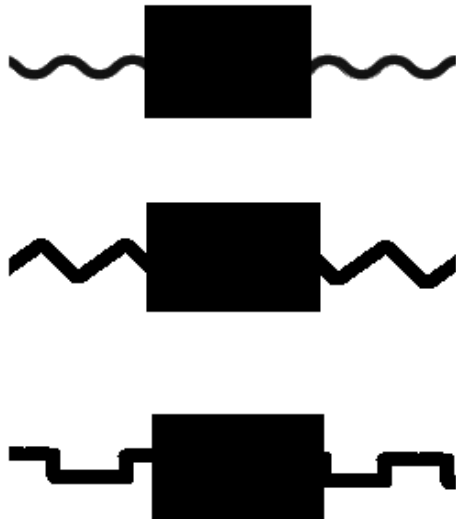
$$f(i, P) = \min_{j \in P} \{a_{ij} + f(j, P \setminus \{j\})\} \quad (1)$$

---

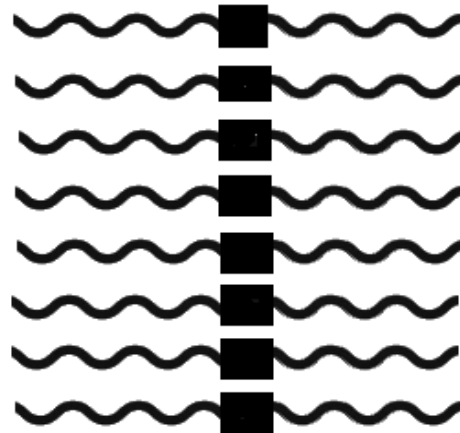
Held M., Karp R.M. A dynamic programming approach to sequencing problems // Journ. of the Soc. for Ind. and Appl. Math. V. 10. - 1962.- P. 196-210.

# Графический процессор (GPU)

**CPU**  
Multiple Instruction  
Multiple Data



**GPU**  
Single Instruction  
Multiple Data



# Решение задачи коммивояжера алгоритмом ДП

| <b><math>n</math></b> | <b>CPU AMD<br/>EPYC 7502 (2019 г.)</b> | <b>GPU Tesla V100<br/>(2017 г.)</b> |
|-----------------------|--|-------------------------------------|
| <b>24</b>             | <b>13.5 s.</b>                         | <b>14 ms.</b>                       |
| <b>25</b>             | <b>30 s.</b>                           | <b>29 ms.</b>                       |
| <b>26</b>             | <b>71 s.</b>                           | <b>65 ms.</b>                       |
| <b>27</b>             | <b>153 s.</b>                          | <b>140 ms.</b>                      |

# Алгоритм ветвей и границ для задачи с несколькими устройствами

Ветвление по продуктам и устройствам. На каждом шаге очередной продукт поочередно размещается на каждом подходящем устройстве. Частичное решение представлено в виде множеств продуктов для каждого устройства. Для оценки суммарных переналадок используется таблица  $F(i, P)$ , полученная алгоритмом ДП на GPU.

| size<br>$ S  \times  U  \times  I $ | CPLEX     |        |       | B&B  |
|-------------------------------------|-----------|--------|-------|------|
|                                     | time lim. | # opt. | r     | time |
| 30x6x120                            | 300       | 0      | 20%   | 1.2  |
| 40x8x140                            | 900       | 2      | 12.9% | <1   |
| 40x8x150                            | 900       | 0      | 18.8% | 75   |
| 40x8x160                            | 900       | 0      | 20.5% | 7.8  |

Borisovsky P. Exact Solution of One Production Scheduling Problem // In: OPTA 2018. Communications in Computer and Information Science, Springer, Cham. 2018. Vol. 871, pp. 56–67.

## Вариант 2: продукты могут быть произведены по частям несколькими операциями

Пусть решение представлено в виде:

- Перестановка операций  $\pi = (i_1, i_2, \dots, i_k)$
- Вектор  $L = (L_1, L_2, \dots, L_m)$  задает сколько операций можно назначить на соответствующее устройство.

Для вычисления целевой функции нужно сначала расставить операции на устройства, вычислить времена переналадок. Затем для продуктов, которым назначена только одна операция, вычислить длительность производства. Для определения длительностей остальных операций нужно решить задачу ЛП.

