Вычислительная сложность и конструктивные алгоритмы для задачи составления расписаний с распараллеливанием и учетом расхода энергии

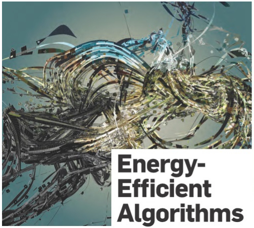
Ю.В. Захарова, М.Ю. Сахно

Институт математики им. С.Л. Соболева СО РАН

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

Motivation (Parallel and Multiprocessor Jobs)

- Parallel jobs require more than one processor at the same time.
- Some jobs can not be performed asynchronously on modern computers. Such situation takes place in multiprocessor graphics cards, where the memory capacity of one processor is not sufficient.
- Many computer systems offer some kinds of parallelism. The energy efficient scheduling of parallel jobs arises in testing and reliable computing, parallel applications on graphics cards, computer control systems and others.



Previous Research: Classic

Makespan

Drozdowski (2009): poly for rigid jobs, pmtn, r_j
approx for rigid jobs, r_j
Brucker (2000), Du, Leung (1989): rigid jobs: NP-hard, strongly NP-hard for prec

Total Completion Time

Lee and Cai (1999): rigid jobs: strongly NP-hard
Schwiegelshohn et. al. (1998), J. Turek et. al. (1994): approximation algorithms for rigid jobs
Hoogeveen (1994): single-mode jobs: NP-hard
Cai (1998): 2-approximation algorithm for single-mode jobs

Previous Research: Energy

Makespan

Pruhs, van Stee (2007), Bunde (2009): poly for single processor, r_j
approx for multiple processors, r_j
Bampis et.al. (2014): approx for prec, r_j

Total Completion Time

Pruhs et. al. (2008), Bunde (2009): poly for single processor
Shabtay, Kaspi (2006): approx for multiple processors

Parallel jobs

Kononov, Zakharova (2017-2022): NP-hardness and approx

Report Structure

- Problem Statement
- Previous Research
- NP-hardness
- Approximation algorithm and lower bound
- Local improvements and experimental evaluation
- Conclusion and Further Research

Speed Scaling Scheduling

Processors and Jobs

$m = 2$ speed-scalable processors

$\mathcal{J} = \{1, \ldots, n\}$ is the set of jobs:
V_j is the processing volume (work) of job j
$W_j := \frac{V_j}{m_j}$ is the work on one processor

E is the energy budget

Parameters

Preemption and migration are characterized for the systems with single image of the memory.
Non-preemptive instances arise in systems with distributed memory.

Homogeneous Model in Speed-scaling

If a processor runs at speed s then the energy consumption is s^α units of energy per time unit, where $\alpha > 1$ is a constant (practical studies show that $\alpha \leq 3$).

It is supposed that a continuous spectrum of processor speeds is available.

E is the energy budget.

The aim is to find a feasible schedule with minimum makespan (total completion time) so that the energy consumption is not greater than a given energy budget.

NP-hardness

Even-Odd Partition Problem

$A = \{a_1, a_2, \ldots, a_{2n_0}\}$ is the ordered set such that
$\sum_{a_i \in A} a_i = 2C$, $a_i < a_{i+1}$, $i = 1, \ldots, 2n_0 - 1$
$a_{2i+1} > 3a_{2i}$ for $i = 1, \ldots, n_0 - 1$.
Question: whether A can be partitioned into two subsets A_1 and A_2

$$\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i = C, \ |A_1| = |A_2| = n_0,$$

A_1 contains only one element from each pair $a_{2i-1}, \ a_{2i}, \ i = 1, \ldots, n_0$.

Theorem

Problem $P2|size_j, energy|\sum C_j$ is NP-hard.
Proof is based on the reduction of the Even-Odd Partition Problem.

2-Approximation Algorithm

Scheme

Step 1: Given an instance $(P2)$, we generate the instance $(P1)$: reindex jobs in non-decreasing of volumes W'_j, and find optimal durations p'_j.

Step 2: Calculate processing times of jobs for $(P2)$: $p_j = 2p'_j$ for single-processor jobs and $p_j = p'_j$ for two-processor jobs. Assign job j to the first available processor if j requires one processor or to the two processors when both of them are available if j is a two-processor job while keeping the order of job starting times the same as those generated in Step 1.

