

**Параллельный алгоритм
«Иди с победителями» для некоторых
задач составления расписаний**

П.А. Борисовский

**Институт математики им. С.Л. Соболева СО РАН
(Омский филиал)**

Исследование выполнено за счет гранта Российского научного фонда
№ 22-71-10015,

Цель работы

Разработка параллельного алгоритма решения широкого класса задач составления расписаний (перестановочных).

Качества хорошего алгоритма:

- Точность решений
- Скорость работы
- Универсальность
- Простота реализации
- Масштабируемость

Задача составления производственного расписания с временными окнами (Production Scheduling with Time Windows, PSTW)

Постановка задачи^a.

J – множество работ, 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 .

^a J. Berndorfer, S. N. Parragh. Modeling and solving a real world machine scheduling problem with due windows and processing set restrictions. Procedia Computer Science, V. 200, 2022, P. 1646–1653. <https://doi.org/10.1016/j.procs.2022.01.365>

Задача составления производственного расписания

Критерий.

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

где L – суммарное запаздывание

Z – суммарная стоимость запуска всех работ.

Классическая задача Flow Shop

Множество работ $J = \{j_1, \dots, j_n\}$;

множество машин $I = \{i_1, \dots, i_m\}$.

Каждая работа состоит из m операций, которые должны быть выполнены на машинах i_1, i_2, \dots, i_m . Порядок выполнения работ должен быть одинаковым на всех машинах.

Длительность операции работы j на машине i равна p_{ij} .

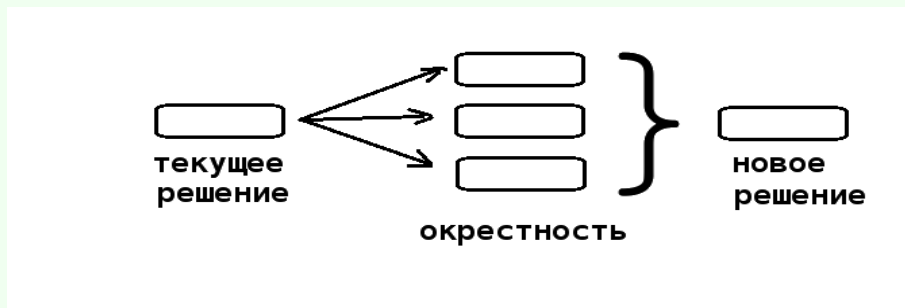
Требуется найти такое упорядочение работ, при котором момент окончания последней работы минимален.

Случайный локальный поиск (Hillclimbing, (1+1)-EA)

1. Построить начальное решение $S^{(0)}$.
2. На каждой итерации t выполнять:
 - 2.1. Сгенерировать случайное решение S' из окрестности текущего решения $S^{(t)}$ (т.е. применить небольшие случайные изменения $S^{(t)}$).
 - 2.2. Если $f(S') < f(S^{(t)})$, то $S^{(t+1)} := S'$, иначе $S^{(t+1)} := S^{(t)}$.

