A Fourier space algorithm for solving quadratic assignment problems

Risi Kondor *

Abstract

The quadratic assignment problem (QAP) is a central problem in combinatorial optimization. Several famous computationally hard tasks, such as graph matching, partitioning, and the traveling salesman all reduce to special cases of the QAP.

In this paper we propose a new approach to the QAP based on the theory of non–commutative Fourier analysis on the symmetric group. Specifically, we present a branch–and–bound algorithm that performs both the branching and the bounding steps in Fourier space.

By exploiting the band–limited nature of the QAP objective function and using FFT techniques, the algorithm runs in $O(n^3)$ time per branch–and–bound node. The techniques underlying the algorithm generalize to a range of other combinatorial optimization problems.

1 Introduction

Given a pair of non-negative matrices $A, A' \in \mathbb{R}^{n\times n}$, the **quadratic assignment problem** (QAP) is to solve

$$\widehat{\sigma} = \arg\max_{\sigma\in\mathbb{S}_n} \sum_{i,j=1}^{n} A_{\sigma(i),\,\sigma(j)}\, A'_{i,j}, \tag{1.1}$$

where $\mathbb{S}_n$ is the group of permutations of $\{1, 2, \ldots, n\}$, also called the symmetric group of degree n [18]. An equivalent way to write (1.1) is $\widehat{\sigma} = \arg\max_{\sigma\in\mathbb{S}_n} \mathrm{tr}\left(P_\sigma A P_\sigma^\top A'\right)$, where P_σ is the permutation matrix corresponding to σ,

$$[P_\sigma]_{i,j} = \begin{cases} 1 & \text{if } \sigma(j) = i, \\ 0 & \text{otherwise.} \end{cases}$$

To keep the exposition as simple as possible, in this paper we assume that the diagonal elements of A and A' are zero. Extending the ideas of the paper to general matrices is relatively straightforward. Note that instead of (1.1), some authors consider the minimum quadratic assignment problem $\widehat{\sigma} = \arg\min_{\sigma\in\mathbb{S}_n} \sum_{i,j=1}^{n} A_{\sigma(i),\,\sigma(j)}\, A'_{i,j}$. The two problems are trivially related and for the rest of this paper we focus exclusively on (1.1).

Several famous NP-complete problems are natural to cast in the form (1.1), making the QAP one of the classic NP-hard problems of combinatorial optimization. For example, if A and A' are the adjacency matrices of two weighted graphs, then (1.1) finds the best matching between the two graphs. If A is the adjacency matrix of a single weighted graph and

$$A'_{i,j} = \begin{cases} -1 & \text{if } j = (i \bmod n) + 1, \\ 0 & \text{otherwise,} \end{cases} \tag{1.2}$$

then (1.1) is the traveling salesman problem (TSP).

The time honored strategy for finding the exact solution of QAPs is to use a so-called "branch and bound" algorithm on a tree with the permutations $\{\sigma \in \mathbb{S}_n\}$ at the leaves. Branch–and–bound requires that at each branch point we have an upper bound on the objective function (in our case $\sum_{i,j=1}^{n} A_{\sigma(i),\,\sigma(j)}\, A'_{i,j}$) restricted to the leaves below. The algorithm proceeds by descending from the root in a depth first manner, always choosing the most promising branch until it reaches a leaf that is a tentative maximizer. It then backtracks to check all other branches which, according to the bound, could yield a better maximum than the current maximizer.

Branch–and–bound algorithms for the QAP show variation both in what branching strategy they employ and what bounds they compute at each branch point. However, for a given tree structure, such as the simplest one, where each branching point corresponds to fixing a single $i \mapsto \sigma(i)$, small differences in the tightness of the bounds can make a huge difference in overall running time, especially towards the top of the tree. Consequently, a great deal of effort has been directed at devising improved bounds, even some that require $O(n^5)$ or more operations per branch [18].

The Fourier transform of a general function $f\colon \mathbb{S}_n \to \mathbb{C}$ is the collection of matrices

$$\widehat{f}(\lambda) = \sum_{\sigma\in\mathbb{S}_n} f(\sigma)\, \rho_\lambda(\sigma),$$

where λ extends over the integer partitions of n, and ρ_λ are the corresponding irreducible representations

*Center for the Mathematics of Information, California Institute of Technology, MC 305-16, Pasadena, CA 91125, U.S.A, risi@caltech.edu. Present work mostly done while at the Gatsby Computational Neuroscience Unit, University College London, 17 Queen Square, WC1N 3AR, U.K..

of the symmetric group. The starting point for the present paper is the observation in [20] that if f is the objective function of the QAP, then the only non-zero components of $\widehat{f}$ are the four matrices $\widehat{f}((n))$, $\widehat{f}((n-1,1))$, $\widehat{f}((n-2,2))$ and $\widehat{f}((n-2,1,1))$.

The novel aspect of our branch–and–bound algorithm is that its bounding strategy is based on these Fourier matrices. Specifically, each branch of the tree of permutations is identified with a certain coset of $\mathbb{S}_n$, and the bound is computed from the Fourier transform of f restricted to that coset. The key to computational efficiency is to exploit the representation theory of the symmetric group, allowing us to (a) compute the Fourier transform at the root of the tree in $O(n^3)$ time; (b) compute each Fourier transform at level k of the tree in $O(k^3)$ time from the Fourier transform immediately above; (c) compute the corresponding bound by singular value decomposition in $O(k^3)$ time.

Together, these techniques result in an overall time complexity of $O(n^3)$ per visited node, which is considered cheap by the standards of modern QAP algorithms. In particular, it matches the complexity of the classic, but still widely used (and in some situations hard to beat) Gilmore–Lawler (GL) bound based QAP algorithms [7][14].

In general, it is very difficult to derive theoretical results on how many nodes a branch–and–bound algorithm will visit, so most authors resort to empirical comparisons. Our preliminary experiments indicate that in general the Fourier algorithm performs similarly to the GL approach, while on at least one special class of QAPs it is superior.

Related work The QAP is a bottleneck for both theoreticians and empiricists. Despite some approximation results for special cases [2][6][1] no general approximation result was known for the QAP until the very recent work of Nagarajan and Sviridenko [17]. In fact, for the min-QAP problem there is a series of results showing that it remains hard for any approximation factor [23][19].

On the empirical front the situation is not much better. Most exact methods proposed to solve the QAP over the course of the last fifty years use branch–and–bound techniques, and only differ in their bounds, ranging from Gilmore–Lawler bounds, via eigenvalue based bounds to more sophisticated and expensive bounds that involve solving a semi-definite-program or other optimization problem at each branch point. Despite these achievements, solving a general QAP of size $n \geq 20$ remains hard, and solving a QAP with $n \geq 30$ is considered state–of-the-art. Some excellent surveys of the QAP are [3] and [18].

2 Graph functions and the graph correlation

To present our algorithm in the most intuitive manner, without loss of generality, we regard A and A' in (1.1) as the adjacency matrices of a pair of weighted, directed, loopless graphs $\mathcal{G}$ and $\mathcal{G}'$ of n vertices. The objective function

$$f(\sigma) = \sum_{i,j=1}^{n} A_{\sigma(i),\sigma(j)}\, A'_{i,j} \tag{2.3}$$

of the quadratic assignment problem then captures the degree of "alignment" between the two graphs when overlaying them by identifying vertex i of A' with vertex $\sigma(i)$ of A. We will also refer to (2.3) as the **graph correlation** between $\mathcal{G}$ and $\mathcal{G}'$. The quadratic assignment problem is to find the maximum of (2.3) over the symmetric group $\mathbb{S}_n$. Throughout the paper for the definitions of terms from group theory and representation theory, the reader is referred to Appendix A.

The natural "relabeling" action of $\mathbb{S}_n$ on the vertices of $\mathcal{G}$ is $i \mapsto \pi(i)$; on the edges of $\mathcal{G}$ is $(i,j) \mapsto (\pi(i),\pi(j))$; and on the adjacency matrix is $A \mapsto A^\pi$, where

$$A^\pi_{i,j} = A_{\pi^{-1}(i),\,\pi^{-1}(j)}. \tag{2.4}$$

Thus, (2.3) can be regarded as the elementwise product between a relabeled version of A and the original A'. Since the action $(i,j) \mapsto (\pi(i),\pi(j))$ is transitive on the off-diagonal entries of A and $(n,n-1)$ is stabilized by the subgroup $\mathbb{S}_{n-2}$, the off-diagonal part of A, viewed as a homogeneous space of $\mathbb{S}_n$, is isomorphic to $\mathbb{S}_n/\mathbb{S}_{n-2}$.

From an operational point of view, the above imply that A induces a function

$$f_A \colon \mathbb{S}_n \to \mathbb{R}, \qquad f_A(\sigma) = A_{\sigma(n),\,\sigma(n-1)}, \tag{2.5}$$

called the **graph function** of $\mathcal{G}$, which has the property that under relabeling it transforms by left-translation. This means that if A and A^π are related by (2.4), then the corresponding graph functions are related by $f_{A^\pi} = f_A^\pi$, where f_A^π is the **left-translate** of f_A by π, defined $f_A^\pi(\sigma) = f_A(\pi^{-1}\sigma)$. Graph functions were introduced in [11] and [13] to define efficiently computable graph invariants. In the present context they enter by way of the following result.

