О ПОДХОДАХ К РЕШЕНИЮ ЗАДАЧ СОСТАВЛЕНИЯ РАСПИСАНИЙ В МНОГОПРОЦЕССОРНЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ С УЧЕТОМ РАСПАРАЛЛЕЛИВАНИЯ И РЕСУРСНЫХ ОГРАНИЧЕНИЙ

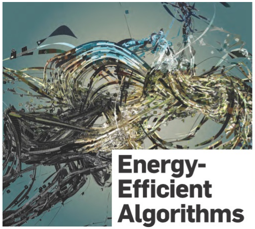
Ю.В. Захарова

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.



Multiprocessor and Parallel Jobs

Parallel Jobs

- **Rigid jobs**: the number of required processors is given and fixed ($size_j$).
- **Moldable jobs**: the number of required processors is chosen by the scheduler before starting a job, and is not changed until the job termination (δ_j).
- **Malleable jobs**: the number of required processors is chosen by the scheduler, and can be changed at runtime (δ_j).

Multiprocessor Jobs

- **Single mode jobs**: the set of required processors is given and fixed (fix_j).
- **Multimode jobs**: alternative sets of processors may be used (set_j).

Uniform and non-uniform partition of work between processors.

Resources

Energy $E = \int\limits_{t_0}^{t_1} s^{\alpha}(t)\mathrm{d}t$, where $s(t)$ is the speed of a processor at time t and $\alpha > 1$ is the constant.

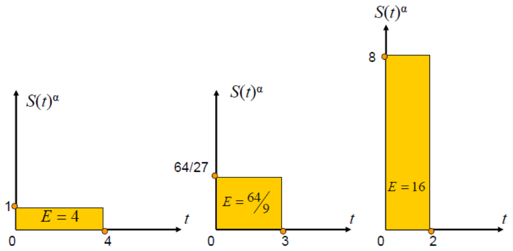
A data bus is a part of the system bus that is used to transfer data between computer components.

Cash memory influences on sequential execution of jobs.

Speed Scaling Scheduling

Modern microprocessors ($\alpha = 1.11$ for Intel PXA 270, $\alpha = 1.62$ for Pentium M770, $\alpha = 1.66$ for a TCP offload engine, $\alpha = 3$ for CMOS devices) can run at variable speed.
High speeds result in higher performance but also high energy consumption. Lower speeds save energy but performance degrades.

$$Energy = \int_{t_0}^{t_1} s^{\alpha}(t)\mathrm{d}t$$



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.

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

Parallelizable Jobs

Kononov, Zakharova (2015-2022): NP-hardness, approx
Li (2010, 2012): prec, heuristics
Kong, Guan, Deng, Yi (2011): approx

Report Structure

Single-Processor Jobs and Fully-Parallelizable Jobs

Single-Processor Jobs

$$\sum_{j \in \mathcal{J}} C_j = \sum_{j=1}^{\frac{n}{2}} (\frac{n}{2} - j + 1)(p_{2j-1} + p_{2j}).$$

Fully-Parallelizable Jobs

$$\sum_{j \in \mathcal{J}} C_j = \frac{1}{2} \sum_{j=1}^{n} (n - j + 1) p_j.$$

Rigid Jobs: MIP model

$$w_{jk} = \begin{cases} 1, & \text{if job } j \text{ is executed in event point } k, \\ 0 & \text{otherwise.} \end{cases}$$

T^{st}_{jk} and T^{f}_{jk} is the start and completion time of job j in event point k.

$$\sum_{j \in J} x_{jik} \leq 1, i \in I, k \in K,$$

$$\sum_{k \in K} w_{jk} = 1, j \in J,$$

$$\sum_{i \in I} x_{jik} = size_j w_{jk}, k \in K, j \in J,$$

$$T^{f}_{jk} \geq T^{st}_{jk}, k \in K, j \in J,$$

$$T^{f}_{jk} - T^{st}_{jk} = w_{jk} p_j, k \in K, j \in J,$$

$$\sum_{k \in K} (T^{f}_{jk} - T^{st}_{jk}) \geq p_j, j \in J,$$

$$T^{st}_{jk} \geq T^{f}_{j'k'} - T_{max}(2 - x_{jik} - x_{j'ik'}),\ j \neq j' \in J, i \in I, k' < k \in K, k \neq 1.$$

Rigid Jobs: Dynamic Programming

Preliminaries

$A_j = \sum_{i=1}^{j} p_j,\ j = 1, \dots, n$; n is the two-processor job.
Running time is $O(nA_n^{3n_2+1})$.

