Optimization Report-Course IT3020-Topic 24

Pham Quang Trung · Nguyen Huy Hoang · Nguyen Viet Quy Anh

February 2023

1 Introduction to important files in this projects

- solution/cp_model.py: The implementation of CP.
- solution/mip_model_minimize_max_salary.py: The implementation of MIP to minimize the maximum payment for a group of worker.
- solution/mip_model_minimize_time.py: The implementation of MIP to minimize the execution time of the project.
- solution/greedy_alg.py: The greedy approach for this problem.
- tests/mixed integer programming/test_for_mip.py: Test generator for MIP model with real number input.
- tests/test_generator.py: Test generator for this problem.

2 The algorithms

2.1 Greedy Algorithm

The greedy criteria: Longest task will be assigned first to the team of worker who can start the soonest.
The algorithm will have a list that store the tasks that have all of their prerequisite tasks sorted. Then the tasks in the list will be sorted by their execution time. The longest task will be assigned to the team who can start working on the task the soonest. After all the task in the list have been sorted, the algorithm continues with next set of tasks having all of their prerequisites sorted. The algorithm keep iterating until there is no task left.

2.2 Constraint Programming Algorithm

2.2.1 Notation

- Q: the set contains the pairs (i, j) such that task i must be completed before task j can start.
- $W(i)$: the set of teams who can execute the task i.
- $d(i)$: the execution time of task i.
- $s(i)$: the time worker i can start working.
- N: the set of tasks
- I: the set of workers

2.2.2 Decision Variables

- $X(i, j) = 1$ if the task j be assigned to worker i
- $St(i)$: the start time of task i
- $Y(i)$: the worker working on task i

2.2.3 Constraint

The problem has the following constraints:

1. If the worker cannot perform the task, he is not assigned to it.
2. The task having prerequisite tasks need to be performed after the all the prerequisite tasks have been done
3. The task must be done by exactly one worker
4. If the task i is done by worker j, then the time it starts must be after the time worker j can start working
5. If two tasks are performed by the same worker, they have to be non-overlapping
6. The obvious constraint: If the worker i performs the task j, the task is performed by the worker i

Mathematically stating:

1. $\forall j \notin W(i), X(j,i) = 0$
2. $\forall (i,j) \in Q, St(j) \geq St(i) + d(i)$
3. $\sum_{k=1}^{N} X(k,i) = 1$
4. $X(i,j) = 1 \Rightarrow St(j) \geq s(i)$
5. $Y(i) = Y(j) \Rightarrow St(i) + d(i) \leq St(j) \vee St(j) + d(j) \leq St(i)$
6. $X(i,j) = 1 \Leftrightarrow Y(j) = i$

2.2.4 Objective Function

There are two objective function to be minimize:

1. The execution time of the project:
   $T = max_{i \in N}(St(i) + d(i))$
2. The maximum amount of money paid to a group of workers:
   $M = max_{k \in I}(\sum_{i \in N} X(k,i) * c(i,k))$

In the program, we will minimize $T + M$. We will have a set of feasible solutions. From that, we will compare all solutions to find the solution with the least execution time and least maximum payment for a group.

2.3 Mixed Integer Programming Model

The above model of Constraint Programming can be transformed to be a Mixed Integer Programming model by the process called linearization. The constraint can be restated as follow:

1. $\forall j \notin W(i), X(j,i) = 0$
2. $\forall (i,j) \in Q, St(j) \geq St(i) + d(i)$
3. $\sum_{k=1}^{N} X(k,i) = 1$
4. $St(j) \geq s(i) - M \times (1 - X(i,j)), \forall (i,j) \in I \times N$
5. - $St(i) + d(i) \leq St(j) + M(1 - x_{ij_1})$
   - $St(j) + d(j) \leq St(i) + M(1 - x_{ij_2})$
   - $x_{ij_1} + x_{ij_2} \geq M * b_{ij}$
   - $x_{ij_1} + x_{ij_2} + M(1 - b_{ij}) \leq 1$
   - $Y(i) + t_{ij_0} \leq Y(j) + M * t_{ij_1}$

- $Y(j) + t_{ij_0} \leq Y(i) + M * t_{ij_2}$
- $t_{ij_1} + t_{ij_2} = t_{ij_0}$
- $b_{ij} = 1 - t_{ij_0}$

with $x_{ij_1}, x_{ij_2}, b_{ij}, t_{ij_0}, t_{ij_1}, t_{ij_2}$ are binary variables, M is a sufficiently large constant