Proposition 2.1. *The objective function of the QAP given in (2.3) can be expressed as*

$$f(\sigma) = \frac{1}{(n-2)!} \sum_{\pi \in \mathbb{S}_n} f_A(\sigma\pi)\, f_{A'}(\pi).$$

Proof. For each (i,j) pair (with $i,j, \in \{1,2,\ldots,n\}$ and $i \neq j$), there are exactly $(n-2)!$ permutations in

$\mathbb{S}_n$ mapping n to i and $n-1$ to j (to be specific, such permutations form a left coset $\sigma_{i,j}\mathbb{S}_{n-2}$) Hence,

$$\sum_{i,j=1}^{n} A_{\sigma(i),\sigma(j)}\, A'_{i,j} = \frac{1}{(n-2)!} \sum_{\pi\in\mathbb{S}_n} A_{\sigma\pi(n),\sigma\pi(n-1)}\, A'_{\pi(n),\pi(n-1)} = \frac{1}{(n-2)!} \sum_{\pi\in\mathbb{S}_n} f_A(\sigma\pi) f_{A'}(\pi).$$

■

For general $g, h\colon \mathbb{S}_n \to \mathbb{R}$, $\sum_{\pi\in\mathbb{S}_n} g(\sigma\pi)\, h(\pi)$ is called the **correlation** of g with h. Thus, Proposition 2.1 establishes a connection between the correlation of two graphs and the correlation of the corresponding graph functions.

3 Formulating the QAP in Fourier space

Instead of working directly with the objective function (2.3), which is an object of size $n!$, in this paper we solve the QAP by manipulating its Fourier transform. In general, the **Fourier transform** of a function $g\colon \mathbb{S}_n \to \mathbb{R}$ is the collection of matrices

$$\widehat{g}(\lambda) = \sum_{\sigma\in\mathbb{S}_n} g(\sigma)\, \rho_\lambda(\sigma), \qquad \lambda \vdash n, \tag{3.6}$$

where $\lambda \vdash n$ denotes that $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ extends over all integer partitions of n, and for each partition, $\rho_\lambda\colon \mathbb{S}_n \to \mathbb{C}^{d_\lambda \times d_\lambda}$ is the corresponding irreducible representation (irrep) of $\mathbb{S}_n$. The inverse transform is

$$g(\sigma) = \frac{1}{n!} \sum_{\lambda\vdash n} d_\lambda \operatorname{tr}\big[\, \rho_\lambda(\sigma)^{-1}\, \widehat{g}(\lambda)\,\big]. \tag{3.7}$$

As explained in Appendix A, there is considerable freedom in the choice of irreps, and naturally, this will influence the form of the Fourier transform. However, several of our results demand that the ρ_λ satisfy some special properties, namely that they be real, orthogonal, and adapted to the chain of subgroups $\mathbb{S}_n > \mathbb{S}_{n-1} > \ldots > \mathbb{S}_1$. Therefore, in the following, unless stated otherwise, we assume that the ρ_λ are given in **Young's Orthogonal Representation (YOR)**, which fulfills all of these requirements. The rows/columns of ρ_λ, and hence of $\widehat{g}(\lambda)$ are usually indexed by standard Young tableaux, or, equivalently, Yamanouchi symbols. If Y is a Yamanouchi symbol of shape λ, then $[\widehat{g}]_Y$ will denote the column of $\widehat{g}(\lambda)$ indexed by Y. Once again, see Appendix A for the exact definitions of these terms.

While for a general function g the total number of entries across all $\{\widehat{g}(\lambda)\}_{\lambda\vdash n}$ matrices is $n!$, some important functions on permutations are **band–limited** in the sense that their Fourier transform is identically zero except for a small set of matrices $\{\widehat{g}_\lambda\}_{\lambda\in\Lambda}$, or a small set of columns $\{[\widehat{g}]_Y\}_{Y\in\mathcal{Y}}$. Functions induced from homogeneous spaces generally turn out to be band–limited in this sense. In [11] we have shown that the graph function f_A is severely band-limited, and introduced the term **graph Fourier transform** for $\widehat{f_A}$.

PROPOSITION 3.1. ***(for proof, see [11])*** *The Fourier transform of a (directed, weighted, loopless) graph $\mathcal{G}$ with adjacency matrix $A \in \mathbb{R}^{n\times n}$ (expressed in YOR) is identically zero except for*

1. *the one dimensional Fourier component $\widehat{f}((n))$;*
2. *the $[\widehat{f_A}]_{[2\ldots]}$ and $[\widehat{f_A}]_{[12\ldots]}$ columns of the $n-1$ dimensional component $\widehat{f}((n-1,1))$;*
3. *the $[\widehat{f_A}]_{[22\ldots]}$ column of the $n(n-3)/2$ dimensional component $\widehat{f}((n-2,2))$; and*
4. *the $[\widehat{f_A}]_{[32\ldots]}$ column of the $(n-1)(n-2)/2$ dimensional component $\widehat{f}((n-2,1,1))$.*

We now show that Propositions 2.1 and 3.1 together imply that the graph correlation is also band–limited. This is crucially important, because otherwise any algorithm attempting to maximize f in Fourier space would require $O(n!)$ storage. The fact that the objective function of the QAP is band–limited was first discussed in [20]. The following result goes further in that it also describes the structure of the $\widehat{f}(\lambda)$ matrices, and relates them to $\widehat{f_A}$ and $\widehat{f_{A'}}$.

PROPOSITION 3.2. *(cf. [20]) Given a pair of graphs $\mathcal{G}$ and $\mathcal{G}'$ of n vertices with graph Fourier transforms $\widehat{f_A}$ and $\widehat{f_{A'}}$, the Fourier transform of their graph correlation (2.3) can be expressed as*

$$\widehat{f}(\lambda) = \frac{1}{(n-2)!}\ \widehat{f_A}(\lambda) \cdot (\widehat{f_{A'}}(\lambda))^\top, \qquad \lambda \vdash n. \tag{3.8}$$

In particular, $\widehat{f}(\lambda)$ is identically zero unless $\lambda = (n)$, $(n-1,1)$, $(n-2,2)$ or $(n-2,1,1)$, and $\widehat{f}((n-2,2))$ and $\widehat{f}((n-2,1,1))$ are dyadic (rank one) matrices.

Proof. Given $g\colon \mathbb{S}_n \to \mathbb{R}$, let $g^-\colon \mathbb{S}_n \to \mathbb{R}$ be defined $g^-(\sigma) = g(\sigma^{-1})$. By the orthogonality of YOR, $\rho_\lambda(\sigma^{-1}) = \rho_\lambda(\sigma)^\top$, hence $\widehat{g^-}(\lambda) = \widehat{g}(\lambda)^\top$. Therefore,

$$\sum_{\pi\in\mathbb{S}_n} f_A(\sigma\pi)\, f_{A'}(\pi) = \sum_{\pi\in\mathbb{S}_n} f_A(\sigma\pi^{-1})\, f_{A'}(\pi^{-1}) = \sum_{\pi\in\mathbb{S}_n} f_A(\sigma\pi^{-1})\, f^-_{A'}(\pi) = f_A * f^-_{A'},$$

where $*$ denotes convolution, and (3.8) follows by the convolution theorem (see Appendix A). Plugging in

Proposition 3.1 gives the explicit expressions

$$\begin{aligned}
\widehat{f}((n)) &= \tfrac{1}{(n-2)!}\,[\widehat{f}_A]_{[\dots]}\cdot[\widehat{f}_{A'}]^\top_{[\dots]},\\
\widehat{f}((n-1,1)) &= \tfrac{1}{(n-2)!}\,\big([\widehat{f}_A]_{[2\dots]}\cdot[\widehat{f}_{A'}]^\top_{[2\dots]}+\\
&\qquad\qquad [\widehat{f}_A]_{[12\dots]}\cdot[\widehat{f}_{A'}]^\top_{[12\dots]}\big),\\
\widehat{f}((n-2,2)) &= \tfrac{1}{(n-2)!}\,[\widehat{f}_A]_{[22\dots]}\cdot[\widehat{f}_{A'}]^\top_{[22\dots]},\\
\widehat{f}((n-2,1,1)) &= \tfrac{1}{(n-2)!}\,[\widehat{f}_A]_{[32\dots]}\cdot[\widehat{f}_{A'}]^\top_{[32\dots]},
\end{aligned}$$

where the product of a column vector v with a row vector $w^\top$ is to be interpreted as the outer product matrix, $\big[vw^\top\big]_{i,j} = v_i w_j$. All other components of $\widehat{f}$ are zero. ■

4 Searching the tree of cosets

Letting $[\![i,j]\!]$ denote the special permutation (called a contiguous cycle)

$$[\![i,j]\!] = \begin{cases} k+1 & \text{for } k = i, i+1, \dots, j-1,\\ i & \text{for } k = j,\\ k & \text{otherwise,}\end{cases}$$

the symmetric group can be partitioned into a union of (disjoint) $\mathbb{S}_{n-1}$–cosets as $\mathbb{S}_n = \bigcup_{i_n=1,\dots,n}[\![i_n,n]\!]\,\mathbb{S}_{n-1}$. Each $[\![i_n,n]\!]\,\mathbb{S}_{n-1}$ itself partitions into $\mathbb{S}_{n-2}$–cosets as

$$[\![i_n,n]\!]\,\mathbb{S}_{n-1} = \bigcup_{i_{n-1}=1,\dots,n-1} [\![i_n,n]\!][\![i_{n-1},n-1]\!]\,\mathbb{S}_{n-2},$$

and so on, forming a tree of cosets with individual permutations at the leaves.

