A New Line Balancing Method Considering Robot Count and Operational Costs in Electronics Assembly

Ryo Murakami, Sachio Kobayashi, Hiroki Kobayashi and Junji Tomita
Fujitsu Laboratories
Nakahara-ku, Kawasaki-shi, Kanagawa, Japan

Abstract

Automating electronics assembly is complex because many devices are not manufactured on a scale that justifies the cost of setting up robotic systems, which need frequent readjustments as models change. Moreover, robots are only appropriate for a limited part of assembly because small, intricate devices are particularly difficult for them to assemble. Therefore, assembly line designers must minimize operational and readjustment costs by determining the optimal assignment of tasks and resources for workstations. Several research studies address task assignment issues, most of them dealing with robot costs as fixed amount, ignoring operational costs. In real factories, the cost of human resources is constant, whereas robot costs increase with uptime. Thus, human workload must be as large and robot workload as small as possible for the given number of humans and robots. We propose a new task assignment method that establishes a workload balancing that meet precedence and further constraints. The following must be determined before using our method: which tasks robots can perform, and which workstations robots are assigned to. We assume that humans can perform every task and consider the constraints that restrict the tasks robots can perform. By applying our method to several case studies, problems involving 20 humans were solved within 1 minute and 1% dispersion. These results indicate that our method can be used in actual factories where a short-term planning period corresponding to frequent production fluctuations is required. We also applied our method to real assembly data for laptops manufactured by our company and obtained task assignment that reduces the operational costs by 30%. This suggests that our method can contribute to promoting the automation of electronics assembly by demonstrating its cost reduction potential.

Introduction

In product development, the current global market continuously pressures manufacturers to compete with competitors from around the world. Manufacturers must speed up the time to market while minimizing manufacturing costs to ensure that their products remain competitive [1]. Automating assembly processes and optimizing assembly lines are essential to survival in the electronics market.

Skilled robotic systems are key components in fully automated assembly processes and necessary for highly efficient production. In electronics assembly, however, robots are available to a limited part of assembly because small, intricate devices are particularly difficult for robots to assemble. Moreover, many devices are not produced on the scale necessary to justify the cost of setting up robotic systems that need frequent readjustments as models change. Therefore, which tasks robots handle must be decided very carefully.

Assembly lines consist of sequences of workstations performing repetitive sets of tasks typically for the industrial production of high-quantity commodities; they are even gaining importance in low-volume production of customized products. Because of the high investment and operational costs involved, the design of assembly lines is of considerable practical importance [2]–[5]. A number of crucial decisions must be made in assembly line design, including product design, process selection, line layout configuration, line balancing and resource planning [6]–[8]. The first two decisions, product design and process selection, provide information about the work that must be performed on the assembly line, that is, a set of indivisible tasks related by constraints. The main sources of these constraints are technological, economic and environmental in nature. Which tasks robots can handle is a particularly important constraint in electronics assembly. The next decision deals with choosing the line layout (straight, U-shaped, with circular transfer, asymmetric, etc.). This defines how workstations will be situated on the line as well as what flow directions and rules are used. Finally, the final two crucial decisions determine the optimal assignment of tasks and resources (human, machine, robot, etc.) to workstations. This is a complex combinatorial problem whose solution determines, for the most part, the efficiency of the line. In addition to designing a new line, operational lines must be redesigned periodically or after changes to the production process or production program.

There are relatively few studies that address task assignment problems with variations of resources [9]–[15], although many recent publications looking at the simple assembly line balancing problem exist [16]–[27]. Ref. [15] is one of the few studies that deal with an assignment problem for manual and robotic assembly mixing lines. Although the authors minimize total line costs under some constraints, including robot constraints, they deal with robot costs as fixed amount, ignoring the operational costs. The longer a robot operates, the higher its costs because of power consumption. By contrast, the costs of human resources are constant, except overtime pay. Thus, the human workload should be as large and the robot workload as small as possible for the given number of humans and robots. We propose a new task assignment method that establishes the workload balancing, addressing precedence and further constraints. This represents a major improvement in real factories.

Methodology: Definition of Problem

In [15], the authors assign tasks as well as resources to workstations for hybrid manual and robotic assembly mixing lines. Here, we assume that the resource assignment, i.e. which workstations the robots are assigned to (humans assigned to the remaining workstation), is established. This is because a desirable assignment of robots to workstations is decided depending on the given environment and conditions. For example, robot positions are rarely changed once robots are set up because readjustment of robots is time-consuming and costly. Our method can compute an optimal assignment more quickly by focusing on task assignment.

In our task assignment problem, we consider two kinds of constraints: precedence and robot. There are order relationships among the tasks, which are illustrated in a precedence graph such as **Figure 1**, where an arc (i, j) exists if task j cannot be started before the end of task i. Precedence constraints are the restrictions on task j when assigned to the workstation allocated to task i or subsequent tasks. Robot constraints represent the tasks that the robot can perform. Note that we do not impose an upper limit on each workstation processing time T_n, while most line balancing studies use the assembly time for a product, called the cycle time C, as the upper limit. This reflects the fact that overtime work recovers some operational delays in real factories.

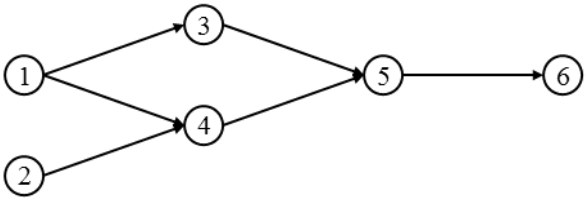


Figure 1 – Example of Precedence Graph

Here, we introduce the objective of our problem. We propose two objectives corresponding to the number of resources in the line.

The first reflects our idea that humans should be assigned to as many tasks as possible. We use the objective

$$E_1 \equiv \sum_{n \in H} (T_n - C)^2, \tag{1}$$

when there are enough resources to satisfy $T_n < C$ for any workstation. Here, H is the subset of the workstations to which humans are assigned. The n-th workstation processing time T_n is the sum of the duration t_i of the tasks assigned to the workstation:

$$T_n \equiv \sum_{i \in S_n} t_i, \tag{2}$$

where S_n is the subset of the tasks assigned to the n-th workstation. Each workstation time approximates the cycle time C by minimizing this objective. Since wages are independent of their workstation times, unless they exceed the cycle time C, assigning as many tasks as possible to humans maximizes line efficiency. Such assignment minimizes robot operating time. We assume that the total operational cost C_{R} are proportional to the robot operating time T_{R}:

$$C_{\mathrm{R}} \propto T_{\mathrm{R}}. \tag{3}$$

Thus, we minimize T_{R} instead of C_{R}.

The second objective is to perform conventional assignments. When the given resources are so few or the given cycle time is so tight that the inequality $T_n < C$ for all the workstations cannot be satisfied, the workstation loads should be equalized to finish work as soon as possible. Thus, we use the objective

$$E_2 \equiv \max_n T_n, \tag{4}$$

which maximum is operated for all workstations, including the robot workstations. Note that a variance such as E_1

$$E_2' \equiv \sum_{n=1}^{N} T_n^2, \tag{5}$$

(N: the number of workstations) is inapplicable here. Generally, it takes robots more time to complete a task than humans. Therefore, E_2' induces unbalanced workstation times between humans and robots, in which humans are assigned to more tasks than robots.

Our task assignment problem is defined as follows:

Given a set of tasks, a set of the duration and robot availability for each task, and resource assignment to each workstation, and given a cycle time and possible precedence constraints, we try to find assignment of tasks to workstations on the line so that:

- No precedence constraints are violated;
- No robot constraints are violated; and
- The following objectives are met:
  - If there are enough resources to satisfy $T_n < C$ for any workstation, each human processing time approximates the cycle time C as close as possible.
  - If the given resources are so few or the given cycle time is so tight that the inequality $T_n < C$ for all the workstations cannot be satisfied, the variance of a set of workstation times is as low as possible.

Methodology: Algorithm

Here, we introduce an algorithm to solve our task assignment problem. We adopt an approximate approach: metaheuristics. There are so many important criteria for making task assignments that are difficult to represent in the objective that the difference between the exact and approximate solutions is inconsequential. The approximate solution is obtained more quickly. Moreover, some solutions obtained from metaheuristics corresponding to the local minima of the objective are often useful for manual selection based on the above criteria.

Our algorithm requires the following input:

- Desired number of workstations
- Set of IDs for the robot-assigned workstations
- Desired cycle time
- Durations of each operation
- Precedence constraints between operations
- Robot availability for each operation

Our method is based on a tabu search for a constraint satisfaction problem. We propose the following algorithm to generate possible solutions to the problem:

1. Generate the initial assignment of tasks to the workstations using first-fit heuristics.
2. Exchange two tasks, or move a task to another workstation randomly.
3. Count the number of tasks violating the constraints, and calculate the objective.
4. Stochastically undo the changes based on the metaheuristics (tabu search).

Data

We chose a laptop as an industrial case study. It is manufactured on an assembly line in a factory. Table 1 presents the data of this case study. The grayed tasks are robot available. The durations for humans and robots are given for these tasks. The predecessors of each task are shown in the Precedents column.

We also used an artificial data set containing 1000 tasks to confirm the scalability of our method. Each task has random duration and precedents. Each duration $0 < t_i < 10$ is a uniform random number. For the precedents, we used two normal random numbers that represent a group number G and an order O in the group. **Error! Reference source not found.** shows the precedence graph generated from a set of group numbers and orders (G, O). An arc exists based on orders in the group,

Table 1 – Data on Laptop Assembly

	Durations [s] for		
Task	**Human**	**Robot**	**Precedents**
1	1		
2	1		
3	2	2	
4	3		
5	5		3
6	1	3	3
7	3	6	6
8	3	6	6
9	4		6
10	3	5	9
11	5	6	6
12	7	6	11
13	5		
14	3	6	5
15	2	6	12
16	1		14,15
17	2	4	16
18	1		17
19	3		18
20	1		19
21	3	15	20
22	1		21
23	3		3
24	1	5	
25	4		24
26	4		25
27	4		
28	1	2	26
29	3		28,27
30	2	5	29
31	2		3
32	3		22,13,8,10,7
33	3	6	
34	8		33
35	10		4
36	6		6
37	3		32
38	4		37
39	2		32
40	6	15	39
41	5	10	39
42	1		40,41
43	7		34
44	3		43,32
45	2		

	Durations [s] for		
Task	**Human**	**Robot**	**Precedents**
46	4		45
47	4		30,46
48	4		46
49	5		47
50	4	6	49
51	5		43
52	5	5	51,44
53	4		
54	2	5	13,53
55	4		32
56	1		54,55
57	5		13
58	2	8	57
59	4	8	32
60	2	5	
61	2	6	60
62	3		61
63	6		62,13
64	2	5	63
65	2	5	13,31
66	2	5	54
67	4	10	47
68	5		56,66,42,38,36,59,58, 52,23,64,67,65,50,48
69	6		
70	4		69
71	4		70
72	1		71
73	4		72
74	5		69
75	5		
76	3	6	75
77	5		75
78	2		75
79	5		73,77,78
80	5		
81	9		68
82	5	5	68
83	6	10	79
84	5	10	79
85	6		83,74,84
86	3		85
87	3		80,84,83
88	5		87
89	5		88
90	2		89

	Durations [s] for		
Task	**Human**	**Robot**	**Precedents**
91	6		90
92	6	6	91
93	7		92
94	5		93
95	2		
96	7		86,94,95,76
97	6	10	96
98	3		97,68
99	4		95
100	6		95
101	7	20	68
102	4		98
103	6	6	46
104	6		98
105	8		98
106	4		104
107	10		106
108	4		107,68
109	7		108
110	3		102
111	6		102
112	12		109
113	4		35
114	5	10	113
115	5	5	114
116	3		68
117	5	5	68
118	4		103,82,81,110,117,116
119	3		118
120	10		95
121	7		98,111,112,115,101
122	2		121,119,120,99,100
123	2		
124	8		13
125	3	5	13
126	2		122
127	7	15	126
128	1		127
129	3		128
130	3		122
131	3		130
132	7	5	75
133	1		129,132,125,131,124
134	1		133
135	1		134

and we assume that there is no arc between tasks belonging to separate groups. In **Error! Reference source not found.**, for example, task 1 has $(G, O) = (20, 1)$ and task 2 has $(G, O) = (20, 2)$, and there exists an arc from task 1 to task 2. It is possible for two or more tasks to have the same set (G, O) as task 3 and 4 in this figure. No group and no order can occur as $G = 19$ and $(G, O) = (10, 8)$ in this figure. Here, each group number is $1 \leq G \leq 20$ sampling from the normal distribution whose mean and variance are $21/2$ and $19/4$, respectively. On the other hand, each order is $1 \leq O \leq 10$ sampling from the normal distribution whose mean and variance are $11/2$ and $9/4$, respectively.

Note that we checked the balancing speed under the precedence constraints here. Thus, we assumed that all resources are humans and ignored the robot constraints and unbalanced assignment using the objective E_1.

