The Approximated Dynamic Programming Approach to the Dynamic Quadratic Assignment Problem

Sirirat Muenvanichakul and Peerayuth Charnsethikul
Department of Industrial Engineering, Kasetsart University
50 Paholyothin Rd., Jatujak, Bangkok 10900 Thailand
Email: sirirat.m@ku.ac.th, fengprc@ku.ac.th,

Abstract

This paper presents an algorithm combining dynamic programming (DP), benders decomposition and metaheuristics for solving a dynamic facility layout problem. The problem is proposed as an extended model of quadratic assignment problem (QAP) called the dynamic quadratic assignment problem (DQAP). Solving for an optimal solution is extremely difficult since DQAP is an NP-hard problem. In approximated DP, the search is incomplete as in the original DP. For each period, a set of best solutions provided by metaheuristics is determined as the initial solution set, and the number of all possible solutions is the product of the initial solution set of all periods. In order to improve the quality of solutions, Benders decomposition is then implemented with given initial solution from the approximated DP. The experimental results and the quality of the obtained solutions will be illustrated.

Keywords: dynamic quadratic assignment, approximated dynamic programming, benders decomposition

1 Introduction

The dynamic quadratic assignment problem (DQAP) is an extended problem of the original quadratic assignment problem (QAP) proposed by Koopmans and Beckman (1975). DQAP is a decision problem formulated to find the optimal location assignment among a set of facilities over discrete time periods. During the periods, many parameters in the problem such as demands and distribution costs are likely to change. The objective is to minimize the sum of flow costs and rearrangement costs over all discrete time periods. The application of this problem is necessary not only for the design of new facilities, but also for the redesign of existing facilities due to the introduction of new products, the installation of new equipment or processes, or the realization of an increase or decrease in throughput volume. DQAP is mathematically formulated as a modified QAP as follows:

Minimize:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{T} C_{ijklt} X_{ijt} X_{klt} + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{T-1} R_{ijlt} X_{ijt} X_{il(t+1)} \qquad (1)$$

Subject to

$$\sum_{i=1}^{n} X_{ijt} = 1 \qquad \forall j,\, t \qquad (2)$$

$$\sum_{j=1}^{n} X_{ijt} = 1 \qquad \forall i,\, t \qquad (3)$$

$$X_{ijt} \in \{0,1\} \qquad \forall i, j, t \qquad (4)$$

where

n represents the number of facilities or locations in each period t.

T represents the number of discrete time periods.

$C_{ijklt} : C_{ijklt} = f_{ikt} * d_{jlt}$ represents the cost of assigning facility i to location j and facility k to location l at period t,

f_{ikt} is the workflow cost from facility i to facility k in period t.

d_{jlt} is the distance from location j to location l at period t.

R_{ijlt} represents the rearranging cost when facility I, located at location j at period t, is moved to location l at period $(t+1)$.

X_{ijt} is 1, if facility i is assigned to location l at period t. Otherwise, X_{ijt} is 0.

2 Approximated Dynamic Programming

In the fundamental paper on the dynamic facility layout problem, Rosenblatt (1986) proposed a dynamic programming to develop an optimal solution methodology and identify bounding procedures. Using dynamic programming terminology, a stage corresponds to a specific layout and a state corresponds to a period. This procedure can be used to reduce the number of candidate static layouts to be examined. The effectiveness of this procedure depends on the relative magnitude of the shifting costs.

The following recursive relationship was established:

$$L^{*}_{tm} = \min_{k} \left\{ L^{*}_{t-1,k} + R_{km} \right\} + C_{tm} \quad , \forall t \qquad (5)$$

where:

L^{*}_{tm} is the minimum total cost for layout m at up to period t, where layout m is being used in period t.

R_{km} is the rearrangement cost from layout k to layout m.

C_{tm} is the material handling cost for layout m at period t.

With Rosenblatt's dynamic programming model, each period in the planning discrete time corresponds to a stage and each particular layout arrangement corresponds to a state. Therefore, there are $n!$ states in each of the T stages. The total number of possible solutions is $(n!)^T$. Restricting the state space of the model, it was determined that any layout arrangement for a given period does not need to be considered if the difference between the total cost of the arrangement, L_{tm}, and the cost of optimal static solution for that period, L^{*}_{tm}, is greater than the difference between the values of the upper bound and the lower bound of the model. Therefore, only the best static solutions for each period need to be considered. For a small problem, the optimal solution can be obtained. For a larger problem, using all $n!$ static layouts will result in an intractable problem.

N_t is the number of the static assigning layouts of period t. The main concept of this methodology is providing the optimal solution for the layouts included in its procedure. In the case that n is very large, exploring all possible solutions as $n!$ at each period needs the capability of software and hardware to solve the problem. If $N_t < n!$, it then cannot guarantee the optimal solution for the problem since all the possible static layouts are not determined. Concerning selecting the best N_t layouts in each period, the larger N_t in the set of the best ranked solutions should lead to better solutions. There are some suggestions on the method of selecting the N_t layouts where $N_t < n!$. One method is to choose them randomly, but the quality of the solutions is usually not good.

