Optimal Approximation of Doubly Stochastic Matrices
Supplementary Material

Nikitas Rontsis
Department of Engineering Science
University of Oxford, OX1 3PJ, UK

Paul J. Goulart
Department of Engineering Science
University of Oxford, OX1 3PJ, UK

1 Proof of Theorem 3.2

In this section we will prove Theorem 3.2 from the main paper, i.e. we will show that

$$A \operatorname{diag}(H_u \operatorname{vec}(D))A^T = S \odot (D + D^T) + \operatorname{diag}(S \odot (D + D^T)\mathbf{1})$$

where S (S_u) is an integer $(0-1)$ matrix representing the sparsity pattern of C (C_u) and D is an upper triangular $n \times n$ matrix.

Using the fact that $H_u H_u^T = I$, the definition of A and defining $d := H_u \operatorname{vec} D$ we get

$$\begin{aligned} A \operatorname{diag}(d)A^T &= A \operatorname{diag}(\sqrt{d}) H_u H_u^T \operatorname{diag}(\sqrt{d})A^T \\ &= \Big(A \operatorname{diag}(\sqrt{d}) \operatorname{diag}(H_u)\Big)\Big(A \operatorname{diag}(\sqrt{d}) \operatorname{diag}(H_u)\Big)^T \end{aligned}$$

where $\sqrt{d}$ denotes the element-wise square root of d. Recalling that $A = A_1 + A_2$ we have

$$A \operatorname{diag}(\sqrt{d}) \operatorname{diag}(H_u) = \underbrace{A_1 \operatorname{diag}(\sqrt{d}) \operatorname{diag}(H_u)}_{:=B_1} + \underbrace{A_2 \operatorname{diag}(\sqrt{d}) \operatorname{diag}(H_u)}_{:=B_2}$$

thus

$$A \operatorname{diag}(d)A^T = \begin{bmatrix} I_n & I_n \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}^T \begin{bmatrix} I_n \\ I_n \end{bmatrix}$$

We now focus on the first part of the above symmetric product, To this end, define $\sqrt{s}_i$ the i^{th} column of $S_u \odot \sqrt{D}$ where $\sqrt{D}$ denotes the element-wise square root of D. Using the fact that

$H_u^T \operatorname{diag}(x) H_u = \operatorname{diag}(\operatorname{vec}(S_u) \odot (H_u^T x))$ for any vector x of appropriate dimensions we get

$$\begin{aligned} H_u^T \operatorname{diag}(\sqrt{d}) H_u &= \operatorname{diag}(\operatorname{vec}(S_u) \odot (H_u^T \sqrt{d})) = \operatorname{diag}(\operatorname{vec}(S_u) \odot (H_u^T H \operatorname{vec}(\sqrt{D}))) \\ &= \operatorname{diag}(\operatorname{vec}(S_u \odot S_u \odot \sqrt{D})) = \operatorname{diag}(\operatorname{vec}(S_u \odot \sqrt{D})). \end{aligned} \tag{1}$$

Use the definitions of A_1, A_2 and (1) to get

$$\begin{aligned} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} \mathbf{1}^T \otimes I \\ I \otimes \mathbf{1}^T \end{bmatrix} H_u^T \operatorname{diag}(d) H_u &= \begin{bmatrix} I & \cdots & I \\ \mathbf{1}^T & & \\ & \ddots & \\ & & \mathbf{1}^T \end{bmatrix} \begin{bmatrix} \operatorname{diag}(\sqrt{s_1}) & & \\ & \ddots & \\ & & \operatorname{diag}(\sqrt{s_n}) \end{bmatrix} \\ &= \begin{bmatrix} \operatorname{diag}(\sqrt{s_1}) & \cdots & \operatorname{diag}(\sqrt{s_n}) \\ \sqrt{s_1}^T & & \\ & \ddots & \\ & & \sqrt{s_n}^T \end{bmatrix} \end{aligned}$$