Параллельный случайный локальный поиск

1. Параллельный просмотр окрестности



Алгоритм $(1+K)$ -EA

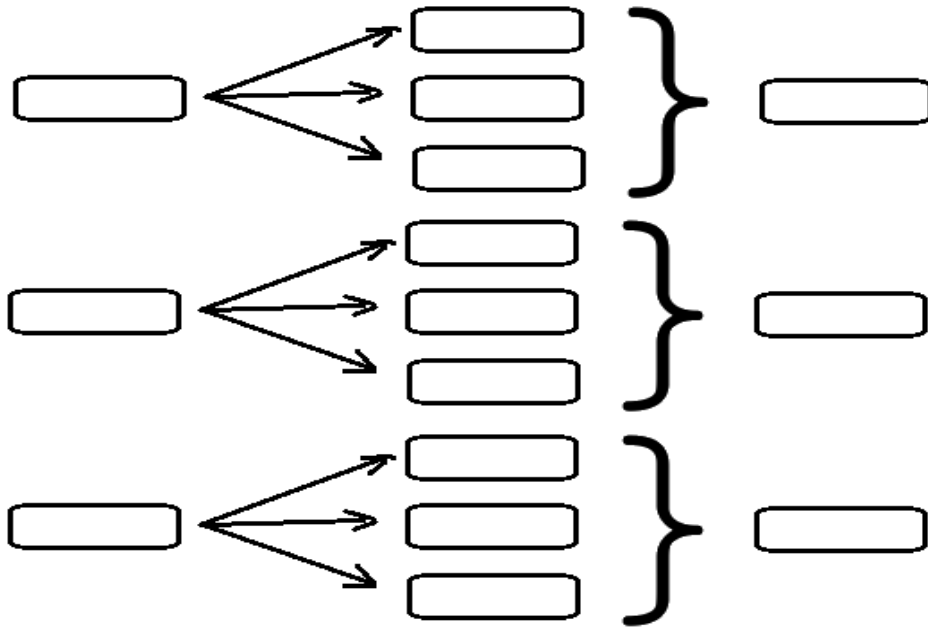
1. Построить начальное решение $S^{(0)}$.

2. На каждой итерации t выполнять:

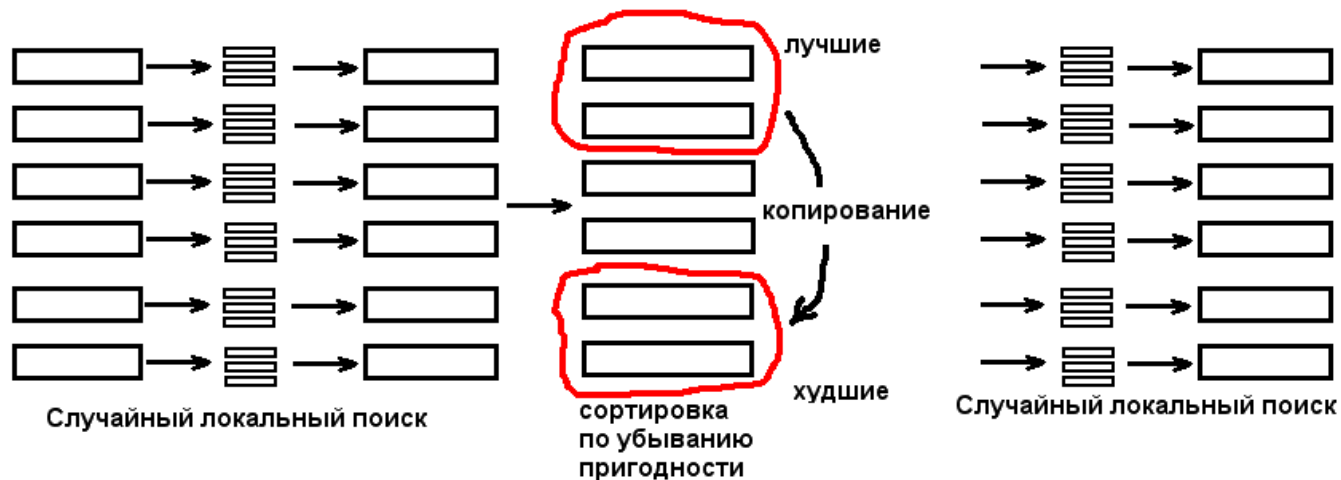
2.1. Сгенерировать K случайных решений S'_1, \dots, S'_K из окрестности текущего решения $S^{(t)}$. Из них выбрать лучшее S^* .

2.2. Если $f(S^*) < f(S^{(t)})$, то $S^{(t+1)} := S^*$, иначе $S^{(t+1)} := S^{(t)}$.

