The Distribution of Values in Combinatorial Optimization Problems

by

Tamon Stephen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mathematics)
in The University of Michigan
2002

Doctoral Committee:

Professor Alexandre I. Barvinok, Chair
Professor Joseph G. Conlon
Assistant Professor Marina A. Epelman
Professor Philip J. Hanlon
Professor John R. Stembridge

To my parents.

ACKNOWLEDGEMENTS

I thank my supervisor, Alexander Barvinok for his patient guidance and humor throughout this project, and for several terms of funding.

I also thank Joe Conlon, Marina Epelman, Phil Hanlon and John Stembridge for their interest in, and attention to, my thesis. There are many other faculty at Michigan who I have had good interactions with, and I would particularly like to thank the graduate students, too numerous to list, who have made this and a friendly and enjoyable five years.

Many people outside of Michigan have provided me with help and encouragement. Levent Tunçel got me started on research and in optimization. I was encouraged by many others at Waterloo, including faculty, coworkers and above all fellow students. Further back, in high school, thanks go to my teachers, who were unusually enthusiastic about mathematics, and in the end, to my family, who trusted my judgments and rarely insisted that I do anything useful with my time.

I'd like to give special mentions to the Fields Institute (Fall 1999), Mike Bandurchin, I.D. Brown, Kimberly Fair, Stephen Le, Todd Parsons, Alek Seelemann, Jennifer Svarckopf, Ming-wei Wang, Song Wang and Wei Yu.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER I

Background

In discrete optimization problems, an objective function is given on a large finite set. The goal is to find "good" (say particularly large or small) values efficiently.

Computational complexity theory has identified a class of problems which are called *NP-complete*. The NP-complete problems contain many of the combinatorial problems that arise in practice, such as in operations research, but seem to be difficult to solve efficiently. The famous $P = NP$? question, which asks whether there is any polynomial time algorithm for NP-complete problems, remains the central problem in computational complexity. However, there is considerable progress in understanding NP-complete problems by using tools such as approximation, randomness and non-polynomial algorithms.

The goal of this thesis is to introduce a method of studying the distribution of objective values for (usually NP-complete) combinatorial optimization problems defined on groups. This approach shows us some very general, yet non-trivial properties of the optimization function. In particular, it allows us to produce guarantees for simple polynomial and non-polynomial approximation algorithms, evaluate heuristics, and helps us understand why some hard problems are simpler than others.

We focus on two important examples of combinatorial optimization problems, the

Quadratic Assignment Problem (QAP) and the Traveling Salesman Problem (TSP). We obtain results on both the *frequency* and *location* of good values.

Our approach is to use techniques from representation theory (see Section 3.2). The main object of study is the function obtained by averaging the objective function on the conjugacy classes of permutations.

This Chapter contains background material, including definitions and some history. We state our main results in Chapter II. The necessary results from representation theory and the proofs are in Chapter III. Discussion and some ancillary results, including computational data and comments on derandomization, are included in Chapter IV.

1.1 Problem Definitions

Let Mat_n be the vector space of all real $n \times n$ matrices $A = (a_{ij})$, $1 \leq i, j \leq n$. The input data for a QAP is a pair of matrices in Mat_n; we refer to such a problem as having size n. To measure the complexity of an algorithm, we count the number of arithmetic operations performed (see for example [AHU74]). This is the usual model of complexity if the data is integer, and can also be used with real numbers [BCSS97]. From now on, we assume that $n \geq 4$.

Let S_n be the group of all permutations σ of the set $\{1, \dots, n\}$. We are interested in the action of S_n on the space Mat_n by simultaneous permutations of rows and columns: we let $\sigma(A) = B$, where $A = (a_{ij})$ and $B = (b_{ij})$, provided $b_{\sigma(i)\sigma(j)} = a_{ij}$ for all $i, j = 1, \dots, n$. Notice that $(\sigma\tau)A = \sigma(\tau A)$ for any two permutations σ and τ. There is a standard scalar product on Mat_n:

$$\langle A, B \rangle = \mathrm{trace}(AB^t) = \sum_{i,j=1}^{n} a_{ij} b_{ij}, \quad \text{where} \qquad A = (a_{ij}) \quad \text{and} \quad B = (b_{ij})$$

Definition 1.1.1. Let us fix two matrices $A = (a_{ij})$ and $B = (b_{ij})$ and let us consider a real-valued function $f : S_n \longrightarrow \mathbb{R}$ defined by

$$f(\sigma) = \langle B, \sigma(A) \rangle = \sum_{i,j=1}^{n} b_{\sigma(i)\sigma(j)} a_{ij} = \sum_{i,j=1}^{n} b_{ij} a_{\sigma^{-1}(i)\sigma^{-1}(j)} \tag{1.1}$$

The problem of finding a permutation σ where the maximum or minimum value of f is attained is known as the *Quadratic Assignment Problem* (QAP). It is one of the hardest problems of Combinatorial Optimization.

We say that a QAP is *symmetric* if the matrix A is symmetric. If B is symmetric, but A is not, then we can switch their roles to get a symmetric problem.

The QAP is a special case of a more general problem.

Definition 1.1.2. Suppose we are given a 4-dimensional array (tensor) $C = \left\{ c_{kl}^{ij} : 1 \leq i, j, k, l \leq n \right\}$ of n^4 real numbers. The general problem is to optimize the function f is defined by:

$$f(\sigma) = \sum_{i,j=1}^{n} c_{\sigma(i)\sigma(j)}^{ij} \tag{1.2}$$

If $c_{kl}^{ij} = a_{ij} b_{kl}$ for some matrices $A = (a_{ij})$ and $B = (b_{kl})$, we get the special case (1.1) we started with. The convenience of working with the generalized problem is that the set of objective functions (1.2) is a vector space.

We say that the generalized QAP defined by $C = (c_{kl}^{ij})$ is *symmetric* if $c_{kl}^{ij} = c_{kl}^{ji}$ for all $i, j, k, l = 1, 2, \ldots, n$.

1.1.1 Motivation for the QAP

Many natural optimization problems can be formulated as QAP's. In fact, the QAP was first posed by the economists Koopmans and Beckmann [KB57]. They consider assigning n facilities to n locations so as to minimize the total cost. Costs arise from transporting materials between facilities; they are proportional to the "flow" of goods a_{ij} required from facility i to facility j and the distances b_{kl} from location k to location l. Then for a given assignment σ of facilities to locations, the total transportation cost is the sum of these costs over all pairs of facilities, equal to the objective function (1.1).

Koopmans and Beckmann also consider a linear cost term d_{ik} for building facility i at location k. Then the Koopmans-Beckmann formulation of the `QAP` objective is:

$$f(\sigma) = \sum_{i,j=1}^{n} b_{\sigma(i)\sigma(j)} a_{ij} + \sum_{i=1}^{n} d_{i\sigma(i)} \tag{1.3}$$

In some circumstances (for example, if the matrix $D = (d_{ij})$ has rank 1, and either $A = (a_{ij})$ or $B = (b_{ij})$ has zero diagonal), the linear term can be absorbed into the quadratic term by modifying the diagonals of A and B. In general, the equation (1.3) can be considered as a special case of the tensor (1.2) by adding the constants d_{ik} to the "diagonal" terms c^{ii}_{kk}.

Lawler [Law63] introduced the generalized objective function (1.2) in the context of management science. This general problem is hard to handle in practice, because a problem of size n is given by a tensor of n^4 numbers, which is expensive to store and manipulate. Since the general problem has not shown a substantial advantage in modeling practical problems, most work is done on the special cases (1.1) and (1.3), which require only $O(n^2)$ storage.

Definition 1.1.3. One of the most prominent problems in computational complexity is the *Traveling Salesman Problem* (`TSP`). For combinatorial optimization, the `TSP` is considered as an *optimization* problem on the the complete graph on n vertices, K_n. A weight is assigned to each edge, and the objective is to find the Hamiltonian cycle (a cycle containing all n vertices) with maximum sum of weights on its edges.