6.
- $Y(i) - k \leq M * (1 - X(k,i))$
- $Y(i) - k \geq M * (1 - X(k,i))$
- $Y(i) \geq k + 1 - M * \gamma_{ij} - M * X(k,i)$
- $Y(i) \leq k - 1 + M * (1 - \gamma_{ij}) + M * X(i,j)$

with γ_{ij} is a binary variable, M is a sufficient large constant

Objective function

For optimizing the execution time

$U = max_{i \in I} s(i) + \sum_{i \in N} d(i)$
$L = max_{i \in N} d(i) + min_{i \in I} s(i)$
$h(i)$ are binary variables, $h(i) = 1$ if $St(i) + d(i) = max_{j \in N}(St(j) + d(j))$
$T \geq St(i) + d(i), \forall i \in N$
$T \leq St(i) + d(i) + (U - L) * (1 - h(i)), \forall i \in N$
$\sum_{i \in N} h(i) = 1$

The objective function to be minimize: T

For optimizing the maximum payment for a group

$U = max_{j \in I}(sum_{i \in N} c_{ij})$
$L = 0$
$Xc(i) = \sum_{j \in N} X(i,j) * c_{ji}, \forall i \in I$
$h(i)$ are binary variables, $h(i) = 1$ if $Xc(i) = max_{j \in N} Xc(j)$
$M \geq Xc(i), \forall i \in I$
$M \leq Xc(i) + (U - L) * (1 - h(i)), \forall i \in I$
$\sum_{i \in I} h(i) = 1$

The objective function to be minimize: M

3 Testing Result

The result table

We collect a total number of 56 tests input with different size to run the experiment. The results are shown here:

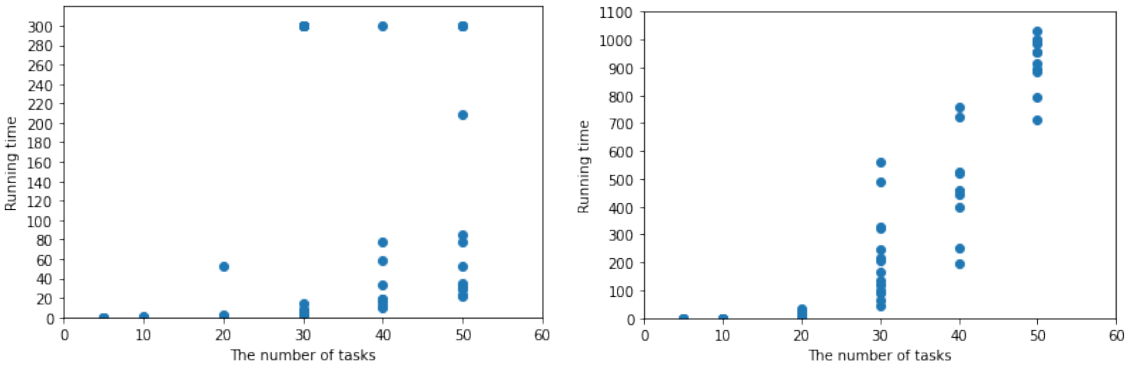
Table 1: Testing results for Mixed Integer Programming method

	number of tasks	number of worker	number of pairs (i,j)	MIP	CP
1	5	2	1	213	640
2	5	3	2	189	439
3	5	3	3	191	465
4	5	4	3	178	420
5	10	5	5	268	608
6	10	5	10	279	828
7	10	7	5	516	736
8	10	7	10	288	607
9	10	7	15	280	702