Recursive equation

$f(j, t, P_1, P_2, Q_1)$ is the minimum total completion time
(i) we have assigned $1, 2, \dots, j$ and n;
(ii) job n starts at time t;
(iii) the total processing times of single-processor jobs processed before t on M_1 and M_2 are P_1 and P_2, and the total processing time of single-processor jobs processed after t on M_1 is Q_1.

$$f(j, t, P_1, P_2, Q_1) = min \begin{cases} f(j-1, t, P_1 - p_j, P_2, Q_1) + P_1, \\ f(j-1, t, P_1, P_2 - p_j, Q_1) + P_2, \\ f(j-1, t, P_1, P_2, Q_1 - p_j) + (t + p_n + Q_1), \\ f(j-1, t, P_1, P_2, Q_1) + (t + p_n + A_j - P_1 - P_2 - Q_1), \end{cases}$$

$j = 1, \dots, n-1,\ t = 0, \dots, A_{n-1},\ P_1 = 0, \dots, \min\{A_j, t\},$
$P_2 = 0, \dots, \min\{A_j - P_1, t\},\ Q_1 = 0, \dots, A_j - P_1 - P_2.$

Rigid Jobs: $\frac{3}{2}$-Approximation Algorithm

Single-Processor Problem

Given an instance $P2$, we generate instance $P1$:
$p'_j = \frac{p_j}{2}$ for $size_j = 1$, $p'_j = p_j$ for $size_j = 2$.

Reindex jobs in non-decreasing of processing times W'_j:

Two-processor Problem

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.

C. Lee, X. Cai Scheduling one and two-processor tasks on two parallel processors // IIE Transactions (1999)

Rigid Jobs: $P2|size_j, energy|\sum C_j$

Single-Processor Problem

Given an instance $P2$, we generate instance $P1$:
$W'_j = \frac{W_j}{2}$ for $size_j = 1$, $W'_j = W_j$ for $size_j = 2$ and $E' = \frac{E}{2}$.

Reindex jobs in non-decreasing of volumes W'_j, and find optimal durations p'_j:

$$\sum_{j=1}^{n}(n-j+1)p'_j \to \min,$$

$$\sum_{j\in\mathcal{J}}(p'_j)^{\alpha-1}(W'_j)^{\alpha} = E'.$$

Two-processor Problem

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.

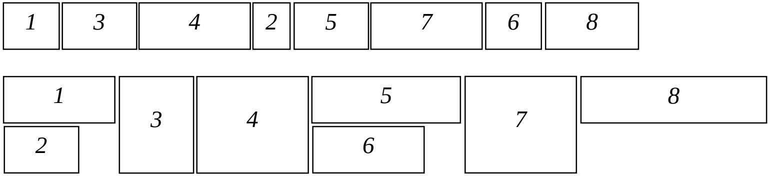
Rigid Jobs: Approximation Algorithm

Lower Bound

$\sum C_j^*(P1) \leq \sum C_j^*(P2)$

Theorem

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



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

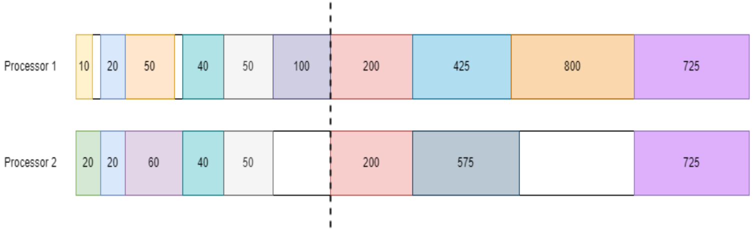
Series 11

SMALL_1 = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)

SMALL_2 = (10, 11, 12, 13, 14, 15, 16, 17, 18, 19)

LARGE_1 = (200, 275, 350, 425, 500, 575, 650, 725, 800, 875)

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	**2.69**	0.90
0.3	0.3	6.16	3.10	**4.06**	2.76
0.3	0.5	8.78	6.09	**5.25**	4.00
0.3	0.7	9.48	10.39	**6.10**	8.33
0.3	1.0	7.26	5.49	**2.88**	1.00
0.5	0.0	5.57	2.04	**3.22**	1.29
0.5	0.3	8.18	4.16	**5.20**	2.23
0.5	0.5	9.87	6.80	**5.69**	2.33
0.5	0.7	11.41	5.63	**7.67**	5.39
0.5	1.0	9.49	6.51	**3.14**	30.49
0.7	0.0	5.91	1.05	**4.41**	0.81
0.7	0.3	7.58	3.50	**5.91**	2.03
0.7	0.5	8.75	4.24	**7.15**	3.77
0.7	0.7	12.45	8.67	**9.14**	13.70
0.7	1.0	8.93	3.18	**5.42**	1.80