It is natural to describe a cycle on the vertices $\{1, 2, \ldots, n\}$ as a permutation giving the order in which the vertices are visited. This allows us to reduce the symmetric TSP to the (symmetric) QAP. Take:

$$A = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 1 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 0 & \ldots & 1 & 0 \end{pmatrix}, \quad a_{ij} = \begin{cases} 1 & \text{if } |i-j| = 1 \mod n \\ 0 & \text{otherwise} \end{cases}$$

This encodes the cycle $(1, 2, \ldots, n)$. The action of permuting the rows and columns of A by σ permutes the cycle to σ. The edge weights are entered in a symmetric matrix B:

$$b_{ij} = \frac{1}{2}(\text{Weight of edge } \{i, j\})$$

Then $\langle B, \sigma(A) \rangle$ gives the weight of cycle σ.

Similarly, the asymmetric TSP reduces to an asymmetric QAP. Take:

$$A = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 0 & \ldots & 0 & 0 \end{pmatrix}, \quad a_{ij} = \begin{cases} 1 & \text{if } j-i = 1 \mod n \\ 0 & \text{otherwise} \end{cases}$$

Let B the contain the (asymmetric) weights:

$$b_{ij} = \text{Weight of edge } (i, j)$$

Then $\langle B, \sigma(A) \rangle$ gives the weight of cycle σ.

Definition 1.1.4. It is interesting to compare the `QAP` to the *Linear Assignment Problem* (`LAP`) of maximizing:

$$f(\sigma) = \sum_{i=1}^{n} d_{i\sigma(i)} \tag{1.4}$$

This models the problem of assigning n jobs (to be executed in parallel) to n processors, where the cost of running job i on processor j is d_{ij}. In this example, the objective is to minimize the sum of the costs.

The `LAP` is again an optimization problem on the set of permutations. However, there is a well known polynomial time algorithm for the `LAP`, called the *Hungarian Algorithm* described in [Kuh55].

There are three special cases of the `QAP` that reduce to the `LAP`. First, if A in (1.1) has constant columns, then σ acts on A by permuting these columns, and the objective function is:

$$f(\sigma) = \sum_{i,j=1}^{n} a_{ij} b_{\sigma(i)\sigma(j)} = \sum_{i,j=1}^{n} a_{1j} b_{\sigma(i)\sigma(j)}$$

This reduces to the `LAP` (1.4) defined by the matrix $D = (d_{jl})$ where:

$$d_{jl} = \sum_{k=1}^{n} a_{1j} b_{kl}$$

Then the objective functions (1.1) and (1.4) coincide:

$$f(\sigma) = \sum_{i=1}^{n} d_{i\sigma(i)} = \sum_{i,j=1}^{n} a_{ij} b_{\sigma(i)\sigma(j)}$$

Similarly, consider a generalized problem (1.2), where for any k and l the matrix $A = (a_{ij})$ given by $a_{ij} = c_{kl}^{ij}$ has constant columns. Then for any permutation σ, the objective value is:

$$f(\sigma) = \sum_{i,j=1}^{n} c_{\sigma(i)\sigma(j)}^{ij} = \sum_{i,j=1}^{n} c_{\sigma(i)\sigma(j)}^{1j}$$

To reduce to the linear assignment problem, we take the matrix $D = (d_{jl})$ to be defined by:

$$d_{jl} = \sum_{k=1}^{n} c_{kl}^{1j}$$

Now if A in (1.1) has constant rows, then σ acts on A by permuting rows. We reduce to (1.4) by taking $D = (d_{ik})$ given by:

$$d_{ik} = \sum_{l=1}^{n} a_{i1} b_{kl}$$

For the generalized problem (1.2), if for all k and l, the matrix $A = (a_{ij})$ given by $a_{ij} = c_{kl}^{ij}$ has constant rows, we reduce to the `LAP` by defining $D = (d_{ik})$ by:

$$d_{ik} = \sum_{l=1}^{n} c_{kl}^{i1}$$

Finally, if A is a diagonal matrix, we get an `LAP` by defining $D = (d_{ik})$ as:

$$d_{ik} = a_{ii} b_{kk}$$

And if in (1.2), for all k and l, the matrix $A = (a_{ij})$ is diagonal, we reduce to to the `LAP` by defining $D = (d_{ik})$ as:

$$d_{ik} = c_{kk}^{ii}$$

We can separate an arbitrary `QAP` into a linear component and a "pure quadratic" component. Given a problem of type (1.1), we decompose the matrix A into $A = A_1 + A_2$, where A_1 is the projection of A into the vector space spanned by the matrices with constant rows, the matrices with constant columns, and the diagonal matrices. We call the `QAP` defined by (A_1, B) the *linear* part of the problem. If $A = A_1$, then the `QAP` defined by (1.1) can be decomposed into the three linear components described above, and hence reduced to the `LAP`. For a generalized problem of type (1.1) defined by the tensor $C = (c_{kl}^{ij})$, we write $C = C_1 + C_2$, where C_1 is the projection of C into the vector space spanned by tensors where for all k, l the matrices $A = (a_{ij}) = (c_{kl}^{ij})$ have constant columns, tensors where all such A have constant rows, and tensors where all such A are diagonal. We call C_1 the *linear* part of the problem, and note that if $C = C_1$, then the problem (1.2) reduces to the `LAP`. We give formulas for computing A_1 (or C_1) given A (or C) in Section 3.2.3.

The remaining component (A_2, B) for a problem of type (1.1) defines a `QAP`. We call this the *pure* component of the `QAP`. In particular we call a `QAP` *pure* if in the decomposition above we have $A = A_2$. We note that A_2 must have constant row and column sums, and a constant diagonal. Similarly, for the generalized problem (1.2), the component C_2 defines the *pure* component of the `QAP`, and we call a generalized problem *pure* if in the decomposition we have $C = C_2$. For any k, l the matrix $A = (A_{ij}) = (C_2)_{kl}^{ij}$ must have constant row and column sums, and a constant diagonal.

Definition 1.1.5. We introduce the standard Hamming metric on the symmetric group S_n. For two permutations $\tau, \sigma \in S_n$, let the distance $\mathrm{dist}(\sigma, \tau)$ be the number of indices $1 \leq i \leq n$ where σ and τ disagree:

$$\mathrm{dist}(\sigma, \tau) = |i : \sigma(i) \neq \tau(i)|.$$

One can observe that the distance is invariant under the left and right actions of S_n:

$$\mathrm{dist}(\sigma\sigma_1, \sigma\sigma_2) = \mathrm{dist}(\sigma_1, \sigma_2) = \mathrm{dist}(\sigma_1\sigma, \sigma_2\sigma)$$

for all $\sigma_1, \sigma_2, \sigma \in S_n$.

If we are modelling a problem using the `QAP` (such as the facility location example introduced in Section 1.1.1), then permutations are nearby in the Hamming distance if they involve making many of the same assignments. The Hamming distance is exactly the number of assignments that differ between the permutations. In particular, the distance does not depend on the choice of the numbering of the plants from the set $\{1, 2, ..., n\}$, since the distance is invariant under conjugation by any permutation ω:

$$\mathrm{dist}(\sigma, \tau) = \mathrm{dist}(\omega\sigma\omega^{-1}, \omega\tau\omega^{-1})$$

In this case, ω permutes the labels of the plants.

The notion of "nearness" measured by the Hamming distance is appropriate for the context of a heuristic "local search" approach to solving an optimization problem on permutations. We discuss heuristic approaches to the `QAP` in Section 4.2.

There are several other metrics on S_n, see for example chapter 6B of [Dia88]. Most do not make sense for our problem (for example, they change substantially under relabeling). One metric that is quite similar to the Hamming metric is the *Cayley metric* of the (generating) set of all transpositions. This metric counts the minimum number of 2-cycles (transpositions) required to change τ to σ. Our results would not look too different if we used the Cayley metric.

Definition 1.1.6. For a permutation τ and an integer $k > 1$, we consider the "k-th ring" around τ:

$$U(\tau, k) = \big\{\sigma \in S_n : \mathrm{dist}(\sigma, \tau) = n - k\big\}.$$

Hence for any permutation τ the group S_n splits into the disjoint union of n rings $U(\tau, k)$ for $k = 0, 1, 2, \ldots, n-2, n$.