We can then form the symmetric product of this matrix with itself to obtain

$$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}^T = \begin{bmatrix} \operatorname{diag}(\sqrt{s_1}) & \cdots & \operatorname{diag}(\sqrt{s_n}) \\ \sqrt{s_1}^T & & \\ & \ddots & \\ & & \sqrt{s_n}^T \end{bmatrix} \begin{bmatrix} \operatorname{diag}(\sqrt{s_1}) & \sqrt{s_1} & & \\ \vdots & & \ddots & \\ \operatorname{diag}(\sqrt{s_n}) & & & \sqrt{s_n} \end{bmatrix}$$

or, noting that $\sqrt{s_i}^T \sqrt{s_i} = \mathbf{1}^T s_i$, $\operatorname{diag}^2(\sqrt{s_i}) = \operatorname{diag}(s_i)$, $\operatorname{diag}(\sqrt{s_i})\sqrt{s_i} = s_i$ where s_i denotes the $i-$th column of $S_u \odot \operatorname{mat}(d)$,

$$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}^T = \begin{bmatrix} \operatorname{diag}(\sum_i s_i) & s_1 & \dots & s_n \\ s_1^T & \mathbf{1}^T s_1 & & \\ \vdots & & \ddots & \\ s_n^T & & & \mathbf{1}^T s_n \end{bmatrix} = \begin{bmatrix} \operatorname{diag}((S_u \odot D)\mathbf{1}) & (S_u \odot D) \\ (S_u \odot D)^T & \operatorname{diag}((S_u \odot D)^T \mathbf{1}) \end{bmatrix}.$$

Thus

$$\begin{aligned} A \operatorname{diag}(d) A^T &= \begin{bmatrix} I & I \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} B_1^T B_2^T \end{bmatrix} \begin{bmatrix} I \\ I \end{bmatrix} \\ &= \operatorname{diag}((S_u \odot D)\mathbf{1}) + (S_u \odot D) + (S_u \odot D)^T + \operatorname{diag}((S_u \odot D)^T \mathbf{1}) \end{aligned}$$

or, recalling that D is upper triangular,

$$A \operatorname{diag}(H_u \operatorname{vec}(D)) A^T = S \odot (D + D^T) + \operatorname{diag}(S \odot (D + D^T)).$$

2 Detailed Results for the SuiteSparse Matrices

In this Section we provide detailed results for §4.3 of the main paper.

Table 1: Results of Algorithm 1 for Undirected Weighted Graph Matrices from the SuiteSparse collection. A tolerance of 10^{-4} is used for the termination of our Algorithm. The timings are in seconds and they are compared against the solution times of Gurobi with its default settings (on $\mathcal{P}1$). Hardware used: a single thread running on an Intel Gold 5120 with 192GB of memory.

Problem Name	n	n_{nz}	t_{admm}	t_{gurobi}
GD97_b	47	311	1.14×10^{-3}	4.07×10^{-2}
Journals	124	12 068	1.24×10^{-2}	7.80×10^{-2}
MISKnowledgeMap	2427	59 449	8.75×10^{-2}	1.47
Sandi_authors	86	334	1.17×10^{-3}	3.87×10^{-2}
Stranke94	10	100	6.40×10^{-4}	9.81×10^{-3}
USAir97	332	4584	5.26×10^{-3}	3.27×10^{-2}
ak2010	45 292	262 390	4.60×10^{-1}	2.22
al2010	252 266	1 482 748	2.87	1.61×10^{1}
ar2010	186 211	1 090 521	1.39	1.17×10^{1}
astro-ph	16 706	259 208	9.47×10^{-1}	2.31×10^{1}
az2010	241 666	1 437 760	2.71	1.45×10^{1}
ca2010	710 145	4 199 511	5.65	5.57×10^{1}
co2010	201 062	1 175 636	2.19	1.24×10^{1}
cond-mat-2003	31 163	271 221	1.12	3.26×10^{1}
cond-mat-2005	40 421	391 803	2.31	9.90×10^{1}
cond-mat	16 726	111 914	1.90×10^{-1}	3.35
ct2010	67 578	403 930	5.07×10^{-1}	3.76
de2010	24 115	140 171	1.23×10^{-1}	1.21
fl2010	484 481	2 830 775	5.20	3.04×10^{1}
ga2010	291 086	1 709 142	3.18	2.10×10^{1}
geom	7343	31 139	1.21×10^{-1}	2.49×10^{-1}
hep-th	8361	39 863	6.28×10^{-2}	5.76×10^{-1}
hi2010	25 016	149 142	2.82×10^{-1}	1.25
human_gene1	22 283	24 669 643	2.47×10^{2}	8.94×10^{3}
human_gene2	14 340	18 068 388	1.21×10^{2}	3.36×10^{3}
ia2010	216 007	1 237 177	2.78	1.69×10^{1}
id2010	149 842	878 106	1.01	8.30
il2010	451 554	2 616 018	5.14	3.70×10^{1}