Results

Here, we summarize the assignment results for the above data. Three robots were placed in the 1st, 4th and 7th workstations. We showed assignment results using stacked bar charts, such as Figure 3 and Figure 4. The horizontal axis is the workstation number, and the vertical axis is the workstation time. A bar corresponds to a task, and the number written in each bar shows

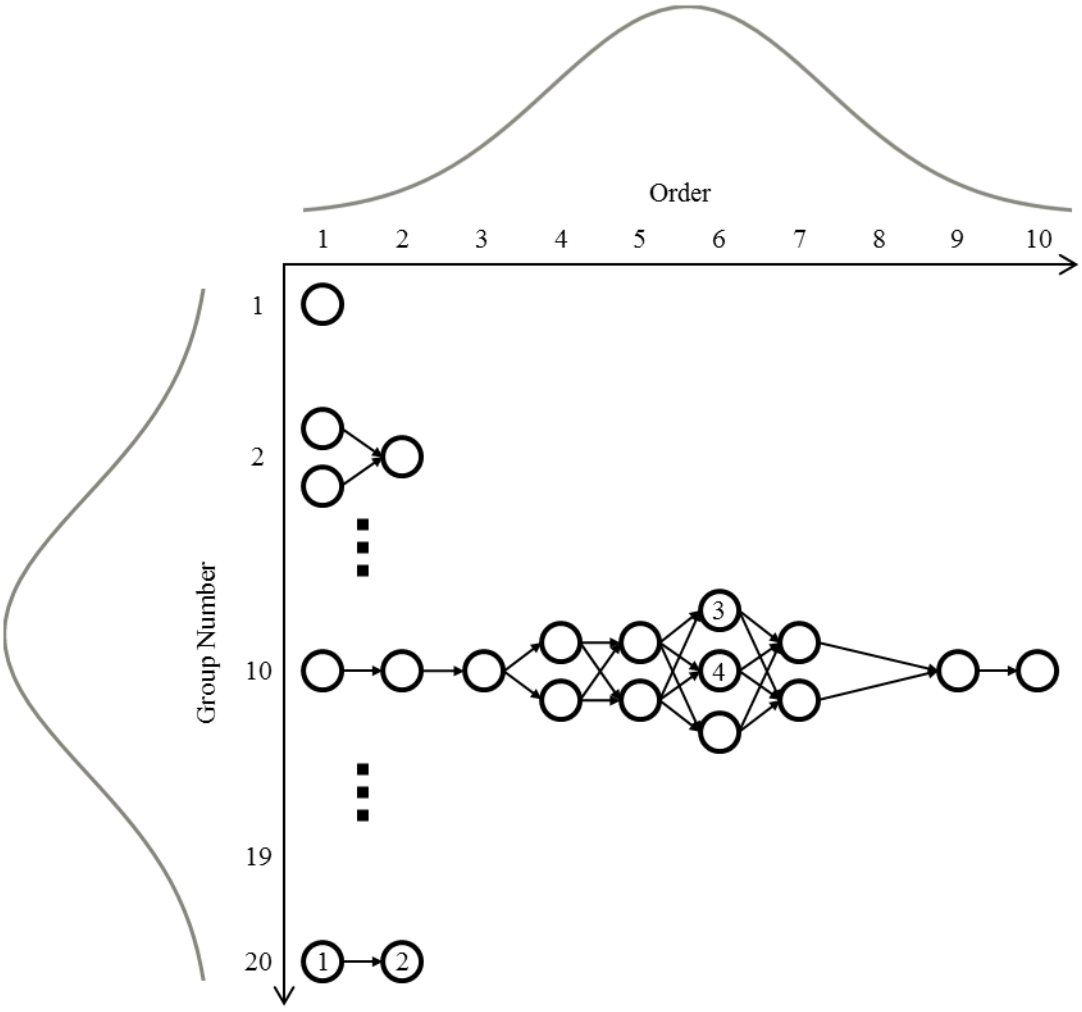


Figure 2 – Precedence Graph Generated from Group Numbers and Orders

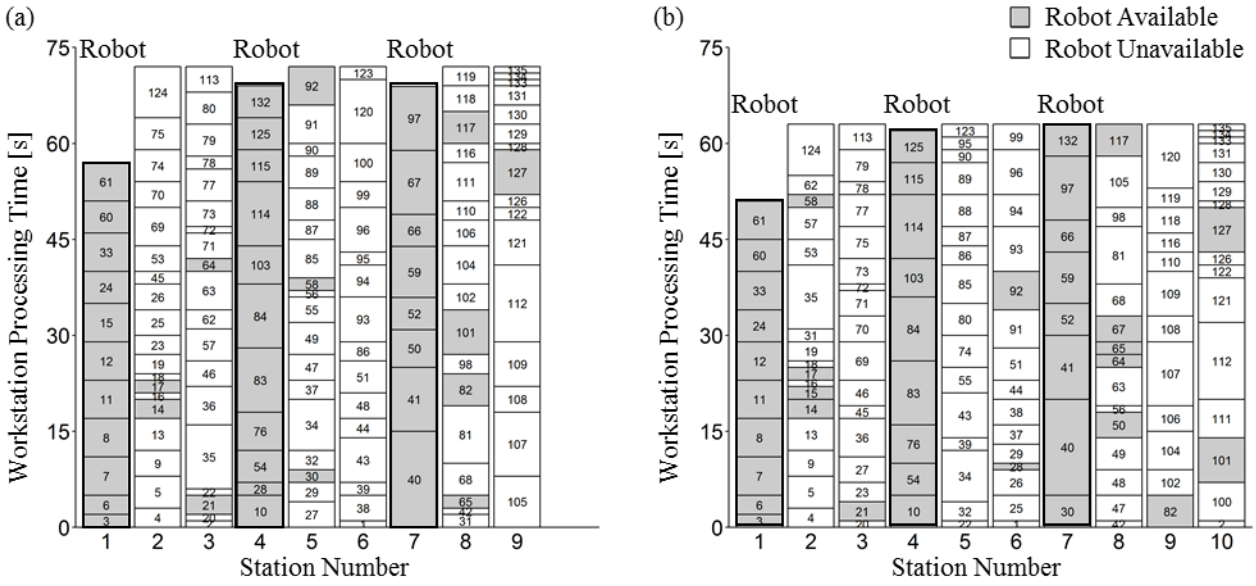


Figure 3 – Charts of Assignment Using the Objective E_2 for (a) $N = 9$, (b) $N = 10$

which task corresponds to the bar. The height represents the duration of the task. The color of each box represents the robot availability for the task; gray is available, and white is unavailable. In Figure 3 and Figure 4, the only available tasks are assigned to the robot workstations, and the workstation times are well balanced except for the first workstation to which assignment is restricted because of the precedence and robot constraints in this data. The precedence constraints are also satisfied by all the tasks, although it is difficult to confirm this from the charts.

Figure 3 (a) and (b) show the results of conventional assignments using the objective E_2 for the number of workstations $N = 9, 10$, respectively. Their maximum workstation times are 72 and 63 s, respectively. When the cycle time is nearly equal to 72 or 63 s, the assignment of Figure 3 (a) or (b) can be adopted, unless the number of workstations or resources is restricted. When the cycle time is $63\ \mathrm{s} < C < 72\ \mathrm{s}$, for example, $C = 67\ \mathrm{s}$, are the number of workstations appropriate? If

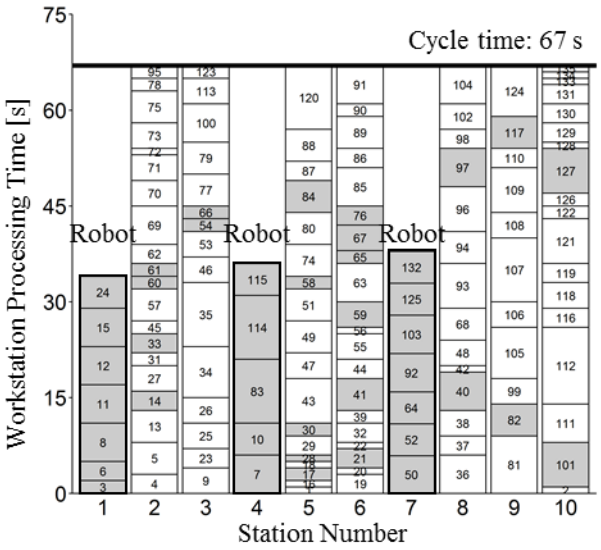


Figure 4 – Chart of Assignment Using the Objective E_1 for $N = 10, C = 67$ s

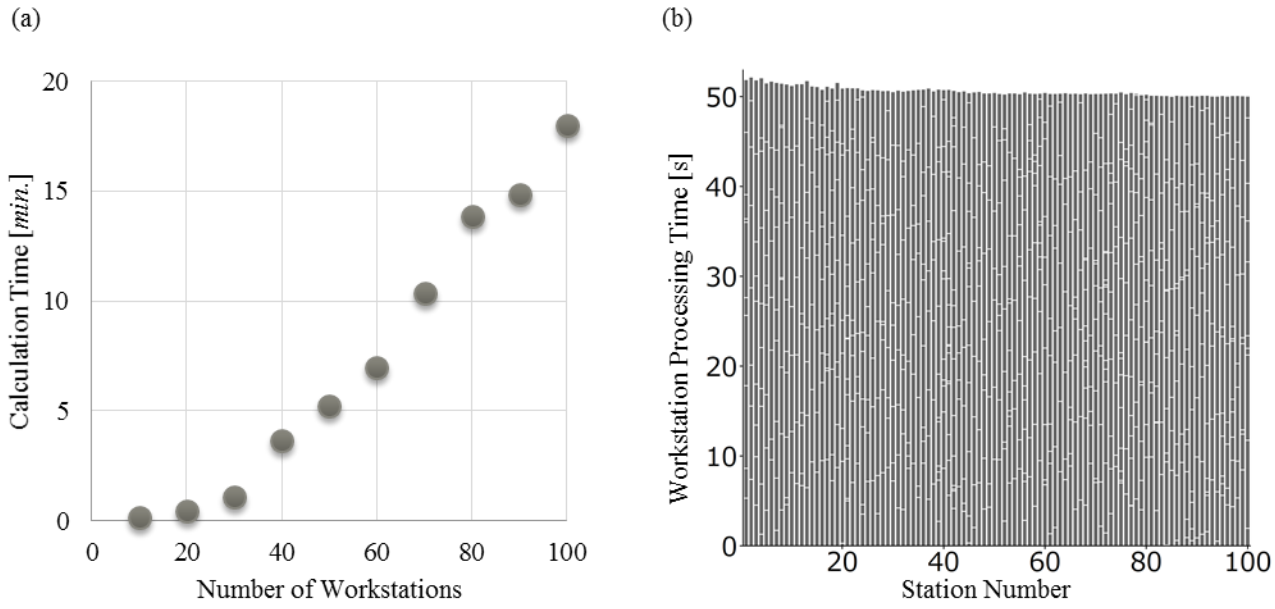


Figure 5 – (a) Relation between Calculation Time and Number of Workstations, (b) Assignment Chart for $N = 100$

overtime cannot be done, $N = 10$ has to be chosen. The assignment of Figure 3 (b), however, is not appropriate because of long idle times at human-assigned workstations.

Thus, our new assignment using the objective E_1 for $N = 10, C = 67$ s is displayed in Figure 4. Here, the workstation times for the humans are balanced around the cycle time $C = 67$ s. As a result, the total robot processing time for our assignment (Figure 4) is 108 s, which is approximately 40% shorter than that of the conventional assignment's 176 s (Figure 3 (b)). These are the main results in this paper. Note that our method cannot determine which of the two assignments, Figure 3 (a) and Figure 4, is better. The number of workstations can be decided considering various factors in the line.

Finally, we checked the scalability of our method. We applied our method to the artificial data set under Data, changing the number of workstations and measuring their calculation times, where each calculation stops when the standard deviation of a set of station times reaches 1 s. Figure 5 (a) summarizes the results. The vertical axis is the calculation time, and the horizontal axis is the number of workstations. Our method can finish the calculation within 1 minute when the number of workstations is $N = 20$ and within 20 minutes when the number of workstations is $N = 100$. Figure 5 (b) shows the assignment chart for $N = 100$ as an example of our calculation result. Each workstation time is well balanced.

Conclusions

In this paper, we presented a new method of handling an assignment problem for hybrid manual and robotic assembly mixing lines. The method is based on a tabu search for a constraint satisfaction problem. The aim is to assign tasks to workstations, each of which is occupied by either a human or a robot. The focus is on how to deal with the idle time of each workstation. By assigning tasks to the workstations of humans so that no idle time remains, our method reduces the total processing time of

robots by approximately 40% in the laptop case study presented. In the future, further research will be undertaken on reassigning existing assembly lines in response to changes to the production process or production program.

Acknowledgments

This work was supported by the Grant-in-Aid of the New Energy and Industrial Technology Development Organization (NEDO) of Japan (Project Code P15008).

References

[1] M. Padrón, M. de los A. Irizarry, P. Resto, and H. P. Mejía, "A methodology for cost-oriented assembly line balancing problems," *J. Manuf. Technol. Manag.*, vol. 20, no. 8, pp. 1147–1165, Oct. 2009.

[2] Ronald G. Askin and Charles R. Standridge, *Modeling and Analysis of Manufacturing Systems*. 1993.

[3] S. Y. Nof, W. E. Wilhelm, and H.-J. Warnecke, *Industrial Assembly*. Chapman & Hall, 1997.

[4] A. Scholl, *Balancing and Sequencing of Assembly Lines*. Physica-Verlag, Heidelberg, 1999.

[5] J. P. A. Dolgui, *Supply Chain Engineering: Useful Methods and Techniques*. Springer, 2010.

[6] A. Kimms, "Minimal investment budgets for flow line configuration," *IIE Trans.*, vol. 32, no. 4, pp. 287–298, 2000.

[7] G. W. Zhang, S. C. Zhang, and Y. S. Xu, "Research on flexible transfer line schematic design using hierarchical process planning," *J. Mater. Process. Technol.*, vol. 129, no. 1–3, pp. 629–633, 2002.

[8] O. Battaïa, A. Dolgui, N. Guschinsky, and G. Levin, "A decision support system for design of mass production machining lines composed of stations with rotary or mobile table," *Robot. Comput. Integr. Manuf.*, vol. 28, no. 6, pp. 672–680, Dec. 2012.

[9] S. C. Graves and D. E. Whitney, "A mathematical programming procedure for equipment selection and system evaluation in programmable assembly," *Decis. Control Incl. Symp. Adapt. Process. 1979 18th IEEE Conf.*, vol. 2, pp. 531–536, 1979.

[10] S. C. Graves and B. W. Lamar, "An integer programming procedure for assembly system design problems," *Oper. Res.*, vol. 31, no. 3, pp. 522–545, Jun. 1983.