Finding the optimal solution for DQAP relies on the solution to the QAP. It is well known that static QAP is an NP-complete problem, so heuristic procedures will be necessary for providing good solutions in an efficient executing time. In this paper, the method of selecting the N_t layouts is using the best layouts provided by the metaheuristics. Muenvanichakul (2000), genetic algorithm, tabu search method and simulated annealing method were applied to solve DQAP. Next, approximated dynamic programming, which is based on the concept of incomplete dynamic programming, is applied as an optimal solution procedure. The procedure eliminates all repeating static layouts in each period. It attempts to reduce exploring solution time and avoid constructing repeating layouts. In approximated dynamic programming, the N static layouts (states) in each period can be different each other. Thus, N_t represents the number of static layouts in period (stage) t. A framework of approximated dynamic programming is presented in Figure 1.

However, an appropriate lower bound proposed to cut off useless solutions is restricted in approximated dynamic programming because N layouts are not all possible solutions, and they are the best solutions found by metaheuristics. It attempts keeping the largest N. Logically, the larger N should lead to better solutions.

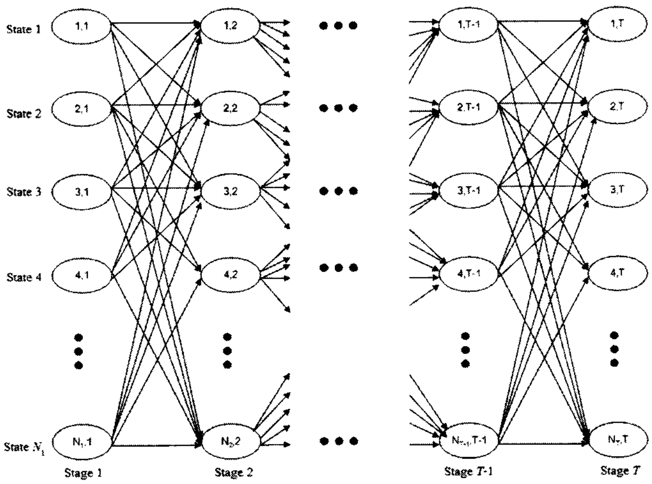

Figure 1: Approximated dynamic programming framework.

3 Benders Decomposition

The 0-1 DQAP model has a quadratic objective function and assignment constraints. The nonlinearity can be removed from the objective function by introducing two new nonnegative variables, Y_{ijklt} and $M_{ijl(t+1)}$. The variables Y_{ijklt} take the value 1 when facility i, j are respectively assigned to location k, l at period t. Similarly, variables $M_{ijl(t+1)}$ represent the facility i assigned to location j at period t, and relocated to location l at period t+1. Thus, two new constraints are added:

$$Y_{ijklt} \geq X_{ijt} + X_{klt} - 1 \qquad ,\forall i, j, t \qquad (6)$$

$$M_{ijl(t+1)} \geq X_{ijt} + X_{il(t+1)} - 1 \,,\forall i, j, t \qquad (7)$$

Because of the quadruple and quintuple subscripts of the new variables, there are a very large number of variables. The sets of variables Y_{ijklt} and $M_{ijl(t+1)}$ may take the value (-1). They will be automatically eliminated due to the integrality constraints.

For solving large scale mixed integer programming problems, the Benders decomposition technique is one of the possible approaches. The algorithm was applied to solve iteratively the linearization of its structure by decomposing a mixed integer problem into two problems as an integer master problem (called MP2) and a linear sub-problem or dual problem.

Then, the corresponding dual problem of linearized DQAP is:

Maximize

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{T}(X_{ijt} + X_{klt} - 1)U_{ijklt} + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{T-1}(X_{ijt} + X_{il(t+1)} - 1)V_{ijl(t+1)}$$

Subject to

$$0 \leq U_{ijklt} \leq C_{ijklt} \qquad , i=1,..,n, j = 1,\ldots, n \;, t = 1,\ldots T$$

$$0 \leq V_{ijl(t+1)} \leq R_{ijlt} \qquad , i=1,..,n, j = 1,\ldots, n \;, t = 1,\ldots T$$

$$X_{ijt} \in \{0,1\} \qquad ,\forall \; i, j, t$$

In this case, it easily deals with the dual variables. The optimal solutions are obvious as follows:

$$U^{*}_{ijklt} = C_{ijklt} \quad \text{If } (X_{ijt} + X_{klt} - 1) = 1,$$
$$= 0 \quad \text{Otherwise.} \qquad (8)$$

$$V^{*}_{ijl(t+1)} = R_{ijlt} \quad \text{If } (X_{ijt} + X_{il(t+1)} - 1) = 1,$$
$$= 0 \quad \text{Otherwise.} \qquad (9)$$

For an obtained optimal U^*_{ijklt} and $V^*_{ijl(t+1)}$, the master problem (MP2) can be solved for a new solution. Iteratively, the process can be continued by adding the cutting plane for a candidate solution in order to improve local bound Z. Thus, the new adding cut is:

$$Z \geq \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{T}(X_{ijt}+X_{klt}-1)U^*_{ijklt}+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{T-1}(X_{ijt}+X_{klt}-1)V^*_{ijl(t+1)}$$

Theoretically, the objective of MP2 is equivalent to the objective of the problem. The MP2 model is as follows:

Minimize: Z

Subject to:

$$Z \geq \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{T}(X_{ijt}+X_{klt}-1)U^*_{ijklt}+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{T-1}(X_{ijt}+X_{il(t+1)})V^*_{ijl(t+1)}$$

$$\sum_{i=1}^{n} X_{ijt} = 1 \qquad \forall j, t$$

$$\sum_{j=1}^{n} X_{ijt} = 1 \qquad \forall i, t$$

$$X_{ijt} \in \{0,1\} \qquad \forall i, j, t$$

In this paper, there are two aspects of benders decomposition. One is the original concept of the Benders decomposition technique. Since DQAP is a hard problem, dealing with the linearized DQAP is still hard. The relaxation model of Benders decomposition is applied. The relaxation idea is to relax the master problem into T - linear assignment problems while the arrangement cost is ignored. A new cut is a combination of all T solutions of linear assignment problems.

