Scheduling Parallel Machines With a Single Server

Frank Werner[1]

Keramat Hasani[2] **Svetlana A. Kravchenko**[2]

[1] Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg

[2] United Institute of Informatics Problems, Minsk, Belarus

Outline of the Talk

- Introduction
- Makespan Problem
  - MILP models
  - Metaheuristics and heuristics
- Mean Flow Time Problem
  - MILP models
  - Metaheuristics

Problem Definition

The problem can be formulated in the following way:

- n jobs $J_1, J_2, \ldots, J_n$ have to be scheduled on two identical parallel machines without preemptions.
- Each machine can process at most one job at a time.
- For each job j, there are given:

  p_j - processing time , s_j - setup time
- Before processing any job, it has to be loaded on a machine and it takes s_j time unit.
- All setups have to be done by a single server which can handle at most one job at a time.

Maximum Completion Time Problem (cont'd)

Authors	Year	Problem	Approaches
Hall et al.	1996	P2,S1||Cmax	• Unary NP-hardness proof
Kravchenko&Werner	1997	P2,S1|S_i = 1|Cmax	• A pseudo-polynomial algorithm
Abdekhodaee&Wirth	2002	P2,S1||Cmax	• A MILP model • Two backward/forward heuristics
Abdekhodaee et al.	2004	P2,S1|s_j = s|Cmax P2,S1|p_j = p|Cmax	• Complexity analysis • Some heuristics • Some lower bounds
Abdekhodaee et al.	2006	P2,S1||Cmax	• Two greedy heuristics • A genetic algorithm • A Gilmory-Gomory algorithm

Maximum Completion Time Problem (cont'd)

Authors	Year	Problem	approaches
Zhang & Wirth	2009	P2,S1||Cmax	• Consideration of the online version
Kim&Lee	2012	P,S1||Cmax	• Some heuristics for small-sized instances
Gan et al.	2012	P2,S1||Cmax	• Two MILP models • Two variants of a Branch and Price algorithm

Maximum Completion Time Problem (cont'd)

- Our approaches:
  - Mixed integer programming models
    - Setup sequence model
      - In this model, the loading order of the jobs is used.
    - Block models
      - In this model we consider the jobs as a set of blocks.
  - Metaheuristics
    - Simulated annealing algorithm (SA)
      - A composite neighborhood is used.
    - Genetic algorithm (GA)
  - Heuristics
    - Algorithm Min-idle
    - Algorithm Min-loadgap

Setup Sequence Model (M0)

- In the following model, the loading order of the jobs is used as in Gan et al. (2012).

$$x_{ij} = \begin{cases} 1, & if\ \ J_j\ is\ the\ i-th\ \ job\ to\ be\ setup, \\ 0, & otherwise. \end{cases}$$

$$\sum_{j=1}^{n} x_{i,j} = 1 \qquad\qquad \sum_{i=1}^{n} x_{i,j} = 1$$

Let ss_i be the loading time of the ith loading job and pp_i be the processing time of the ith loading job, i.e., we have

$$ss_i = \sum_{j=1}^{n} s_j x_{i,j} \qquad pp_i = \sum_{j=1}^{n} p_j x_{i,j}.$$

Setup Sequence Model (M0) (cont'd)

Now for the first and the second loading jobs, we can introduce the inequality

$$F_{1,2} \geq ss_1 + ss_2$$

If the processing part of the first loading job is large enough, then one can introduce the inequality

$$L_{1,2} \geq pp_1 - ss_2$$

and to denote the time interval when only one machine is busy, one can introduce L_2 with the inequalities

$$L_2 \geq L_{1,2} - pp_2,\ \ L_2 \geq pp_2 - L_{1,2}.$$

Setup Sequence Model (M0) (cont'd)

let $$x_j = \begin{cases} 1, & \text{if } J_j \text{ is finished last among the jobs } J_1, J_2, \ldots, J_j, \\ 0, & \text{otherwise}. \end{cases}$$

To estimate the overlapping part for the first two jobs, we introduce the inequalities:

$$OF_2 \geq L_{1,2} - M(1 - x_2), \ OF_2 \geq pp_2 - Mx_2$$

where $M = \max_j \{p_j\}$

To know the earliest time when one of the machines is available, we introduce the inequality

$$F_2 \geq F_{1,2} + OF_2.$$

Setup sequence model (M0) (cont'd)

