Online supplement to: "A column generation algorithm with dynamic constraint aggregation for minimum sum-of-squares clustering"

Antonio M. Sudoso
Department of Computer, Control and Management Engineering "A. Ruberti", Sapienza University of Rome, Via Ariosto 25, Rome, 00185, Italy, antoniomaria.sudoso@uniroma1.it

Daniel Aloise
Department of Computer and Software Engineering, Polytechnique Montréal, 2500 Chem. de Polytechnique, Montréal, H3T 1J4, Canada, daniel.aloise@polymtl.ca

---

Appendix A Pseudo-code of the algorithm for updating the partition Q

Algorithm 1 presents the pseudo-code for updating the aggregating partition Q from an incompatible selected column $t^\star$. Note that the inclusion of any incompatible column $t^\star \in T$ results in an increase of the size of the agRMP problem by the number of incompatibilities it has with respect to Q. After the update, Q includes the index sets $\mathcal{I}_1, \dots, \mathcal{I}_m, \mathcal{I}_{m+1}, \dots, \mathcal{I}_{u(t^\star, Q)}$. Moreover, we estimate lower and upper bounds for the disaggregated dual variables and construct a new agRMP.

Algorithm 1: Aggregating partition update

Input: Partition Q with index sets $\mathcal{I}_1, \dots, \mathcal{I}_m$, incompatible column $t^\star$

$h \leftarrow m$;

for $j = 1, \dots, m$ **do**

  if $(t^\star, \mathcal{I}_j) \in \Omega$ **then**

    $h \leftarrow h + 1$; /* Increase the number of components */

    $\mathcal{I}_h \leftarrow \mathcal{I}_j \setminus (t^\star \cap \mathcal{I}_j)$; /* Add a new component */

    $\mathcal{I}_j \leftarrow t^\star \cap \mathcal{I}_j$; /* Update the old component */

  end

end

Output: Updated partition Q with index sets $\mathcal{I}_1, \dots, \mathcal{I}_m, \mathcal{I}_{m+1}, \dots, \mathcal{I}_{u(t^\star, Q)}$

Appendix B Proof of Proposition 1

To prove that the constraint matrix A is totally unimodular, we need to show that every square submatrix of A has a determinant that is either 0, +1, or -1. Note that a row of A can have either two +1 if it corresponds to a CL constraint or one +1 and one -1 if it corresponds to an ML constraint. Let B be a $k \times k$ submatrix of A. We show the assertion by induction on the size k that $\det(B) \in \{0, +1, -1\}$. If $k = 1$, then B has a single entry that, by assumption, is 0, +1 or -1, and therefore $\det(B) \in \{0, +1, -1\}$. Now take $k \geq 2$ and assume by induction that the assertion is true for all $(k-1) \times (k-1)$ submatrices of A. We consider four cases.

1. B has at least one row with all zero coefficients. In this case $\det(B) = 0$.
2. B has at least one row with one non-zero coefficient, assume it is b_{ij}. Let B' be the $(k-1) \times (k-1)$ submatrix obtained by removing the i-th row and the j-th column from B. Developing the determinant with respect to the i-th row of B yields

$$\det(B) = b_{ij}(-1)^{i+j}\det(B') = \begin{cases} (-1)^{i+j}\det(B') & \text{if } b_{ij} = +1, \\ (-1)^{i+j+1}\det(B') & \text{if } b_{ij} = -1. \end{cases}$$

   By induction, $\det(B') \in \{0, +1, -1\}$, and therefore $\det(B) \in \{0, +1, -1\}$ holds.
3. Every row of B has exactly two non-zero coefficients, which are +1 and -1. In this case, the sum of all the columns of B yields a vector with all 0 entries. This implies that the columns of B are linearly dependent, and therefore $\det(B') = 0$.
4. Every row of B has exactly two non-zero coefficients of value 1. In this case, each row of B corresponds to a CL constraint. Let V_1 and V_2 be the partition of the nodes of G such that $V = V_1 \cup V_2$ and every edge $(i, j) \in E$ (i.e., cannot-link constraint) has one end in V_1 and the other in V_2. Due to the bipartition of G, we have a partition of the columns of B as $B = (B_1|B_2)$ where the columns in B_1 resp. B_2 correspond to nodes in V_1 resp. V_2. Adding up all columns of B_1 resp. B_2 yields a vector with all 1-entries in both cases. This implies that the columns of B are linearly dependent, and therefore $\det(B') = 0$.