[11] R. E. Gustavson, "Design of cost effective assembly systems," in *C. S. Draper Laboratory Report*, Cambridge, 1986, p. 2661.

[12] S. C. Graves and C. H. Redfield, "Equipment selection and task assignment for multiproduct assembly system design," *Int. J. Flex. Manuf. Syst.*, vol. 1, no. 1, pp. 31–50, Sep. 1988.

[13] E. Falkenauer, "Solving equal piles with the grouping genetic algorithm," pp. 492–497, Jul. 1995.

[14] J. Bukchin and M. Tzur, "Design of flexible assembly line to minimize equipment cost," *IIE Trans.*, vol. 32, no. 7, pp. 585–598, 2000.

[15] B. Rekiek, P. De Lit, F. Pellichero, T. L'Eglise, P. Fouda, E. Falkenauer, and A. Delchambre, "A multiple objective grouping genetic algorithm for assembly line design," *J. Intell. Manuf.*, vol. 12, no. 5–6, pp. 467–485, 2001.

[16] A. C. Nearchou, "Balancing large assembly lines by a new heuristic based on differential evolution method," *Int. J. Adv. Manuf. Technol.*, vol. 34, no. 9–10, pp. 1016–1029, Nov. 2006.

[17] S. B. Liu, K. M. Ng, and H. L. Ong, "Branch-and-bound algorithms for simple assembly line balancing problem," *Int. J. Adv. Manuf. Technol.*, vol. 36, no. 1–2, pp. 169–177, Nov. 2006.

[18] O. Kilincci and G. M. Bayhan, “A P-invariant-based algorithm for simple assembly line balancing problem of type-1,” *Int. J. Adv. Manuf. Technol.*, vol. 37, no. 3–4, pp. 400–409, Mar. 2007.

[19] D. D. Sheu and J.-Y. Chen, “Line balance analyses for system assembly lines in an electronic plant,” *Prod. Plan. Control*, Mar. 2008.

[20] U. Özcan and B. Toklu, “A new hybrid improvement heuristic approach to simple straight and U-type assembly line balancing problems,” *J. Intell. Manuf.*, vol. 20, no. 1, pp. 123–136, May 2008.

[21] C. Blum, “Beam-ACO for simple assembly line balancing,” *INFORMS J. Comput.*, vol. 20, no. 4, pp. 618–627, Nov. 2008.

[22] R. Pastor and L. Ferrer, “An improved mathematical program to solve the simple assembly line balancing problem,” *Int. J. Prod. Res.*, vol. 47, no. 11, pp. 2943–2959, Jun. 2009.

[23] O. Kilincci, “A Petri net-based heuristic for simple assembly line balancing problem of type 2,” *Int. J. Adv. Manuf. Technol.*, vol. 46, no. 1–4, pp. 329–338, May 2009.

[24] W. Ho and A. Emrouznejad, “A mathematical model for assembly line balancing model to consider disordering sequence of workstations,” *Assem. Autom.*, vol. 29, no. 1, pp. 49–51, Feb. 2009.

[25] J. Bautista and J. Pereira, “A dynamic programming based heuristic for the assembly line balancing problem,” *Eur. J. Oper. Res.*, vol. 194, no. 3, pp. 787–794, May 2009.

[26] O. Kilincci, “Firing sequences backward algorithm for simple assembly line balancing problem of type 1,” *Comput. Ind. Eng.*, vol. 60, no. 4, pp. 830–839, May 2011.

[27] E. C. Sewell and S. H. Jacobson, “A branch, bound, and remember algorithm for the simple assembly line balancing problem,” *INFORMS J. Comput.*, vol. 24, no. 3, pp. 433–442, Aug. 2012.

A New Line Balancing Method Considering Robot Count and Operational Costs in Electronics Assembly

Ryo Murakami, Sachio Kobayashi, Hiroki Kobayashi and Junji Tomita

FUJITSU LABORATORIES LTD.

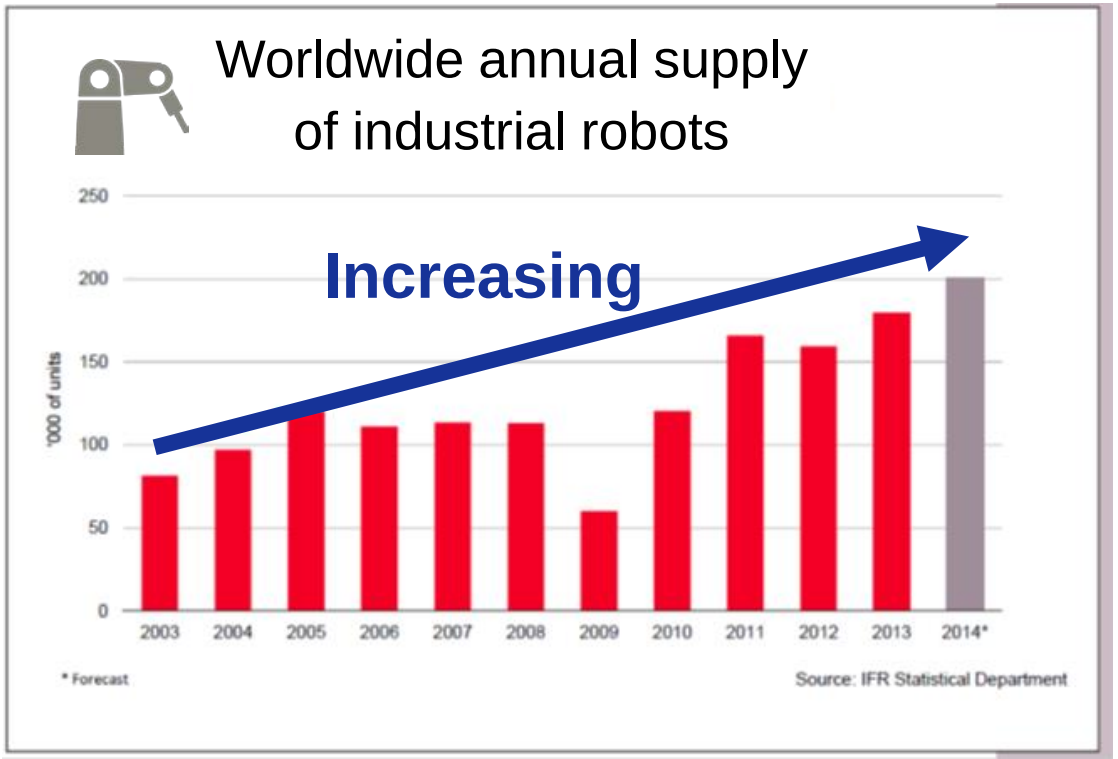
Contents

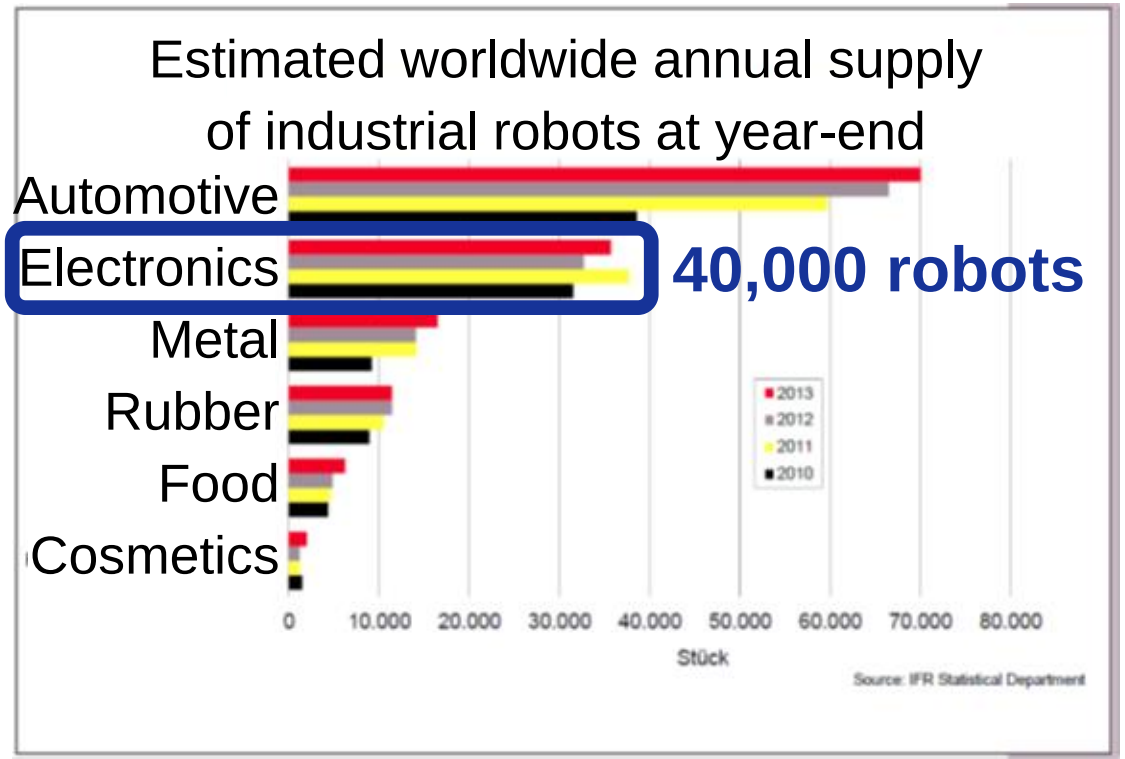
Contents

Robot Labor on Electronics Assembly Lines

- Increasing due to improvement of robot costs and utilization

Proceeding **robotic automation** is essential to reduce manufacturing costs and win a place in the electronics market.

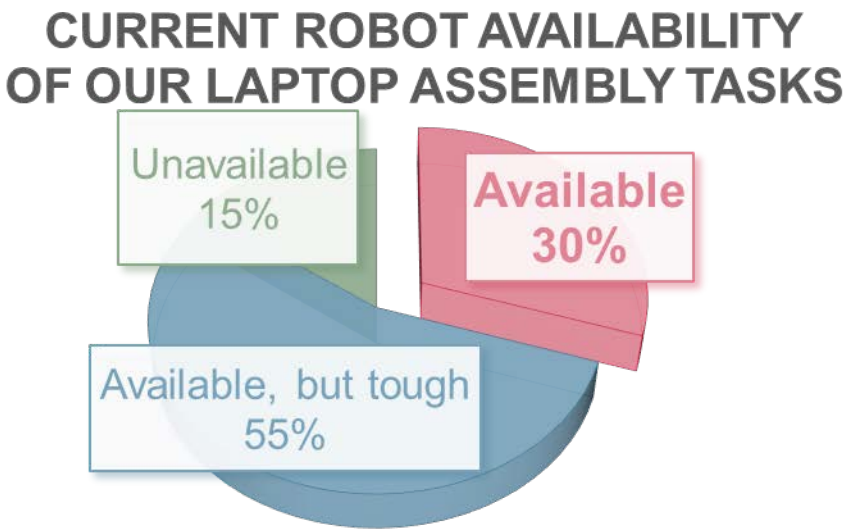




Barrier of Robotic Automation: Different from Humans

- Limitation of ability to do
  - *robot available only for 30% of total tasks in our laptop assembly*
- Difference of processing speed
  - *from almost same one to too slow one*

Which tasks are assigned to robots, that is **Line Balancing**, need to be carefully performed.



Average task time in our laptop assembly

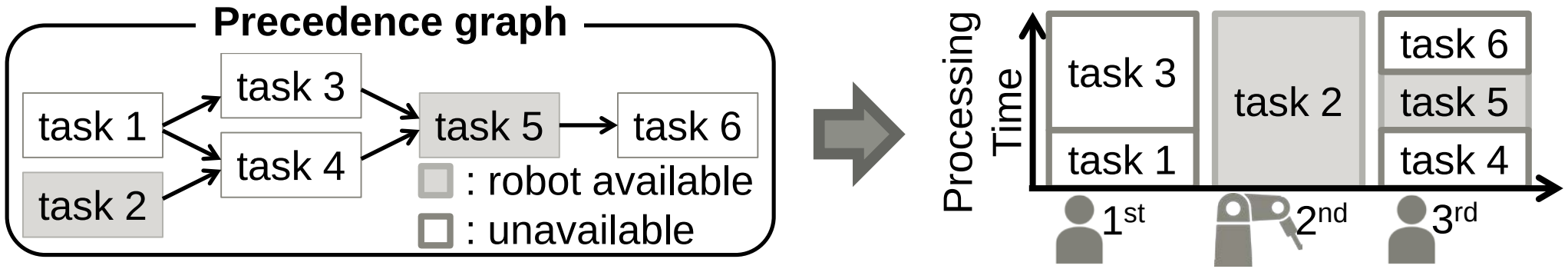
Robot	Human
7.2 s	3.9 s

Twice slower

Difficulty of Line Balancing

- We must find the best assignment in **enormous combinations** under **some constraints**.
  - *Each processing time balanced in the assignment*
- Experts manually make only good assignments.
  - *The room for improvement corresponding to $1M annual loss in a factory as will be shown later*

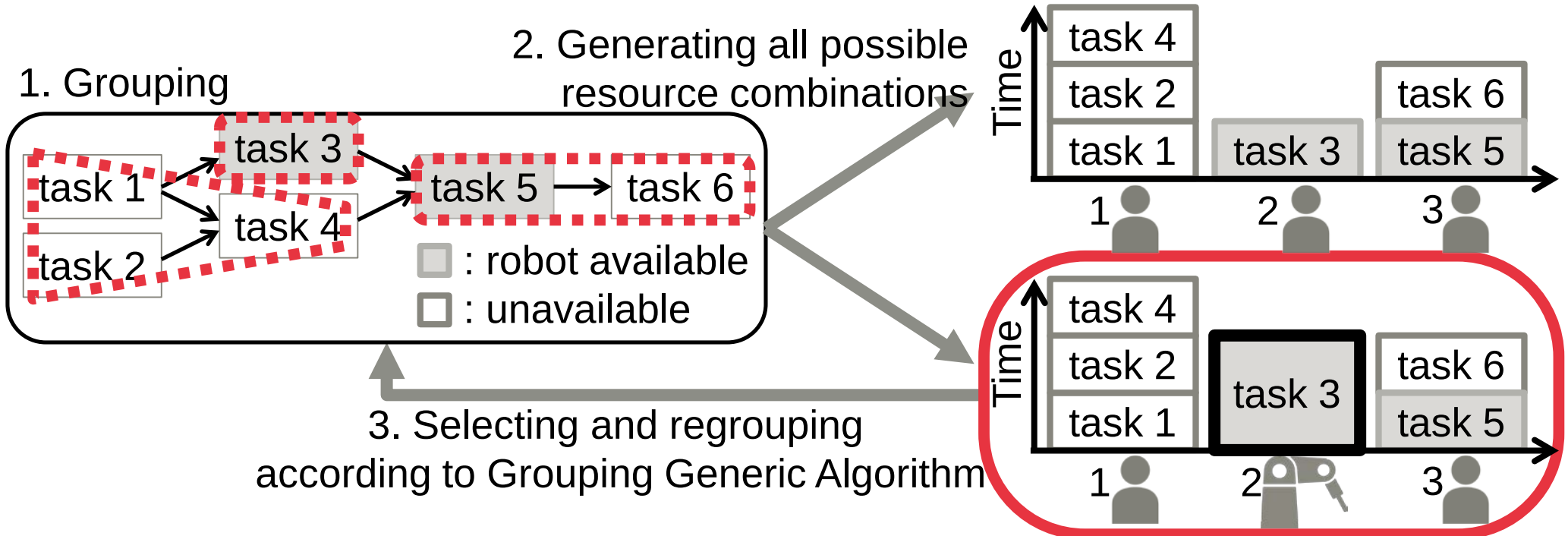
Automated high quality balancing in robot contained line is needed.