For $j = 2,\ldots,n-1$ we have the following inequalities:

$$F_{j,j+1} \geq F_j + ss_{j+1},$$

$$L_{j,j+1} \geq L_j - ss_{j+1},$$

$$L_{j+1} \geq L_{j,j+1} - pp_{j+1},$$

$$C_{max}(s) = F_n + L_n$$

$$L_{j+1} \geq pp_{j+1} - L_{j,j+1},$$

$$OF_{j+1} \geq L_{j,j+1} - M(1 - x_{j+1}),$$

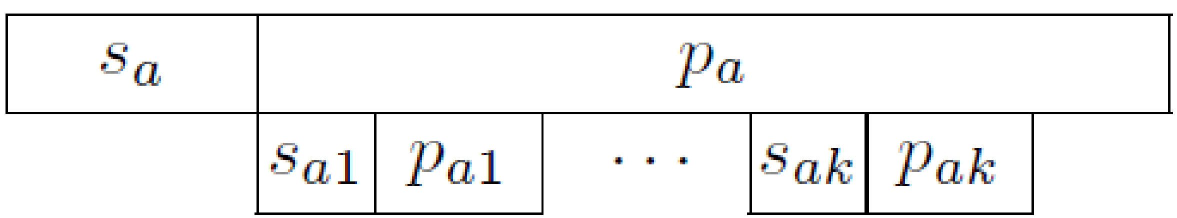
$$OF_{j+1} \geq pp_{j+1} - Mx_{j+1},$$

$$F_{j+1} \geq F_{j,j+1} + OF_{j+1}.$$

Block Models (M1,M2)

- The problem $P2, S1 \| C_{max}$ can be considered as a unit of blocks $B_1, \ldots, B_z$, where $z \leq n$.

- Each block B_k can be completely defined by the first level job J_a and a set of second level jobs $\{J_{a1}, \ldots, J_{ak}\}$, where inequality $p_a \geq s_{a1} + \ldots + s_{ak} + p_{a1} + \ldots + p_{ak}$ holds.



Block Models (cont'd)

The variable $B_{k,f,j}$ is used for a block, $k = 1, \ldots, n$, $j = 1, \ldots, n$:

$$B_{k,f,j} = \begin{cases} 1, & \text{if job } J_j \text{ is scheduled in level } f \text{ in the } k\text{-th block,} \\ 0, & \text{otherwise.} \end{cases}$$

$$f = \begin{cases} 1, & \text{if the level is the first one,} \\ 2, & \text{if the level is the second one.} \end{cases}$$

Block Models (cont'd)

Each job belongs to some block, i.e., for $j=1,\ldots,n$, we have

$$\sum_{k=1}^{n}\sum_{y=1}^{2} B_{k,y,j} = 1$$

There is only one job of the first level for each block:

$$\sum_{j=1}^{n} B_{k,1,j} \le 1$$

The loading part of the block Bk has the length $ST_k \ge 0$,

$$ST_k \ge \sum_{j=1}^{n} s_j B_{k,1,j}$$

The objective part of the block Bk has the length

$$\sum_{j=1}^{n} (s_j + p_j) B_{k,2,j}.$$

Block Models (cont'd)

The processing part of the block has the length $PT_k \geq 0$,

$$PT_k \geq \sum_{j=1}^{n} p_j B_{k,1,j} - \sum_{j=1}^{n} (s_j + p_j) B_{k,2,j}$$

$$st_j + ST_j \leq st_{j+1}$$
$$st_j + ST_j + PT_j \leq st_{j+2}$$

We denote by F the total length of the modified schedule:

$$F \geq st_n + ST_n + PT_n$$
$$F \geq st_{n-1} + ST_{n-1} + PT_{n-1}$$

$ch[j]$ denotes the maximal number of second level jobs for the same block **(only in model M1):**

$$(a) B_{x,2,1} + \ldots + B_{x,2,n} \leq ch[1] B_{x,1,1} + \ldots + ch[n] B_{x,1,n}$$

Block Models (cont'd)

- M1 contains all constraints.
- M2 contains all except (a).
- The objective function is :

$$C_{max}(s) = F + \sum_{x=1}^{n}\sum_{j=1}^{n}(s_j + p_j)B_{x,2,j}$$

Metaheuristics (SA)

- A composite neighborhood has been selected which includes three operators: Swap, Insert, Block.
- Several neighbors are randomly generated. Then the neighbor with the best makespan value among them is taken as generated neighbor.
- Number of neighbors =

  1 Swap + 1 Insert + $\left\lfloor \frac{n}{2} \right\rfloor - 1$ Blocks.

Metaheuristics (GA)

Parameters:

- **Initialization of the population**
  - Population size (PS) = 15.
  - Initial population contains only randomly generated job sequences (to have a sufficient large diversity)
- **Evaluation of the population**
  - An individual with smaller objective function value has a higher fitness.
- **Selection of individuals**
  - Tournament selection
  - This process is done PS – 1 times.
- **Crossover**
  - One-point crossover
- **Mutation**
  - 1 Swap + 1 Insert + 1 Block.
- **Formation of new population**
  - The elitist strategy was considered.