4 Computational Experience

4.1 Results of Applying Approximated dynamic programming

The algorithm was coded in Visual Basic 6 and run on the PC Pentium IV 2.4 GHz. 512 MB RAM. Consider a first plant layout with twenty facilities over five planning time periods, the facilities are assumed to be equal size. The set of initial layouts are provided by the metaheuristics genetic algorithm, tabu search and simulated annealing. In genetic algorithm, the process of selecting a solution to be an initial solution for the next generation used a Roulette wheel for setting an opportunity of the population, while the mutation parameter was set to zero. For large problems, the process of generating new solutions in tabu search method will take too much time for evaluations. It is therefore controlled by creating a random number of a specific time period to create the new solutions. Nevertheless, both methods are not able to obtain a better solution than those obtained by the simulated annealing (Luangpaiboon, 1995). Thus, combining three heuristic algorithms including simulated annealing, aims to obtain a better solution and to increase the power of the quality of solutions.

A set of a hundred and six initial solutions, which the best cost is 4,193,928 units was firstly determined to eliminate the repeating layouts in each period. The first experiment for the problem of n = 20, T = 5 was set by controlling the state space. The actual number of states was determined by various trials. The number of states of each trial was randomly selected from all initial solutions. The best solution was then found in the 7th trial, which had a maximum number of states and six good solutions can be also found as 4,180,226, 4,189,747, 4,192,084, 4,192,514, 4,192,589, and 4,192,675 units Thus, it shows that if the best initial solutions were included in an experiment, it can improve the quality of solutions as in Table 1. In addition, the result of the test problem for n =20, T=3, and n=20, T=5, the good solutions (but not optimal) are found in Table 2. The layouts of best solutions of (n =20, T=3) and (n=20, T=5) are shown in Tables 3 and 4, respectively.

4.2 Results of Applying Benders Decomposition

All experiments in this part are investigated as problem size n=20, T=3. An initial solution for testing in this part is initiated by the approximated dynamic programming. After adding a group of cuts (not all possible cuts), there is consequently no better solution as compared to the initial solution. On the other hand, for some cases which started the algorithm with different initial worse solutions, there are a set of better solutions found. More solutions are in progress with the expectation that this scheme will provide a further opportunity to generate more effective static layouts leading to a better solution overall for the DQAP.

Table 1: The experiments of approximated dynamic programming for n=20, T=5

n	T	Trial	Search space	Number of states (excluding repeating layout)	Executing time. (sec)	By approximated dynamic programming
20	5	1	100,000	(10x10x10x10x10)	216	4,354,420
		2	100,000	(10x10x10x10x10)	209	4,419,156
		3	100,000	(10x10x10x10x10)	208	4,321,477
		4	100,000	(10x10x10x10x10)	209	4,370,316
		5	100,000	(10x10x10x10x10)	210	4,384,428
		6	258,720	(8x10x21x11x14)	539	4,340,098
		7	759,375	(15x15x15x15x15)	1,551	4,177,595

Table 2: The experiments of (n=20, T=3) and (n =20 T=5)

		Initial Solution		Time		Best total cost	
N	T	Search space	Number of states (excluding repeating layout)	Executing time. (sec)	AddIn Table time. (sec)	From initial solutions	After applied approximated dynamic programming
20	3	39,168	(32x36x34)	43	228	2,593,775	2,575,963
20	5	759,375	(15x15x15x15x15)	1551	580	4,193,928	4,177,595

Table 3: Layouts of n = 20, T =3

	Layout of n = 20, T = 5	Total cost
Best Initial Layout	10,7,19,8,13,14,11,2,5,6,1,12,3,15,17,20,9,16,4,18 7,20,10,11,5,3,15,4,16,6,18,8,13,1,9,17,2,19,14,12 16,6,9,18,8,5,19,4,13,14,15,20,2,7,17,3,10,12,11,1	2,593,775
Best Layout	3,14,11,8,15,10,4,7,16,17,12,13,1,18,19,5,2,9,20,6 6,4,12,17,2,14,9,10,11,15,19,16,18,3,5,7,1,12,8,20 16,6,9,18,8,5,19,4,13,14,15,20,2,7,17,3,10,12,11,1	2,575,963