Our Fourier space branch–and–bound algorithm maximizes (2.3) by searching in this tree. To formulate the algorithm, to each vertex we associate a restricted objective function $f_{i_n,i_{n-1},\dots,i_{k+1}}\colon \mathbb{S}_k \to \mathbb{R}$ defined

$$\begin{aligned}(4.9)\quad & f_{i_n,i_{n-1},\dots,i_{k+1}}(\tau) =\\ &\qquad f([\![i_n,n]\!][\![i_{n-1},n-1]\!]\dots[\![i_{k+1},k+1]\!]\,\tau),\end{aligned}$$

and assume that we have access to corresponding upper bounds

$$(4.10)\qquad \max_{\tau\in\mathbb{S}_k} f_{i_n,i_{n-1},\dots,i_{k+1}}(\tau) \le B(\widehat{f}_{i_n,i_{n-1},\dots,i_{k+1}}).$$

We also introduce the shorthand f^k_I for $f_{i_n,i_{n-1},\dots,i_{k+1}}$.

The actual algorithm (Figure 1) follows the classic branch–and–bound strategy of searching in a depth–first manner, always following the branch with the highest bound. Having found a tentative maximizer σ_{best}, it backtracks, considering the second, third, etc. most promising branch at each level and updating σ_{best} whenever it finds a leaf with higher objective function value than the current maximizer. However, there is no need to explore branches with $B(\widehat{f}^k_I) \le f(\sigma_{\text{best}})$. The efficiency of branch–and–bound algorithms depends on what fraction of the branches they can eliminate in this way.

The novel feature of our algorithm is that all the above steps are performed in Fourier space, without ever computing a full inverse Fourier transform. To demonstrate that for the QAP objective function this is a viable strategy, we need to

(1) show that the f^k_I functions are band–limited just like f is,

(2) provide a way to efficiently compute each $\widehat{f}^k_I$ from the node above it,

(3) propose an efficiently computable bound $B(\widehat{f}^k_I)$, which is as tight as possible.

Of these goals (1) and (2) hinge on the following general result, which is implicit in Clausen's inverse FFT [4][16][9]. Once again, for the definition of restricted partitions $\lambda{\downarrow_{n-1}}$, extended partitions $\mu{\uparrow^n}$ and the μ–blocks $[\,\cdot\,]_\mu$, the reader is referred to Appendix A.

PROPOSITION 4.1. *Given any $f\colon \mathbb{S}_n \to \mathbb{R}$, and letting f_i be defined as in (4.9) (with $k = n-1$ and $f_i \equiv f_{i_n}$), in YOR,*

$$(4.11)\qquad \widehat{f}_i(\mu) = \sum_{\lambda\in\mu\uparrow^n} \frac{d_\lambda}{n\,d_\mu}\big[\,\rho_\lambda([\![i,n]\!])^\top\cdot\widehat{f}(\lambda)\,\big]_\mu$$

for all $\mu \vdash n{-}1$. In particular, if f is band–limited to the set of Fourier components indexed by $\Lambda \subset \{\lambda\vdash n\}$, then f_i is band–limited to the Fourier components indexed by $\Lambda{\downarrow_{n-1}} = \bigcup_{\lambda\in\Lambda}\lambda{\downarrow_{n-1}}$.

Proof. (Sketch) By definition of the Fourier transform,

$$\begin{aligned}\widehat{f}(\lambda) &= \sum_{\sigma\in\mathbb{S}_n}\rho_\lambda(\sigma)\,f(\sigma) =\\ &\qquad \sum_{i=1}^{n}\sum_{\tau\in\mathbb{S}_{n-1}}\rho_\lambda([\![i,n]\!]\tau)\,f([\![i,n]\!]\tau) =\\ &\qquad\qquad \sum_{i=1}^{n}\rho_\lambda([\![i,n]\!])\sum_{\tau\in\mathbb{S}_{n-1}}\rho_\lambda(\tau)\,f_i(\tau),\end{aligned}$$

which, by the branching rule, $\rho_\lambda(\tau) = \bigoplus_{\mu\in\lambda\downarrow_{n-1}}\rho_\mu(\tau)$ we can write as

$$\widehat{f}(\lambda) = \sum_{i=1}^{n}\rho_\lambda([\![i,n]\!]) \bigoplus_{\mu\in\lambda\downarrow_{n-1}} \widehat{f}_i(\mu).$$

The proposition follows by inverting this mapping from $(\widehat{f}_1,\dots,\widehat{f}_n)$ to $\widehat{f}$. The full proof is in Appendix B. ■

COROLLARY 4.1. *If $f\colon \mathbb{S}_n \to \mathbb{R}$ is the QAP objective function (2.3), then $\widehat{f}_I^k(\lambda)$ is identically zero unless $\lambda = (k)$, $(k-1,1)$, $(k-2,2)$ or $(k-2,1,1)$. Moreover, $\widehat{f}_I^k(k-2,2)$ and $\widehat{f}_I^k(k-2,1,1)$ are dyadic.*

Proof. Proposition 3.2 has shown that f is band–limited to $\{(n),(n-1),(n-2,2),(n-2,1,1)\}$. Since $(n)\downarrow_{n-1} = \{(n-1)\}$, $(n-1,1)\downarrow_{n-1} = \{(n-1),(n-2,1)\}$, $(n-2,2)\downarrow_{n-1} = \{(n-2,1),(n-3,2)\}$, and $(n-2,1,1)\downarrow_{n-1} = \{(n-2,1),(n-3,1,1)\}$, each f_i (with $i = 1,2,\ldots,n$) is band–limited to $\{(n-1),(n-2,1),(n-3,2),(n-3,1,1)\}$. Moreover, since

$$\widehat{f}_i((n-3,2)) = \frac{d_{(n-2,2)}}{n\, d_{(n-3,2)}} \times \big[\, \rho_{(n-2,2)}([\![i,n]\!])^\top\, \widehat{f}((n-2,2))\,\big]_{(n-3,2)},$$

$$\widehat{f}_i((n-3,1,1)) = \frac{d_{(n-2,1,1)}}{n\, d_{(n-3,1,1)}} \times \big[\, \rho_{(n-2,1,1)}([\![i,n]\!])^\top\, \widehat{f}((n-2,1,1))\,\big]_{(n-3,1,1)},$$

and $\widehat{f}((n-2,2))$ and $\widehat{f}((n-2,1,1))$ are dyadic matrices, so will $\widehat{f}_i((n-3,2))$ and $\widehat{f}_i((n-3,1,1))$ be. This proves our statement for $k = n-1$. The $k = n-2,\ldots,1$ cases follow by induction. ∎

5 Fourier space bounds

We leave the discussion of the complexity of computing $\widehat{f}_I^k(\lambda)$ to the next section, and now move on to our third goal of bounding $f_I^k(\tau)$. While the general trace–norm bound presented below is stated in terms of functions on $\mathbb{S}_n$, since any finite group has unitary irreps and non-commutative Fourier transforms, it would hold equally well in this wider setting. We are not aware of any bounds of this type having previously been published in the non-commutative harmonic analysis literature.

PROPOSITION 5.1. *For any function $f\colon \mathbb{S}_n \to \mathbb{R}$,*

$$\max_{\sigma\in\mathbb{S}_n} f(\sigma) \le \frac{1}{n!} \sum_{\lambda\vdash n} d_\lambda \big\|\, \widehat{f}(\lambda)\,\big\|_1 \tag{5.12}$$

where $\|M\|_1$ denotes the trace norm of the matrix M.