10	20	5	5	53300	29100
11	20	5	10	1500	24800
12	20	10	5	1700	32100
13	20	10	10	1900	26700
14	20	10	15	1400	9600
15	20	10	20	748	6100
16	20	10	25	1100	10600
17	20	15	5	2600	16800
18	20	15	10	1800	4800
19	20	15	15	954	11600
20	20	15	20	1800	3800
21	20	15	25	1100	10600
22	30	5	5	300000	218000
23	30	5	10	299500	329100
24	30	5	20	300300	122900
25	30	5	35	299700	487500
26	30	10	5	299900	98900
27	30	10	20	8700	561200
28	30	10	35	299700	207500
29	30	15	5	5800	92100
30	30	15	10	4800	324500
31	30	15	20	3400	246000
32	30	15	35	4900	166100
33	30	20	5	14300	134100
34	30	20	20	3100	62500
35	30	20	35	4000	44700
36	40	10	5	300300	445200
37	40	10	25	57900	519000
38	40	10	40	33700	398600
39	40	20	5	19000	252900
40	40	20	25	16100	459100
41	40	20	40	19200	721200
42	40	25	5	78100	198700
43	40	25	25	9300	525100
44	40	25	40	11000	757400
45	50	15	5	300400	912500
46	50	15	25	300600	983300
47	50	15	50	35300	956722
48	50	15	60	208000	995062
49	50	25	5	300500	712343
50	50	25	25	77700	882345
51	50	25	50	32500	1002347
52	50	25	60	22400	992343
53	50	30	5	85300	792342
54	50	30	25	53200	892901
55	50	30	50	23000	956234
56	50	30	60	29700	1028900

Comparison between CP and MIP model

Overall, the results given by 2 models are excellent. The running time of CP model is greater than that of MIP model. One advantage of MIP model is that it can work with real number input. The MIP model, however, can only optimize one objective function.



(a) Running time of MIP model (b) Running time of CP model

Figure 1: Relation between running time and number of tasks of 2 models, each dot represents a test case

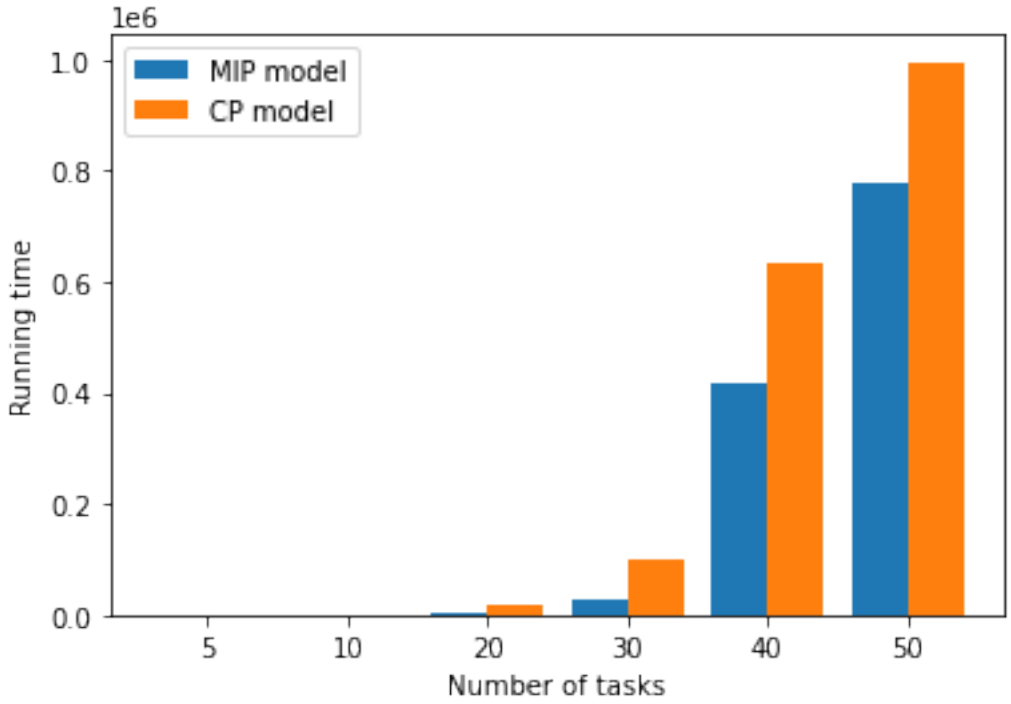


Figure 2: Comparison of MIP and CP avarage running time for different test sizes

4 Contributors

Pham Quang Trung 20214935

MIP model(100%)
Greedy Algorithm(100%)
Report(100%)
Data Analysis(10%)
Slide(30%)

Nguyen Huy Hoang 20214959

CP model(50%)
Data Analysis(90%)

4.1 Nguyen Viet Quy Anh 20214947

CP model(50%)
Slide(70%)