Previous Study - Rekiek *et al*. (1999)

- Iteratively grouping tasks and generating all possible resource combinations for the groups
  - *Robot count optimized by the generation*

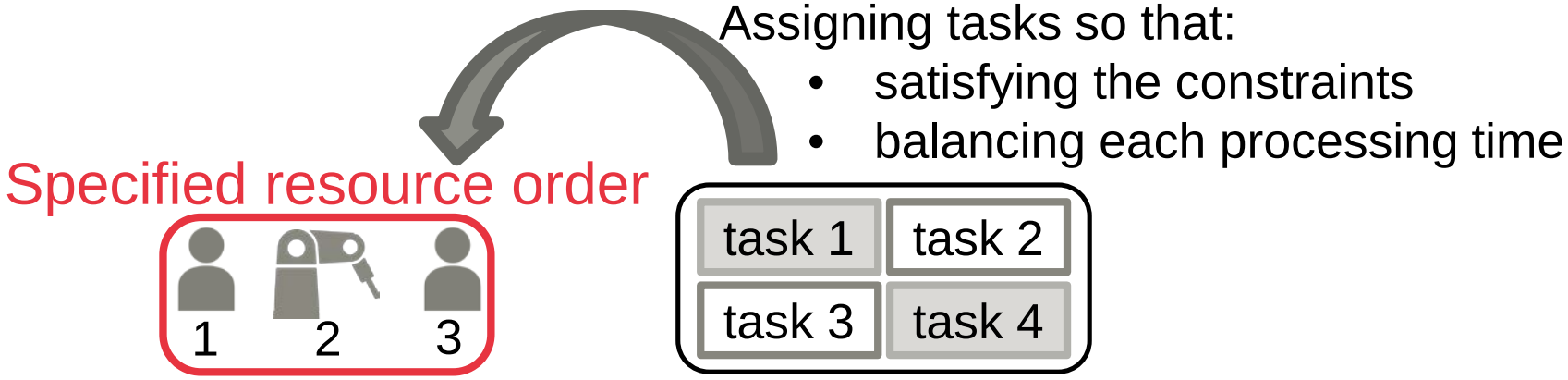
Robot count has been usually decided by a budget, but It is impossible to get the answer with specified robot count in this method.



Our Approach

- Solving the balancing problem as Constraint Satisfaction Problem:
  - ***For specified resource order**, assigning tasks so that the constraints are satisfied and each processing time is balanced*

Achieving line balancing with specified robot count

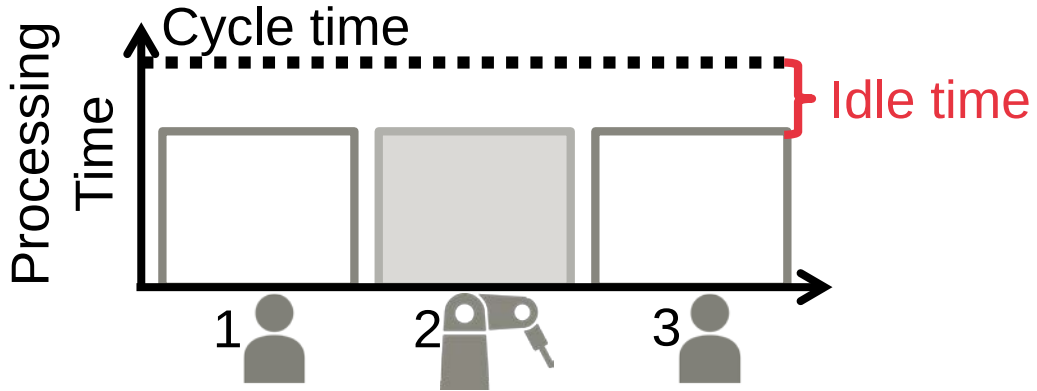


One More New Point: Effective Use of Idle Time

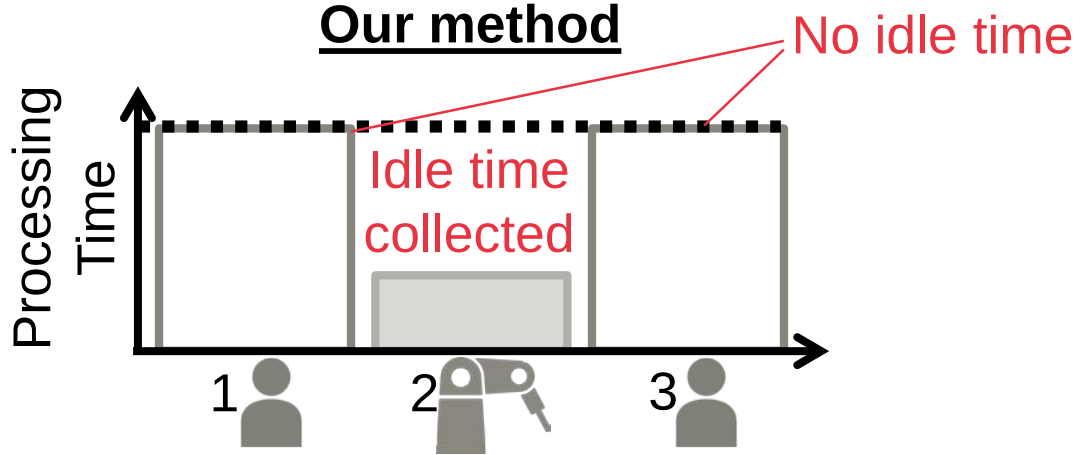
- Cycle time: processing time par a product given by the top
  - ***Highest priority**: to save it, sufficient human count is secured.*
- Idle time: difference between processing time and cycle time
  - *Wasting human resources*
  - ***Saving robot operational costs and extending robot life time***

We **collect all idle time into robots**,
although the previous study remains it in human parts.

Rekiek et al. (1999)



Our method

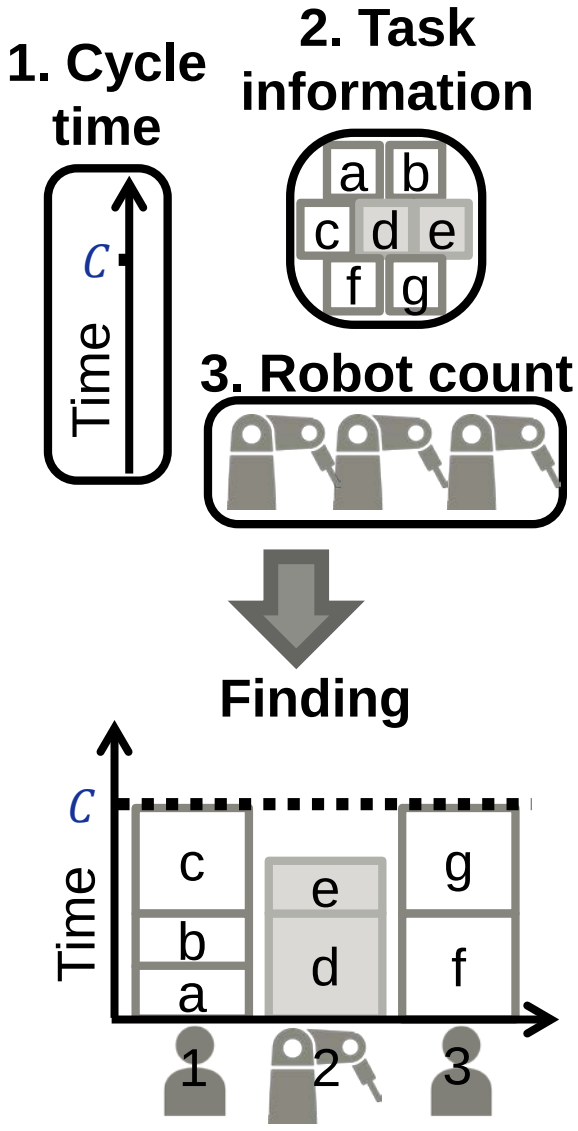


Contents

Problem Definition

- Given:
  1. *Task information*
  2. *Cycle time*
  3. *Robot count*
- Find our idle time collecting assignment
  - *Any resource order is acceptable, but humans are necessary and sufficient count.*

Given:	Task information
	Cycle time
	Robot count
Optimizing:	Human count
	Task assignment



About Task Information

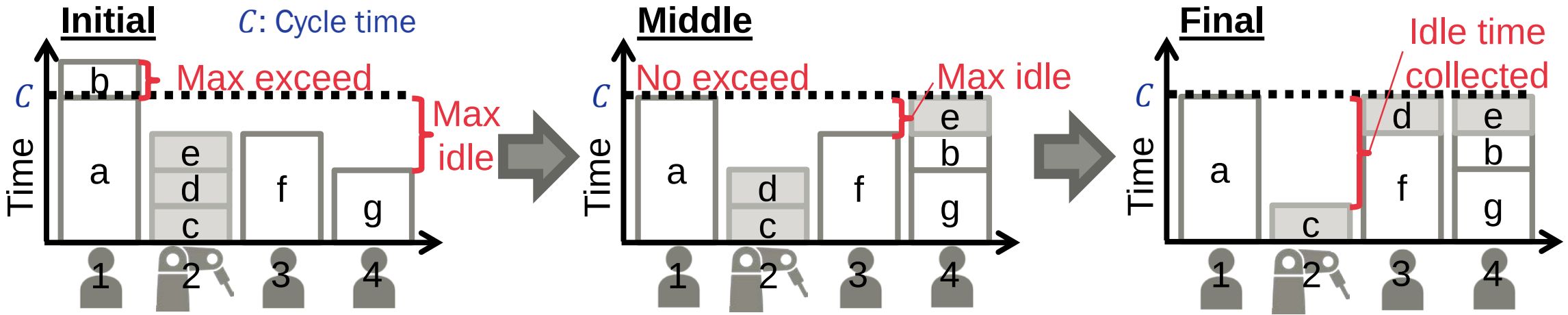
- 3 properties required:
  - *Precedents*
  - *Robot availability*
  - *Task time*

Task	Precedents	Robot Availability	Task Time [s]	
			Human	Robot
1		Unavailable	2	
2		Unavailable	3	
3	1	Available	7	10
4	1, 2	Unavailable	5	
5	3, 4	Available	4	7
6	5	Unavailable	6	

Robots' task time required for available tasks

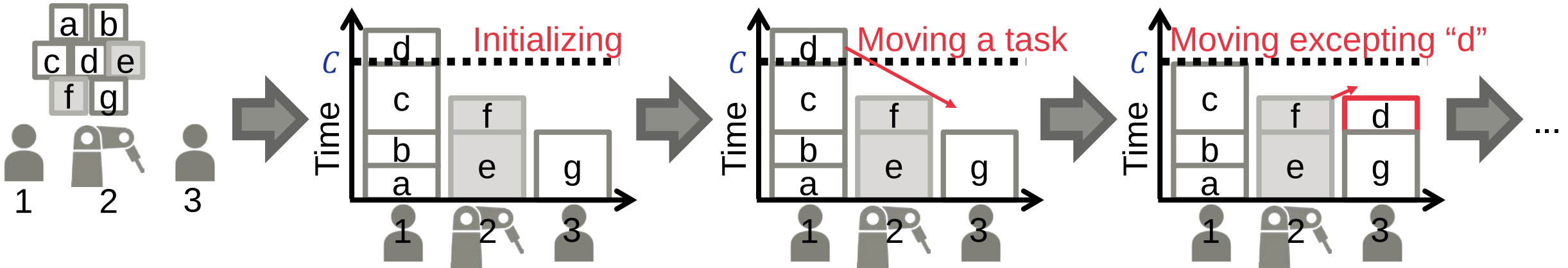
Idle Time Collecting Objective Function

- If **both exceed and idle time of all humans** can be minimized, idle time is collected into robots. (if sufficient human count)
- Thus, we adopted these values as the index to minimize, that is, objective function.
  - *Actually, the objective include only **the max exceed time** and **the max idle time**.*