Proof. For a general matrix $A \in \mathbb{R}^{d\times d}$ with singular values $s_1, s_2, \ldots, s_d$, the trace norm of A is $\|A\|_1 = \sum_{i=1}^d s_i$, while the spectral norm of A is $\|A\|_2 = \max_{1\le i\le d} s_i$. It is well known that these two norms are dual to one-another in the sense that $\|A\|_1 = \sup_{\|B\|_2\le 1} \operatorname{tr}(AB)$. In particular, since $\rho_\lambda(\sigma)$ is orthogonal, $\|\rho_\lambda(\sigma)^{-1}\|_2 = 1$, and hence $\operatorname{tr}\big(\widehat{f}(\lambda)\cdot\rho_\lambda(\sigma)^{-1}\big) \le \|\,\widehat{f}(\lambda)\,\|_1$ for all $\sigma\in\mathbb{S}_n$ and $\lambda\vdash n$ in the inverse Fourier transform (3.7). ∎

```
function find_max(f̂, n) {
    σ ← best_leaf(f̂, n, f(e)) ;
    if σ ≠ ∅ then return σ;
    return e ;
}
function best_leaf(f̂, k, maxsofar) {
    σ ← ∅ ;
    if k=1 then
        if f(e) > maxsofar then σ ← e ;
    else
        for j = 1 to k do
            compute f̂_i ;
        end
        s ← (1, 2, ..., k) ;
        sort s so that B(f̂_{s_1}) ≥ ... ≥ B(f̂_{s_k}) ;
        for j = 1 to k do
            if B(f̂_{s_j}) ≤ maxsofar then return σ ;
            τ ← best_leaf(f̂_{s_j}, k−1, maxsofar) ;
            if τ ≠ ∅ then
                σ ← [[j, k]] τ ;
                maxsofar ← f(σ) ;
            end
        end
    end
    return σ ;
}
```

Figure 1: Our main branch–and–bound algorithm for solving the QAP.

COROLLARY 5.1. *For the QAP objective function (2.3), each upper bound $B(\widehat{f}_I^k) = n!^{-1}\sum_{\lambda\vdash k} d_\lambda \|\widehat{f}_I^k(\lambda)\|_1$ can be computed in $O(k^3)$ time.*

Proof. By Corollary 4.1, $B(\widehat{f}_I^k) = n!^{-1}(b_{(k)} + b_{(k-1,1)} + b_{(k-2,2)} + b_{(k-2,1,1)})$, where $b_\lambda = d_\lambda\|\widehat{f}_I^k(\lambda)\|_1$. Since $\widehat{f}_I^k((k))$ is one-dimensional, $b_{(k)}$ is trivial. The second component $\widehat{f}_I^k((k-1))$ is k–1–dimensional, so computing its SVD takes $O(k^3)$ time. The third component $\widehat{f}_I^k((k-2,2))$ is $O(k^2)$–dimensional but diadic, therefore, $\|\widehat{f}_I^k((k-2,2))\,\|_1 = \|u\cdot v^\top\|_1 = \|u\|\cdot\|v\|$ takes only $O(k^2)$ operations to compute. Finally, $\widehat{f}_I^k((k-2,1,1))$ is analogous. ∎

Together, $\widehat{f}((n))$ and $\widehat{f}((n-1,1))$ determine the matrix of "first order marginals" $F_{i,j} = \sum_{\sigma:\sigma(i)=j} f_I^k(\sigma)$ (see, e.g., [5][8] or [12]). Hence, a slightly tighter version of (5.12) for the QAP is

$$\max_{\sigma\in\mathbb{S}_n} f(\sigma) \le \frac{1}{(n-2)!}\max_{\sigma\in\mathbb{S}_n} \operatorname{tr}(P_\sigma F) + \frac{1}{n!}\sum_{\lambda\in\{(n-2,2),(n-2,1,1)\}} d_\lambda\|\,\widehat{f}(\lambda)\,\|_1. \tag{5.13}$$

In effect, this not only constrains the matrix multiplying $\widehat{f}((n-1,1))$ in the bound to be orthogonal, but also to be an actual representation matrix. The complexity of evaluating (5.13) is still $O(n^3)$, though, since $\max_{\sigma\in\mathbb{S}_n} \operatorname{tr}(P_\sigma F)$ is a linear assignment problem, which can be solved in $O(n^3)$ time by the Kuhn–Munkres (Hungarian) algorithm.

6 Computational complexity

To bound the complexity of our Fourier space branch–and–bound algorithm, we need to quantify the number of operations needed to compute $\widehat{f}$ and each $\widehat{f}^k_I$. In addition to the structural results of the previous section, the key to efficient computation is to exploit the special properties of YOR.

In particular, letting $\tau_i \in \mathbb{S}_n$ be the **adjacent transposition** that swaps i with $i+1$ and leaves everything else fixed, $[\![i,j]\!]$ can be written as $\tau_i\tau_{i+1}\dots\tau_{j-1}$, and hence for any $\lambda \vdash n$ and any $M \in \mathbb{R}^{d_\lambda \times c}$,

$$\rho_\lambda([\![i,j]\!])\, M = \rho_\lambda(\tau_i)\cdot\rho_\lambda(\tau_{i+1})\cdot\ldots\cdot\rho_\lambda(\tau_{j-1})\cdot M. \tag{6.14}$$

Since in YOR each row of $\rho_\lambda(\tau_k)$ has at most two non-zero entries, the right-hand side of this equation can be computed in just $2(j-i)\,c\,d_\lambda$ scalar operations.[1] If $M \in \mathbb{R}^{d_\lambda\times d_\lambda}$ is a dyadic matrix stored in the form $uv^\top$ (where u and v are column vectors), then the computational cost is further reduced to $2(j-i)\,d_\lambda$. Exploiting the sparsity of YOR in this way is at the heart of Clausen's FFT for $\mathbb{S}_n$ [4][16]. In the proof of the following result we also refer to Maslen's FFT [15], which uses somewhat more involved techniques, and under certain circumstances can shave off an extra factor of n from the complexity of Fourier transforms.

Proposition 6.1. *The Fourier transform of the QAP objective function (2.3) can be computed from A and A' in $3(2n-3)n(n-1)+2(n-1)^2+1$ scalar operations.*

Proof. By definition, f_A is right $\mathbb{S}_{n-2}$–invariant ($f_A(\sigma\tau) = f_A(\sigma)$ for all $\tau \in \mathbb{S}_{n-2}$), therefore it is induced from the corresponding $f_A^{\mathrm{restr}}\colon \mathbb{S}_n/\mathbb{S}_{n-2} \to \mathbb{R}$. By Theorem 4.1 of [15] the Fourier transform of such functions can be computed in $3(2n-3)n(n-1)/2$ operations. The same holds for $\widehat{f}_{A'}$. Once we have $\widehat{f}_A$ and $\widehat{f}_{A'}$, by Proposition 3.2, $\widehat{f}((n))$ can be computed by a single multiplication; $\widehat{f}((n-1,1))$ can be computed in $2(n-1)^2$ scalar operations; and $\widehat{f}((n-2,2))$ and $\widehat{f}((n-2,1,1))$, assuming that we store them in dyadic form, can be computed without any scalar operations at all. ■

Proposition 6.2. *If f is the objective function of the QAP, then $\widehat{f}_{i_n,i_{n-1},\dots,i_k}$ can be computed from $\widehat{f}_{i_n,i_{n-1},\dots,i_{k+1}}$ in $O(k^3)$ scalar operations for any $k = n, n-1,\dots,2$ and any choice of $i_n, i_{n-1},\dots,i_k$ (with $1 \le i_j \le j$).*

Proof. To simplify notation, let $f^k_I = f_{i_n,i_{n-1},\dots,i_{k+1}}$ and $f^{k-1}_{I'} = f_{i_n,i_{n-1},\dots,i_k}$. By Corollary 4.1, the only non-zero components of $\widehat{f}^k_I$ are $\widehat{f}^k_I((k))$, $\widehat{f}^k_I((k-1,1))$, $\widehat{f}^k_I((k-2,2))$ and $\widehat{f}^k_I((k-2,1,1))$. Furthermore, the last two of these components are dyadic, and we assume that they are stored in dyadic form. Since $\rho_{(k)}$ is the trivial representation, $\rho_{(k)}([\![i_k,k]\!])^\top \cdot \widehat{f}^k_I((k)) = \widehat{f}^k_I((k))$. By the hook rule $d_{(k-1,1)} = k-1$, $d_{(k-2,2)} = k(k-3)/2$ and $d_{(k-2,1,1)} = (k-1)(k-2)/2$. Applying the analog of (6.14) with $\rho_\lambda([\![i_k,k]\!])^\top = \rho_\lambda(\tau_{k-1})\dots\rho_\lambda(\tau_{i_k})$ then gives that $\rho_{(k-1)}([\![i_k,k]\!])^\top \cdot \widehat{f}^k_I((k-1))$ can be computed in $2(k-i_k)(k-1)^2$ operations; $\rho_{(k-2,2)}([\![i_k,k]\!])^\top \cdot \widehat{f}^k_I((k-2,2))$ can be computed in $(k-i_k)k(k-3)$ operations; and $\rho_{(k-2,1,1)}([\![i_k,k]\!])^\top \cdot \widehat{f}^k_I((k-2,1,1))$ can be computed in $(k-i_k)(k-1)(k-2)$ operations. Now applying Proposition 4.1 gives that from these products $\widehat{f}^{k-1}_{I'}((k-1))$ can be computed in 1 scalar operation (2 scalars to be added); $\widehat{f}^{k-1}_{I'}((k-2,2))$ can be computed in $2(k-1)^2$ operations (two dyadic products to be added to a third matrix); and $\widehat{f}^{k-1}_{I'}((k-3,2))$ and $\widehat{f}^{k-1}_{I'}((k-3,1,1))$ can be computed by just copying numbers. In total this is $2k(k-3)(k-i_k)+2(k-1)^2+1$ scalar operations. ■

Combining Propositions 6.1 and 6.2 with Corollary 5.1 gives our main complexity result.