# Экспериментальное сравнение ГА и CPLEX на случайных задачах

| # | CPLEX |         | GA   |         | GA-DP      |             |
|---|-------|---------|------|---------|------------|-------------|
|   | Cmax  | ch.time | Cmax | ch.time | Cmax       | ch.time     |
| 1 | 297   | 76      | 288  | 59      | <b>281</b> | <b>48</b>   |
| 2 | 334   | 143     | 307  | 88      | <b>291</b> | <b>57</b>   |
| 3 | 274   | 192     | 251  | 129     | <b>234</b> | <b>75</b>   |
| 4 | 327   | 181     | 304  | 114     | <b>295</b> | <b>92</b>   |
| 5 | 311   | 675     | 301  | 539     | <b>281</b> | <b>290</b>  |
| 6 | —     | —       | 603  | 2130    | <b>502</b> | <b>1026</b> |

---

Borisovsky P.A., Ereemeev A.V., Kallrath J. Multi-product continuous plant scheduling: combination of decomposition, genetic algorithm, and constructive heuristic // International Journal of Production Research. V. 58. N.9. - 2020. - P. 2677-2695.

# Задача составления расписания в сборочном производстве

$J$  – множество работ (операций),

$I$  – множество машин.

Каждая работа должна быть выполнена ровно один раз.

$p_{ij}$  – длительность выполнения работы  $j$  на машине  $i$ .

$[e_j, d_j]$  – временное окно для окончания работы  $j$ .

$s_j$  – максимально возможное запаздывание работы  $j$ .

$z_i$  – «стоимость» запуска любой работы на машине  $i$ .

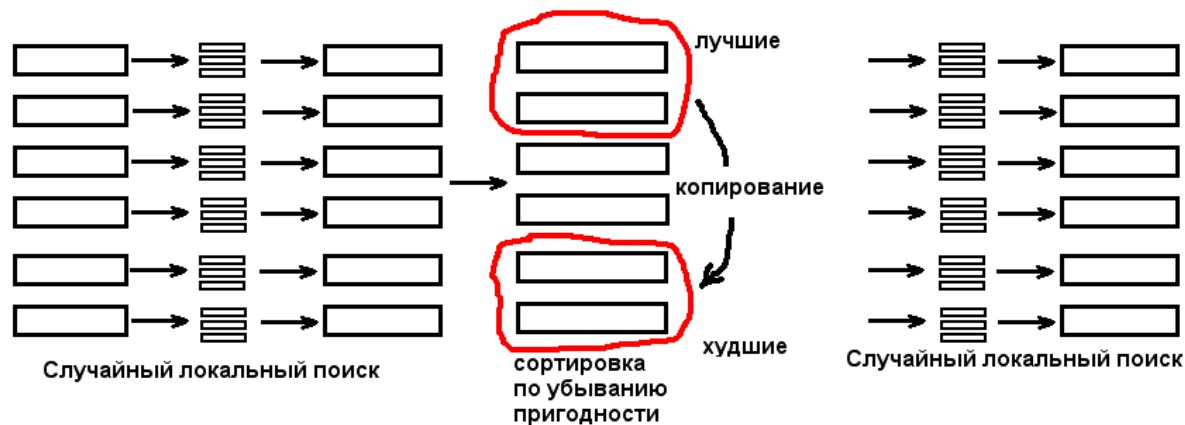
Критерий

$$L + 0.001 \cdot Z \rightarrow \min,$$

где  $L$  – суммарное запаздывание

$Z$  – суммарная стоимость запуска всех работ.

# Гибридный алгоритм итеративного локального поиска со схемой «Иди с победителями»



Алгоритм специально разработан для выполнения на GPU.

# Сравнение алгоритмов

| №  | Gurobi |        | GWW    |        |        | ГА     |        |        |
|----|--------|--------|--------|--------|--------|--------|--------|--------|
|    | ниж.   | верх.  | мин.   | сред.  | макс.  | мин.   | сред.  | макс.  |
| 1  | 121,6  | 402,1  | 122,08 | 122,14 | 122,29 | 125,34 | 132,33 | 145,37 |
| 2  | 128,79 | 128,89 | 128,98 | 129,00 | 129,02 | 129,27 | 138,74 | 148,26 |
| 3  | 140,29 | 480,8  | 140,71 | 141,27 | 142,73 | 143,90 | 152,05 | 162,98 |
| 4  | 84,20  | 84,28  | 84,34  | 84,36  | 84,38  | 85,57  | 91,76  | 104,59 |
| 5  | 70,49  | 70,56  | 71,64  | 72,30  | 73,67  | 71,99  | 80,25  | 89,92  |
| 6  | 119,63 | 119,73 | 119,83 | 119,86 | 120,02 | 121,17 | 131,79 | 142,14 |
| 7  | 124,23 | 124,26 | 124,35 | 124,55 | 125,38 | 124,65 | 127,93 | 133,66 |
| 8  | 70,40  | 70,41  | 70,52  | 71,47  | 72,71  | 73,82  | 79,86  | 87,84  |
| 9  | 191,75 | 520,1  | 192,16 | 192,92 | 195,15 | 198,45 | 209,89 | 227,52 |
| 10 | 68,90  | 109,4  | 69,35  | 70,09  | 71,38  | 69,605 | 73,646 | 82,63  |

---

Борисовский П.А. Параллельный алгоритм «Иди с победителями» для некоторых задач составления расписаний // Дискретный анализ и исследование операций. Том 30, Номер 4. С. 5–23.