Search Algorithm: Tabu Search

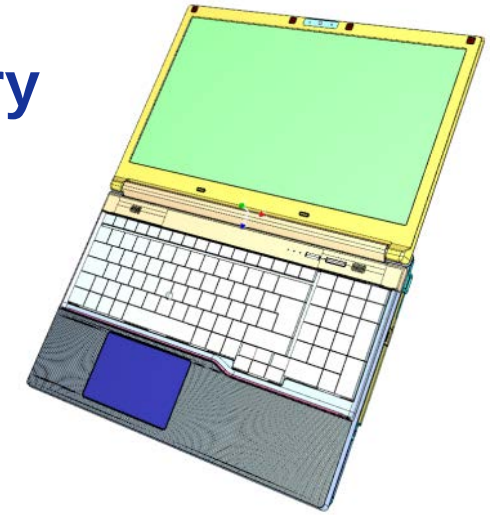
- Generating initial assignment using heuristics, and repeating possible local changes under tabus to minimize our objective
  - *Used heuristic: first fit*
    - often used for bin packing problems
  - *Used tabus: "resent moved **20%** tasks must not move."*



Contents

Used Data: an Industrial Case Study

- Assembly data of a **laptop, manufactured in a factory**
- Task count: **135** (robot available: **40**)
- Total task time by humans: **559 s**
  - *Each task time is an Integer.*



Task	Precedents	Robot Availability	Task Time [s]	
			Human	Robot
1		Unavailable	1	
⋮				
7	6	Available	3	6
⋮				
135	134	Unavailable	1	

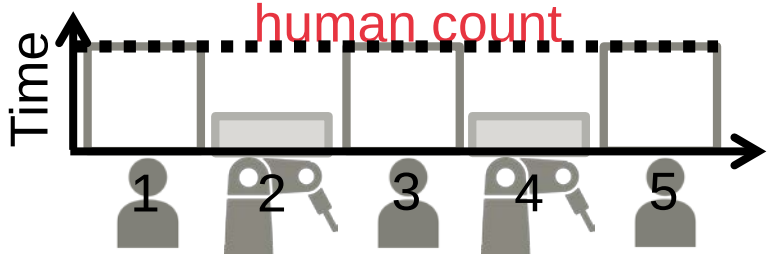
Overview of Results

1. An optimal assignment according to robot count
   - ***minimum human count*** *as well as 0 objective*
2. Balancing performance compared with manual
   - *How much our objective can be reduced*
     - Here human count and resource order was fixed for comparison.

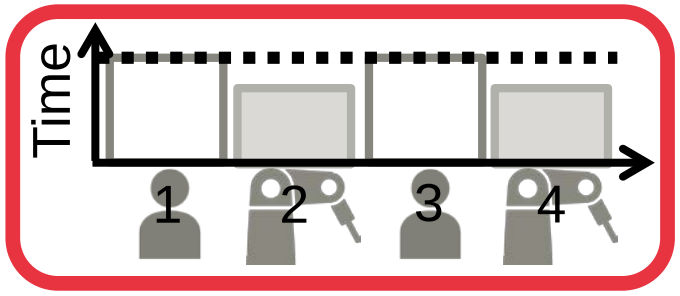
Problem Definition

Given:	Task information
	Cycle time
	Robot count
Optimizing:	Human count
	Task assignment

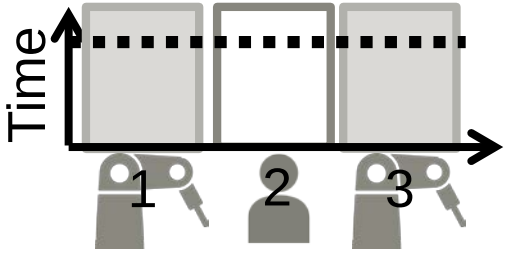
Not minimum human count



0 objective & minimum human count



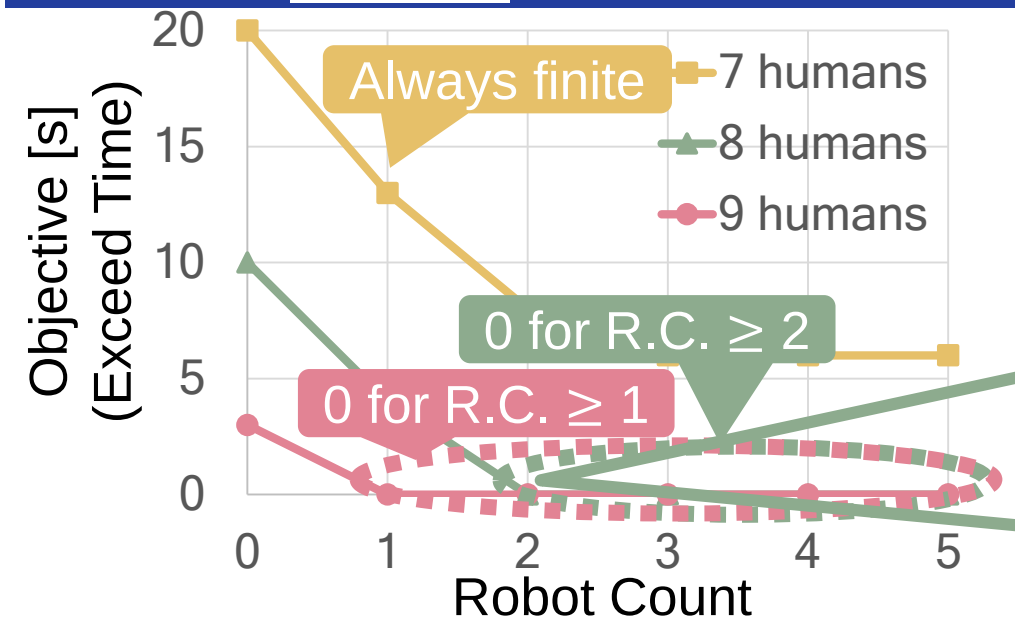
Not 0 objective

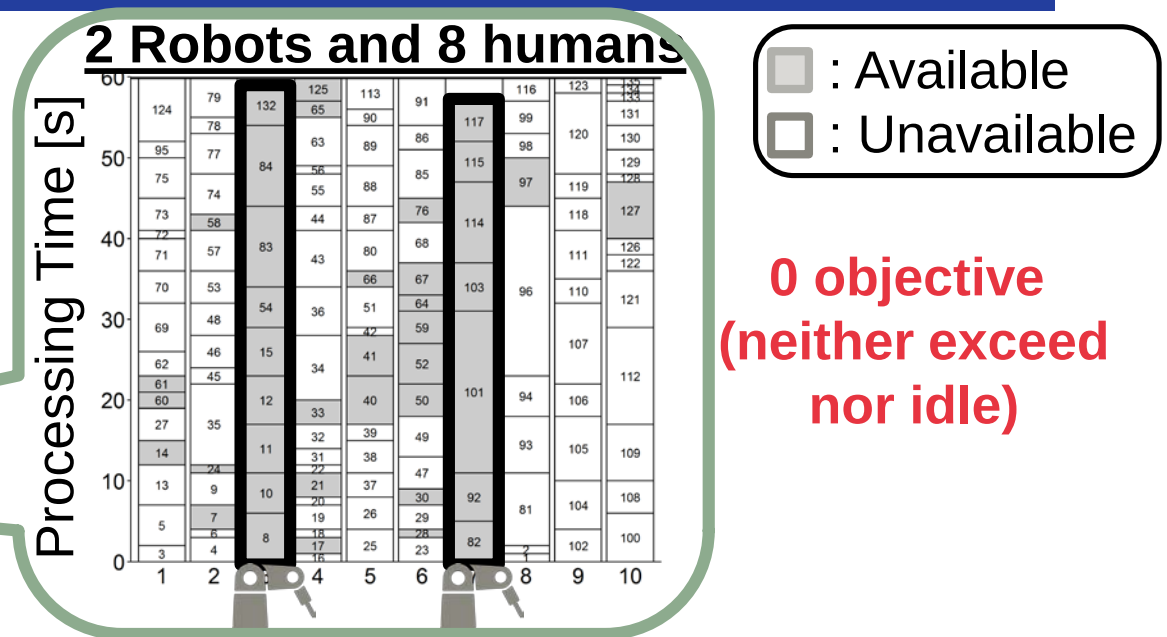


1. An Optimal Assignment According to Robot Count

- Cycle time: 60 s Robot count: from 0 to 5
- Left figure: relation between our objective and robot count for some human count
  - *7 humans: finite objective for any robot count*
  - *8 humans: **0 objective** for robot count ≥ 2*
  - *9 humans: **0 objective** for robot count ≥ 1*

We specified that an optimal assignment is that of **8 human** for robot count ≥ 2 and that of **9 human** for robot count =1.

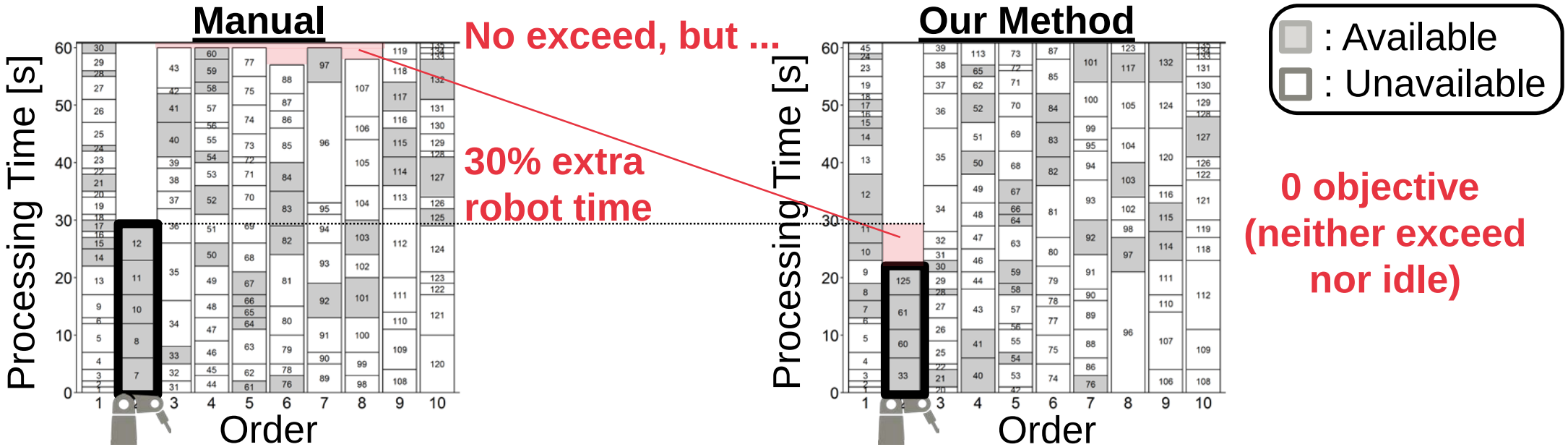




2. Balancing Performance Compared with Manual

- Condition: 61 s cycle time for 9 humans and 1 robot
- Manual: no exceed, but idle time in human parts

  Causing **30% extra robot time**
- Our method: **0 objective (neither exceed nor idle)**



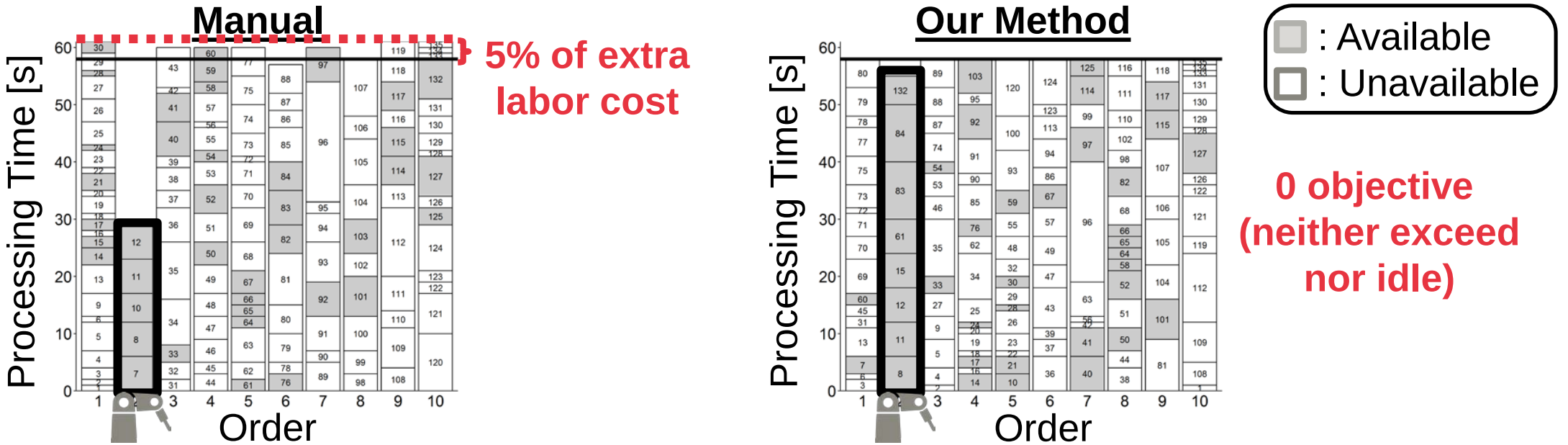
2. Balancing Performance Compared with Manual

- Condition (strict): 58 s cycle time for 9 humans and 1 robot
- Manual: 3 s exceeded

Causing an **extra cost for $3\ s/58\ s \simeq 5\%$** overtime working

- Our method: **0 objective (neither exceed nor idle)**

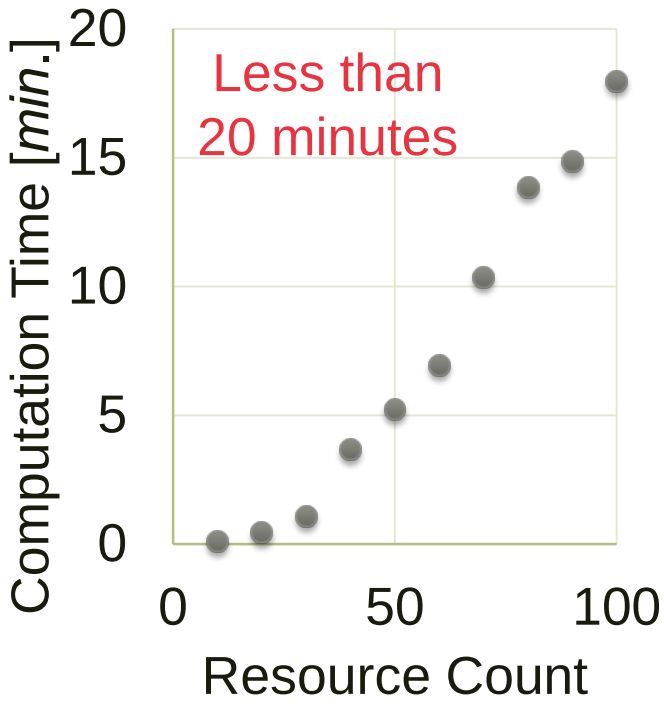
Our method can **automate high quality** assignments.