Theorem 6.1. *The running time of Algorithm 1 is $O((1+c)n^3)$, where c is the total number of calls made to the recursive function* `best_leaf`.

Theorem 6.1 tells us that our Fourier space algorithm is efficient at evaluating individual nodes of the branch–and–bound tree. However, as is often the case with branch–and–bound algorithms, proving a meaningful bound for the number of tree nodes that need to be visited seems very challenging. For this reason, QAP algorithms are generally evaluated by empirical comparison on benchmark examples or random problems drawn from some ensemble.

[1] Here and in the following, in accordance with the majority of the algebraic complexity literature, by a single **scalar operation** we mean multiplying two scalars and adding them to a third. Multiplication by constants and copying are assumed to be free.

7 Comparison to other algorithms, generalizations, and empirical results

To the best of our knowledge, apart from [20], the idea of using non–commutative harmonic analysis to solve QAPs has not been previously explored, and both the branching and bounding strategies described above appear very different to those employed by other algorithms. However, there are some interesting connections.

7.1 Alternative coset trees. The coset $[\![i_n, n]\!]\mathbb{S}_{n-1}$ is exactly the set of permutations that map n to position i_n. Similarly, $[\![i_n, n]\!][\![i_{n-1}, n-1]\!]\mathbb{S}_{n-2}$ is the coset of permutations that map n to i_n and $n-1$ to either i_{n-1} or $i_{n-1}+1$, depending on whether $i_n > i_{n-1}$ or not, and so on, down to lower levels in the tree. In other words, descending the coset tree corresponds to successively assigning the vertices $n, n-1, n-2, \ldots$ of $\mathcal{G}$ to vertices of $\mathcal{G}'$. Many classical QAP algorithms subdivide the assignment problem in a very similar way, by assigning vertices of one graph to vertices of the other. Thus, even though the computations involved in branching are very different, some aspects of the behavior of the Fourier space algorithm can be expected to be similar to that of its classical counterparts. Moreover, at each branch point one could evaluate both the Fourier and a classical bound and hope to improve performance by using the lesser of the two.

An advantage of the Fourier framework is that it permits a range of variations on the branching strategy. The simplest of these is to branch to two–sided cosets

$$[\![i_n, n]\!] \ldots [\![i_{k+1}, k+1]\!]\mathbb{S}_k\, [\![j_{k+1}, k+1]\!]^{-1} \ldots [\![j_n, n]\!]^{-1},$$

corresponding to assigning j_n to i_n, etc.. While by a slight generalization of Proposition 6.2 the objective function can still be restricted to these cosets in $O(k^3)$ operations, being able to vary both the i's and the j's can make it easier for the algorithm to narrow its search to the most promising subset of permutations. Branching to cosets induced by subgroups of $\mathbb{S}_n$ other than $\mathbb{S}_k$ is also possible, but would make the branching step somewhat more involved.

7.2 Comparison to GL–bounds. It is interesting to compare (5.12) with classical QAP bounds, in particular, the GL–bound mentioned in the introduction, which at the top of the tree takes the form $f(\sigma) \leq \min_{\tau\in\mathbb{S}_n} \mathrm{tr}\,(P_\tau M)$, where $M_{i,j} = \tilde{A}_i^\top \tilde{A}'_j$, $\tilde{A}_i$ is a column vector composed of the entries of the the i'th column of A excepting A_{ii} sorted in decreasing order, and $\tilde{A}'_j$ is defined similarly. At lower levels of the tree, as expected, n would be replaced by k, A and A' by their submatrices indexed by the unassigned vertices, and $\mathrm{tr}\,(P_\tau M)$ is replaced by $\mathrm{tr}\,(P_\tau(M+L))$, where L captures the interaction between the vertex next to be assigned and those already assigned. It is remarkable that both the modified Fourier bound (5.13) and the GL–bound hinge on solving a linear assignment problem, but the objective functions are quite different.

7.3 Peak amplification. Composing f with a convex positive function ϕ accentuates its peaks, and can thus improve the performance of branch–and–bound search. Unfortunately, in our Fourier setting there is no general guarantee that $\phi \circ f$ will be band–limited. For polynomial ϕ the Fourier space multiplication theorem [8] does imply a band–limit, but in general $\widehat{\phi\circ f}$ involves many more components than f, and computing each component requires Clebsch–Gordan transforms, which are themselves very expensive.

The following proposition shows that for the simplest case of $\phi(x) = x^2$, $\widehat{\phi\circ f}$ can be computed directly from a sequence of "higher order graph functions" $f_A^{ijk\ell}, f_A^{ijik}, f_A^{ijjk}, \ldots$ without any Clebsch-Gordan transforms. In a branch–and–bound setting $\widehat{f^2}_{i_n, i_{n-1}, \ldots, i_{k+1}}$ can then be computed by Proposition 4.1, and the corresponding bounds can be computed by Proposition 5.1.

PROPOSITION 7.1. *The square of the QAP objective function (2.3) can be expressed as*

$$\begin{aligned}
\widehat{f^2}(\lambda) &= \frac{1}{(n-4)!}\ \widehat{f_A^{ijk\ell}}(\lambda)\cdot\big(\widehat{f_{A'}^{ijk\ell}}(\lambda)\big)^\top \\
&+\frac{1}{(n-3)!}\ \Big[\ \widehat{f_A^{ijik}}(\lambda)\cdot\big(\widehat{f_{A'}^{ijik}}(\lambda)\big)^\top + \widehat{f_A^{ijjk}}(\lambda)\cdot\big(\widehat{f_{A'}^{ijjk}}(\lambda)\big)^\top \\
&\quad + \widehat{f_A^{ijki}}(\lambda)\cdot\big(\widehat{f_{A'}^{ijki}}(\lambda)\big)^\top + \widehat{f_A^{ijkj}}(\lambda)\cdot\big(\widehat{f_{A'}^{ijkj}}(\lambda)\big)^\top\ \Big] \\
&+\frac{1}{(n-2)!}\ \Big[\ \widehat{f_A^{ijji}}(\lambda)\cdot\big(\widehat{f_{A'}^{ijji}}(\lambda)\big)^\top + \widehat{f_A^{ijij}}(\lambda)\cdot\big(\widehat{f_{A'}^{ijij}}(\lambda)\big)^\top\ \Big],
\end{aligned}$$

where

$$\begin{aligned}
f_A^{ijk\ell}(\pi) &= A_{\pi(n),\,\pi(n-1)}\, A_{\pi(n-2),\,\pi(n-3)}, \\
f_A^{ijik}(\pi) &= A_{\pi(n),\,\pi(n-1)}\, A_{\pi(n),\,\pi(n-2)}, \\
f_A^{ijjk}(\pi) &= A_{\pi(n),\,\pi(n-1)}\, A_{\pi(n-1),\,\pi(n-2)}, \\
f_A^{ijki}(\pi) &= A_{\pi(n),\,\pi(n-1)}\, A_{\pi(n-2),\,\pi(n)}, \\
f_A^{ijkj}(\pi) &= A_{\pi(n),\,\pi(n-1)}\, A_{\pi(n-2),\,\pi(n-1)}, \\
f_A^{ijji}(\pi) &= A_{\pi(n),\,\pi(n-1)}\, A_{\pi(n-1),\,\pi(n)}, \\
f_A^{ijij}(\pi) &= (A_{\pi(n),\,\pi(n-1)})^2.
\end{aligned}$$

In particular, $\widehat{f^2}$ is band limited to the ten Fourier components $\{\widehat{f}(\lambda)\}_{\lambda\in(n-4)\uparrow^n}$. Moreover, $\widehat{f}((n-4,4))$,

$\widehat{f}((n-4,2,2))$,$\widehat{f}((n-4,2,1,1))$ *and* $\widehat{f}((n-4,1,1,1))$ *are dyadic.*

Proof. The squared objective function

$$f^2(\sigma) = \Big(\sum_{i,j=1}^{n} A_{\sigma(i),\,\sigma(j)}\, A'_{i,j}\Big)^2 = \sum_{i,j,k,\ell=1}^{n} A_{\sigma(i),\,\sigma(j)}\, A_{\sigma(k),\,\sigma(\ell)}\, A'_{i,j}\, A'_{k,\ell}$$

can be broken down into a sum of three expressions. First, we write the contribution to the sum on the right hand side form terms where i, j, k and ℓ are all different as

$$\frac{1}{(n\,-4)!} \sum_{\pi\in\mathbb{S}_n} f_A^{ijk\ell}(\sigma\pi)\, f_A^{ijk\ell}(\pi).$$