Consequently, $\det(B) \in \{0, +1, -1\}$ follows for each square submatrix B of A and, hence, the constraint matrix A is totally unimodular □

Appendix C Numerical results on DCA evaluation

The columns in the tables refer to the aggregation level (ℓ), to the number of sought clusters (k), the number of CG iterations (iter), the initial number of aggregated covering constraints (m_{start}), the final number of aggregated constraints in the final RMP after convergence (m_{end}), the average number of aggregated covering constraints per iteration during the execution of the CG+DCA algorithm (m_{avg}), the number of times the partition Q was updated (Q updates), the CPU time spent in the resolution of the RMP (in seconds), and the total time spent by the CG+DCA algorithm (also in seconds).

k	ℓ	iter	m_{start}	m_{end}	m_{avg}	Q updates	time RMP (s)	time (s)
2	150	10400	150	150	150.00	0	332.62	740.41
2	75	2386	77	142	97.53	32	7.80	95.56
2	37	2030	39	142	97.67	53	4.19	74.15
2	2	2527	2	141	103.72	61	7.85	85.28
3	150	3779	150	150	150.00	0	23.20	126.96
3	75	2137	77	143	99.60	42	3.68	61.11
3	37	2037	41	138	99.15	62	3.24	58.69
3	3	2200	3	137	100.13	71	3.75	61.26
4	150	3273	150	150	150.00	0	14.27	86.00
4	75	1230	79	143	101.01	42	1.36	19.57
4	37	1837	42	137	102.18	54	2.45	28.06
4	4	2296	4	137	102.51	77	3.48	31.87
5	150	2272	150	150	150.00	0	7.02	38.15
5	75	1120	79	137	99.84	37	1.11	15.37
5	37	1548	42	135	99.72	54	1.75	19.62
5	5	1188	5	126	98.83	75	1.02	14.64
6	150	1642	150	150	150.00	0	3.61	23.00
6	75	795	79	136	100.30	42	0.61	9.04
6	37	734	43	131	94.35	60	0.44	7.83
6	6	973	6	137	99.25	84	0.72	10.03

Table 1 **Results for `ch150` and $k \in \{2, 3, 4, 5, 6\}$.**

k	ℓ	iter	m_{start}	m_{end}	m_{avg}	Q updates	time RMP (s)	time (s)
2	202	7874	202	202	202.00	0	170.65	1062.22
2	101	10239	102	185	107.84	43	257.52	938.31
2	50	3434	52	185	108.94	66	16.18	309.65
2	2	4420	2	168	99.89	95	20.18	275.26
3	202	6974	202	202	202.00	0	167.25	698.38
3	101	6376	104	194	122.11	50	52.52	524.03
3	50	4435	53	189	129.31	68	28.48	348.91
3	3	4228	3	172	108.55	80	17.22	312.87
4	202	8477	202	202	202.00	0	256.64	866.20
4	101	4510	102	188	113.78	44	16.61	302.78
4	50	3490	54	193	124.12	68	15.72	240.16
4	4	3675	4	182	120.59	76	14.14	214.56
5	202	10390	202	202	202.00	0	291.07	901.91
5	101	3389	106	187	119.14	48	9.42	200.48
5	50	2488	56	175	115.88	61	5.32	144.01
5	5	2836	5	191	123.43	81	4.17	113.77
6	202	7069	202	202	202.00	0	108.18	428.87
6	101	2863	106	176	121.14	40	5.44	120.27
6	50	2019	58	174	119.55	53	3.56	83.56
6	6	1824	6	157	109.81	63	2.49	70.91