Contents

Scalability of Our Method

- Used data: artificial 1000 tasks
  - *Precedents: substituted for set of group number and order in the group generated by normal random number*
  - *Robot availability: all available*
  - *Task time: uniform random number from 0 to 10*
- Convergence condition: objective < 1 s



Resource count < 50 in large system (server, storage, ...) usually.
Our method can compute these assignments in practical time.

Future Work

- Implementation of experts' knowledge into our balancing algorithm
- Dual arm robots
- Mixed line

Conclusions

I presented a new balancing method for robot contained line.

- Main contribution for this research field: realization of idle time collecting assignment for **specified robot count**.
- Showed for our laptop assembly data set:
  - *An optimal assignment according to robot count*
  - *Balancing performance compared with manual assignment:*
    - Reduction of **30 % robot time** by collecting idle time into robots
    - Reduction of **5 % cost of human resources** by searching better assignments in enormous combinations exhaustively

Acknowledgements

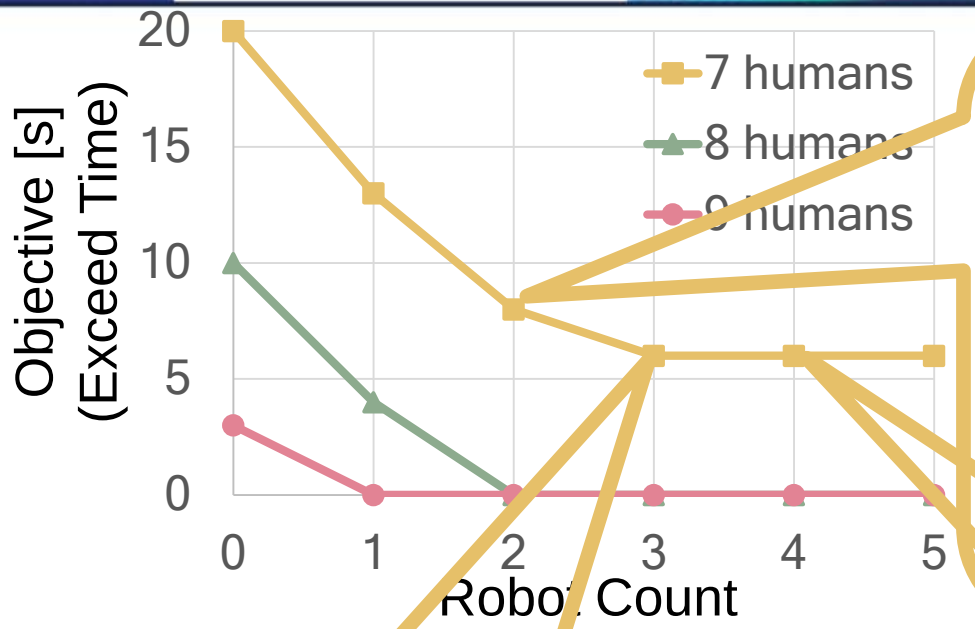
- **Satoshi Tomita**, Fujitsu Limited, Japan: providing us data of laptop assembly tasks
- **Yosuke Korotsune**, Shimane Fujitsu Limited, Japan: executing manual line balancing
- Colleagues, Companies
- This work was supported by the Grant-in-Aid of the New Energy and Industrial Technology Development Organization (NEDO) of Japan. (Project Code P15008)

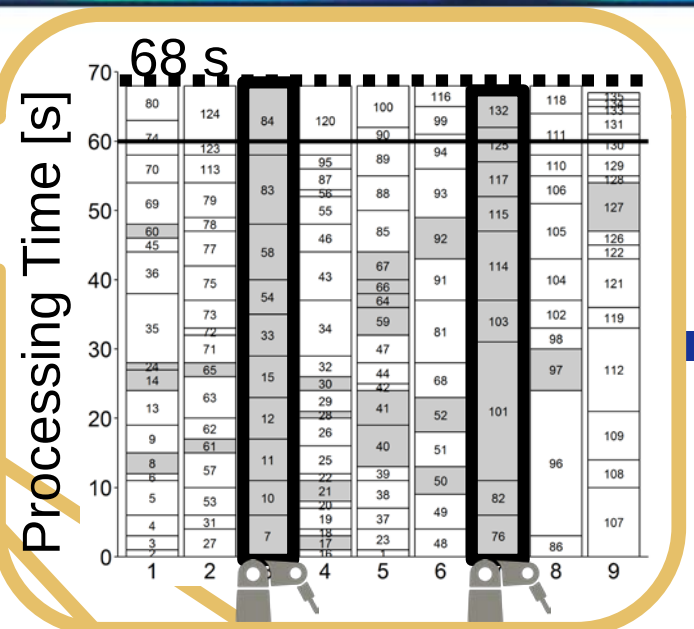
Collaborators

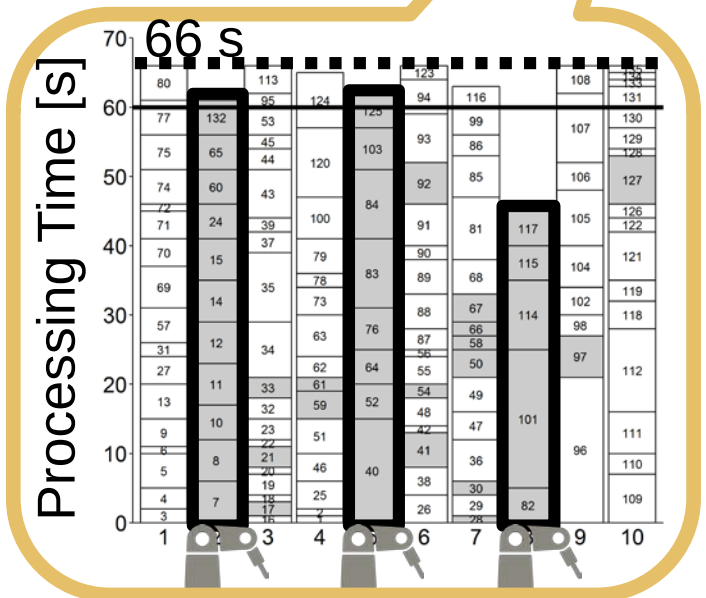
Sachio Kobayashi

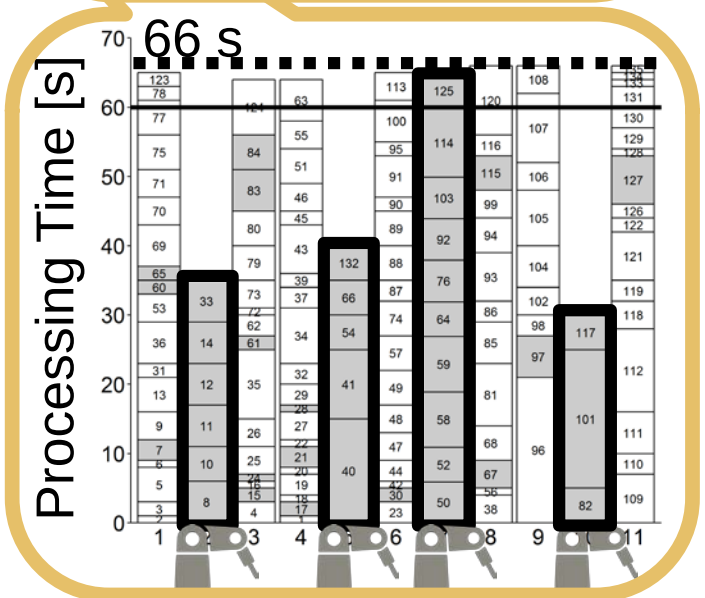
Hiroki Kobayashi

Junji Tomita

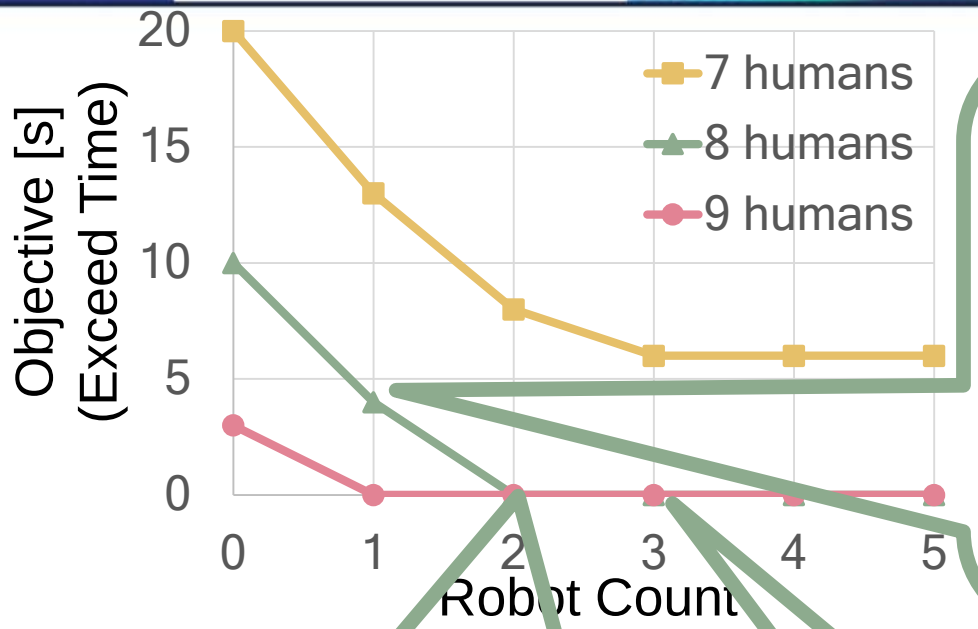


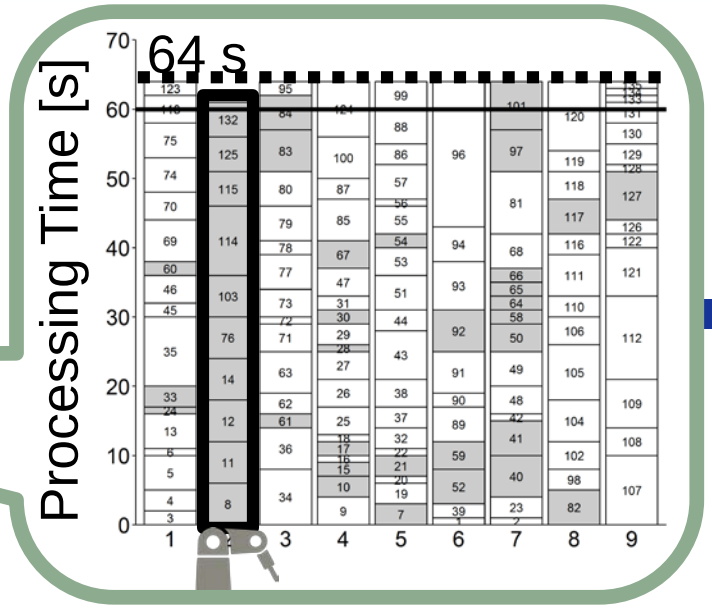


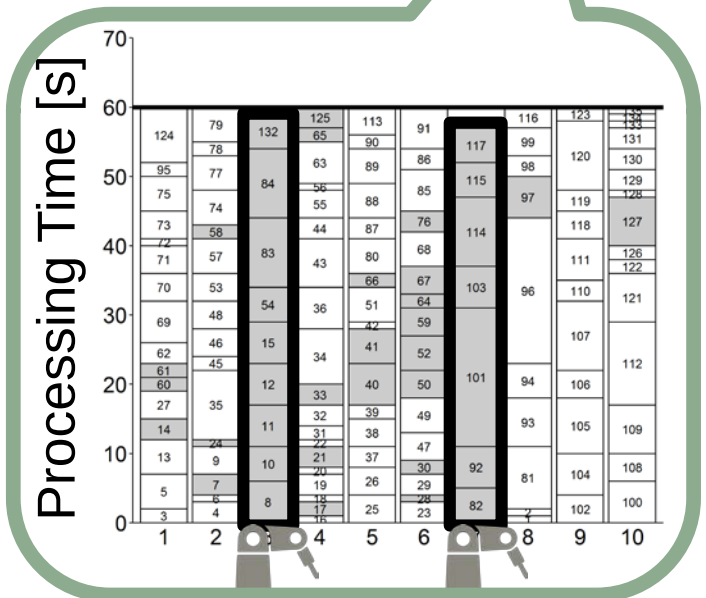


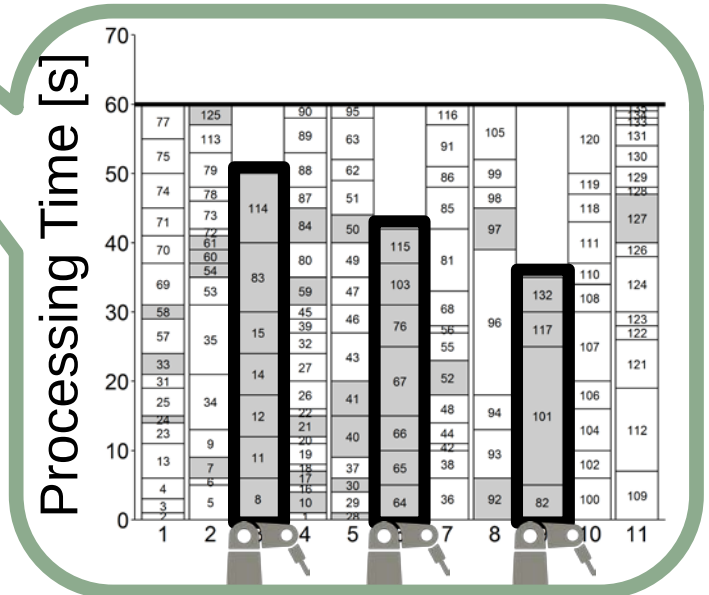


- For 7 humans:
  - *Top right:* ***2 robots***
  - *Bottom left:* ***3 robots***
  - *Bottom right:* ***4 robots***