Secondly, the contribution from terms corresponding to the four ways that either i or j can match k or ℓ we write as $(n-3)!^{-1}$ times

$$\sum_{\pi\in\mathbb{S}_n} f_A^{ijik}(\sigma\pi)\, f_A^{ijik}(\pi) + \sum_{\pi\in\mathbb{S}_n} f_A^{ijjk}(\sigma\pi)\, f_A^{ijjk}(\pi) + \sum_{\pi\in\mathbb{S}_n} f_A^{ijki}(\sigma\pi)\, f_A^{ijki}(\pi) + \sum_{\pi\in\mathbb{S}_n} f_A^{ijkj}(\sigma\pi)\, f_A^{ijkj}(\pi).$$

Finally, we have the contribution from terms where both i and j match one of k and ℓ, which is $(n-2)!^{-1}$ times

$$\sum_{\pi\in\mathbb{S}_n} f_A^{ijji}(\sigma\pi)\, f_A^{ijji}(\pi) + \sum_{\pi\in\mathbb{S}_n} f_A^{ijij}(\sigma\pi)\, f_A^{ijij}(\pi).$$

■

Our experiments suggest that for a variety of ensembles of random QAP problems, squaring the objective function does indeed lead to tighter bounds, but at least in the $n \le 20$ regime, the resulting reduction in the number of branches visited does not justify the extra cost of carrying around much larger Fourier transforms and having to compute larger SVDs. On large benchmark problems the situation might be different. Proposition 7.1 has natural generalizations to f^3, f^4, and so on.

7.4 Higher order assignment. The third–order version of the QAP is

$$\widehat{\sigma} = \arg\max_{\sigma\in\mathbb{S}_n} \sum_{i,j,k=1}^{n} A_{\sigma(i),\sigma(j),\sigma(k)}\, A'_{i,j,k},$$

and corresponds to finding the best alignment between two hypergraphs. From an algebraic point of view, however, the only difference is that solving this problem involves searching over $\mathbb{S}_n/\mathbb{S}_{n-3}$, as opposed to $\mathbb{S}_n/\mathbb{S}_{n-2}$. While we do not have space to describe this case in detail, the results of the paper generalize to third and higher order assignment problems with relative ease, except that naturally the per–node computational complexity will increase.

7.5 Empirical results. We have implemented our algorithm as an extension to the $\mathbb{S}_n$ob[9] open source object–oriented $\mathbb{S}_n$ FFT library, and will make the code publicly available. Experiments to compare the performance of the Fourier method to that of established QAP algorithms are under way, but at this stage our results are too preliminary to report in detail.

Our experience is that in general the performance of the basic Fourier algorithm is comparable to that of the GL method in terms of the tightness of the bounds, number of nodes visited, as well as total CPU time. Restricting ourselves to specific subclasses of QAPs, however, significant differences start to emerge. In particular, when A and A' are both symmetric, the GL algorithm seems to do better, while the case when they are anti–symmetric favors the Fourier algorithm.[2] For example, on 100 randomly generated anti–symmetric QAP instances of size $n = 12$ (where each entry of A above the diagonal is chosen independently from the uniform distribution on $[0, 1]$ and $A'_{i,j} = i - j$), on average, the mean Fourier bound at the top of the tree was 293.4 ± 22.8, while the mean GL–bound was 358.3 ± 18.2, and the mean number of nodes visited by the two algorithms was 51.3 ± 29.2 vs. 484.5 ± 247.0.

Clearly, more extensive experiments are needed to find out how useful our algorithm can be for solving QAPs arising in practical applications, and how competitive it is on standard benchmarks.

8 Conclusions

We presented a new algorithm for solving the quadratic assignment problem, which, to the best of our knowledge, is the first attempt to attack the QAP in the Fourier domain. The feasibility of the algorithm hinges on the following points. (1) The Fourier transform of the objective function of the QAP is very sparse (this was already pointed out in [20]) and can be computed in $O(n^3)$ time by FFT methods. (2) The Fourier space objective function can be restricted to each node of the search tree from the node above in $O(n^3)$ time. (3) Each

[2] The anti-symmetric case appears in ranking problems. If $A_{i,j}$ is the number of voters preferring candidate i to candidate j and $A'_{i,j} = i - j$, then the solution of QAP gives the optimal ranking of the candidates with respect to a natural metric. Also note that when one of the two matrices is (anti–)symmetric, the other one can also be (anti–)symmetrized without affecting the objective function.

branch can be upper bounded from its Fourier transform in $O(n^3)$ time.

Preliminary experiments suggest that on certain classes of QAPs, such as anti–symmetric ones, the Fourier approach is more efficient than other algorithms of the same per–node complexity, but more empirical work is needed to establish where the new algorithm fits in among established QAP solvers. One attractive feature of the Fourier approach is its generality: the framework is flexible enough to accommodate a variety of branching strategies, modifications to the objective function, and higher order assignment problems. In fact, it is applicable to optimizing over any finite group.

Acknowledgments

The author would like to thank Rocco Servedio, Cliff Stein, Garud Iyengar, Daniel Bienstock, John Shawe-Taylor and Marconi Soares Barbosa for discussions and suggestions related to this work, as well as the anonymous reviewers for some important observations.

References

[1] Esther M. Arkin, Refael Hassin, Shlomi Rubinstein, and Maxim Sviridenko. Approximations for maximum transportation with permutable supply vector and other capacitated star packing problems. *Algorithmica*, 39(2):175–187, 2004.

[2] Esther M. Arkin, Refael Hassin, and Maxim Sviridenko. Approximating the maximum quadratic assignment problem. *Inf. Process. Lett.*, 77(1):13–16, 2001.

[3] Erand Çela. *The Quadratic Assignment Problem: Theory and Algorithms*. Springer, 1998.

[4] Michael Clausen. Fast generalized Fourier transforms. *Theor. Comput. Sci.*, 67:55–63, 1989.

[5] Persi Diaconis. *Group Representation in Probability and Statistics*, volume 11 of *IMS Lecture Series*. Institute of Mathematical Statistics, 1988.

[6] Uriel Feige and Michael Langberg. Approximation algorithms for maximization problems arising in graph partitioning. *Journal of Algorithms*, 41:174–211, 2001.

[7] P. C. Gilmore. Optimal and suboptimal algorithms for the quadratic assignment problem. *J. Soc. Indust. Appl. Math.*, 10(2):305–313, 1962.

[8] Jonathan Huang, Carlos Ernesto Guestrin, and Leonidas Guibas. Fourier theoretic probabilistic inference over permutations. *Journal of Machine Learning Research (JMLR)*, 10:997–1070, May 2009.

[9] Risi Kondor. $\mathbb{S}_n$`ob`: a C++ library for fast Fourier transforms on the symmetric group, 2006. Currently downloadable from `http://www.gatsby.ucl.ac.uk/~risi/Snob/`.

[10] Risi Kondor. *Group theoretical methods in machine learning*. Ph.D. thesis, Columbia University, 2008.

[11] Risi Kondor and Karsten Borgwardt. The skew spectrum of graphs. In Andrew McCallum and Sam Roweis, editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 496–503. Omnipress, 2008.

[12] Risi Kondor, Andrew Howard, and Tony Jebara. Multi-object tracking with representations of the symmetric group. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

[13] Risi Kondor, Nino Shervashidze, and Karsten Borgwardt. The graphlet spectrum. In Léon Bottou and Michael Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 529–536, Montreal, June 2009. Omnipress.

[14] Eugene L. Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963.

[15] David K. Maslen. The efficient computation of Fourier transforms on the symmetric group. *Mathematics of Computation*, 67(223):1121–1147, 1998.

[16] David K. Maslen and Daniel N. Rockmore. Generalized FFTs — a survey of some recent results. In *Groups and Computation II*, volume 28 of *DIMACS Ser. Discrete Math. Theor. Comput. Sci.*, pages 183–287. AMS, Providence, RI, 1997.

[17] Viswanath Nagarajan and Maxim Sviridenko. On the maximum quadratic assignment problem. In *SODA '09: Proceedings of the Nineteenth Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 516–524, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[18] Panos M. Pardalos, Franz Rendl, and Henry Wolkowicz. The quadratic assignment problem: A survey and recent developments. *The Quadratic Assignment Problem and Related Problems*, 16, 1993.

[19] M. Queyranne. Performance ratio of polynomial heuristics for triangle inequality quadratic assignment problems. *Operations Research Letters*, 4:231–234, 1986.

[20] Dan Rockmore, Peter Kostelec, Wim Hordijk, and Peter F. Stadler. Fast fourier transforms for fitness landscapes. *Applied and Computational Harmonic Analysis*, 12(1):57–76, 2002.