The innermost ring $U(\tau, n)$ contains the single permutation τ, and the next ring $U(\tau, n-2)$ has $\binom{n}{2}$ permutations that differ from τ by a swapping a pair of elements (that is, by a 2-cycle). The size of the ring $U(\tau, k)$ increases as k decreases, and most of the permutations are contained in the outermost rings. In Lemma 3.4.2, we show that the number of permutations in the outermost ring $U(\tau, 0)$ is at least $\frac{n!}{3}$ for

$n \geq 2$. Using this fact, it is easy to show that the number of permutations in the ring $U(\tau, k)$ is between $\frac{n!}{3k!}$ and $\frac{n!}{k!}$, see the proof of Lemma 3.4.3.

Definition 1.1.7. Let $f : S_n \longrightarrow \mathbb{R}$ be a function of type (1.1) or (1.2). Let

$$\overline{f} = \frac{1}{n!} \sum_{\sigma \in S_n} f(\sigma)$$

be the average value of f on the symmetric group and let

$$f_0 = f - \overline{f}$$

be the "shifted" function. Hence the average value of f_0 is 0. Let τ be a permutation where the maximum value of f_0 is attained, so $f_0(\tau) \geq f_0(\sigma)$ for all $\sigma \in S_n$ and $f_0(\tau) > 0$ unless $f_0 \equiv 0$ (the problem with minimum instead of maximum is completely similar).

We remark that it is easy to compute the average value $\overline{f}$ (see Lemma 3.1.1).

1.1.2 Objectives

We are interested in the following two types of questions:

i. What fraction of the objective values are relatively large (close to optimal)? Specifically, given a constant $0 < \gamma < 1$ possibly depending on n, how many permutations $\sigma \in S_n$ satisfy $f_0(\sigma) \geq \gamma f_0(\tau)$?

ii. Where are other relatively large values located with respect to the optimum? Specifically, how does the average value of f_0 over the k-th ring $U(\tau, k)$ compare with the optimal value $f_0(\tau)$? In particular, is a random permutation from the vicinity of the optimal permutation better than a random permutation from the whole group S_n?

The answer to the first question tells us how well the sample optimum of a set of randomly chosen permutations approximates the true optimum. The answer to the second question gives us information on how we can understand the problem heuristically – whether to look for large values near other large values, or to sample permutations as broadly as possible.

Definition 1.1.8. We introduce a function $\nu(m)$ that occurs in our results. For an integer $k \geq 0$, let

$$d_k = \sum_{j=0}^{k} (-1)^j \frac{1}{j!}.$$

For $k \geq 2$ let

$$\nu(m) = \frac{d_{m-2}}{d_m}.$$

It is convenient to agree that $\nu(0) = 0$ (and $\nu(1)$ is not defined). One can see that

$$\lim_{m \longrightarrow +\infty} \nu(m) = 1$$

and that $0 \leq \nu(m) < 1$ if m is odd and $1 < \nu(m) \leq 2$ if $m \geq 2$ is even.

Definition 1.1.9. The following functions p and t on the symmetric group S_n play a special role in our approach. For a permutation $\sigma \in S_n$, let p be the number of *fixed points* of σ:

$$p(\sigma) = \left| i : \sigma(i) = i \right|$$

and let t be the number of *transpositions* (or 2-cycles) in σ:

$$t(\sigma) = \left| i < j : \sigma(i) = j \quad \text{and} \quad \sigma(j) = i \right|$$

One can show that $p(\sigma)$, $p^2(\sigma)$ and $t(\sigma)$ are functions of type (1.2) for some particular tensors $\{c_{kl}^{ij}\}$, see Remark 3.2.5.

In the next two sections, we highlight some of the work that has been done on the

TSP and QAP. We mention both theoretical approximation results (guarantees and hardness) and empirical data.

1.2 Remarks on the TSP

The TSP is one of the most studied NP-complete problems, due to its long history, frequent occurrence in practical situations, and accessibility to non-specialists. Some notes on the early history of the TSP are found in [ABCC98]. The NP-completeness of the TSP decision problem was shown in Karp's famous paper on reducibility [Kar72].

The important ideas in complexity theory of reductions and completeness, were originally used (for example in Karp's paper) in the context of "decision problems", that is problems with a 'yes' or 'no' answer. The TSP *decision* problem is to determine if a given graph has any Hamiltonian cycle. We are interested in the TSP *optimization* problem of finding a cycle of maximum (or minimum) weight. For the optimization problem, we can assume that weights are given on the complete graph, K_n.

1.2.1 Complexity of Optimization Problems

The ideas of reductions and completeness can be extended from decision problems to optimization problems, see for example [ACG$^+$99]. More detailed complexity information is available from optimization problems than from the corresponding decision problems. In particular, optimization problems can be compared with respect to the degree of *approximation* possible in polynomial time. A variety of behaviors are observed among optimization problem whose decision problems are NP-complete, including both positive approximation results and negative completeness results.

However, it is tricky to work with optimization problems because they are sensitive to formulation. A given decision problem may have several natural analogous optimization problems with very different behaviors with respect to approximation.

Consider the case of the TSP. An approximation algorithm will return a permutation σ which we expect to have objective value comparable to that of the optimal permutation τ, that is, the ratio $\frac{f(\sigma)}{f(\tau)}$ should be close to one. To ensure that this ratio is meaningful, we can not allow arbitrary edge weights, because the optimal value could be zero (for example).

One way to fix this is to assume that all the weights are positive. Then the TSP *minimization* problem is *NPO-complete*, that is any polynomial time algorithm guaranteeing a value σ with $\frac{f(\sigma)}{f(\tau)} \leq 2^{-n^\kappa}$, for some $\kappa > 0$ implies $P = NP$ [ACG+99]. This is the strongest type of non-approximability result.

In contrast, the (asymmetric) TSP *maximization* problem with positive weights admits a polynomial time constant approximation guaranteeing $\frac{f(\sigma)}{f(\tau)} \geq \frac{8}{13}(1 - \frac{1}{n})$ where n is the size of the problem [Blä02]. If the TSP is symmetric, this can be improved to 3/4 [Ser84], and there is a randomized algorithm which returns a solution with expected ratio r for any ratio $r < 25/33$ [HR00]. Reportedly, these ratios have recently been improved to 5/8 [LS] in the asymmetric case, and 7/8 in the symmetric case.

1.2.2 Approximation With Respect to the Average

Our approach is to look at approximation results with respect to the average function value over all permutations. In the cases of TSP and QAP, it is easy to compute the average over all permutations, see Lemma 3.1.1. Equivalently, we can consider a normalized maximization problem with average value on all permutations of 0. In this case some edge weights will be negative.

Our results show that this zero average problem is not as difficult as the positive weights minimization problem. At the same time, it does not appear to be as easy as the positive weight maximization problem, where greedy algorithms can already give a good approximation.

1.2.3 Empirical Results

There is a large body of empirical results on the `TSP`. These consist of difficult `TSP` instances that have been proposed and provably solved (or not) to optimality. Many problems of this type are two-dimensional Euclidean problems, where the input is a set of points in the plane. These problems can be attacked very effectively with geometric heuristics, and it is now common to see problems on more than 10,000 points solved. See for example [ABCC01].

In fact, Arora has shown [Aro98] that a grid-based algorithm can be used to construct a polynomial time approximation scheme (PTAS) for the Euclidean `TSP` in fixed dimension. A PTAS gives, for any $\epsilon > 0$, some polynomial time algorithm (with running time depending on ϵ) which guarantees a $(1-\epsilon)$ approximation for the minimum.

Less empirical work has been done on non-Euclidean and asymmetric instances of the `TSP`. An interesting survey of asymmetric `TSP`'s drawn from various sources is [MP91]. In these experiments, various asymmetric `TSP`'s are solved, some random, some structured. The authors report success in solving randomly generated problems on thousands of nodes using branch-and-bound heuristics, as well as some structured problems of similar sizes.

More recently [CJMZ01] studied heuristics on a variety of difficult (some random, some structured) asymmetric `TSP` instances. They routinely solve problems on 100 nodes, and are usually able to solve problems of size 1000.

1.2.4 Non-polynomial Algorithms

To help understand the complexity of problems, we can also consider the performance of non-polynomial algorithms. The `TSP` can be solved in time $O(n^2 2^n)$ by dynamic programming. This is not polynomial, but is substantially faster than the $O(n!)$ required to enumerate the solutions.