Lower Bound

To evaluate the results obtained, we use the known lower bound:

$$LB = \max\{LB_1, LB_2\},$$

$$LB_1 = \frac{1}{2}\left(\sum_{i \in J}(s_i + p_i) + \min_{i \in J}\{s_i\}\right),$$

$$LB_2 = \sum_{i \in J} s_i + \min_{i \in J}\{p_i\},$$

Two Heuristics for Very Large Instances

- **Algorithm Min-idle**
  - tries to minimize the gap between completing a job and starting the next job (corresponds to LB_1), for instances with L = {0.1, 0.5, 0.8, 1}. The complexity is $O(n^2)$.
- **Algorithm Min-loadgap**
  - tries to minimize the gap between the completion time of the loading of a job and the start of the loading of the next job (corresponds to LB_2), for instances with L = {1, 1.5, 1.8, 2}. The complexity is $O(n^2)$.

Computational Results

Generation of instances

- The performance of the models M0, M1 and MP was tested on the data generated in the same way as it is described in Abdekhodaee and Wirth (2002) and Gan et al. (2012).
- For $n \in \{8,10,14,16,18,20\}$, 10 instances were generated

  for $L \in \{0.1, 0.5, 0.8, 1, 1.5, 1.8, 2.0\}$,

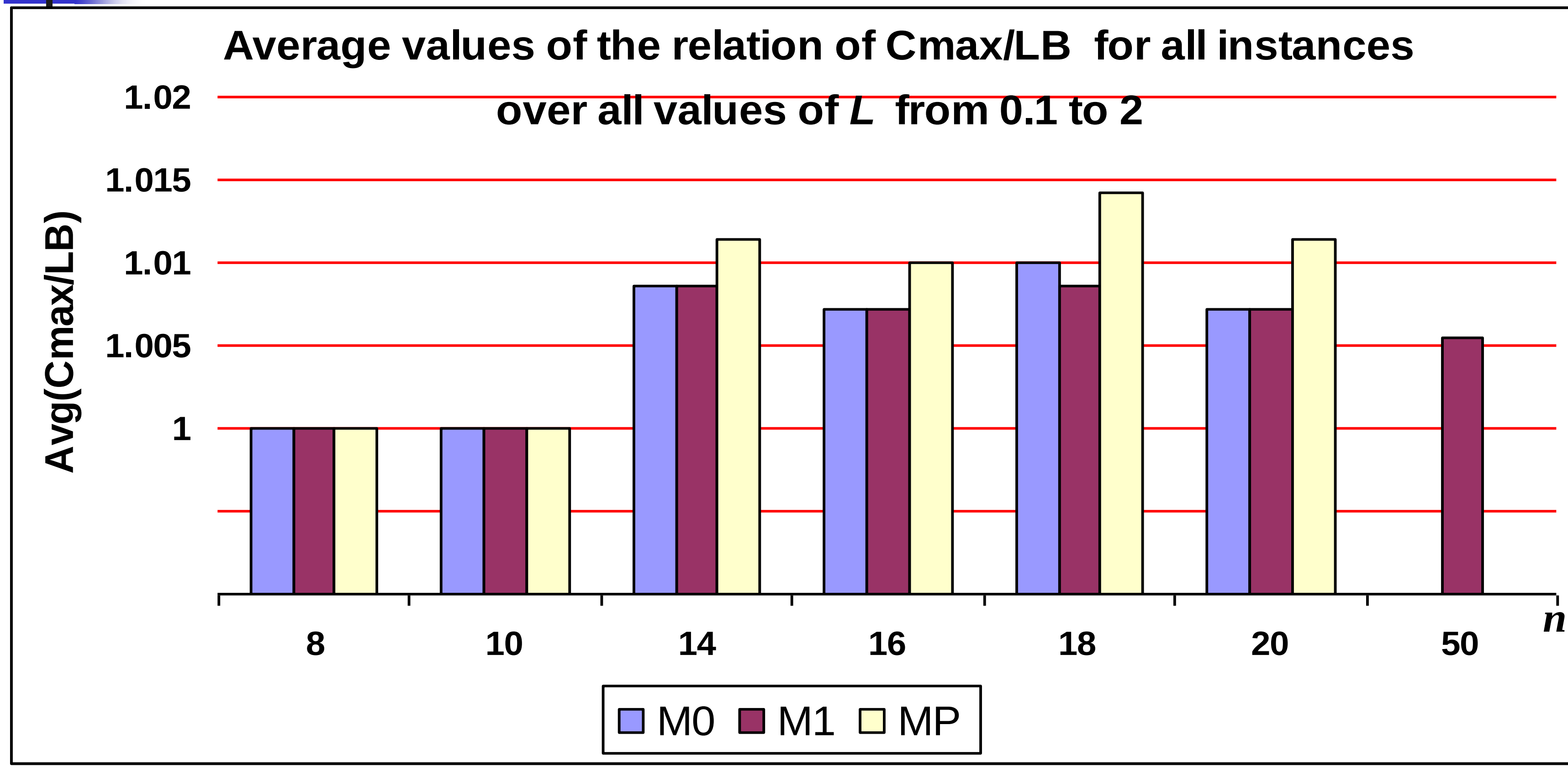
  and 5 instances for larger problems were generated for