Table 4: Layouts of n = 20, T =5

	Layout of n = 20, T = 3	Total cost
Best Initial Layout	2,5,17,18,15,4,12,19,16,9,7,8,6,20,1,13,11,3,14,10 15,19,13,11,7,10,9,4,14,18,2,8,17,3,20,6,16,1,12,5 16,6,13,15,8,5,4,7,9,17,10,18,3,19,11,1,12,2,20,14 14,7,8,9,3,10,16,5,13,4,19,12,1,11,17,6,18,15,20,2 6,14,5,15,19,3,10,1,9,12,20,11,16,17,13,4,7,8,18,2	4,193,928
Best Layout	2,5,17,18,15,4,12,19,16,9,7,8,6,20,1,13,11,3,14,10 15,19,13,11,7,10,9,4,14,18,2,8,17,3,20,6,16,1,12,5 16,6,13,15,8,5,4,7,9,17,10,18,3,19,11,1,12,2,20,14 14,7,15,16,2,10,3,5,13,4,19,12,1,11,17,6,18,8,20,9 6,14,5,15,19,3,10,1,9,12,20,11,16,17,13,4,7,8,18,2	4,177,595

5 Conclusions

As investigated in a few experiments, it was found that better solutions to the dynamic quadratic assignment problems can be detected. The number of states required in the applied approximated dynamic programming is provided by the combining with Benders decomposition. The quality of the solutions depends on the quality of the initial solutions.

The proposed algorithms, however, are still far from being completed. The study should consider the quality of solutions and improve the efficiency of algorithms. Providing the new cut-methods for Benders decomposition may be investigated.

6 References

[1] Koopmans, T.C. and Beckman, M., Assigment Problems and the Location of Economic Activities, Econometrica, Vol.25, No.1, 1957.

[2] Benders, J. F., Partitioning Procedures for Solving Mixed Variables Programming Problems, Numerische Mathematik, Vol.4, pp.238-252, 1962.

[3] Dunker, Thomas, Randons, Gunter, and Westkamper, Engelber, Combining Evolutionary Computation and Dynamic Programming for Solving a Dynamic Facility Layout Problem, European Journal of Operational Research, Vol.165, pp.55-69, 2005.

[4] Lacksonen, T. A., and Enscore, E. E., Quadratic Assignment Algorithms for the Dynamic Layout Problem, International Journal of Production Research,Vol.31, No.3, pp.503-517, 1993.

[5] Luangpaiboon, P., Dynamic Process Layout Planning, Master Thesis, Kasetsart University, 1995.

[6] Muenvanichakul, S., and Charnsethikul, P., A Hybrid Approach of Genetic Algorithm/ Simulated Annealing and Tabu Search Method for Dynamic Process Layout Planning, Proceeding of the International Workshop on the Quadratic Assignment Problem and Extension, Kasetsart University, Bangkok, Thailand., pp. 85-94, 2000.

[7] Rosenblatt, M. J, The dynamics of plant layout, Management Science, Vol.32, No.1, pp.76-82, 1986.

[8] Urban, T. L., A Heuristic for the Dynamic Facility Layout Problem, IIE Transaction, Vol.25, No.4, pp.57-63,1993.

7 Appendix: Data

Table 5: $[F_{ik1}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	36	6	66	12	83	13	58	66	80	60	45	76	80	27	14	11	47	95	47
2	59	0	49	91	65	22	87	53	55	61	72	39	36	99	17	69	8	48	5	9
3	94	36	0	82	48	83	40	89	81	50	68	32	65	50	8	72	85	34	68	30
4	74	84	33	0	69	39	64	68	6	76	46	49	78	3	9	22	73	68	94	7
5	73	60	26	0	0	54	49	98	69	52	20	51	48	44	71	90	81	88	38	22
6	37	59	62	87	98	0	67	79	43	34	83	82	98	82	32	51	40	57	35	87
7	7	28	66	36	57	70	0	40	89	77	22	38	12	76	19	80	84	50	46	16
8	97	66	36	55	98	92	48	0	25	54	75	99	26	25	84	89	25	92	54	4
9	5	41	0	52	96	8	10	37	0	47	5	21	38	45	79	9	43	40	81	95
10	14	85	88	59	45	37	23	46	65	0	51	64	81	72	52	7	9	67	87	10
11	50	96	55	24	46	30	32	83	93	1	0	38	0	8	84	74	50	4	83	72
12	20	38	72	15	14	56	3	26	41	73	7	0	31	59	23	86	64	26	81	42
13	41	68	17	63	14	34	24	69	45	77	44	15	0	22	93	55	32	42	89	82
14	54	23	98	69	94	37	65	51	31	71	9	7	26	0	4	42	77	65	46	5
15	96	73	57	8	62	92	89	85	0	27	87	85	62	86	0	93	54	66	78	60
16	62	22	51	15	11	38	32	30	57	50	59	63	40	91	46	0	50	0	75	53
17	92	91	17	0	22	3	90	34	82	88	11	32	9	73	23	12	0	3	91	95
18	80	43	62	37	29	48	26	84	30	67	75	8	43	62	18	54	24	0	84	27
19	92	47	57	28	85	30	20	89	42	99	97	14	54	20	51	91	62	33	0	9
20	4	47	0	44	95	93	18	25	60	78	17	15	93	49	89	46	5	48	88	0