Table 1: Continued.

Problem Name	n	n_{nz}	t_{admm}	t_{gurobi}
in2010	267 071	1 548 787	3.00	1.92×10^{1}
ks2010	238 600	1 360 398	2.58	1.69×10^{1}
ky2010	161 672	949 450	1.14	9.44
la2010	204 447	1 185 081	5.54	1.40×10^{1}
lesmis	77	585	8.87×10^{-3}	4.28×10^{-2}
ma2010	157 508	934 118	1.08	9.49
md2010	145 247	845 625	9.59×10^{-1}	8.17
me2010	69 518	404 994	4.99×10^{-1}	3.69
mi2010	329 885	1 907 975	3.60	2.53×10^{1}
mn2010	259 777	1 486 879	2.89	1.95×10^{1}
mo2010	343 565	2 000 133	3.84	2.57×10^{1}
mouse_gene	45 101	28 967 291	5.97×10^{2}	3.52×10^{4}
ms2010	171 778	1 011 758	3.78	1.12×10^{1}
mt2010	132 288	770 956	9.05×10^{-1}	7.16
nc2010	288 987	1 705 607	3.05	1.88×10^{1}
nd2010	133 769	759 715	1.35	9.30
ne2010	193 352	1 107 206	2.03	1.35×10^{1}
netscience	1589	7073	1.36×10^{-2}	8.19×10^{-2}
nh2010	48 837	283 387	3.87×10^{-1}	2.46
nj2010	169 588	999 500	1.23	1.02×10^{1}
nm2010	168 609	999 579	1.61	9.92
nopoly	10 774	70 842	7.05×10^{-2}	5.35×10^{-1}
nv2010	84 538	501 536	4.66×10^{-1}	4.61
ny2010	350 169	2 059 713	3.66	2.29×10^{1}
oh2010	365 344	2 133 584	3.93	2.88×10^{1}
ok2010	269 118	1 543 266	2.83	1.68×10^{1}
or2010	196 621	1 176 133	1.44	1.22×10^{1}
pa2010	421 545	2 480 007	4.80	2.76×10^{1}
rgg_n_2_15_s0	32 768	353 248	3.44×10^{-1}	2.69
ri2010	25 181	150 931	1.42×10^{-1}	1.29
sc2010	181 908	1 075 068	1.30	1.12×10^{1}
sd2010	88 360	499 082	9.10×10^{-1}	5.57
tn2010	240 116	1 434 082	1.77	1.53×10^{1}
tx2010	914 231	5 370 503	1.00×10^{1}	7.65×10^{1}

Table 1: Continued.

Problem Name	n	n_{nz}	t_{admm}	t_{gurobi}
ut2010	115 406	687 472	9.04×10^{-1}	6.15
va2010	285 762	1 687 890	1.99	1.69×10^{1}
vt2010	32 580	188 178	2.98×10^{-1}	1.58
wa2010	195 574	1 143 006	2.05	1.25×10^{1}
wi2010	253 096	1 462 500	3.21	1.98×10^{1}
wv2010	135 218	798 140	9.75×10^{-1}	7.99
wy2010	86 204	513 790	4.89×10^{-1}	4.93