  $L \in \{0.1, 0.5, 0.8, 1, 1.5, 1.8, 2.0\}$.

$$p_j \overset{d}{=} U(0,100)$$

$$s_j \overset{d}{=} U(0,100L)$$

Computational Results (cont'd)



M0 (Setup sequence model),
M1 (Block model),
MP (MILP model from Gan et al. 2012)

Computational Results (cont'd)

For the Model M2 we obtained the following results:

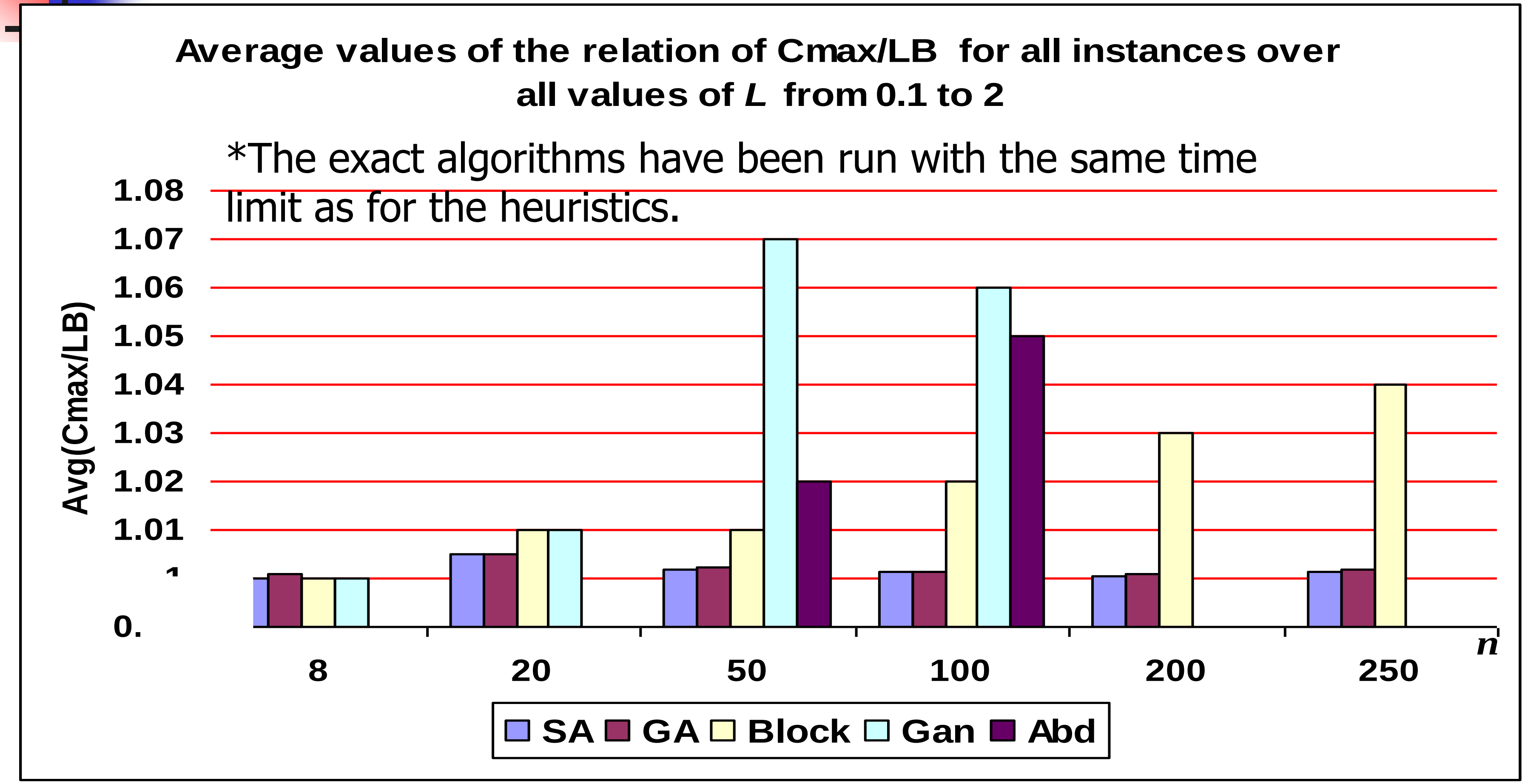
The average and the maximal gaps for $n = 200$