1.3 Remarks on the `QAP`

Like the `TSP`, the `QAP` is known as one of the most difficult problems in combinatorial optimization. A good recent survey of the `QAP` literature is [BÇPP99].

1.3.1 Complexity Results for `QAP`

We noted in Section 1.2 that the `QAP` generalizes the `TSP`. Hence `QAP` is at least as hard as `TSP`, and the hardness results we have for `TSP` apply to `QAP`. For example, minimizing the `QAP` with positive weights is NPO-complete [ACG$^+$99].

In fact, the `QAP` appears much more difficult than the `TSP` both theoretically and practically. There are few positive approximation results for special cases of the `QAP`.

For the problem of maximizing a `QAP` with positive coefficients, we can get a trivial approximation guarantee of $1/n^2$ by picking a permutation which includes the largest possible single term in the objective function. Specifically, we take i', j', k', l' so that:

$$a_{i'j'}b_{k'l'} = \max\left\{\left(\max_{i\neq j} a_{ij}\right)\left(\max_{k\neq l} b_{kl}\right), \left(\max_{i} a_{ii}\right)\left(\max_{k} b_{kk}\right)\right\}$$

We can choose i', j', k', l' so that either $i' \neq j'$ and $k' \neq l'$ or $i' = j'$ and $k' = l'$. Then for any permutation σ such that $\sigma(i') = k'$, and if $i' \neq j'$, such that $\sigma(j') = l'$, we have $f(\sigma) \geq f(\tau)/n^2$. By fixing more terms, this guarantee can be improved by a constant factor (at the price of a worse polynomial running time), but we know of no algorithm in the literature that improves on this guarantee for all QAP's with positive weights.

Arkin, Hassin and Sviridenko [AHS00] consider the problem of maximizing the QAP where one of the matrices is assumed to satisfy the triangle inequality, and the other is non-negative (analogous to a metric TSP). They get a $\frac{1}{4}$-approximation guarantee. It is NP-complete to get a constant approximation for the minimization version of this problem [Que86].

Ye [Ye99] gives an algorithm using semi-definite programming that gives a $\dfrac{1}{n^2\ln(4n^4)}$ approximation guarantee for maximizing (1.2), when the tensor $C = \left\{c_{kl}^{ij} : 1 \leq i,j,k,l \leq n\right\}$ is positive semi-definite when considered as an $n^2 \times n^2$ matrix.

There is another notion of a "good" approximation called the domination number. The *domination number* of a permutation σ is the number of permutations whose objective value is at most $f(\sigma)$. For example, if the objective function is maximized at τ, then τ has domination number $n!$. The problem of finding a permutation with good domination number for the QAP is examined in [GY02]. They show that at least in the case where n is a prime power, there is a polynomial algorithm that returns a permutation whose domination number is at least $(n-2)!$. Additionally, they show that such an algorithm exists for TSP for all n. We discuss their algorithm in Section 4.4.

1.3.2 Empirical Results

The contrast between QAP and TSP in empirical tests is even sharper. A problem library has been developed for QAP, called *QAPLIB* [BKR97]. Most of the problems in the QAPLIB have a simple structure, say that the distance matrix is obtained from an arrangement of points in the plane, and some component that is either random or drawn from applied data. Despite considerable effort, it has proved difficult to solve instances of size $n = 20$, and solutions of problems of size $n = 30$ are considered remarkable. See for example [ABGL02] and [BMCP98].

The heuristics that are effective in solving moderately sized `TSP`'s do not work as well on `QAP`'s. One of the goals of this thesis is to provide some explanation of why this is, and to distinguish the cases where we might expect heuristics to perform well.

In [AZ01], the authors compare the `QAP` and the `TSP` via a "ruggedness coefficient" which describes a type of local variability. They take the view that problems with lower local variability are easier for local search. They show that the ruggedness of the `QAP` lies between the ruggedness of the `TAP` (which they consider an easy problem for local search) and the ruggedness of the binary string problem, which they take as an example of a very hard problem for local search.

1.3.3 Non-polynomial Algorithms

No counterpart of the dynamic programming algorithm that solves `TSP` is known for the `QAP`, and, in fact, we know of no algorithm that solves the `QAP` essentially faster than $O(n!)$, the time required to enumerate the feasible solutions.

CHAPTER II

Summary of Results

In this chapter we state our main results on the distributions of the `QAP` and the `TSP`. The proofs are presented in Chapter III.

The results are divided into four cases, which arise from considering the representations of Mat_n under the action of S_n (see Section 3.2). We begin with the most special case, which we call the "bullseye", and which already includes the symmetric `TSP`. We then consider two different generalizations of the bullseye – the "pure" `QAP` and the symmetric `QAP`. Finally, we consider the general `QAP`.

For each case, we bound the frequency of relatively large values, and describe the possible locations of relatively large values with respect to the optimum. The frequency bounds allow us to analyze how well we can approximate the optimum by taking the best value from a random set of points. We state algorithmic results of this type for the "pure" and general cases. The location of large values helps in designing and understand heuristics for the `QAP`, and is discussed further in Section 4.2

2.1 Bullseye Special Case

Suppose that the matrix $A = (a_{ij})$ in (1.1) is symmetric and has constant row and column sums and a constant diagonal:

$$
\begin{aligned}
a_{ij} = a_{ji} \qquad & \text{for all} \quad 1 \le i, j \le n; \\
& \text{for some} \quad a \\
\sum_{i=1}^{n} a_{ij} = a \qquad & \text{for all} \quad j = 1, \dots, n \quad \text{and} \\
\sum_{j=1}^{n} a_{ij} = a \qquad & \text{for all} \quad i = 1, \dots, n; \\
a_{ii} = b \qquad & \text{for some} \quad b \quad \text{and all} \quad i = 1, \dots, n.
\end{aligned}
$$

In particular, we noted in Section 1.1, that this case includes the symmetric traveling salesman problem (TSP).

Similarly, for the generalized problem (1.2) we assume that for any k and l the matrix $A = (a_{ij})$, where $a_{ij} = c_{kl}^{ij}$, is symmetric with constant row and column sums and a constant diagonal. Retaining the assumption that A is symmetric for all k, l, we can weaken the remaining assumptions to the following $(n-1)^2$ linear conditions:

$$
\begin{aligned}
& (n-2)(c_{11}^{11} + c_{kk}^{ii}) + \sum_{j\neq 1} \sum_{l \neq 1} (c_{1l}^{1j} + c_{l1}^{j1}) + \sum_{j \neq i} \sum_{l \neq k} (c_{kl}^{ij} + c_{lk}^{ji}) \\
= \quad & (n-2)(c_{kk}^{11} + c_{11}^{ii}) + \sum_{j\neq 1} \sum_{l \neq k} (c_{kl}^{1j} + c_{lk}^{j1}) + \sum_{j \neq i} \sum_{l \neq 1} (c_{1l}^{ij} + c_{l1}^{ji}) \qquad (2.1) \\
& \text{for all } 2 \le i, k \le n
\end{aligned}
$$

It turns out that the optimum has a characteristic "bullseye" feature in the Hamming metric on S_n (see Definition 1.1.5). This is illustrated in Figure 2.1.

Theorem 2.1.1 (Bullseye Distribution). *Let*

$$
\alpha(n, k) = \frac{k^2 - 3k + \nu(n-k)}{n^2 - 3n},
$$

where $k = 0, 1, \dots, n-2, n$ and ν is the function of Definition 1.1.8. Then we have

$$
\frac{1}{|U(\tau, k)|} \sum_{\sigma \in U(\tau, k)} f_0(\sigma) = \alpha(n, k) f_0(\tau)
$$

for $k = 0, 1, \dots, n-2, n$.

We observe that as the ring $U(\tau, k)$ contracts to the optimal permutation τ (hence k increases), the *average* value of f on the ring steadily improves (as long as $k \geq 3$).