Lemma

$\sum_{j \in \mathcal{J}} C_j(\mathcal{A}) \leq 2 \sum C^*_j(P2) \leq 2 \sum C^*_j(P1)$.

Theorem

A 2-approximate schedule can be found in $O(n \text{log} n)$ time for scheduling problem $P2|size_j, energy|\sum C_j$.

Lower Bounds

Two-processor Jobs

$$\frac{1}{2}\sum_{i=1}^{n}(n-i+1)p_{\pi_i} \to \min,$$

$$\sum_{i=1}^{n}(V_{\pi_i})^{\alpha}p_{\pi_i}^{1-\alpha} = E.$$

Agreeable Sizes and Volumes

$$LB(\pi) = \frac{1}{m}\sum_{j=1}^{n}\sum_{i=1}^{j} size_{\pi_i}p_{\pi_i} + \frac{1}{2}\sum_{j=1}^{n}p_{\pi_j} - \frac{1}{2m}\sum_{j=1}^{n} size_{\pi_j}p_{\pi_j} \to \min,$$

$$\sum_{j=1}^{n} W_j^{\alpha}p_j^{1-\alpha}size_j = E.$$

Test examples

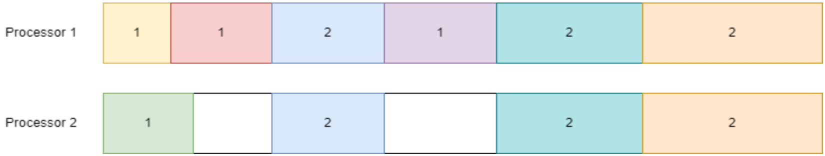
- alpha (1.5, 2.0, 2.5, 3.0)
- jobs count (50, 100)
- small jobs probability (0.0, 0.3, 0.5, 0.7, 1.0)
- single jobs probability (0.3, 0.5, 0.7) or blocks (2, 4, 6, 8, 10)
- series (11, 12, 21, 22)

Examples count $= 30$

Local improvements

1. Find blocks

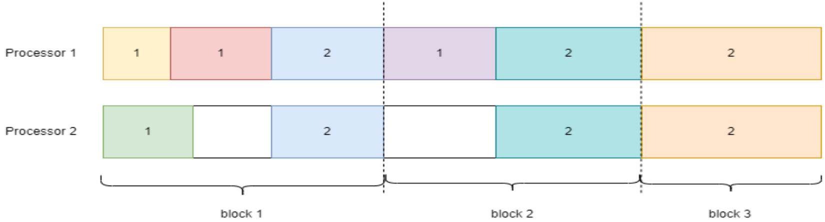
2. If a block consists of an odd number of single-processor jobs, move the last job to the next block if possible



Local improvements

1. Find blocks

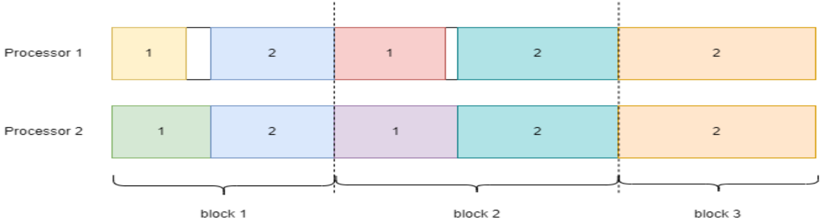
2. If a block consists of an odd number of single-processor jobs, move the last job to the next block if possible