# Задача выполнения заказов клиентов с одной машиной и переналадками

$n$  – количество клиентов;

$m$  – количество продуктов;

$p_{ij} \geq 0$  – длительность производства продукта  $j$  для клиента  $i$ ;

$s_{jj'} \geq 0$  время переналадки при переключении с продукта  $j$  на продукт  $j'$ ;

$s'_j$  – начальная переналадка для продукта  $j$ .

Операция – пара  $(i, j)$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, m$ .

Требуется найти перестановку операций.

Критерий:  $\sum_{i=1}^n C_i$ .

| Series<br><i>n-m</i> | GA   |               |         | ILS-G <sub>w</sub> W |         |               |
|----------------------|------|---------------|---------|----------------------|---------|---------------|
|                      | Time | GA-GR         | GA-RN   | Time                 | CPU     | GPU           |
| 20-20                | 1200 | 2.2530        | 5.355   | 800                  | 6.0774  | <b>1.2315</b> |
| 20-50                | 1800 | <b>0.7844</b> | 3.5733  | 1200                 | 7.5514  | 1.516         |
| 20-100               | 3600 | 2.1624        | 6.2971  | 2400                 | 25.8    | <b>0.8039</b> |
| 50-20                | 1800 | 1.8652        | 4.3787  | 1200                 | 7.6236  | <b>1.2252</b> |
| 50-50                | 5400 | 6.9498        | 12.9446 | 3600                 | 25.4435 | <b>0.6407</b> |
| 50-100               | 7200 | 20.48         | 27.4265 | 4800                 | 88.8814 | <b>2.2458</b> |

---

P.A. Borisovsky, A.O. Zakharov, Yu.V. Zakharova. Evolutionary Algorithms for Customer Order Scheduling // The Bulletin of Irkutsk State University. Series Mathematics (In Press).

# Задача с несколькими машинами

$n$  – количество клиентов;

$m$  – количество продуктов и машин, каждая машина производит свой продукт.

$p_{ij} \geq 0$  длительность производства продукта  $j$  для клиента  $i$ .

| Series<br>$n-m$ | LDR-AS and GAs |        |        |        | ILS-G <sub>w</sub> W |        |                |
|-----------------|----------------|--------|--------|--------|----------------------|--------|----------------|
|                 | Time           | LDR-AS | GA-GR  | GA-RN  | Time                 | CPU    | GPU            |
| 50-10           | 9              | 0.002  | 0.0317 | 0.7117 | 6                    | 0.1253 | <b>0.0002*</b> |
| 50-20           | 17             | 0.005  | 0.0049 | 0.7989 | 12                   | 0.189  | <b>0.0006*</b> |
| 100-10          | 46             | 0.0385 | 0.0385 | 1.1307 | 30                   | 0.3007 | <b>0.01787</b> |
| 100-20          | 91             | 0.0307 | 0.0307 | 1.4994 | 60                   | 0.456  | <b>0.016*</b>  |
| 200-10          | 162            | 0.0232 | 0.0232 | 1.2774 | 108                  | 0.4374 | <b>0.0223</b>  |
| 200-20          | 324            | 0.0688 | 0.2393 | 1.5967 | 216                  | 0.723  | <b>0.0491*</b> |

# Решение серии задач ЛП

Если при вычислении целевой функции требуется решение задачи ЛП, то в алгоритме «иди с победителями» нужно параллельно решать большое количество задач ЛП.

## Эксперимент.

250 тысяч симплекс-таблиц 120x80

GPU Tesla V100, OpenCL

7 секунд

CPU AMD EPYC 7502 (OpenCL, POCL driver)

| Кол-во потоков | Время, с. |
|----------------|-----------|
| 1              | 15        |
| 2              | 10        |
| 3              | 6         |
| 4              | 5         |
| 5              | 5         |
| 6              | 4         |
| 7              | 4         |
| 8              | 4         |
| 12             | 2         |
| 16             | 2         |
| 20             | 1         |

Возможные способы ускорения на GPU:

- Исследование других способов распараллеливания и оптимизации работы с памятью.
- Применение специальных методов работы с редкими матрицами.
- Алгоритмы ЛП: метод внутренней точки, методы первого порядка.

**Спасибо за внимание!**