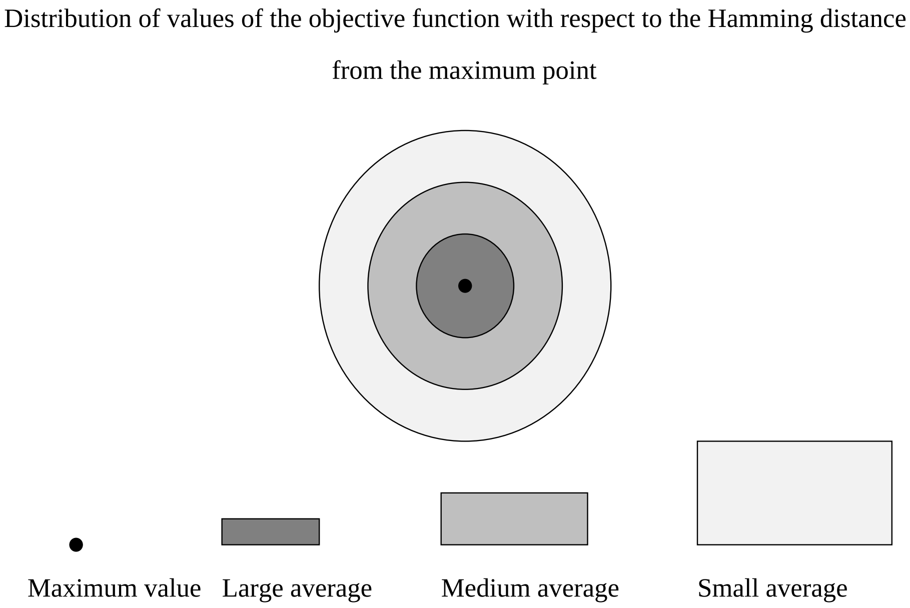

Figure 2.1: Bullseye Distribution

It is easy to construct examples where *some* values of f in a very small neighborhood of the optimum are particularly bad, but as follows from Theorem 2.1.1, such values are relatively rare.

Estimating the size of the ring $U(\tau, k)$, we get the following result.

Theorem 2.1.2 (Frequency of Large Values for Bullseye QAP). *Let us choose an integer $3 \leq k \leq n-3$ and a number $0 < \gamma < 1$ and let*

$$\beta(n, k) = \frac{k^2 - 3k}{n^2 - 3n}.$$

The probability that a random permutation $\sigma \in S_n$ satisfies the inequality

$$f_0(\sigma) \geq \gamma \beta(n, k) f_0(\tau)$$

is at least

$$\frac{(1-\gamma)\beta(n,k)}{3k!}.$$

2.2 Pure Special Case

In this section, we consider a more general case of a not necessarily symmetric matrix $A = (a_{ij})$ in (1.1) having constant row and column sums and a constant diagonal:

$$\begin{aligned} & && \text{for some} \quad a \\ \sum_{i=1}^{n} a_{ij} &= a && \text{for all} \quad j = 1, \dots, n \quad \text{and} \\ \sum_{j=1}^{n} a_{ij} &= a && \text{for all} \quad i = 1, \dots, n; \\ a_{ii} &= b && \text{for some} \quad b \quad \text{and all} \quad i = 1, \dots, n. \end{aligned}$$

We noted in Section 1.1 that this case includes the Asymmetric Traveling Salesman Problem (`ATSP`).

Similarly, for generalized problems (1.2) we assume that for any k and l the matrix $A = (a_{ij})$, where $a_{ij} = c_{kl}^{ij}$ has constant row and column sums and has a constant diagonal. These conditions can be relaxed to the following $(n-1)^2$ linear conditions on C:

$$\begin{aligned} n(n-2)(c_{11}^{11} + c_{kk}^{ii}) \quad & + \sum_{j\neq 1}\sum_{l\neq 1}\left((n-1)(c_{1l}^{1j} + c_{l1}^{j1}) + c_{l1}^{1j} + c_{1l}^{j1}\right) \\ & + \sum_{j\neq i}\sum_{l\neq k}\left((n-1)(c_{kl}^{ij} + c_{lk}^{ji}) + c_{lk}^{ij} + c_{kl}^{ji}\right) \\ = n(n-2)(c_{kk}^{11} + c_{11}^{ii}) \quad & + \sum_{j\neq 1}\sum_{l\neq k}\left((n-1)(c_{kl}^{1j} + c_{lk}^{j1}) + c_{lk}^{1j} + c_{kl}^{j1}\right) \\ & + \sum_{j\neq i}\sum_{l\neq 1}\left((n-1)(c_{1l}^{ij} + c_{l1}^{ji}) + c_{l1}^{ij} + c_{1l}^{ji}\right) \\ & \text{for all } 2 \leq i, k \leq n \end{aligned} \tag{2.2}$$

We call this case pure because the objective function f lacks the linear component (see Definition 1.1.4). In the terminology of the Koopmans-Beckmann problem (see Section 1.3), the total flow of goods (both in and out) is constant across facilities, and the cost of building the facilities does not depend on the location.

The behavior of averages of f_0 over the rings $U(\tau, k)$ is described by the following result.

Theorem 2.2.1 (Pure Distribution).

Let us define three functions of n and k:

$$\alpha_1(n,k) = 1 - \nu(n-k)$$
$$\alpha_{2e}(n,k) = \frac{k^2 - 3k - n - 3\nu(n-k) + \nu(n-k)n + 4}{n^2 - 4n + 4} \quad \textit{and}$$
$$\alpha_{2o}(n,k) = \frac{k^2 - 3k - n - 2\nu(n-k) + \nu(n-k)n + 3}{n^2 - 4n + 3},$$

where $k = 0, 1, \ldots, n-2, n$ and ν is the function of Definition 1.1.8.

If n is even, then for some non-negative γ_1 and γ_2 such that $\gamma_1 + \gamma_2 = 1$ we have

$$\frac{1}{|U(\tau,k)|} \sum_{\sigma \in U(\tau,k)} f_0(\sigma) = \Big(\gamma_1 \alpha_1(n,k) + \gamma_2 \alpha_{2e}(n,k)\Big) f_0(\tau)$$

for $k = 0, 1, \ldots, n-2, n$.

If n is odd, then for some non-negative γ_1 and γ_2 such that $\gamma_1 + \gamma_2 = 1$ we have

$$\frac{1}{|U(\tau,k)|} \sum_{\sigma \in U(\tau,k)} f_0(\sigma) = \Big(\gamma_1 \alpha_1(n,k) + \gamma_2 \alpha_{2o}(n,k)\Big) f_0(\tau)$$

for $k = 0, 1, \ldots, n-2, n$.

We show in Section 3.9 that, at least for even n, for any choice of $\gamma_1, \gamma_2 \geq 0$ such that $\gamma_1 + \gamma_2 = 1$ there is a function f of type (1.1) for which the averages of f_0 over $U(\tau, k)$ are given by the formulas of Theorem 2.2.1.

2.2.1 Damped Oscillator

We observe that there are two extreme cases. If $\gamma_1 = 0$ and $\gamma_2 = 1$ then f exhibits a bullseye type distribution as in Section 2.1. If $\gamma_1 = 1$ and $\gamma_2 = 0$ then f exhibits a “damped oscillator” type of distribution: the average value of f_0 over $U(\tau, k)$ changes its sign with the parity of k and approaches 0 fast as k shrinks. In short, if f has a damped oscillator distribution, there is no particular advantage in choosing a permutation in the vicinity of the optimal permutation τ, see Figure 2.2.