[21] Daniel N. Rockmore. Some applications of generalized FFTs. *Proceedings of the DIMACS workshop on Groups and Computation*, pages 329–370, 1997.

[22] Bruce E. Sagan. *The Symmetric Group*. Graduate Texts in Mathematics. Springer, 2001.

[23] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *J. ACM*, 23(3):555–565, 1976.

[24] Jean-Pierre Serre. *Linear Representations of Finite Groups*, volume 42 of *Graduate Texts in Mathamatics*. Springer-Verlag, 1977.

9 Appendix A: Definitions from group theory and representation theory

In an effort to make our paper as close to self–contained as possible, in the following we give the definitions of some standard terms from group/representation theory, as well as of some not necessarily so standard notations. For a gentle introduction to representation theory, see, e.g., [24]. For information specific to the symmetric group, we recommend [22], and for more details on non–commutative Fourier transforms and their applications, see [5] [21] and [10].

Conjugation. We denote the complex conjugate of a number $z \in \mathbb{C}$ by z^* and the Hermitian conjugate $M^{*\top}$ of a matrix $M \in \mathbb{C}^{d_1 \times d_2}$ by $M^\dagger$.

Groups. A **group** is a set G endowed with an operation $G \times G \to G$ (usually denoted multiplicatively) obeying the following axioms:

G1. for any $x, y \in G$, $xy \in G$ (closure);

G2. for any $x, y, z \in G$, $x(yz) = (xy)z$ (associativity);

G3. there is a unique $e \in G$, called the **identity** of G, such that $ex = xe = x$ for any $x \in G$;

G4. for any $x \in G$, there is a corresponding element $x^{-1} \in G$ called the **inverse** of x, such that $x\,x^{-1} = x^{-1}x = e$.

We *do not* require that the group operation be commutative, i.e., in general, $xy \neq yx$. A subset H of G is called a **subgroup** of G, denoted $H < G$, if H itself forms a group with respect to the same operation as G, i.e., if for any $x, y \in H$, $xy \in H$. If $x \in G$ and $H < G$, then $xH = \{\, xh \mid h \in H \,\} \subset G$ is called a (left–) H–**coset**.

Representations. For the purposes of this paper a **representation** of G over $\mathbb{C}$ is a matrix–valued function $\rho\colon G \to \mathbb{C}^{d_\rho \times d_\rho}$ that satisfies $\rho(x)\,\rho(y) = \rho(xy)$ for any $x, y \in G$. We call d_ρ the **order** or the **dimensionality** of ρ. Note that $\rho(e) = I$ for any representation. Two representations ρ_1 and ρ_2 of the same dimensionality d are said to be **equivalent** if for some invertible $T \in \mathbb{C}^{d \times d}$, $\rho_1(x) = T^{-1}\rho_2(x)\,T$ for any $x \in G$. A representation ρ is said to be **reducible** if it decomposes into a direct sum of smaller representations in the form

$$\begin{aligned}\rho(x) &= T^{-1}\left(\rho_1(x) \oplus \rho_2(x)\right) T \\ &= T^{-1}\left(\begin{array}{c|c} \rho_1(x) & 0 \\ \hline 0 & \rho_2(x) \end{array}\right) T \qquad \forall\, x \in G\end{aligned}$$

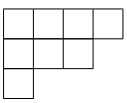
for some invertible $T \in \mathbb{C}^{d_\rho \times d_\rho}$. A maximal system of pairwise inequivalent, irreducible representations we call a system of **irreps**. It is possible to show that if $\mathcal{R}_1$ and $\mathcal{R}_2$ are two different systems of irreps of the same finite group G, then $\mathcal{R}_1$ and $\mathcal{R}_2$ have the same (finite) cardinality, and there is a bijection $\phi\colon \mathcal{R}_1 \to \mathcal{R}_2$ such that if $\phi(\rho_1) = \rho_2$, then ρ_1 and ρ_2 are equivalent. A finite group always has at least one system of irreps in which all the $\rho(x)$ matrices are **unitary**, i.e., $\rho(x)^{-1} = \rho(x)^\dagger = \rho(x)^{*\top}$ for all $x \in G$. The theorem of total reducibility asserts that given a system of irreps $\{\rho_1, \rho_2, \ldots, \rho_k\}$, any representation ρ can be reduced into a direct sum of the ρ_i's in the sense that there is an invertible matrix T and sequence of multiplicities $m_1, m_2, \ldots, m_k$ such that $\rho(x) = T^{-1}\big[\bigoplus_{i=1}^{k} \bigoplus_{j=1}^{m_k} \rho_i(x)\big]T$.

Permutations. A **permutation** of $\{1, 2, \ldots, n\}$ is a bijective mapping $\sigma\colon \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$. The product of two such permutations is defined by composition, $(\sigma_2\sigma_1)(i) = \sigma_2(\sigma_1(i))$ for all $i \in \{1, 2, \ldots, n\}$. With respect to this operation the set of all $n!$ possible permutations of $\{1, 2, \ldots, n\}$ form a group called the **symmetric group** of degree n, which we denote $\mathbb{S}_n$. For $m < n$ we identify $\mathbb{S}_m$ with the subgroup of permutations that only permute $1, 2, \ldots, m$, i.e., $\mathbb{S}_m = \{\, \tau \in \mathbb{S}_n \mid \tau(m+1) = m+1, \ldots, \tau(n) = n \,\}$.

Cycle notation. A **cycle** in a permutation $\sigma \in \mathbb{S}_n$ is a sequence $(c_1, c_2, \ldots, c_k)$ such that for $i = 1, 2, \ldots, k-1$, $\sigma(c_i) = c_{i+1}$ and $\sigma(c_k) = c_1$. Any permutation can be expressed as a product of disjoint cycles. Some special permutations that we will be interested in are

the **adjacent transpositions** $\tau_i = (i, i+1)$ and the **contiguous cycles** $[\![i, j]\!] = (i, i+1, \ldots, j)$.

Partitions and Young diagrams. The sequence of positive integers $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ is said to be an **integer partition** of n (denoted $\lambda \vdash n$) if $\sum_{i=1}^{k} \lambda_i = n$ and $\lambda_i \geq \lambda_{i+1}$ for $i = 1, 2, \ldots, k-1$. The **Young diagram** of λ consists of $\lambda_1, \lambda_2, \ldots, \lambda_k$ boxes laid down in consecutive rows, as in



for $\lambda = (4, 3, 1)$. One natural partial order on partitions is the inclusion order of their diagrams: for $\lambda = (\lambda_1, \ldots, \lambda_k) \vdash n$ and $\mu = (\mu_1, \ldots, \mu_{k'}) \vdash m$, $\lambda \geq \mu$ if and only if $\lambda_i \geq \mu_i$ for all $i = 1, 2, \ldots \max(k, k')$ (for $i > k'$ we take $\mu_i = 0$). We define the **restriction** of $\lambda \vdash n$ to $m \leq n$ as the set of partitions $\lambda\downarrow_m = \{\, \mu \vdash m \mid \lambda \geq \mu \,\}$, and the **extension** of $\mu \vdash m$ to $n \geq m$ as the set of partitions $\mu\uparrow^n = \{\, \lambda \vdash n \mid \lambda \geq \mu \,\}$.

Young tableaux. A Young diagram with numbers in its cells is called a **Young tableau**. A **standard tableau** is a Young tableau in which each of the numbers $1, 2, \ldots, n$ is featured exactly once, and in such a way that in each row the numbers increase from left to right and in each column they increase from top to bottom. For example,

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 5 & 8 \\ \hline 3 & 4 & 7 \\ \cline{1-3} 6 \\ \cline{1-1} \end{array} \qquad (9.15)$$

is a standard tableau of shape $\lambda = (4, 3, 1)$. An alternative way to specify a standard tableau t is by its so-called **Yamanouchi symbol** $t = [r_n r_{n-1} \ldots r_1]$, where r_i is the row index of the number i in t. We use the shorthand $[r_n r_{n-1} \ldots r_{n-k} \ldots]_n$ for the Yamanouchi symbol beginning with $r_n r_{n-1} \ldots r_{n-k}$ and padded with the appropriate number of 1's so as to result in a tableau of n boxes. For example, $[123122 \ldots]_8$ is equivalent to (9.15), and $[\ldots]_n$ is the unique standard tableau of shape $\lambda = (n)$.

Indexing the irreps. The significance of integer partitions and standard Young tableaux is that the former are in bijection with the irreps of $\mathbb{S}_n$, while the latter are in bijection with the rows and columns of the actual representation matrices. Exploiting these bijections, we label the irreps by the partitions $\{\lambda \vdash n\}$, and we label the individual rows and columns of $\rho_\lambda(\sigma)$ by the standard tableaux of shape λ (or, alternatively, the Yamanouchi symbols of shape λ). The dimensionalities $d_\lambda = d_{\rho_\lambda}$ can be computed from the so-called **hook rule**

$$d_\lambda = \frac{n!}{\prod_i \ell(i)}, \qquad (9.16)$$

where i ranges over all the cells of λ, and $\ell(i)$ are the lengths of the corresponding "hooks", i.e., the number of cells to the right of cell i plus the number of cells below i plus one. For example, it is easy to check that $d_{(1)} = 1$, $d_{(n-1,1)} = n-1$, $d_{(n-2,2)} = n(n-3)/2$, and $d_{(n-2,1,1)} = (n-1)(n-2)/2$.