Table 6: $[F_{ik2}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	71	72	30	71	32	67	54	22	62	72	56	13	32	90	59	5	87	35	66
2	2	0	40	90	6	64	84	40	25	13	71	4	47	17	39	71	70	80	7	7
3	85	6	0	56	58	31	96	11	20	27	65	82	57	2	30	98	49	27	28	43
4	54	54	26	0	87	87	99	12	82	99	85	70	89	9	82	83	36	24	89	8
5	22	6	50	55	0	92	75	30	60	44	97	4	85	72	51	59	58	13	24	54
6	71	10	99	14	5	0	49	82	14	24	36	31	67	20	18	61	62	96	14	11
7	91	3	24	97	36	61	0	99	41	56	37	78	66	89	59	14	59	5	14	56
8	19	87	61	9	9	65	70	0	37	2	22	14	40	61	34	92	3	98	31	47
9	36	87	75	38	89	95	90	6	0	55	53	56	80	9	71	9	93	23	94	9
10	70	68	62	36	34	43	7	8	94	0	34	76	7	94	85	28	47	69	65	16
11	42	10	60	63	30	39	99	34	52	16	0	83	79	48	24	52	81	56	15	85
12	50	48	95	25	29	80	19	28	57	91	70	0	74	99	22	40	16	68	6	18
13	19	83	31	15	57	54	94	23	34	72	36	62	0	70	81	27	36	53	89	53
14	18	74	51	43	32	95	77	97	74	52	63	83	39	0	51	50	70	39	39	10
15	12	4	48	34	51	42	16	20	42	12	1	20	61	91	0	76	27	61	66	24
16	95	10	91	94	74	95	58	17	75	52	85	59	47	33	95	0	38	99	85	13
17	32	81	33	19	20	0	74	91	57	71	64	92	12	99	47	67	0	61	86	50
18	20	55	74	43	69	86	0	86	90	7	97	67	39	55	98	62	53	0	0	51
19	28	64	98	84	94	29	79	41	87	9	34	45	42	65	10	95	91	27	0	92
20	70	16	94	11	11	21	8	4	58	6	26	70	17	87	48	35	8	86	97	0

Table 7: $[F_{ik3}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	12	15	9	0	71	5	37	53	97	44	16	80	26	72	99	61	74	42	26
2	39	0	4	43	94	22	38	87	33	94	95	43	96	11	75	69	81	57	48	62
3	54	32	0	3	77	55	0	82	10	95	11	27	13	43	84	50	10	71	79	2
4	23	25	39	0	79	29	5	30	57	53	89	76	19	52	35	59	23	68	25	2
5	10	91	81	67	0	80	55	27	96	13	33	93	26	34	46	54	73	50	41	17
6	11	44	93	95	24	0	79	64	91	26	93	84	46	60	28	25	57	40	44	73
7	91	20	89	78	31	13	0	94	82	9	94	77	49	0	79	5	5	37	79	18
8	74	45	45	87	30	30	79	0	74	4	34	62	32	66	89	27	29	11	74	76
9	67	53	19	69	37	87	97	84	0	6	11	39	23	82	69	25	13	41	69	33
10	3	14	52	62	58	38	49	95	52	0	26	63	16	44	67	50	98	91	57	57
11	75	87	89	37	3	0	20	77	49	21	0	93	18	38	17	38	65	31	61	98
12	80	79	12	43	65	69	23	22	33	48	95	0	59	35	71	28	9	37	43	19
13	21	20	91	93	43	43	81	26	43	81	79	30	0	17	99	72	93	93	7	44
14	41	64	54	98	88	90	71	36	27	79	85	61	49	0	31	31	77	73	85	53
15	62	36	50	0	45	5	22	60	64	74	30	99	64	64	0	78	86	49	63	67
16	15	93	47	90	65	11	19	86	87	77	69	91	93	36	97	0	68	91	58	6
17	98	93	67	19	32	7	47	5	30	74	8	7	57	98	52	51	0	15	67	95
18	77	97	11	44	5	17	52	64	88	82	92	51	95	89	12	31	26	0	44	80
19	16	76	22	71	19	88	1	24	61	35	38	14	12	58	98	74	1	83	0	67
20	89	71	77	76	64	35	39	15	28	96	11	99	25	48	23	75	21	9	9	0

Table 8: $[F_{ik4}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	14	20	36	83	21	21	1	67	12	71	74	34	3	13	38	77	12	7	67
2	32	0	96	38	4	78	31	26	6	7	61	55	87	11	23	75	93	65	87	98
3	71	97	0	24	89	39	31	42	36	3	41	49	87	97	56	10	18	78	89	51
4	28	16	26	0	85	8	78	17	36	25	32	84	35	54	25	15	89	3	7	75
5	56	13	97	37	0	76	73	11	57	84	62	16	93	75	74	22	26	20	53	33
6	98	74	57	8	50	0	86	48	76	83	68	78	0	31	71	55	32	21	15	82
7	24	14	46	75	4	76	0	22	47	30	85	92	12	37	73	61	69	42	83	82
8	53	47	80	16	89	46	88	0	15	46	15	20	1	2	1	17	14	2	96	50
9	68	13	35	6	73	96	8	16	0	90	8	0	38	49	19	73	16	91	74	22
10	63	22	18	33	47	35	67	79	6	0	67	8	68	38	8	89	19	13	53	68
11	72	32	62	34	94	75	13	69	91	17	0	94	43	52	64	52	34	38	70	85
12	83	59	26	26	9	77	44	59	21	26	13	0	85	70	83	38	50	92	38	40
13	26	69	38	91	59	82	64	9	11	58	30	22	0	26	67	5	14	69	55	7
14	85	98	95	88	60	23	64	18	53	67	74	2	11	0	17	65	66	81	9	21
15	39	44	29	81	62	73	14	18	4	23	32	91	27	83	0	50	89	2	26	23
16	54	17	54	61	69	10	77	17	29	75	83	74	73	27	84	0	3	64	39	92
17	97	91	70	35	56	31	32	10	32	33	12	56	9	40	19	43	0	2	5	26
18	33	45	45	54	21	11	27	4	61	32	30	17	99	70	96	40	48	0	25	2
19	36	20	91	68	95	96	53	24	96	58	99	13	0	39	56	35	12	90	0	72
20	25	39	15	83	24	68	39	4	84	92	0	3	69	76	6	3	2	30	13	0