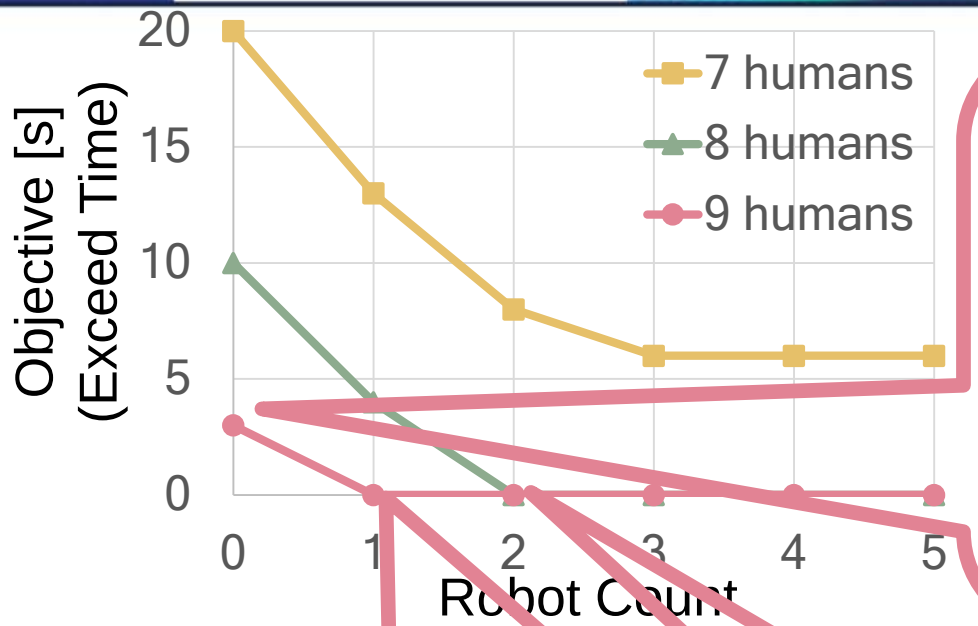


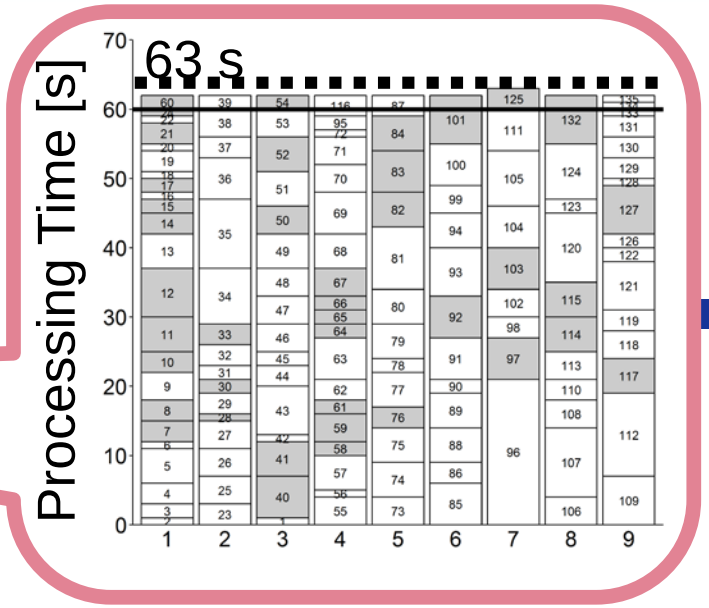


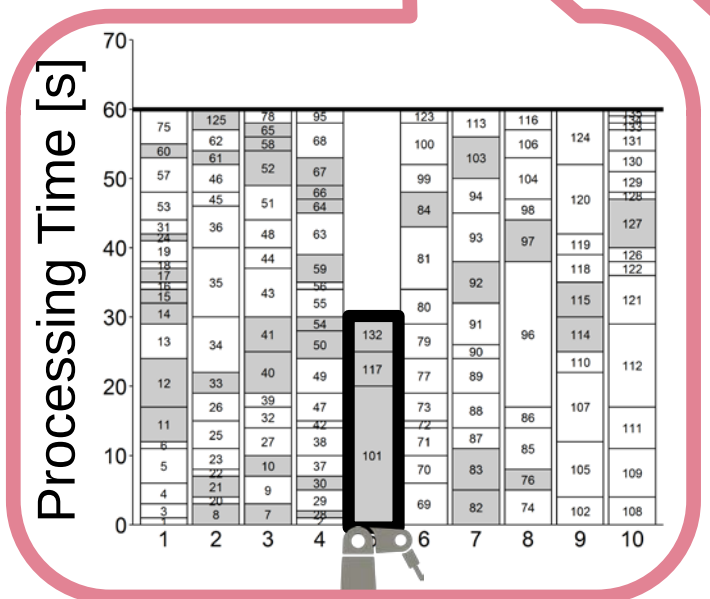


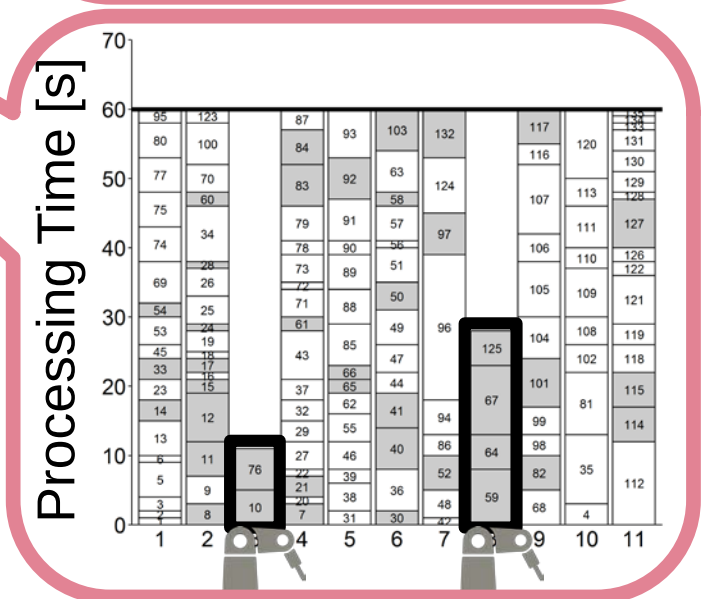
: Available
: Unavailable

- For 8 humans:
  - *Top right:* ***2 robots***
  - *Bottom left:* ***3 robots***
  - *Bottom right:* ***4 robots***







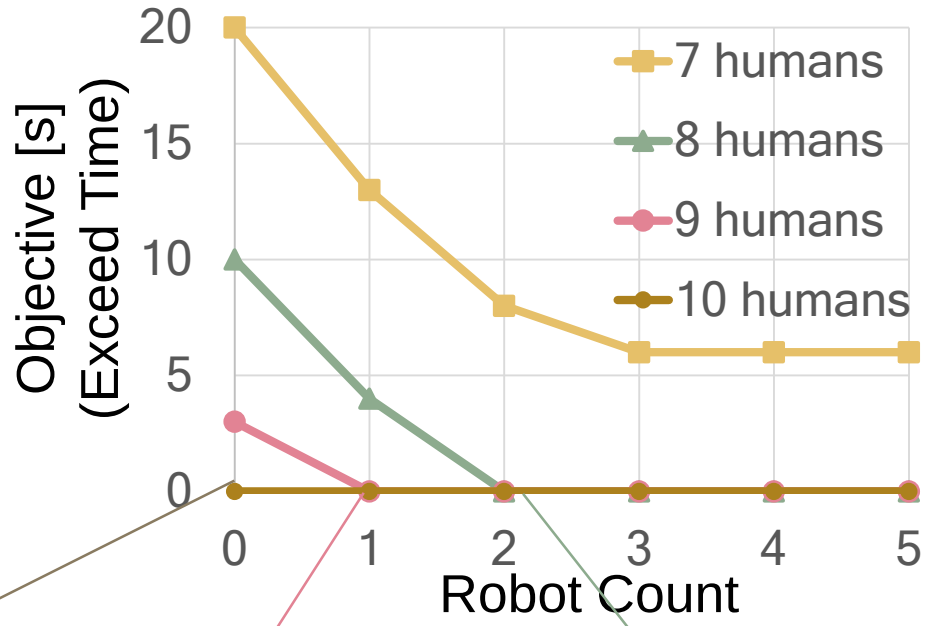


: Available
: Unavailable

- For 9 humans:
  - *Top right:* ***0 robot***
  - *Bottom left:* ***1 robots***
  - *Bottom right:* ***2 robots***

Robot and Human Count

- Cycle time: 60 s



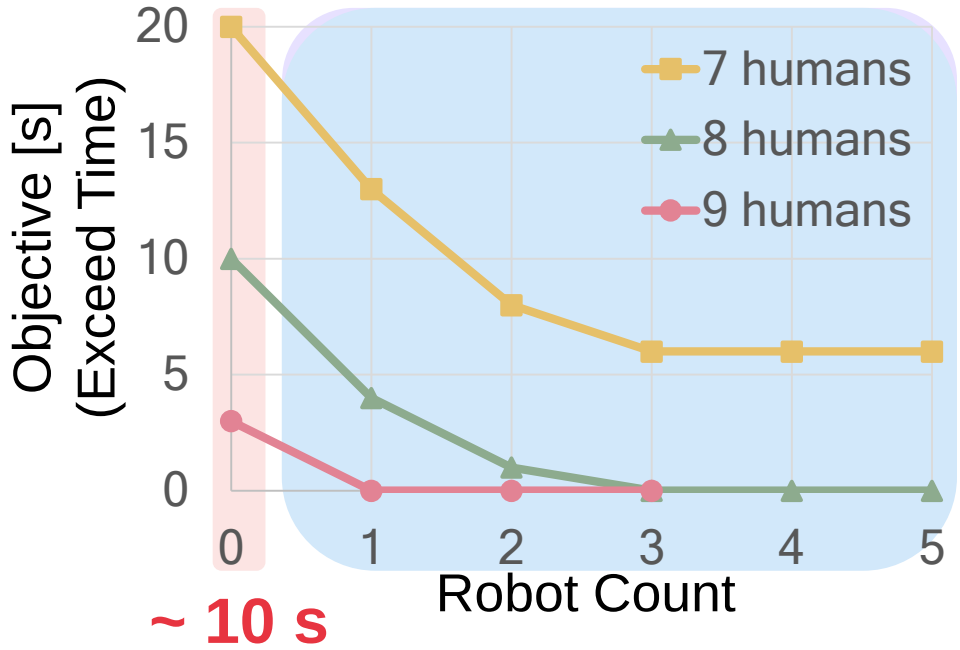
Resource Combinations Obtaining 0 Objective