Young's Orthogonal Representation. The specific system of irreps that we use in this paper is called **Young's orthogonal representation**, or just **YOR**. A special feature of YOR is that its irreps are not only unitary, but also real-valued, hence the $\rho_\lambda(\sigma)$ matrices are orthogonal. YOR is defined by explicitly specifying the representation matrices corresponding to adjacent transpositions. For any standard tableau t, letting $\tau_i(t)$ be the tableau that we get from t by exchanging the numbers i and $i+1$ in its diagram, the rule defining $\rho_\lambda(\tau_i)$ in YOR is the following: if $\tau_i(t)$ is *not* a standard tableau, then the column of $\rho_\lambda(\tau_i)$ indexed by t is zero, except for the diagonal element $[\rho_\lambda(\tau_i)]_{t,t} = 1/d_t(i, i+1)$; if $\tau_i(t)$ is a standard tableau, then in addition to this diagonal element, we also have a single non-zero off-diagonal element, $[\rho_\lambda(\tau_i)]_{\tau_k(t),t} = (1 - 1/d_t(i, i+1)^2)^{1/2}$. All other matrix entries of $\rho_\lambda(\tau_i)$ are zero. In the above $d_t(i, i+1) = c_t(i+1) - c_t(i)$, where $c(j)$ is the column index minus the row index of the cell where j is located in t. Note that the trivial representation $\rho_{\text{triv}}(\sigma) \equiv 1$ is the irrep indexed by $\lambda = (n)$. Also note that for any $\lambda \vdash n$, each row/column of $\rho_\lambda(\tau_i)$ has at most two non-zero entries.

Restricted representations. If ρ is an irrep of a finite group G and H is a subgroup of G, then ρ restricted to H is a representation of H (denoted $\rho\downarrow_H$), but in general it is not irreducible. This means that if $\rho_1, \rho_2, \ldots, \rho_k$ is a system of irreps of H, then by the theorem of total reducibility $\rho\downarrow_H(x) = T^{-1}\big[\bigoplus_{i=1}^{k} \bigoplus_{j=1}^{m_k} \rho_i(x)\big]\, T$ for some T and $m_1, m_2, \ldots, m_k$. We say that ρ is **adapted** to $H < G$ (equivalently, that it is in a Gelfand–Tsetlin basis w.r.t. $H < G$) if in this formula T is the identity. An important property of YOR is that it is adapted to the entire chain $\mathbb{S}_1 < \mathbb{S}_2 < \ldots < \mathbb{S}_n$. In particular, **Young's branching rule** tells us that in YOR the irreps of $\mathbb{S}_n$ restricted to $\mathbb{S}_{n-1}$ are of the particularly simple form $\rho_\lambda\downarrow_{n-1} = \bigoplus_{\mu \in \lambda\downarrow_{n-1}} \rho_\mu$, i.e., for any $\tau \in \mathbb{S}_{n-1}$, $\rho_\lambda(\tau)$ is block–diagonal and each block corresponds to a different $\mu \in \lambda\downarrow_{n-1}$. The block corresponding to μ in $M = \rho_\lambda(\sigma)$ or any square matrix of the same size we denote $[M]_\mu$.

The Fourier transform. The **Fourier transform** of a function $f\colon \mathbb{S}_n \to \mathbb{C}$ is defined as the collection of matrices

$$\widehat{f}(\lambda) = \sum_{\sigma\in\mathbb{S}_n} f(\sigma)\,\rho_\lambda(\sigma) \qquad\qquad \lambda \vdash n. \tag{9.17}$$

As always, we assume that the ρ_λ representations are given in YOR. Since YOR is a system of real valued representations, if f is a real valued function, then the $\widehat{f}(\lambda)$ Fourier components will also be real valued. Non-commutative Fourier transforms such as (9.17) enjoy some of the same properties as the usual Fourier transforms on the real line and the unit circle. In particular (9.17) is a an invertible, unitary mapping $\mathbb{C}^{\mathbb{S}_n} \to \bigoplus_{\lambda\vdash n} \mathbb{C}^{d_\lambda\times d_\lambda}$ with inverse given by

$$f(\sigma) = \frac{1}{n!}\sum_{\lambda\vdash n} d_\lambda\,\mathrm{tr}\big[\widehat{f}(\lambda)\,\rho_\lambda(\sigma)^{-1}\big].$$

Convolution and correlation. The **convolution** of $f\colon \mathbb{S}_n \to \mathbb{C}$ with $g\colon \mathbb{S}_n \to \mathbb{C}$ is defined $(f * g)(\sigma) = \sum_{\tau\in\mathbb{S}_n} f(\sigma\tau^{-1})\,g(\tau)$. The convolution theorem states that $\widehat{f*g}(\lambda) = \widehat{f}(\lambda)\,\widehat{g}(\lambda)$ for each $\lambda \vdash n$. The **correlation** of f with g is defined $(f \star g)(\sigma) = \sum_{\tau\in\mathbb{S}_n} f(\sigma\tau)g(\tau)^*$ and the correlation theorem states that $\widehat{f\star g}(\lambda) = \widehat{f}(\lambda)\,\widehat{g}(\lambda)^\dagger$.

10 Appendix B: Proof of Proposition 4.1

The inverse Fourier transform of $\widehat{f_i}$ on $\mathbb{S}_{n-1}$ in YOR is

$$f_i(\tau) = \frac{1}{(n-1)!}\sum_{\mu\vdash n-1} d_\mu \mathrm{tr}\big[\rho_\mu(\tau)^\top\,\widehat{f_i}(\mu)\big]. \tag{10.18}$$

On the other hand, the inverse Fourier transform on $\mathbb{S}_n$ tells us that

$$\begin{aligned} f_i(\tau) = f([\![i,n]\!]\tau) &= \frac{1}{n!}\sum_{\lambda\vdash n} d_\lambda \mathrm{tr}\big[\rho_\lambda([\![i,n]\!]\,\tau)^\top\,\widehat{f}(\lambda)\big] \\ &= \frac{1}{n!}\sum_{\lambda\vdash n} d_\lambda \mathrm{tr}\big[\rho_\lambda(\tau)^\top\rho_\lambda([\![i,n]\!])^\top\,\widehat{f}(\lambda)\big] \end{aligned}$$

for any $\tau \in \mathbb{S}_{n-1}$, By the branching rule, $\rho_\lambda(\tau) = \bigoplus_{\mu\in\lambda\downarrow_{n-1}} \rho_\mu(\tau)$, hence

$$\begin{aligned} &\mathrm{tr}\big[\rho_\lambda(\tau)^\top\rho_\lambda([\![i,n]\!])^\top\,\widehat{f}(\lambda)\big] \\ &\qquad = \mathrm{tr}\Big[\Big[\bigoplus_{\mu\in\lambda\downarrow_{n-1}} \rho_\mu(\tau)^\top\Big]\rho_\lambda([\![i,n]\!])^\top\,\widehat{f}(\lambda)\Big] \\ = \mathrm{tr}\Big[\Big[\bigoplus_{\mu\in\lambda\downarrow_{n-1}} \rho_\mu(\tau)^\top\Big]\cdot\Big[\bigoplus_{\mu\in\lambda\downarrow_{n-1}} &\big[\rho_\lambda([\![i,n]\!])^\top\,\widehat{f}(\lambda)\big]_\mu\Big]\Big] \\ &\qquad = \sum_{\mu\in\lambda\downarrow_{n-1}} \mathrm{tr}\big[\rho_\mu(\tau)^\top[\rho_\lambda([\![i,n]\!])^\top\,\widehat{f}(\lambda)]_\mu\big]. \end{aligned}$$

Furthermore, the double summation $\sum_{\lambda\vdash n}\sum_{\mu\in\lambda\downarrow_{n-1}}$ is equivalent to $\sum_{\mu\vdash n-1}\sum_{\lambda\in\mu\uparrow^{n-1}}$, hence

$$f_i(\tau) = \frac{1}{n!}\sum_{\mu\vdash n-1}\sum_{\lambda\in\mu\uparrow^n} d_\lambda \mathrm{tr}\big[\rho_\mu(\tau)^\top[\rho_\lambda([\![i,n]\!])^\top\,\widehat{f}(\lambda)]_\mu\big]. \tag{10.19}$$

By the invertibility of the Fourier transform, $\{\rho_\mu(\tau)\}_{\tau\in\mathbb{S}_{n-1}}$ must span $\mathbb{R}^{d_\mu\times d_\mu}$. Hence, for (10.18) and (10.19) to both hold for all $\tau \in \mathbb{S}_{n-1}$, we must have (4.11). ■