Table 9: $[F_{ik5}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	8	41	27	82	41	31	69	81	51	13	62	92	11	47	9	55	70	76	55
2	41	0	6	47	50	96	49	27	95	27	1	52	27	84	21	82	84	18	92	15
3	33	84	0	49	39	84	8	12	51	56	72	94	80	36	58	53	97	62	90	13
4	53	69	67	0	8	53	42	40	30	35	6	59	38	32	21	36	77	73	7	55
5	23	83	55	50	0	82	91	24	34	39	63	70	54	60	26	57	92	64	86	48
6	39	52	63	4	87	0	85	82	82	65	11	95	23	0	99	77	31	72	68	30
7	56	55	12	82	62	83	0	94	40	53	77	59	45	58	0	17	62	52	81	16
8	91	72	80	37	97	36	79	0	8	46	13	93	91	94	46	19	32	61	48	95
9	25	97	62	30	86	14	72	46	0	86	19	90	7	6	45	44	54	76	14	46
10	94	38	59	60	44	46	0	69	42	0	76	4	84	31	0	3	38	2	23	16
11	37	5	97	45	97	52	50	61	55	33	0	74	56	87	21	22	98	20	63	36
12	73	91	27	24	48	96	8	19	87	3	15	0	70	79	69	96	91	16	33	9
13	90	31	91	37	72	8	59	47	2	43	86	21	0	71	56	48	24	23	40	92
14	33	37	79	89	70	98	89	91	68	14	31	84	74	0	7	59	71	72	34	99
15	69	41	3	86	82	43	92	28	11	27	95	38	2	33	0	81	43	90	59	13
16	68	85	61	70	77	84	60	5	95	62	65	78	31	33	22	0	77	15	92	93
17	53	89	37	91	33	23	10	19	72	66	90	83	39	80	10	3	0	34	32	56
18	79	99	4	57	29	86	49	40	54	89	91	18	31	80	1	40	53	0	18	49
19	65	63	4	39	3	10	86	35	1	67	38	70	93	98	29	51	34	18	0	44
20	91	90	65	72	56	44	4	62	93	6	63	81	17	50	73	10	47	98	85	0

Table 10: $[D_{jl1}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	10	35	32	4	54	15	44	49	37	79	24	21	92	95	97	26	37	5	24
2	75	0	3	21	89	1	39	10	84	70	98	76	93	16	54	24	63	70	88	16
3	53	37	0	5	92	56	62	47	47	16	92	72	55	35	64	99	14	17	62	75
4	80	23	82	0	56	53	31	34	67	19	6	90	40	81	69	2	14	58	44	87
5	67	79	37	76	0	82	47	72	78	49	87	44	72	49	42	11	47	15	18	15
6	14	29	64	47	23	0	3	38	48	71	92	52	74	25	5	15	62	77	27	77
7	81	95	68	69	32	87	0	16	32	47	66	34	10	37	29	80	81	86	12	49
8	45	10	63	75	68	37	33	0	34	78	33	79	32	26	47	79	75	94	29	29
9	84	83	69	17	10	25	12	86	0	54	44	18	64	30	40	36	21	28	66	60
10	37	96	12	0	54	7	40	22	73	0	97	9	40	20	32	49	52	25	85	95
11	85	10	42	62	11	40	81	52	3	0	0	33	53	45	94	0	78	48	35	4
12	94	83	46	43	73	89	26	84	36	84	4	0	77	26	67	71	94	46	21	65
13	35	54	62	52	52	76	63	59	20	89	11	93	0	43	72	4	77	58	74	4
14	84	66	17	93	88	64	48	61	6	95	16	72	22	0	51	23	1	15	4	93
15	89	35	47	48	0	65	24	88	44	87	54	66	6	74	0	18	92	99	15	76
16	6	4	61	80	93	89	81	72	38	36	97	68	77	63	86	0	63	54	99	90
17	21	38	90	14	73	32	22	83	26	11	59	93	54	44	22	14	0	16	2	20
18	78	58	72	35	0	42	43	69	20	41	53	76	6	39	38	75	66	0	49	79
19	13	49	45	44	8	99	62	53	82	16	32	13	23	35	24	94	83	43	0	39
20	47	30	21	30	3	11	47	50	30	38	60	13	75	3	25	60	0	10	54	0