Параллельный локальный поиск с перезапуском

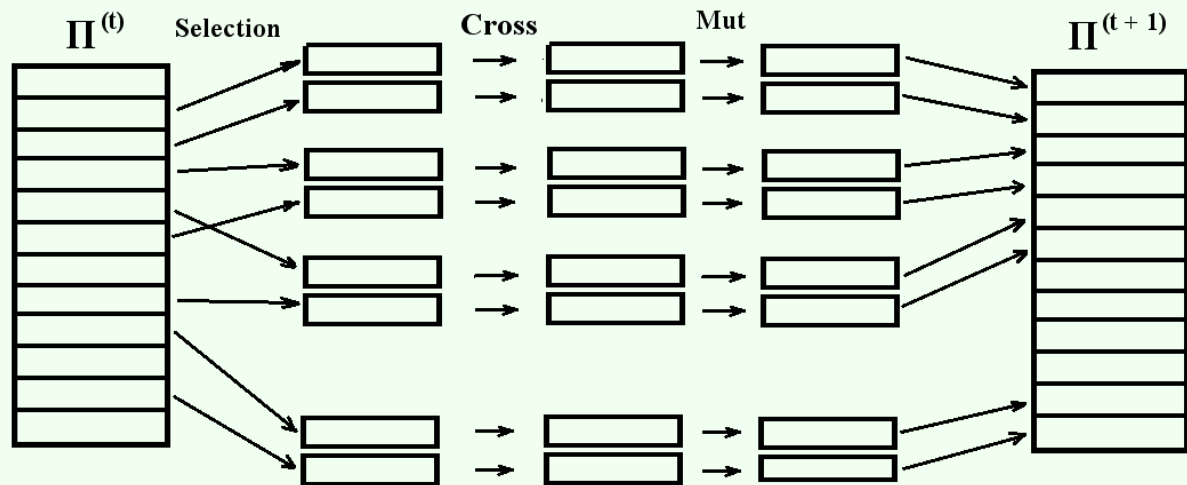


Гибридный алгоритм «Иди с победителями»^a с итеративным локальным поиском



^a Aldous, D., Vazirani, U.: "Go with the winners" Algorithms. Proceedings of 35th IEEE Symposium on Foundations of Computer Science (FOCS), pages 492–501, 1994.

Классический генетический алгоритм



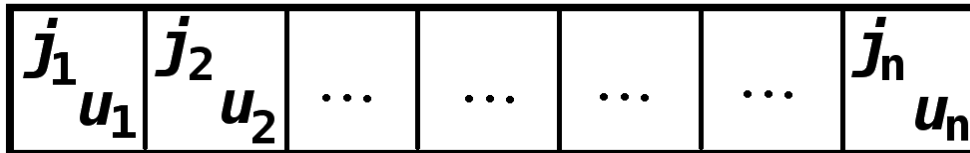
Представление решений

(j_1, j_2, \dots, j_n) – перестановка работ

(u_1, u_2, \dots, u_n) – упорядоченный список машин, например

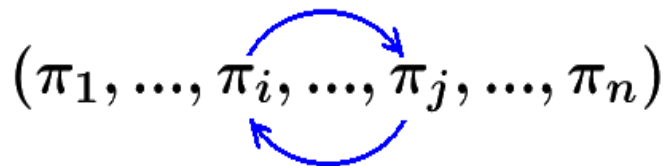
$(1, 1, 2, 2, 2, 3, 3, 3, 3)$.

Решение хранится в виде массива `int [n]`. Для уменьшения обращений к памяти, оба значения (j_k, u_k) записываются в одну ячейку (в старшие и младшие 16 бит).