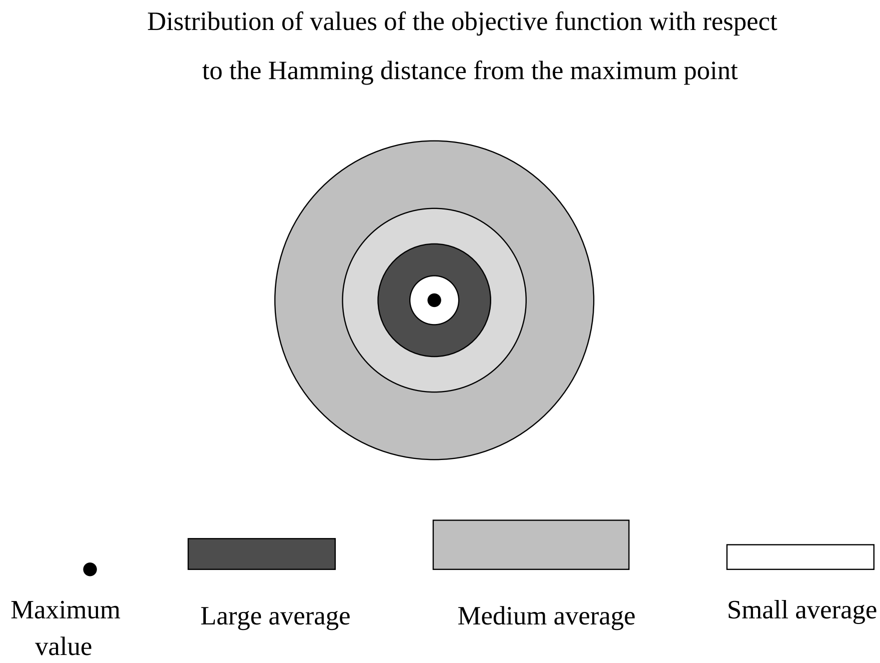

Figure 2.2: Damped Oscillator Distribution

For a typical function f one can expect both γ_1 and γ_2 positive, so f would show a “weak” bullseye distribution: the average value of f_0 over $U(\tau, k)$ improves moderately as k gets smaller, but not as dramatically as in the bullseye case of Section 2.1. Still, it turns out that we can find sufficiently many reasonably good permutations in the vicinity of the optimal permutation (see Remark 3.6.3).

Theorem 2.2.2 (Frequency of Large Values for Pure QAP).

Let us choose an integer $3 \leq k \leq n-3$ and a number $0 < \gamma < 1$ and let

$$\beta(n,k) = \frac{k^2 - 3k + 1}{n^2 - 3n + 1}.$$

The probability that a random permutation $\sigma \in S_n$ satisfies the inequality

$$f_0(\sigma) \geq \gamma \beta(n,k) f_0(\tau)$$

is at least

$$\frac{(1-\gamma)\beta(n,k)}{10k!}.$$

In particular, by choosing an appropriate k, we obtain the following corollary.

Corollary 2.2.3.

i. *Let us fix any $\alpha > 1$. Then there exists a $\delta = \delta(\alpha) > 0$ such that for all sufficiently large $n \geq N(\alpha)$ the probability that a random permutation σ in S_n satisfies the inequality*

$$f_0(\sigma) \geq \frac{\alpha}{n^2} f_0(\tau)$$

is at least δn^{-2}. In particular, one can choose $\delta = \exp\left\{-c\sqrt{\alpha}\ln\alpha\right\}$ for some absolute constant $c > 0$.

ii. *Let us fix any $\epsilon > 0$. Then there exists a $\delta = \delta(\epsilon) < 1$ such that for all sufficiently large $n \geq N(\alpha)$ the probability that a random permutation σ in S_n satisfies the inequality*

$$f_0(\sigma) \geq n^{-\epsilon} f_0(\tau)$$

is at least $\exp\{-n^\delta\}$. In particular, one can choose any $\delta > 1 - \epsilon/2$.

It follows from Corollary 2.2.3 that to get a permutation σ which satisfies (i) for any fixed α, we can use the following straightforward randomized algorithm: sample $O(n^2)$ random permutations $\sigma \in S_n$, compute the value of f and choose the best permutation. With the probability which tends to 1 as $n \longrightarrow +\infty$, we will hit the right permutation. The complexity of the algorithm is quadratic in n for any α, but the coefficient of n^2 grows as α grows. If we are willing to settle for an algorithm of a mildly exponential complexity of the type $\exp\{n^\beta\}$ for some $\beta < 1$ we can achieve a better approximation (ii) by searching through the set of randomly selected $\exp\{n^\beta\}$ permutations. We remarked in Chapter I that no algorithm solving the Quadratic Assignment Problem (even in the special case considered in this section) with an exponential in n complexity $\exp\{O(n)\}$ is known, although such an algorithm does exist for the Traveling Salesman Problem.

We prove these results in Section 3.6.

2.3 General Symmetric Case

In this section, we assume that the matrix $A = (a_{ij})$ in (1.1) is symmetric, that is:

$$a_{ij} = a_{ji} \quad \text{for all} \quad 1 \leq i, j \leq n.$$

In fact, it is sufficient for A to be the sum of a symmetric matrix, a matrix with constant rows, and a matrix with constant columns.

Similarly, in the generalized problem (1.2) we assume that for any $k \neq l$, the matrix $A = (a_{ij})$, where $a_{ij} = c^{ij}_{kl}$, is the sum of a symmetric matrix, a matrix with constant rows, and a matrix with constant columns. We can weaken this assumption slightly to having C satisfy the following $\frac{1}{2}(n^4 - 4n^3 + 5n^2 - 2n)$ linear conditions:

$$\begin{aligned} & nc_{12}^{ij} - nc_{12}^{ji} - \sum_{i'=1}^{n} c_{12}^{i'j} + \sum_{i'=1}^{n} c_{12}^{ji'} - \sum_{j'=1}^{n} c_{12}^{ij'} + \sum_{j'=1}^{n} c_{12}^{j'i} \\ & = nc_{kl}^{ij} - nc_{kl}^{ji} - \sum_{i'=1}^{n} c_{kl}^{i'j} + \sum_{i'=1}^{n} c_{kl}^{ji'} - \sum_{j'=1}^{n} c_{kl}^{ij'} + \sum_{j'=1}^{n} c_{kl}^{j'i} \qquad (2.3) \\ & \text{for all } 1 \leq i < j < n \text{ and } k \neq l \end{aligned}$$

Overall, the distribution of values of f turns out to be much more complicated than in the special cases described in Sections 2.1 and 2.2.

Theorem 2.3.1 (Symmetric Distribution).

Let us define three functions of n and k:

$$\alpha_1(n,k) = \frac{2nk - 2n - k^2 - 3k - \nu(n-k) + 6}{n^2 - 5n + 6}$$

$$\alpha_{2e}(n,k) = \frac{-nk + n + k^2 + k + \nu(n-k) - 4}{2n - 4} \quad \textit{and}$$

$$\alpha_{2o}(n,k) = \frac{-n^2k + nk^2 + n^2 + nk + n\nu(n-k) - 4n - 3k + 3}{2n^2 - 7n + 3},$$

where $k = 0, 1, \ldots, n-2, n$ and ν is the function of Definition 1.1.8.

If n is even, then for some non-negative γ_1 and γ_2 such that $\gamma_1 + \gamma_2 = 1$ we have

$$\frac{1}{|U(\tau,k)|} \sum_{\sigma \in U(\tau,k)} f_0(\sigma) = \Big(\gamma_1 \alpha_1(n,k) + \gamma_2 \alpha_{2e}(n,k)\Big) f_0(\tau)$$

for $k = 0, 1, \ldots, n-2, n$.

If n is odd, then for some non-negative γ_1 and γ_2 such that $\gamma_1 + \gamma_2 = 1$ we have

$$\frac{1}{|U(\tau,k)|} \sum_{\sigma \in U(\tau,k)} f_0(\sigma) = \Big(\gamma_1 \alpha_1(n,k) + \gamma_2 \alpha_{2o}(n,k)\Big) f_0(\tau)$$

for $k = 0, 1, \ldots, n-2, n$.

It follows from our proof (see Section 3.7) that for any choice of $\gamma_1, \gamma_2 \geq 0$ such that $\gamma_1 + \gamma_2 = 1$ there is a function f of type (1.2) for which that averages of f_0 over $U(\tau,k)$ are given by the formulas of Theorem 4.1. Moreover, at least if n is even, one can choose f to be a function of type (1.1), see Section 3.9.