Table 11: $[D_{jl2}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	51	11	55	54	69	14	17	63	20	45	89	74	93	62	21	82	50	62	87
2	31	0	93	59	46	95	53	22	32	59	36	77	89	58	72	98	1	10	72	17
3	41	2	0	14	88	75	59	36	8	19	68	99	73	50	40	22	33	24	18	88
4	22	84	20	0	47	75	98	52	59	77	85	58	62	16	79	88	33	86	17	84
5	47	66	33	94	0	70	65	97	39	61	58	48	3	47	75	75	85	40	76	43
6	4	92	60	74	54	0	21	76	39	9	86	42	59	64	33	34	57	71	30	34
7	53	46	43	63	17	31	0	47	11	16	46	6	32	8	3	40	15	68	8	24
8	5	15	24	44	14	39	76	0	78	20	93	81	47	82	70	27	47	6	4	64
9	33	99	28	95	32	74	98	23	0	26	17	87	68	69	22	80	36	45	76	44
10	46	47	65	75	60	82	18	58	71	0	67	45	38	78	29	49	24	67	24	70
11	98	22	85	73	55	55	85	36	12	84	0	54	36	10	32	49	6	24	90	72
12	87	36	60	60	69	50	18	57	74	70	55	0	88	12	50	48	88	62	40	16
13	50	76	84	79	69	12	24	57	58	83	18	82	0	88	4	76	95	88	69	16
14	84	61	83	74	71	38	6	70	0	49	55	2	5	0	76	2	39	3	78	30
15	69	66	21	7	21	25	36	18	95	76	40	75	15	91	0	66	59	60	56	57
16	78	13	15	93	99	8	90	47	31	87	48	9	41	6	1	0	99	4	67	72
17	65	25	71	87	2	68	23	20	58	69	3	76	68	39	21	40	0	11	58	96
18	16	71	13	32	84	54	80	11	12	49	81	55	24	46	27	29	55	0	32	31
19	25	12	88	77	15	56	86	91	7	58	66	47	71	87	94	10	93	79	0	2
20	71	99	14	97	97	14	88	77	41	89	39	68	92	99	72	94	66	33	58	0

Table 12: $[D_{jl3}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	64	47	83	59	59	12	20	58	57	71	73	99	69	96	54	51	14	72	58
2	90	0	34	97	28	42	8	87	10	63	93	71	21	5	59	48	13	47	95	47
3	53	71	0	20	44	3	39	64	3	0	58	50	70	7	81	75	48	6	83	50
4	67	68	63	0	98	70	7	27	62	23	92	45	8	99	63	72	34	66	25	63
5	86	7	30	48	0	67	35	52	5	23	14	89	54	31	92	88	98	11	3	68
6	4	1	73	28	24	0	51	23	62	24	47	90	86	61	29	35	13	77	24	96
7	81	82	68	79	39	66	0	10	68	20	0	25	74	25	10	0	66	65	54	28
8	85	42	53	86	8	17	11	0	2	25	90	45	2	99	53	50	1	12	68	96
9	98	60	86	30	40	84	17	84	0	86	10	58	65	24	64	59	67	92	13	23
10	90	56	80	62	42	10	2	82	26	0	48	68	1	12	71	58	47	51	26	89
11	80	22	36	88	60	74	4	67	85	90	0	2	70	78	72	60	88	22	7	92
12	33	1	14	23	85	7	70	93	65	92	64	0	30	40	44	60	59	77	8	72
13	39	99	33	82	63	30	1	83	85	81	24	51	0	92	67	26	17	87	37	52
14	14	99	79	93	79	97	25	37	80	5	75	6	62	0	79	77	58	88	31	60
15	6	50	72	90	96	88	9	61	2	27	79	94	11	92	0	75	46	88	27	95
16	0	29	2	82	74	82	20	61	86	65	0	21	60	41	29	0	48	75	40	96
17	22	12	60	1	84	86	19	43	4	82	84	3	28	63	76	51	0	93	33	32
18	48	82	78	35	42	14	49	32	23	69	83	38	53	35	48	22	76	0	98	59
19	96	78	58	10	82	89	91	45	11	2	88	82	66	97	3	62	7	4	0	6
20	80	21	94	80	9	15	98	68	59	59	72	77	83	34	35	70	88	98	76	0

Table 13: $[D_{jl4}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	72	12	39	86	51	64	30	85	20	69	74	86	27	33	19	18	48	80	13
2	33	0	65	9	12	85	36	93	66	27	15	63	23	94	46	34	45	4	64	20
3	41	65	0	66	85	84	55	70	71	5	68	14	54	95	25	35	3	58	48	38
4	77	9	5	0	0	13	85	88	60	92	36	89	55	74	46	26	9	46	24	6
5	45	27	11	53	0	81	81	7	37	80	62	33	19	5	56	99	56	25	86	15
6	2	42	77	81	51	0	70	97	46	61	83	57	0	40	88	30	4	98	26	92
7	28	81	32	22	88	6	0	29	29	76	74	6	55	3	99	90	85	48	15	66
8	48	91	29	28	57	22	20	0	80	42	39	91	98	32	96	43	20	72	27	23
9	3	29	36	10	35	88	70	70	0	20	49	3	85	88	83	24	87	44	98	85
10	70	44	70	9	69	26	4	34	70	0	89	38	49	21	30	57	6	53	9	3
11	81	24	56	60	75	75	2	42	18	58	0	73	69	0	58	66	16	2	27	42
12	38	80	52	84	17	9	25	82	76	45	38	0	82	41	41	7	27	17	74	52
13	20	31	55	38	79	37	64	58	93	38	6	76	0	12	37	84	58	36	97	67
14	78	20	14	41	38	57	91	73	37	47	49	29	72	0	83	46	49	25	36	66
15	46	93	34	38	37	85	13	35	1	26	2	91	19	59	0	92	21	78	73	47
16	52	6	60	63	28	37	40	91	64	1	58	30	8	79	28	0	27	69	7	97
17	46	98	40	90	78	43	37	77	7	71	45	31	74	19	20	87	0	23	74	83
18	70	20	16	40	26	25	14	19	66	68	14	34	95	1	56	65	11	0	72	49
19	16	41	4	31	15	2	23	95	55	24	42	24	8	6	72	92	3	3	0	0
20	94	31	65	17	7	34	11	30	29	45	86	64	2	48	12	47	96	90	62	0