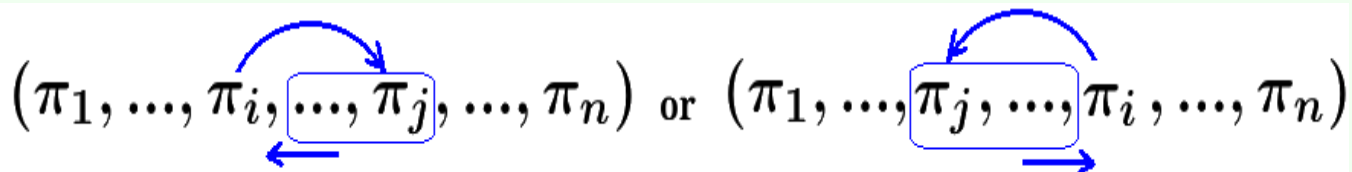


Операторы мутации

1. Swap

$$(\pi_1, \dots, \pi_i, \dots, \pi_j, \dots, \pi_n)$$


2. Insertion

$$(\pi_1, \dots, \pi_i, \dots, \pi_j, \dots, \pi_n) \text{ or } (\pi_1, \dots, \pi_j, \dots, \pi_i, \dots, \pi_n)$$


Сравнение алгоритмов на задаче Flow Shop

Настройки:

GwW: размер популяции **256**, кол-во потомков **$256 \cdot 64 = 16384$**

GA: размер популяции **5000**, размер турнира в селекции **20**.

Таблица 1. Средняя погрешность

$n \cdot m$	T^{\max}	GwW			GA			MA ^a		
		min	avg	max	min	avg	max	min	avg	max
50 x 10	5	0	0.06	0.09	0.09	0.34	0.55	0.03	0.08	0.09
75 x 20	36	0.26	0.56	0.88	1.03	1.51	2.5	0.32	0.67	1.06
50 x 20	14	0.37	0.71	1.07	0.86	1.48	2.24	0.29	0.59	0.92
100 x 10	14	0.02	0.05	0.13	0.15	0.45	0.75	0.03	0.17	0.28
100 x 20	64	0.3	0.55	0.82	0.81	1.36	1.87	0.42	0.79	1.16

^a Borisovsky, P., Kovalenko, Y. A Memetic Algorithm with Parallel Local Search for Flowshop Scheduling Problems // In: Filipic B., Minisci E., Vasile M. (eds) Bioinspired Optimization Methods and Their Applications (BIOMA 2020). 12438 LNCS. - 2020. - P. 201-213.

Сравнение алгоритмов на задаче PSTW

Настройки:

GwW: размер популяции **256**, кол-во потомков **$256 \cdot 512 = 131072$**

GA: размер популяции **5000**, размер турнира в селекции **200**.

Таблица 2. Значения целевой функции

№	Gurobi		GwW			GA		
	ниж.	верх.	min	avg	max	min	avg	max
1	121.6	402.1	122.13	123.00	126.14	125.34	132.33	145.37
2	128.8	128.9	129.02	129.69	133.08	129.27	138.74	148.26
3	140.3	480.8	140.75	141.50	145.73	143.90	152.05	162.98
4	84.2	84.3	84.39	84.95	86.42	85.57	91.76	104.59
5	70.5	70.6	71.70	72.80	74.71	71.99	80.25	89.92
6	119.6	119.7	119.86	120.63	121.91	121.17	131.79	142.14
7	124.2	124.3	124.39	124.56	127.4	124.65	127.93	133.66
8	70.4	70.4	70.56	72.497	76.61	73.82	79.86	87.84
9	191.7	520.1	192.19	193.27	195.22	198.45	209.89	227.52
10	68.9	453.4	69.39	70.10	72.39	69.605	73.646	82.63

Заключение

- Вычисления на GPU могут успешно применяться при решении задач оптимизации в производстве.
- В сравнении с известными аналогами рассмотренный подход позволяет получать похожие по качеству решения, но за значительно меньшее время.
- Особо стоит отметить простоту реализации алгоритма, что позволяет использовать его для широкого класса задач.
- Представляется перспективным опробовать данный подход в комбинации с другими эвристиками, такими как имитация отжига или поиск с запретами.

Спасибо за внимание!