Table 2 **Results for `gr202` and** $k \in \{2,3,4,5,6\}$

k	ℓ	iter	m_{start}	m_{end}	m_{avg}	Q updates	time RMP (s)	time (s)
2	299	16204	299	299	298.98	0	3677.03	7200.00
2	149	7617	149	279	156.59	81	172.62	2108.85
2	74	7811	75	274	172.32	87	92.06	1273.93
2	2	3878	2	250	154.37	116	17.45	565.93
3	299	20100	299	299	299.00	0	1677.23	4596.03
3	149	5682	149	285	173.35	69	44.81	832.01
3	74	7008	75	288	163.59	102	66.46	903.76
3	3	5501	3	267	177.38	109	47.15	689.43
4	299	16745	299	299	299.00	0	821.31	2688.85
4	149	7024	149	291	182.85	69	77.32	904.71
4	74	4921	78	282	163.20	103	31.50	556.38
4	4	4183	4	272	175.62	125	29.00	454.41
5	299	19277	299	299	299.00	0	1117.98	2891.24
5	149	4382	151	288	183.80	76	22.71	379.86
5	74	3476	82	280	161.84	98	12.54	287.29
5	5	3336	5	252	139.84	125	9.54	266.00
6	299	12113	299	299	299.00	0	405.30	1270.82
6	149	2650	153	279	175.45	79	8.01	198.43
6	74	2002	86	270	165.83	111	3.45	119.60
6	6	1751	6	260	143.60	154	2.31	105.36

Table 3 **Results for `pr299` and** $k \in \{2,3,4,5,6\}$

k	ℓ	iter	m_{start}	m_{end}	m_{avg}	Q updates	time RMP (s)	time (s)
2	417	7910	417	417	416.95	0	2931.76	7200.00
2	208	12671	208	299	208.73	82	657.14	6703.35
2	104	2134	105	186	109.37	84	3.68	597.66
2	2	171	2	70	38.77	48	0.04	57.97
3	417	14086	417	417	416.97	0	2506.00	7200.00
3	208	5962	208	317	211.10	105	76.03	1613.87
3	104	2432	104	192	109.21	70	4.26	558.56
3	3	1403	3	134	100.36	99	1.43	306.14
4	417	22756	417	417	417.00	0	2170.16	6339.07
4	208	5132	208	324	210.82	112	49.83	1199.89
4	104	1502	105	197	112.75	78	1.60	241.64
4	4	76	4	41	28.62	24	0.01	13.52
5	417	10798	417	417	417.00	0	232.79	1207.81
5	208	2457	208	317	215.59	96	6.22	216.44
5	104	1070	104	233	131.98	113	0.77	83.92
5	5	174	5	83	49.12	58	0.03	12.92
6	417	10119	417	417	417.00	0	144.55	911.73
6	208	2141	208	334	218.80	110	4.30	157.55
6	104	844	104	212	122.88	95	0.48	54.48
6	6	129	6	73	43.64	55	0.02	9.04

Table 4 **Results for `fl417` and** $k \in \{2,3,4,5,6\}$

Appendix D Bar charts on the DCA dual variable disaggregation strategies

The bar charts of Figures 1-2 present the number of CG iterations, the total CG+DCA CPU time (in seconds), the average number of constraints in the aggregated master problem (m_{avg}), and the final number of constraints when the method completes (m_{end}) for data instances `gr202` and `pr299` for $k \in \{2,3,4,5,6\}$. Similar conclusions are found for instances `ch150` and `fl417`.