2.3.1 The "Spike" Distribution

As in Section 2.2, we see that there are two extreme cases. If $\gamma_1 = 1$ and $\gamma_2 = 0$ then f has a bullseye type distribution as in Section 2.1. If $\gamma_1 = 0$ and $\gamma_2 = 1$ then f has what we call a "spike" distribution, see Figure 2.3. In this case, for $3 \le k \le$

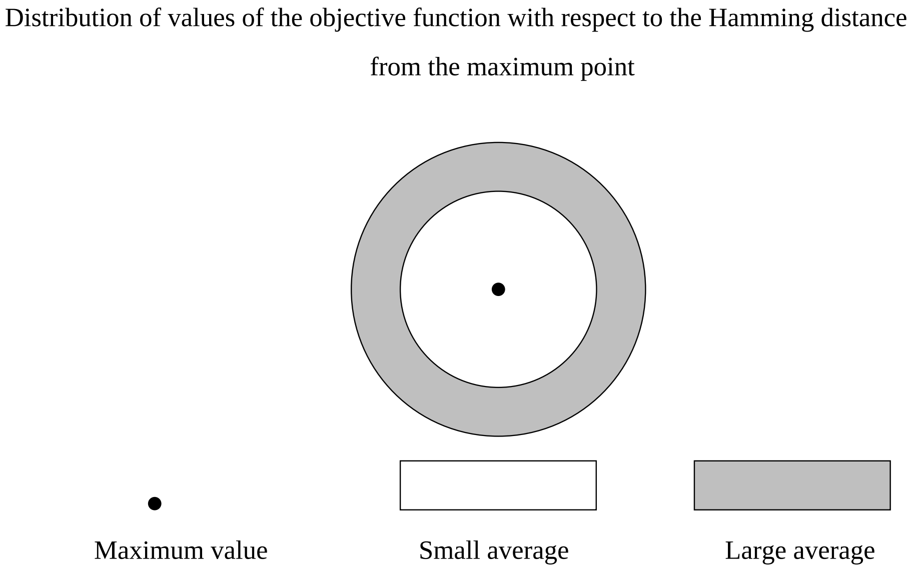

Figure 2.3: Spike Distribution

$n-3$ the average value of f_0 over $U(\tau, k)$ is negative. Thus an average permutation $\sigma \in U(\tau, n-3)$ presents us with a worse choice than the average permutation in S_n. However, the average value of f_0 over $U(\tau, 0)$ is about one half of the maximum value $f_0(\tau)$. Thus there are plenty of reasonably good permutations very far from τ and we can easily get such a permutation by random sampling.

We obtain the following estimate for the number of near-optimal permutations.

Theorem 2.3.2 (Frequency of Large Values for Symmetric QAP). *Let us choose an integer $3 \le k \le n-3$ and a number $0 < \gamma < 1$ and let*

$$\beta(n, k) = \frac{3k-5}{n^2 - kn + k + 2n - 5}.$$

The probability that a random permutation $\sigma \in S_n$ satisfies the inequality

$$f_0(\sigma) \geq \gamma\beta(n,k) f_0(\tau)$$

is at least

$$\frac{(1-\gamma)\beta(n,k)}{5k!2^k}.$$

One can notice that the obtained bound is essentially weaker than the bounds of Theorems 2.1.2 and 2.2.2. We give an example below showing that at least for the generalized problem (1.2), the bounds of Sections 2.1 and 2.2 and Corollary 2.2.3, in particular, cannot hold true. The question whether the estimates can be improved for the problem (1.1) remains open.

2.3.2 Scarcity of Large Values for Symmetric QAP

Let us fix any $0 < \delta < 1$ and any $0 < \epsilon < 1 - \delta$. Let us choose an integer m such that $n^{1-\epsilon} > m > n^{\delta}$. In Remark 3.2.5, we show that there exists a tensor c_{kl}^{ij} such that $c_{kl}^{ij} = c_{kl}^{ji}$ for all k and l and such that for the function f defined by (1.2) we have

$$f(\sigma) = \frac{p^2(\sigma) - mp(\sigma) + p(\sigma) + 2t(\sigma) + m - 5}{n^2 - nm + 2n + m - 5},$$

where $p(\sigma)$ is the number of fixed points in σ and $t(\sigma)$ is the number of 2-cycles in σ, see Definition 1.1.9. We show that $\overline{f} = 0$ and that $f(\varepsilon) = 1$ is the maximum value of f, where ε is the identity permutation.

Then, for all sufficiently large n, the value $f(\sigma) > 2/n$ can be achieved only on permutations σ with $p(\sigma) > m$. The number of such permutations σ does not exceed $\binom{n}{m}(n-m)! = n!/m!$. That is, the probability that a random permutation σ satisfies $f(\sigma) > 2/n$ does not exceed $\exp\{-n^{\delta}\}$ for large n. So, at least for the generalized problem (1.2), the estimate of Corollary 2.2.3 part (ii) does not carry over from the pure special case to the symmetric special case for $\epsilon \leq 1$.

2.4 General Case

It appears that the difference between the general case of problems (1.1) and (1.2) and the symmetric case of Section 2.3 is not as substantial as the difference between the symmetric case and the special cases of Sections 2.1 and 2.2.

First, we describe the behavior of averages of f_0 over the rings $U(\tau, k)$.

Theorem 2.4.1 (General Distribution).

Let us define five functions of n and k:

$$\begin{aligned}
\alpha_1(n,k) &= \frac{-k(n-k)+n-2}{n-2},\\
\alpha_2(n,k) &= 1-\nu(n-k),\\
\alpha_3(n,k) &= \frac{2nk-3k-2n-k^2-\nu(n-k)+6}{n^2-5n+6},\\
\alpha_4(n,k) &= \frac{k+\nu(n-k)-2}{n-2} \quad \text{and}\\
\alpha_{5o}(n,k) &= \frac{-2nk+3k^2-3k+\nu(n-k)n+n-3}{n^2-2n-3},
\end{aligned}$$

where $k = 0, 1, \ldots, n-2, n$ and $\nu(k)$ is the function of Definition 1.1.8.

If n is even, then for some non-negative $\gamma_1, \gamma_2, \gamma_3$ and γ_4 such that $\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 = 1$ we have

$$\begin{aligned}
&\frac{1}{|U(\tau,k)|}\sum_{\sigma\in U(\tau,k)} f_0(\sigma)\\
&= \Big(\gamma_1\alpha_1(n,k)+\gamma_2\alpha_2(n,k)+\gamma_3\alpha_3(n,k)+\gamma_4\alpha_4(n,k)\Big) f_0(\tau)
\end{aligned}$$

for $k = 0, 1, \ldots, n-2, n$.

If n is odd, then for some non-negative $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ and γ_5 such that $\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5 = 1$ we have

$$\begin{aligned}
&\frac{1}{|U(\tau,k)|}\sum_{\sigma\in U(\tau,k)} f_0(\sigma)\\
&= \Big(\gamma_1\alpha_1(n,k)+\gamma_2\alpha_2(n,k)+\gamma_3\alpha_3(n,k)+\gamma_4\alpha_4(n,k)+\gamma_5\alpha_{5o}(n,k)\Big) f_0(\tau)
\end{aligned}$$

for $k = 0, 1, \ldots, n-2, n$.

It follows from our proof (see Section 3.8) that for any choice of $\gamma_1, \ldots, \gamma_4 \geq 0$ (n even) or $\gamma_1, \ldots, \gamma_5 \geq 0$ (n odd) summing up to 1 there is a function f of type (1.2) for which the averages of f_0 over $U(\tau, k)$ are given by the formulas of Theorem 2.4.1. Moreover, at least for combinations of $\gamma_1, \ldots, \gamma_4 \geq 0$, one can choose f to be a function of type (1.1), see Section 3.9.

As we let all but one γ equal to 0, we obtain various extreme distributions: the bullseye (when $\gamma_3 = 1$ or $\gamma_4 = 1$, cf. Section 2.1), damped oscillator (when $\gamma_2 = 1$, cf. Section 2.2) and spike (when $\gamma_1 = 1$ or $\gamma_{5o} = 1$, cf. Section 2.3) types. We do not get any new type of a distribution, but we can find a sharper spike than in the symmetric case.

2.4.1 The Sharp Spike Distribution

Let us consider the function

$$f = \frac{-p(n-p) + n - 2}{n-2},$$

where $p(\sigma)$ is the number of fixed points of σ, see Section 1.1. It follows from Remark 3.2.5 that f is a function of type (1.2) and that $\overline{f} = 0$. The maximum value of 1 is attained at the identity and at the permutations without fixed points. All other values of f are negative. The sharp spike is illustrated in Figure 2.4. In Section 3.9 we construct a function of type (1.1) whose average values over the rings $U(\varepsilon, k)$, where ε is the identity permutation, coincide with those for f.