L	**0.1**	**0.5**	**0.8**	**1.0**	**1.5**	**1.8**	**2.0**
Avg(Cmax/LB)	1.01	1.04	1.07	1.09	1.01	1.00	1.00
Max(Cmax/LB)	1.01	1.08	1.10	1.12	1.02	1.01	1.01

The average and the maximal gaps for $n = 250$

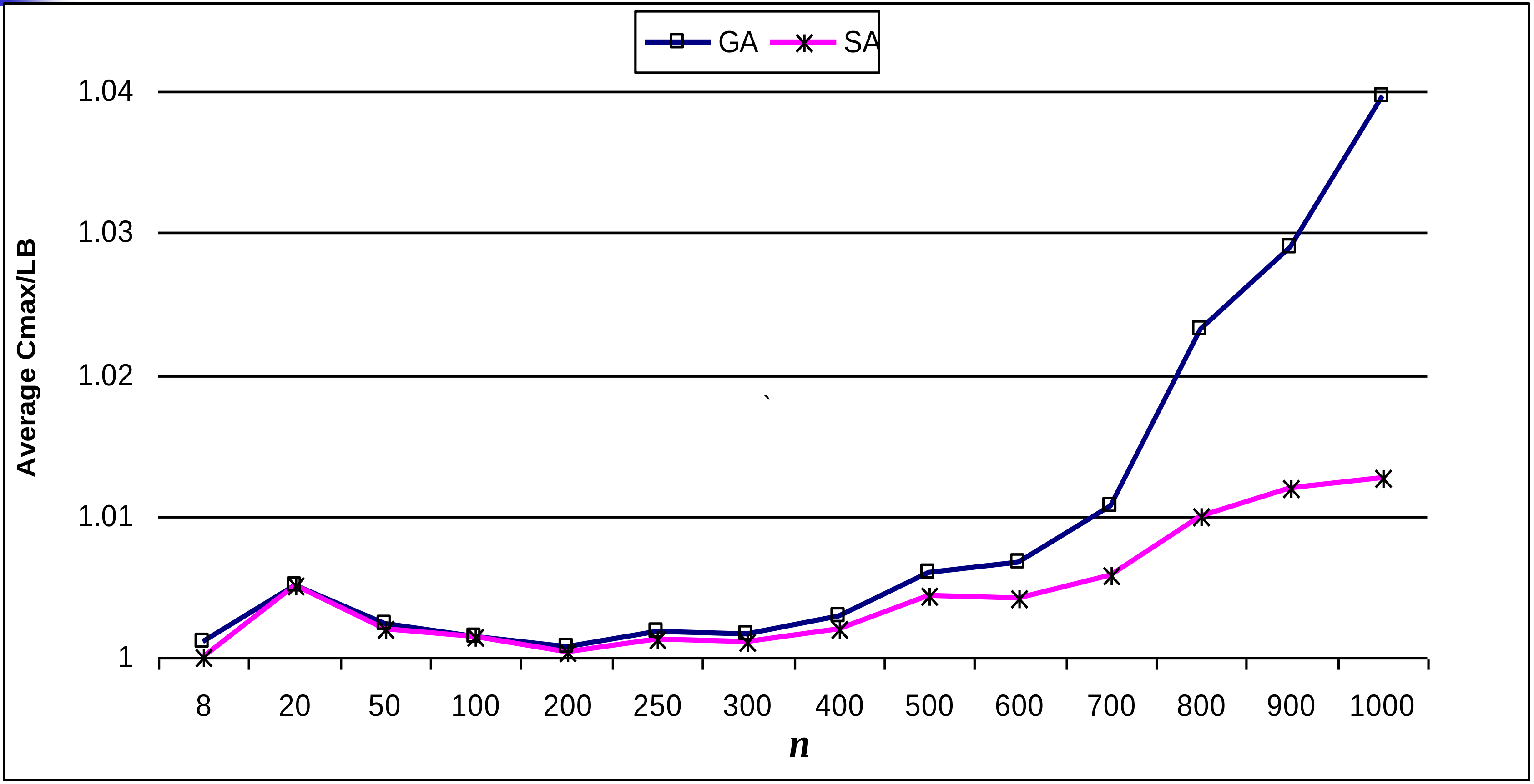
L	**0.1**	**0.5**	**0.8**	**1.0**	**1.5**	**1.8**	**2.0**
Avg(Cmax/LB)	1.02	1.07	1.10	1.10	1.02	1.00	1.00
Max(Cmax/LB)	1.03	1.09	1.11	1.12	1.04	1.00	1.01