Local improvements

1. Find blocks

2. If a block consists of an odd number of single-processor jobs, move the last job to the next block if possible



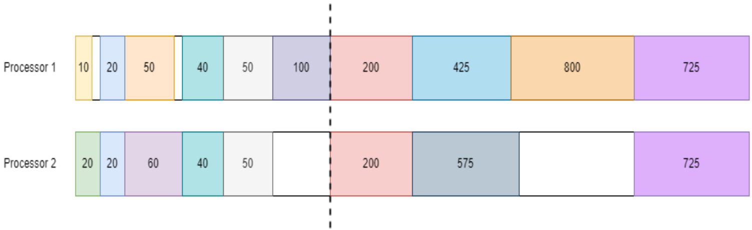
Series 11

$\mathbf{SMALL}_1 = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)$

$\mathrm{SMALL}_2 = (10, 11, 12, 13, 14, 15, 16, 17, 18, 19)$

$\mathbf{LARGE}_1 = (200, 275, 350, 425, 500, 575, 650, 725, 800, 875)$

$\mathrm{LARGE}_2 = (520, 540, 560, 580, 600, 620, 640, 660, 680, 700)$



Test results for series 11

alpha = 1.5, jobs count = 50

single	**small**	E_A	D_A	E_{lo}	D_{lo}
0.3	0.0	4.17	1.22	**2.55**	0.92
0.3	0.3	6.16	3.10	**4.15**	2.43
0.3	0.5	8.78	6.09	**4.99**	2.95
0.3	0.7	9.48	10.39	**6.18**	5.45
0.3	1.0	7.26	5.49	**2.65**	0.91
0.5	0.0	5.57	2.04	**3.01**	1.00
0.5	0.3	8.18	4.16	**4.77**	1.56
0.5	0.5	9.87	6.80	**5.32**	1.76
0.5	0.7	11.41	5.63	**7.07**	3.35
0.5	1.0	9.49	6.51	**4.04**	1.34
0.7	0.0	5.91	1.05	**4.11**	0.67
0.7	0.3	7.58	3.50	**5.36**	1.43
0.7	0.5	8.75	4.24	**6.74**	3.44
0.7	0.7	12.45	8.67	**8.87**	7.60
0.7	1.0	8.93	3.18	**4.55**	0.96

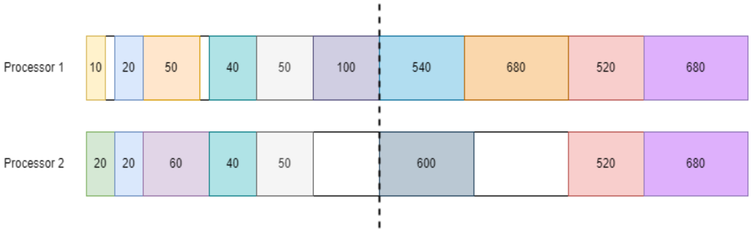
Series 12

$$\mathbf{SMALL}_1 = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)$$

$$\text{SMALL}_2 = (10, 11, 12, 13, 14, 15, 16, 17, 18, 19)$$

$$\text{LARGE}_1 = (200, 275, 350, 425, 500, 575, 650, 725, 800, 875)$$

$$\mathbf{LARGE}_2 = (520, 540, 560, 580, 600, 620, 640, 660, 680, 700)$$



Test results for series 12

alpha = 1.5, jobs count = 50

single	**small**	E_A	D_A	E_{lo}	D_{lo}
0.3	0.0	1.14	0.14	1.14	0.24
0.3	0.3	1.74	0.42	**1.56**	0.50
0.3	0.5	3.16	1.06	**2.53**	0.63
0.3	0.7	4.53	2.20	**3.38**	2.15
0.3	1.0	7.26	5.49	**2.65**	0.91
0.5	0.0	1.41	0.10	1.41	0.16
0.5	0.3	2.79	0.47	**2.40**	0.41
0.5	0.5	3.90	0.86	**3.07**	0.65
0.5	0.7	6.46	1.71	**4.62**	1.52
0.5	1.0	9.49	6.51	**4.04**	1.34
0.7	0.0	1.96	0.08	1.96	0.13
0.7	0.3	3.20	0.20	**2.94**	0.21
0.7	0.5	4.47	0.71	**3.90**	0.69
0.7	0.7	6.49	2.47	**5.39**	2.58
0.7	1.0	8.93	3.18	**4.55**	0.96