General Convex Model

$$\sum_{j \in J} x_{jik} \leq 1, i \in I, k \in K,$$

$$\sum_{k \in K} w_{jk} = 1, j \in J,$$

$$\sum_{i \in I} x_{jik} = size_j w_{jk}, k \in K, j \in J,$$

$$T^f_{jk} \geq T^{st}_{jk}, k \in K, j \in J,$$

$$T^f_{jk} - T^{st}_{jk} = w_{jk} p_j, k \in K, j \in J,$$

$$\sum_{k \in K} (T^f_{jk} - T^{st}_{jk}) \geq p_j, j \in J,$$

$$T^{st}_{jk} \geq T^f_{j'k'} - T_{max}(2 - x_{jik} - x_{j'ik'}),$$

$$j \neq j' \in J, i \in I, k' < k \in K, k \neq 1.$$

$$s_j p_j \geq W_j, j \in J.$$

$$\sum_{j \in J} size_j W_j (s_j)^{\alpha - 1} \leq E.$$

Comparison with BARON solver (8 h)

Instance	Algorithm			BARON Presolve		BARON Record			%
	LB	Obj	%	Obj	Time	LB	Obj	Time	
11	1173.5	1421.3	21.1	1480.5	33.47	394.6	1333.1	2850.3	6.6
12	694.1	786.1	13.2	763.6	28.92	285.2	763.6	28.9	2.9
21	812.4	1011.6	24.5	1082.9	32.23	539.7	981.6	261.9	3.0
22	1984.5	2330.1	17.4	2454.1	28.25	735.9	2231.2	462.5	4.4
block	1682.3	2470.9	46.8	2859.3	41.2	752.8	1970.9	4379.6	25.3

Obtained Results

Total Completion Time

Problem	Complexity	Approx.
$P2\|size_j, energy\|\sum C_j$	NP-hard	2
$P2\|fix_j, energy\|\sum C_j$	?	$2^{\frac{2\alpha-1}{\alpha-1}}$
$P2\|fix_j, pmtn, energy\|\sum C_j$	?	$2^{\frac{\alpha}{\alpha-1}}$

Makespan

Problem	Complexity	Approx.
$P2\|size_j, energy\|C_{\max}$	NP-hard	3/2
$P2\|size_j, prec, energy\|C_{\max}$	NP-hard	2
$P2\|r_j, size_j, pmtn, energy\|C_{\max}$	$Poly$	–
$P2\|r_j, fix_j, pmtn, energy\|C_{\max}$	$Poly$	–
$P2\|fix_j, energy\|C_{\max}$	$Poly$	–

Scheduling Rigid Jobs with Untrusted Predictions

Problem formulation

processing times p_j are not known in advance
x_j is predicted value of the processing time of job j
$\eta_j = |p_j - x_j|$ is the error of the prediction for job j

Monotonic Algorithms

The Shortest Predicted Processing Time First algorithm has competitive ratio at most $\left(\frac{3}{2} + \frac{3\eta}{n}\right)$, where $\eta = \sum_j \eta_j$.
The Round Robin Algorithm is 2-competitive.

Preferential Algorithm

The Preferential Round Robin algorithm with parameter $\lambda \in (0, 1)$ has competitive ratio at most $\min\{\frac{1}{\lambda}\left(\frac{3}{2} + \frac{3\eta}{n}\right); \frac{2}{1-\lambda}\}$. In particular, it is $\frac{2}{1-\lambda}$-robust and $\frac{3}{2\lambda}$-consistent.

Scheduling Fully-Parallelizable Jobs with Untrusted Predictions

Problem formulation

processing times p_j are not known in advance
x_j is predicted value of the processing time of job j
$\eta_j = |p_j - x_j|$ is the error of the prediction for job j

Monotonic Algorithms

The Shortest Predicted Processing Time First algorithm has competitive ratio at most $\left(1 + \frac{2\eta}{n}\right)$, where $\eta = \sum_j \eta_j$.
The Round Robin Algorithm is 2-competitive.

Preferential Algorithm

The Preferential Round Robin algorithm with parameter $\lambda \in (0, 1)$ has competitive ratio at most $\min\{\frac{1}{\lambda}\left(1 + \frac{2\eta}{n}\right); \frac{2}{1-\lambda}\}$. In particular, it is $\frac{2}{1-\lambda}$-robust and $\frac{1}{\lambda}$-consistent.

Conclusion and Further Research

Conclusion

- NP-hardness and polynomial solvability of parallel and dedicated versions with total completion time criterion and the makespan criterion.
- We propose approaches to construct approximation and exact algorithms for various particular cases.
- Algorithm demonstrates competitive results on various series of instances.
- Scheduling with untrusted predictions is investigated.

Further Research

- The problems with more complex structure, where processors are heterogeneous and jobs have alternative execution modes with various characteristics.
- Open question is the complexity status of the problem with single mode jobs on two processors.

Thank you for your attention!