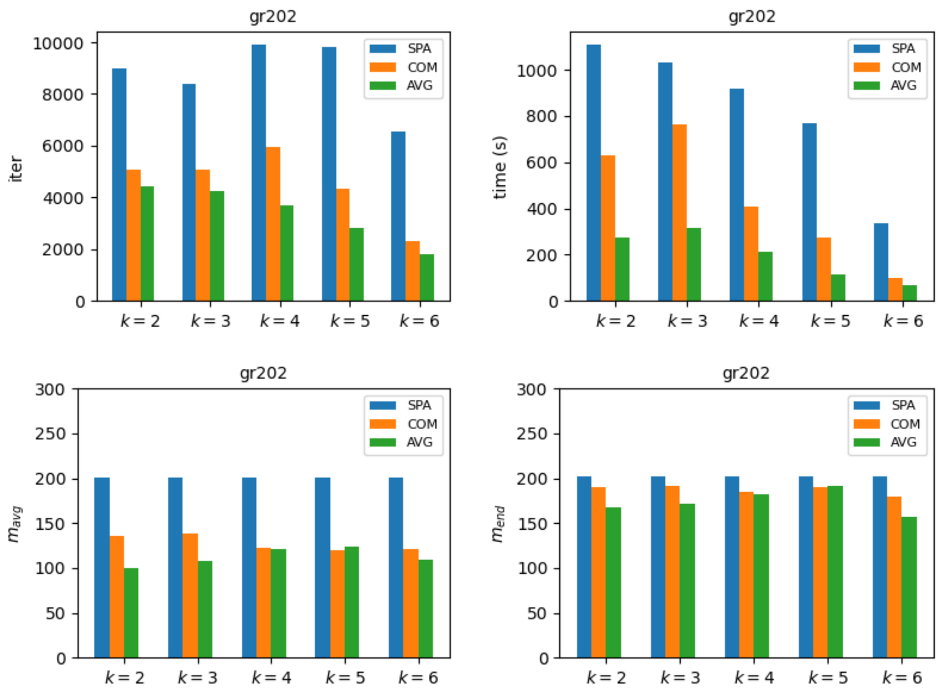

Figure 1 **Results of the CG+DCA algorithm under dual disaggregation strategies *sparse* (SPA), *complementary* (COM), and *average* (AVG) for data instance `gr202` and $k \in \{2,3,4,5,6\}$.**

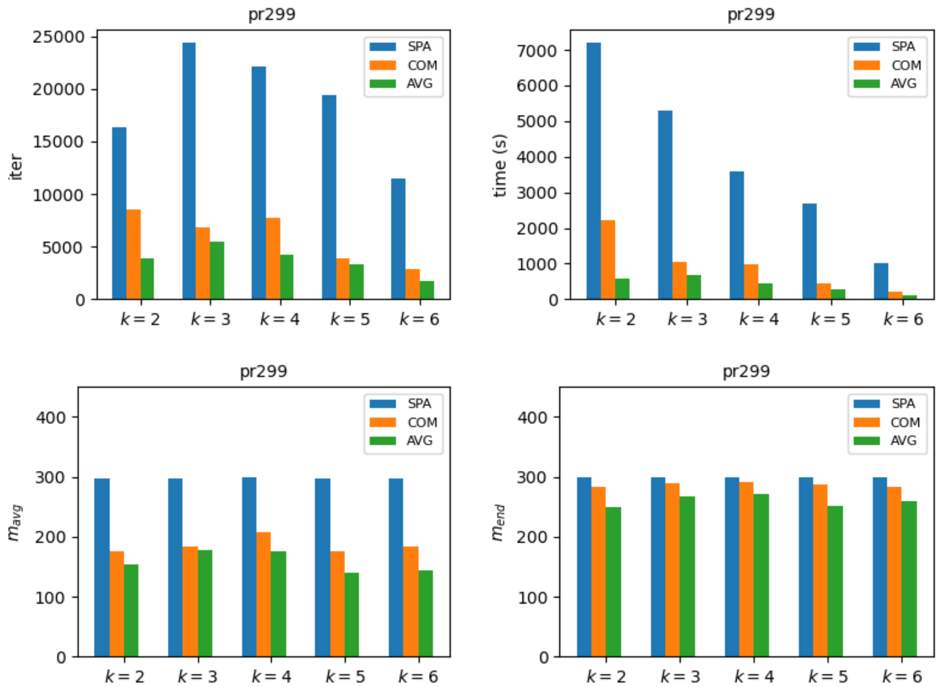

Figure 2 **Results of the CG+DCA algorithm under dual disaggregation strategies *sparse* (SPA), *complementary* (COM), and *average* (AVG) for data instance `pr299` and $k \in \{2,3,4,5,6\}$.**

Appendix E Bar charts on Q disaggregation

The bar charts refer to the average number of incompatibilities of the entering incompatible column (u_{avg}), the total number of times the aggregating partition Q was updated (Q updates), total computing times spent by the algorithm CG+DCA, and the average number of constraints in the aggregated RMP (m_{avg}).