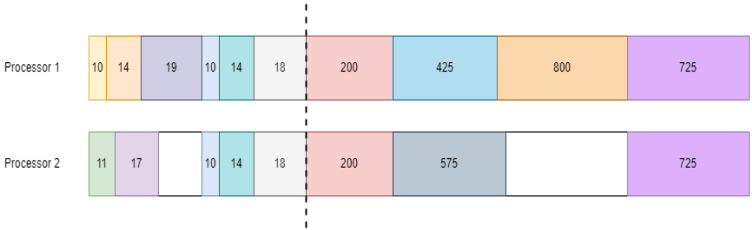
Series 21

$$\mathrm{SMALL}_1 = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)$$

$$\mathbf{SMALL}_2 = (10, 11, 12, 13, 14, 15, 16, 17, 18, 19)$$

$$\mathbf{LARGE}_1 = (200, 275, 350, 425, 500, 575, 650, 725, 800, 875)$$

$$\mathrm{LARGE}_2 = (520, 540, 560, 580, 600, 620, 640, 660, 680, 700)$$



Test results for series 21

alpha = 1.5, jobs count = 50

single	**small**	E_A	D_A	E_{lo}	D_{lo}
0.3	0.0	4.17	1.22	**2.55**	0.92
0.3	0.3	5.86	3.44	**4.33**	3.02
0.3	0.5	8.21	6.68	**5.43**	4.17
0.3	0.7	9.35	14.94	**7.26**	11.05
0.3	1.0	1.07	0.10	1.07	0.16
0.5	0.0	5.57	2.04	**3.01**	1.00
0.5	0.3	7.89	3.92	**4.96**	1.78
0.5	0.5	8.65	8.42	**5.60**	2.21
0.5	0.7	10.28	8.05	**8.32**	5.91
0.5	1.0	1.58	0.13	1.58	0.21
0.7	0.0	5.91	1.05	**4.11**	0.67
0.7	0.3	7.16	3.13	**5.52**	1.60
0.7	0.5	8.09	5.42	**7.20**	4.85
0.7	0.7	12.85	13.69	**10.81**	19.80
0.7	1.0	2.08	0.05	2.08	0.09

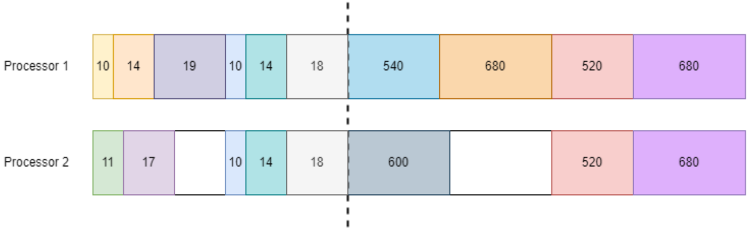
Series 22

$$\mathrm{SMALL}_1 = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)$$

$$\mathbf{SMALL}_2 = (10, 11, 12, 13, 14, 15, 16, 17, 18, 19)$$

$$\mathrm{LARGE}_1 = (200, 275, 350, 425, 500, 575, 650, 725, 800, 875)$$

$$\mathbf{LARGE}_2 = (520, 540, 560, 580, 600, 620, 640, 660, 680, 700)$$

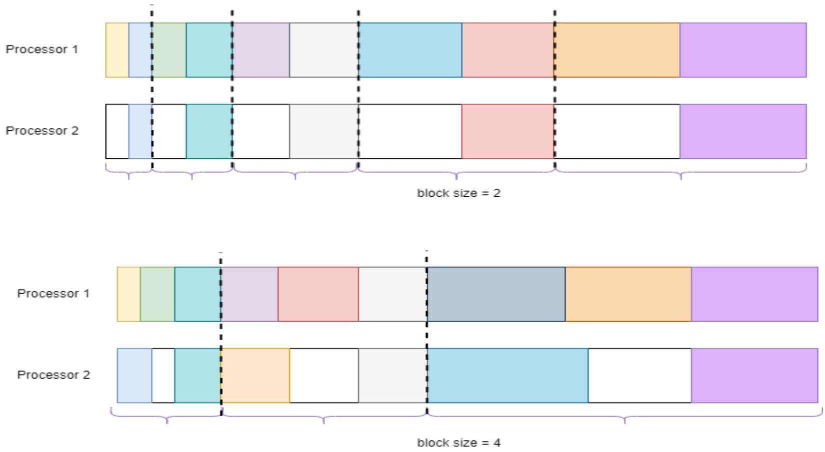


Test results for series 22

alpha = 1.5, jobs count = 50

single	**small**	E_A	D_A	E_{lo}	D_{lo}
0.3	0.0	1.14	0.14	1.14	0.24
0.3	0.3	1.48	0.36	**1.47**	0.58
0.3	0.5	2.46	0.64	2.46	1.01
0.3	0.7	3.37	2.51	3.42	4.08
0.3	1.0	1.07	0.10	1.07	0.16
0.5	0.0	1.41	0.10	1.41	0.16
0.5	0.3	2.35	0.32	**2.32**	0.52
0.5	0.5	3.01	0.55	**2.99**	0.88
0.5	0.7	5.02	1.72	5.03	2.90
0.5	1.0	1.58	0.13	1.58	0.21
0.7	0.0	1.96	0.08	1.96	0.13
0.7	0.3	2.90	0.16	**2.87**	0.25
0.7	0.5	3.90	0.54	**3.88**	0.87
0.7	0.7	5.75	2.61	5.77	4.48
0.7	1.0	2.08	0.05	**2.08**	0.09