Robot count	0	1	2
Human count	10	9	8

+1 robot
-1 human

+1 robots
-1 human

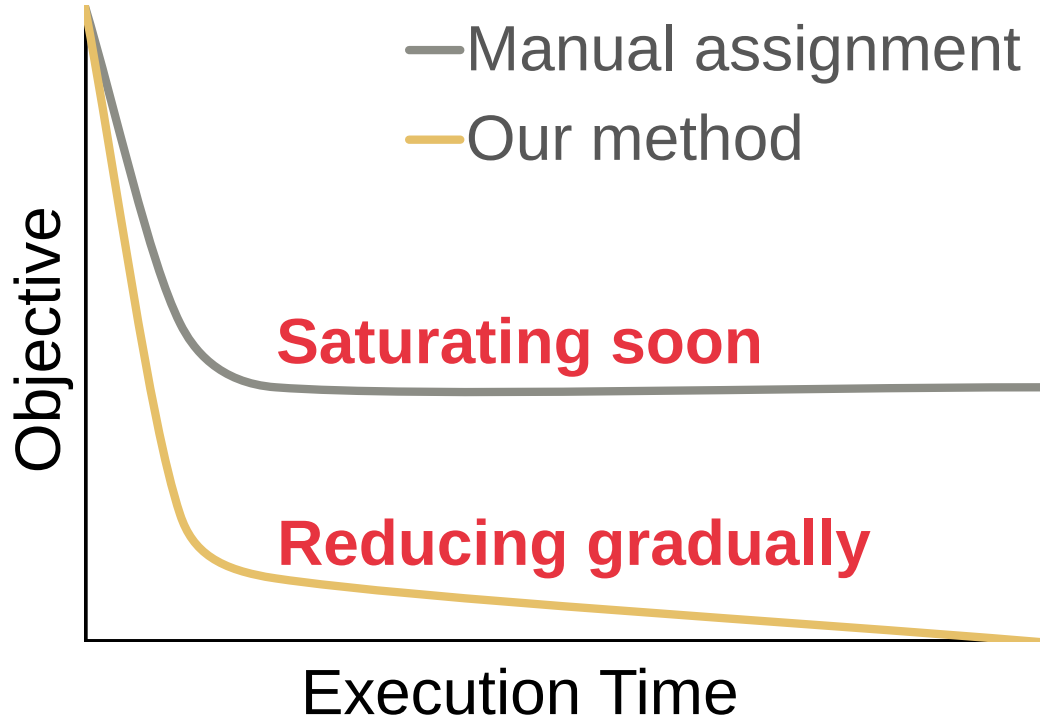
Computational Time

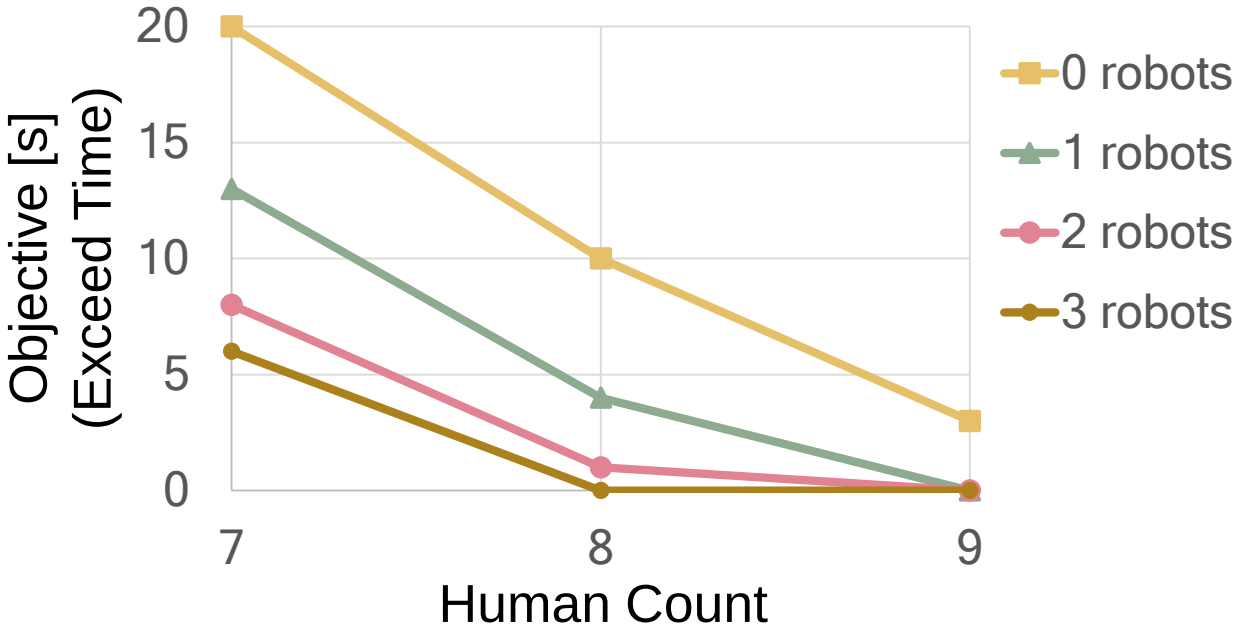


~ 10 s

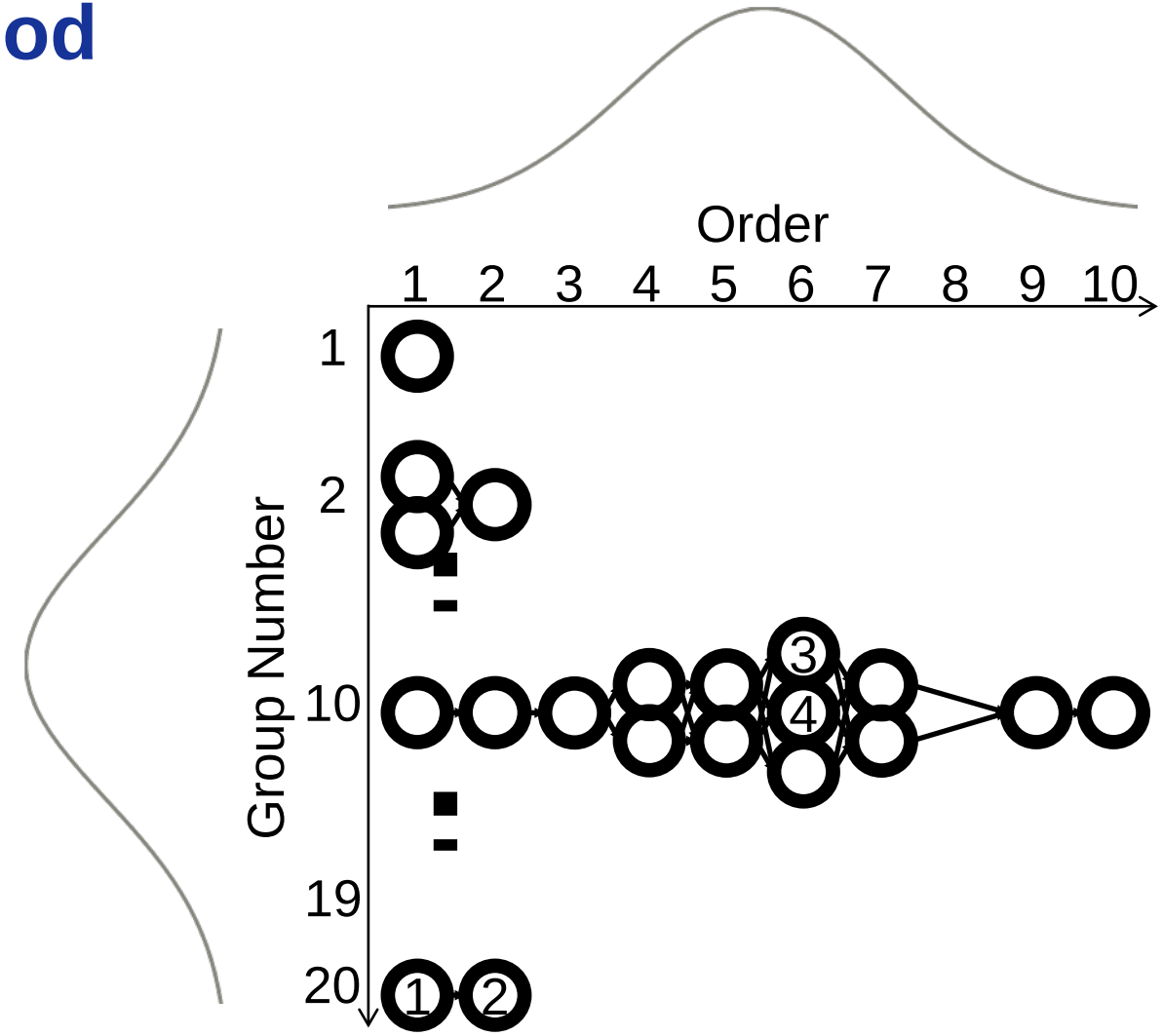
1 ~ 7 hour

Schematic Picture of Execution Time VS Objective

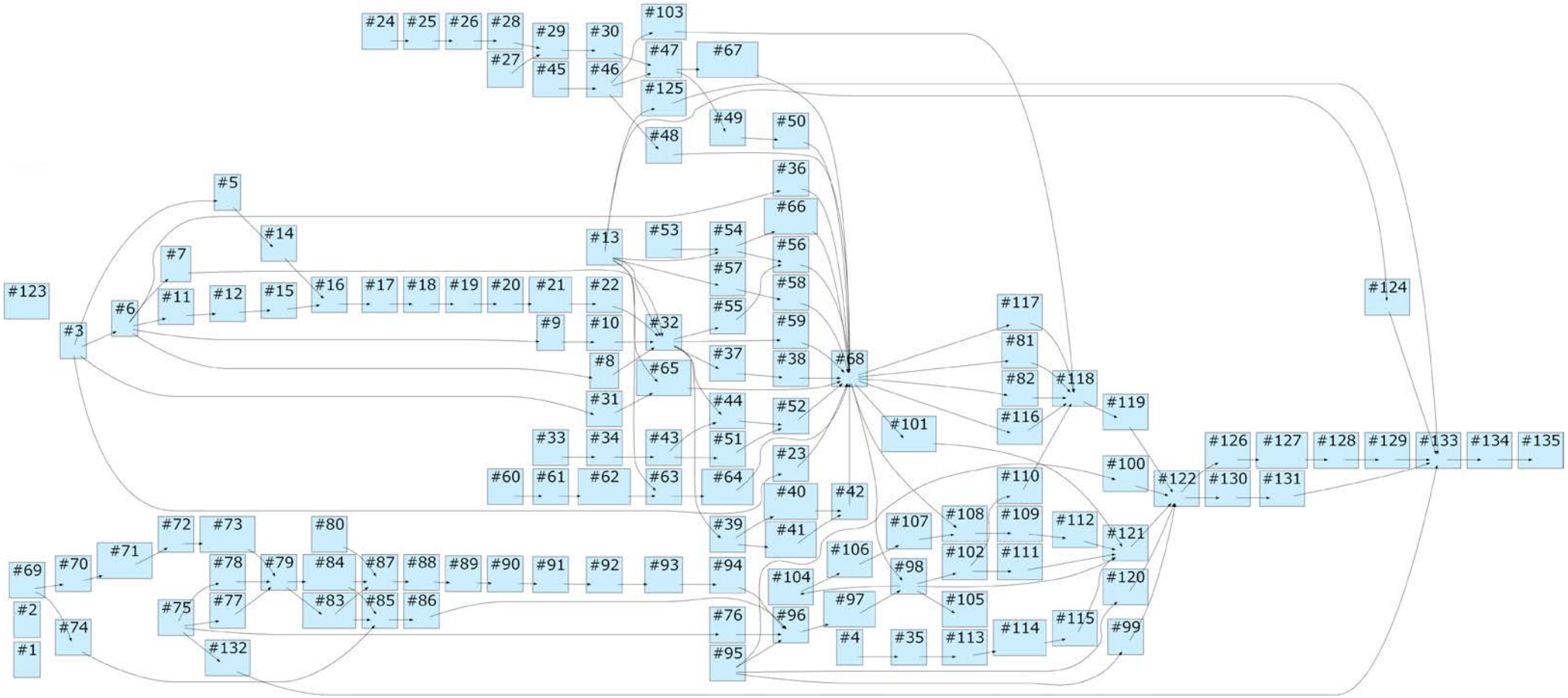



20
15
10
5
0
Objective [s]
(Exceed Time)
7
8
9
Human Count
0 robots
1 robots
2 robots
3 robots

Scalability of Our Method



Precedence Graph of Our Data



Data - Assemble of Laptop

Task ID	Task time (Human) [s]		Task time (Robot) [s]	Precedents
1	1	1		
2	1	6		
3	2	2		
4	3	3		
5	5	6	3	
6	1	3	3	
7	3	6	6	
8	3	6	6	
9	4	6	6	
10	3	5	9	
11	5	6	6	
12	7	6	11	
13	5	5		
14	3	6	5	
15	2	6	12	
16	1	1	14,15	
17	2	4	16	
18	1	1	17	
19	3	6	18	
20	1	1	19	
21	3	15	20	
22	1	1	21	
23	3	6	3	
24	1	5		
25	4	6	24	
26	4	6	25	
27	4	6		
28	1	2	26	
29	3	6	28,27	
30	2	5	29	
31	2	6	3	
32	3	6	22,13, 8,10,7	
33	3	6		
34	8	6	33	
35	10	6	4	
36	6	16	6	
37	3	5	32	
38	4	8	37	
39	2	6	32	
40	6	15	39	
41	5	10	39	
42	1	6	40,41	
43	7	6	34	
44	3	6.4	43,32	
45	2	2		
46	4	4	45	
47	4	5.7	30,46	
48	4	6	46	
49	5	15	47	
50	4	5.9	49	
51	5	6	43	
52	5	5	51,44	
53	4	6		
54	2	5	13,53	
55	4	3	32	
56	1	6	54,55	
57	5	4	13	
58	2	8	57	
59	4	8	32	
60	2	5		
61	2	6	60	
62	3	5	61	
63	6	5	62,13	
64	2	5	63	
65	2	5	13,31	
66	2	5	54	
67	4	10	47	
68	5	5	56,66,42, 38,36,59, 58,52,23, 64,67,65, 50,48	
69	6	6		
70	4	10	69	
71	4	10	70	
72	1	1	71	
73	4	10	72	
74	5	5	69	
75	5	5		
76	3	6	75	
77	5	10	75	
78	2	10	75	
79	5	10	73,77,78	
80	5	5		
81	9	18	68	
82	5	5	68	
83	6	10	79	
84	5	10	79	
85	6	6	83,74,84	
86	3	3	85	
87	3	3	80,84,83	
88	5	5	87	
89	5	5	88	
90	2	2	89	
91	6	6	90	
92	6	6	91	
93	7	7	92	
94	5	5	93	
95	2	2		
96	21	21	86,94, 95,76	
97	6	10	96	
98	3	3	97,68	
99	4	6	95	
100	6	12	95	
101	7	20	68	
102	4	4	98	
103	6	6	46	
104	6	6	98	
105	8	8	98	
106	4	4	104	
107	10	10	106	
108	4	4	107,68	
109	7	7	108	
110	3	3	102	
111	6	6	102	
112	12	12	109	
113	4	10	35	
114	5	10	113	
115	5	5	114	
116	3	15	68	
117	5	5	68	
118	4	4	103,82,81, 110,117,116	
119	3	3	118	
120	10	20	95	
121	7	7	98,111,112, 115,101	
122	2	2	121,119, 120,99,100	
123	2	2		
124	8	12	13	
125	3	5	13	
126	2	2	122	
127	7	15	126	
128	1	1	127	
129	3	3	128	
130	3	3	122	
131	3	3	130	
132	7	5	75	
133	1	1	129,132,125, 131,124	
134	1	1	133	
135	1	1	134	