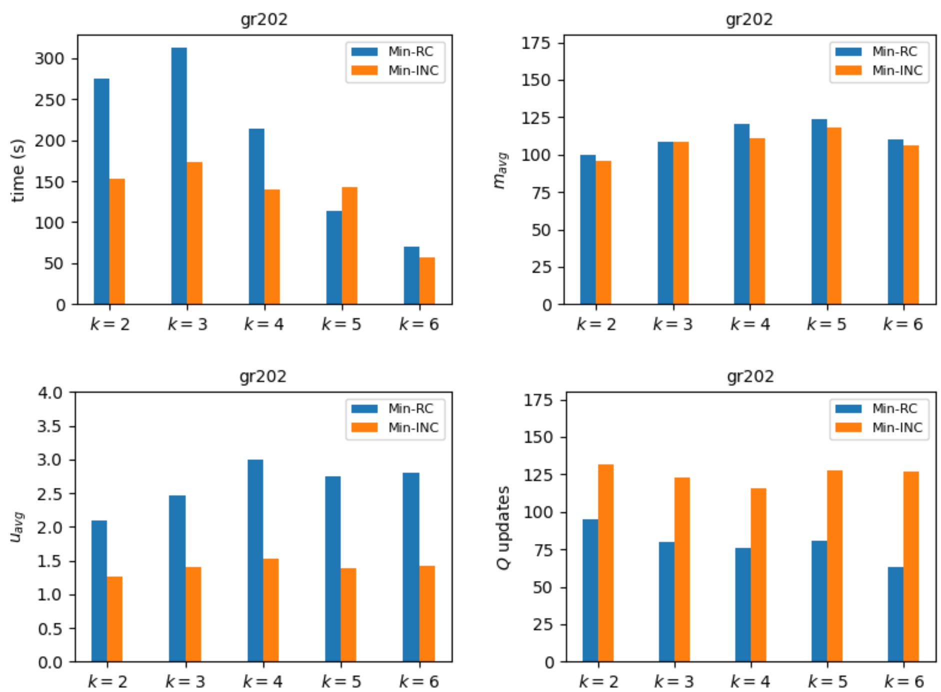

Figure 3 **Results of the CG+DCA algorithm under partition disaggregation strategies *Min-RC* and *Min-INC* for data instance `gr202` and $k \in \{2,3,4,5,6\}$.**

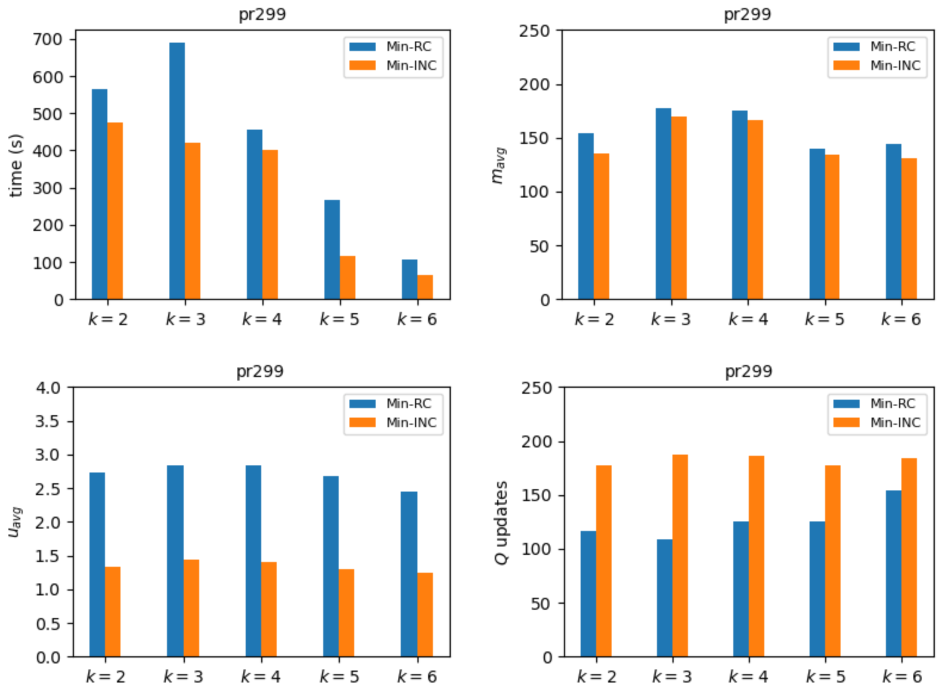

Figure 4 **Results of the CG+DCA algorithm under partition disaggregation strategies *Min-RC* and *Min-INC* for data instance `pr299` and $k \in \{2,3,4,5,6\}$.**