Blocks



Processor 1
Processor 2
block size = 2
Processor 1
Processor 2
block size = 4

Test results for blocks

alpha = 1.5, jobs count = 50

block size	**small**	E_A	D_A	E_{lo}	D_{lo}
2	0.0	50.71	0.00	**6.40**	0.24
4	0.0	25.63	0.00	**6.33**	0.09
6	0.0	17.36	0.00	**6.18**	0.06
8	0.0	13.25	0.00	**5.77**	0.08
10	0.0	10.94	0.00	**6.08**	0.05
2	0.3	50.92	0.00	**9.13**	1.01
4	0.3	25.88	0.00	**9.08**	0.72
6	0.3	17.65	0.00	**8.66**	0.32
8	0.3	13.56	0.00	**7.82**	0.29
10	0.3	11.34	0.00	**8.13**	0.26
2	0.5	51.14	0.01	**11.53**	1.41
4	0.5	26.13	0.01	**11.11**	1.03
6	0.5	17.93	0.01	**10.13**	1.16
8	0.5	13.86	0.01	**9.19**	0.53
10	0.5	11.77	0.03	**9.92**	0.37
2	0.7	51.46	0.01	**15.14**	2.38
4	0.7	26.51	0.02	**13.80**	1.72
6	0.7	18.35	0.02	**12.29**	1.54
8	0.7	14.33	0.02	**11.26**	0.62
10	0.7	12.01	0.02	**9.69**	0.26
2	1.0	50.77	0.00	**7.38**	0.45
4	1.0	25.71	0.00	**7.30**	0.22
6	1.0	17.45	0.00	**7.03**	0.15
8	1.0	13.38	0.00	**6.72**	0.09
10	1.0	11.07	0.00	**7.03**	0.07

Comparison with single processor case

alpha = 1.5, jobs count = 50

single	**small**	E_A	E_1	MAX_w	MAX_b	AVG_w	AVG_b	$COUNT_b$
0.3	0.0	2.49	4.76	–	-3.81	0.00	-2.17	30
0.3	0.3	4.08	6.15	0.74	-4.15	0.74	-2.04	29
0.3	0.5	4.93	7.75	0.12	-5.91	0.07	-2.81	28
0.3	0.7	6.13	9.66	1.15	-7.39	0.74	-3.66	27
0.3	1.0	2.65	5.22	–	-3.45	0.00	-2.45	30
0.5	0.0	2.88	5.07	0.04	-3.35	0.04	-2.16	29
0.5	0.3	4.65	6.61	0.44	-3.86	0.42	-2.00	28
0.5	0.5	5.20	8.16	0.20	-4.81	0.20	-2.83	29
0.5	0.7	6.99	10.29	0.99	-6.54	0.99	-3.13	29
0.5	1.0	4.04	5.57	0.65	-3.64	0.41	-1.65	27
0.7	0.0	3.92	5.15	0.57	-3.15	0.33	-1.28	28
0.7	0.3	5.18	6.60	0.69	-3.32	0.39	-1.68	25
0.7	0.5	6.58	8.24	1.46	-4.58	0.83	-2.12	24
0.7	0.7	8.74	10.51	6.04	-7.03	2.90	-2.50	25
0.7	1.0	4.55	5.67	0.41	-3.44	0.22	-1.25	26

Test conclusions

parameter	best result
alpha	1.5
jobs count	100
small jobs probability	0.3
single jobs probability	0.3, 0.7
blocks	10
series	22

local improvements: blocks with size = 2

single processor case: single jobs probability = 0.3

Further Research

- Comparison with commercial solvers.
- Investigation of lower bounds.
- Statistical analyzes of the experimental results.
- More accurate selections in local improvements.

Thank you for your attention!