Table 14: $[D_{jl5}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	41	95	73	20	7	48	92	98	86	16	91	86	92	59	19	66	0	18	79
2	73	0	32	54	11	7	59	44	85	75	55	44	15	20	30	51	43	62	59	97
3	44	51	0	7	70	84	25	43	39	25	11	21	1	69	90	29	94	82	5	67
4	42	2	24	0	9	29	17	98	42	98	37	6	23	60	62	87	74	10	90	97
5	17	21	26	55	0	0	16	80	75	95	0	32	71	73	27	60	50	34	65	8
6	61	44	33	59	7	0	15	39	33	64	47	22	29	89	28	1	62	52	55	96
7	96	31	97	89	71	88	0	19	74	82	45	29	72	5	1	15	72	76	52	48
8	27	72	52	29	31	40	76	0	9	30	56	17	30	30	22	64	30	37	61	53
9	90	51	79	75	44	17	55	67	0	14	21	28	87	75	44	62	47	62	53	75
10	10	58	10	29	92	83	46	4	63	0	71	56	82	84	12	11	60	12	32	84
11	46	42	10	14	95	30	18	77	16	86	0	75	16	14	90	36	48	10	30	72
12	32	84	19	80	90	15	35	49	30	66	84	0	98	79	39	48	80	40	90	67
13	94	73	39	97	64	67	57	56	72	78	12	50	0	91	65	38	87	31	51	72
14	66	88	22	19	81	64	11	62	40	93	90	69	51	0	73	67	77	8	30	82
15	57	41	25	62	57	92	44	31	66	73	92	68	56	37	0	65	5	90	37	76
16	72	47	44	56	21	75	84	51	76	1	76	19	15	65	55	0	38	47	23	89
17	63	13	4	71	92	52	15	4	70	3	28	67	18	41	58	78	0	62	52	2
18	96	24	17	26	65	16	63	95	39	65	90	6	14	66	80	17	34	0	66	34
19	51	4	8	76	94	4	88	72	33	10	29	82	22	93	89	39	32	58	0	7
20	76	1	9	95	26	46	16	92	75	44	73	58	80	55	78	70	10	65	37	0

Table 15: $[R_{ijlt}]$

Fr\To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	38	74	89	71	39	66	42	78	87	94	5	3	55	43	65	45	16	47	73
2	42	0	62	20	50	41	62	44	35	65	0	80	70	69	29	15	13	12	75	79
3	97	16	0	32	57	42	5	72	40	95	15	0	95	75	13	5	95	96	77	56
4	66	78	6	0	20	3	48	48	93	17	7	98	9	98	59	67	38	72	87	10
5	99	0	71	57	0	8	61	70	23	82	13	35	11	12	55	73	1	97	9	91
6	74	49	34	23	90	0	46	51	2	28	98	0	6	22	46	63	36	25	18	25
7	80	8	34	1	46	63	0	21	31	87	35	15	5	69	50	38	26	68	47	86
8	0	57	97	90	91	0	53	0	0	96	1	95	53	72	58	85	13	60	1	45
9	86	71	0	16	9	13	81	38	0	43	54	39	65	26	60	7	74	76	73	9
10	47	81	37	49	98	67	35	84	65	0	39	63	0	48	12	28	82	0	89	55
11	4	12	99	78	60	46	45	81	43	63	0	27	80	43	5	61	64	66	64	40
12	39	78	84	20	49	40	52	12	77	7	54	0	99	29	12	84	96	32	59	0
13	64	75	44	0	1	93	43	72	65	7	83	83	0	44	22	11	67	50	15	5
14	95	84	46	63	43	51	32	43	37	52	24	98	71	0	86	62	44	85	6	93
15	30	49	98	88	58	54	58	73	19	90	4	56	76	42	0	61	26	69	60	21
16	81	99	34	78	19	97	8	35	20	30	50	88	96	36	26	0	74	70	25	45
17	95	80	1	1	0	35	57	60	19	96	30	82	83	15	12	11	0	0	14	18
18	11	81	14	20	81	28	9	66	99	15	17	40	23	47	0	10	25	0	91	66
19	26	95	70	45	97	7	86	87	10	84	5	4	3	91	47	12	44	93	0	44
20	70	26	12	1	66	2	84	17	27	9	24	64	24	46	23	7	34	82	83	0