Computational Results (cont'd)



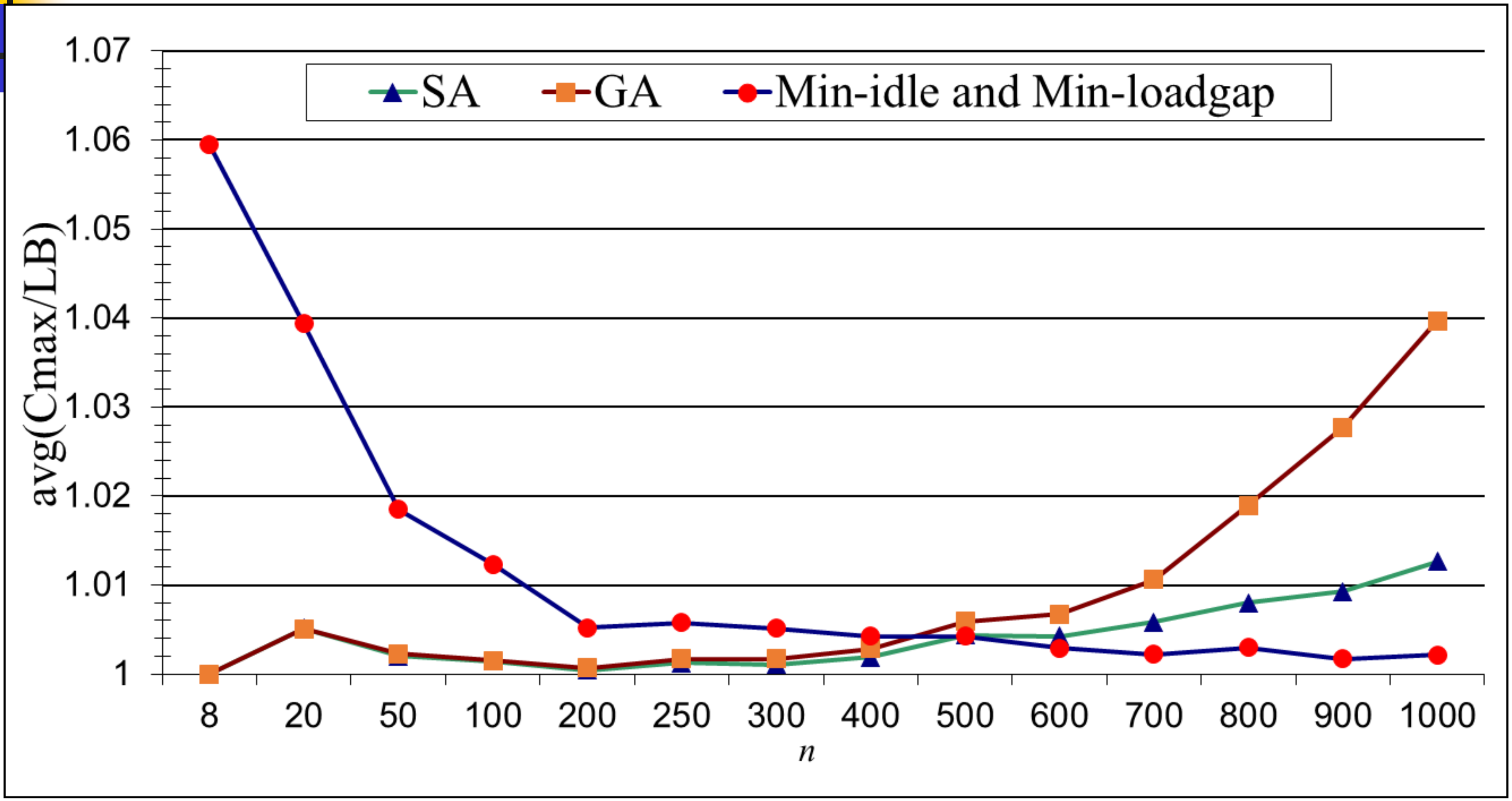
- **Abd** : Genetic algorithm from **Abdekhodaee et al. (2006)**
- **Gan**: The best obtained results from two MILP models and two Branch and price algorithms from **Gan et al. (2012)**

Computational Results (cont'd)



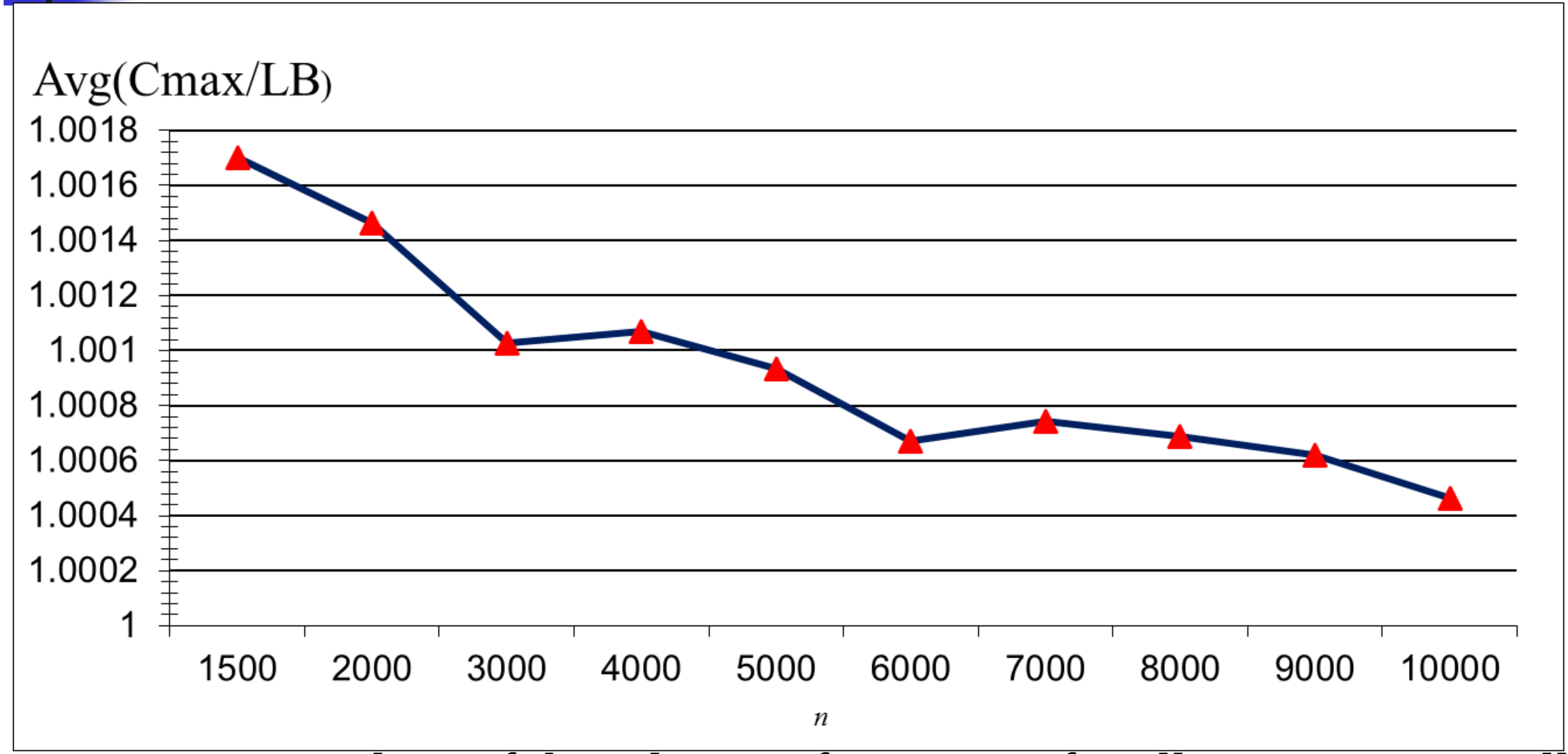
Average values of the relation of *Cmax/LB* for all instances over all values of *L* from 0.1 to 2

Computational Results (cont'd)



Comparison of the average values of the relation Cmax/LB of Algorithms *Min-idle* and *Min-loadgap* with Algorithms SA and GA.

Computational Results (cont'd)



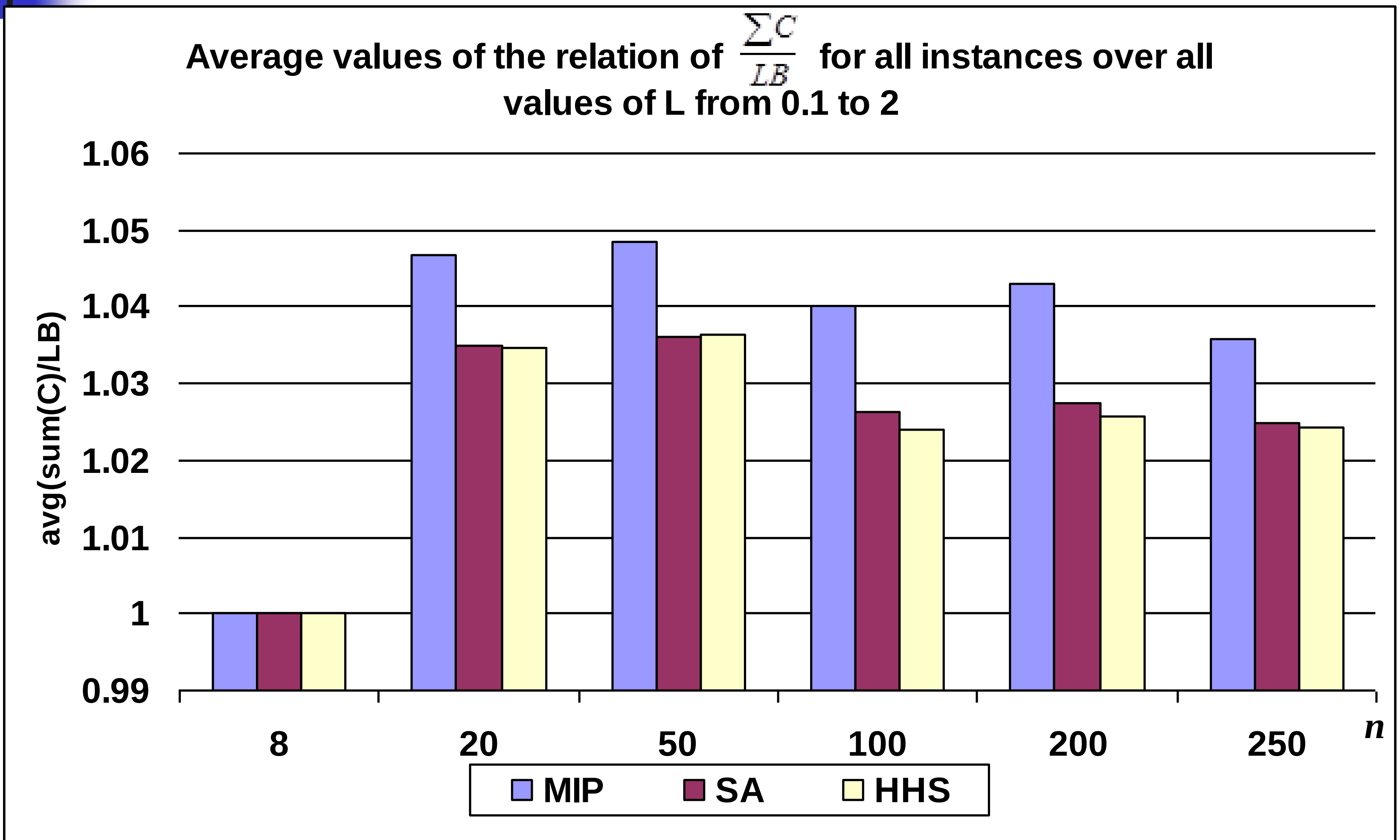
Average values of the relation of *Cmax/LB* for all instances over all values of *L* from 0.1 to 2.

Mean Flow Time Problem

Approaches:

- MILP model:
  - The model is constructed to find an optimal schedule, where all jobs from the list have to be scheduled in a staggered order
- Simulated annealing (SA)
  - requires a substantially different calibration.
- Hybridization of Harmony Search and Simulated Annealing(HHS).
  - SA is used to generate new solutions out of the harmony memory.

Computational Results



Thank you for your attention

QUESTIONS AND SUGGESTIONS