Our bound for the number of nearly optimal permutations is only slightly weaker than the bound of Theorem 2.3.2 in the symmetric case.

Theorem 2.4.2 (Frequency of Large Values for General `QAP`).

Let us choose an integer $3 \leq k \leq n-3$ and a number $0 < \gamma < 1$ and let

$$\beta(n,k) = \frac{k-2}{n^2 - nk + k - 2}.$$

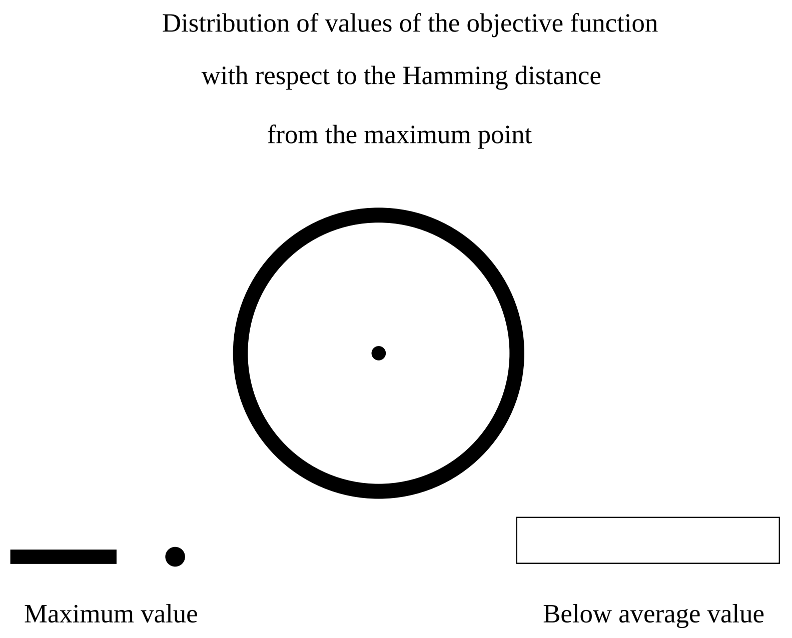

Figure 2.4: Sharp Spike

The probability that a random permutation $\sigma \in S_n$ *satisfies*

$$f_0(\sigma) \geq \gamma\beta(k,n) f_0(\tau)$$

is at least

$$\frac{(1-\gamma)\beta(k,n)}{5k!}.$$

By choosing an appropriate k, we obtain the following corollary.

Corollary 2.4.3.

i. Let us fix any $\alpha > 1$. *Then there exists a* $\delta = \delta(\alpha) > 0$ *such that for all sufficiently large* $n \geq N(\alpha)$ *the probability that a random permutation* σ *in* S_n *satisfies the inequality*

$$f_0(\sigma) \geq \frac{\alpha}{n^2} f_0(\tau)$$

is at least δn^{-2}. *In particular, one can choose* $\delta = \exp\{-c\alpha \ln \alpha\}$ *for some absolute constant* $c > 0$.

ii. Let us fix any $\epsilon > 0$. Then there exists a $\delta = \delta(\epsilon) < 1$ such that for all sufficiently large $n \geq N(\epsilon)$ the probability that a random permutation σ in S_n satisfies the inequality

$$f_0(\sigma) \geq n^{-1-\epsilon} f_0(\tau)$$

is at least $\exp\{-n^\delta\}$*. In particular, one can choose any $\delta > 1 - \epsilon$.*

As in Section 2.2, we conclude that for any fixed $\alpha > 1$ there is a randomized algorithm of $O(n^2)$ complexity which produces a permutation σ satisfying (i). If we are willing to settle for an algorithm of mildly exponential complexity, we can achieve the bound of type (ii), which is weaker than the corresponding bound of Corollary 2.2.3.

For the generalized problem (1.2), we can show that the estimates of Corollary 2.4.3 are tight by constructing a tensor (cf. Remark 3.2.5 and Section 2.3.2) that is constant on the rings $U(\tau, k)$ and attains the averages of Theorem 2.4.1. For functions of type (1.1), we can show that the *frequency* bound of δn^{-2} in Part (i) for the fraction of permutations meeting the guarantee is essentially tight (even in the bullseye case of Section 2.1) by using the functions of Remark 3.9.1. An interesting question is whether the *guarantee* of α/n^2 in Part (i) is also tight for functions of type (1.1).

We prove our results in Chapter III.

CHAPTER III

Methods and Proofs

In this chapter, we prove the results of Chapter II. The facts we use from representation theory are described in Section 3.2.

3.1 The Central Projection

In this section, we prove some technical lemmas for later use.

First, we show that it is indeed easy to compute the average value $\overline{f}$ of a function f defined by (1.1) or (1.2). The result is not new (see [GW70]), we state it here for the sake of completeness.

Lemma 3.1.1. *Let $f : S_n \longrightarrow \mathbb{R}$ be a function defined by (1.1) for some $n \times n$ matrices $A = (a_{ij})$ and $B = (b_{ij})$. Let us define*

$$\alpha_1 = \sum_{1 \leq i \neq j \leq n} a_{ij}, \quad \alpha_2 = \sum_{i=1}^{n} a_{ii} \quad \text{and}$$
$$\beta_1 = \sum_{1 \leq i \neq j \leq n} b_{ij}, \quad \beta_2 = \sum_{i=1}^{n} b_{ii}.$$

Then

$$\overline{f} = \frac{\alpha_1 \beta_1}{n(n-1)} + \frac{\alpha_2 \beta_2}{n}.$$

Similarly, if f is defined by (1.2) for some tensor $C = \left\{c_{kl}^{ij}\right\}$, $1 \leq i, j, k, l \leq n$ then

$$\overline{f} = \frac{1}{n(n-1)} \sum_{1 \leq i \neq j \leq n} \sum_{1 \leq k \neq l \leq n} c_{kl}^{ij} + \frac{1}{n} \sum_{1 \leq i, l \leq n} c_{ll}^{ii}. \tag{3.1}$$

Proof. We prove the first part only, as the proof of the the second part is completely similar. Let us choose a pair of indices $1 \leq i \neq j \leq n$. Then, as σ ranges over the symmetric group S_n, the ordered pair $\big(\sigma(i), \sigma(j)\big)$ ranges over all ordered pairs (k, l) with $1 \leq k \neq l \leq n$ and each such a pair (k, l) appears $(n-2)!$ times. Similarly, for each index $1 \leq i \leq n$, the index $\sigma(i)$ ranges over the set $\{1, \ldots, n\}$ and each $j \in \{1, \ldots, n\}$ appears $(n-1)!$ times. Therefore,

$$\begin{aligned}\overline{f} &= \frac{1}{n!} \sum_{\sigma \in S_n} \sum_{i,j=1}^{n} b_{\sigma(i)\sigma(j)} a_{ij} = \sum_{i,j=1}^{n} \Big(a_{ij} \frac{1}{n!} \sum_{\sigma \in S_n} b_{\sigma(i)\sigma(j)} \Big) \\ &= \frac{1}{n(n-1)} \sum_{i \neq j}^{n} a_{ij} \beta_1 + \frac{1}{n} \sum_{i=1}^{n} a_{ii} \beta_2 = \frac{\alpha_1 \beta_1}{n(n-1)} + \frac{\alpha_2 \beta_2}{n}\end{aligned}$$

and the proof follows. □

Remark 3.1.2. Suppose that $f(\sigma) = \langle B, \sigma(A) \rangle$ for some matrices A and B and all $\sigma \in S_n$ is the objective function in the `QAP` (1.1) and suppose that the maximum value of f is attained at a permutation τ. Let $A^\tau = \tau(A)$ and let $f^\tau(\sigma) = \langle B, \sigma(A^\tau) \rangle$. Then $f^\tau(\sigma) = f(\sigma\tau)$, hence the maximum value of f^τ is attained at the identity permutation ε and the distribution of values of f and f^τ is the same. We observe that if A is symmetric then A^τ is also symmetric, and if A has constant row and column sums and a constant diagonal then so does A^τ. Hence, as far as the distribution of values of f is concerned, without loss of generality we may assume that the maximum of f is attained at the identity permutation ε. The same is true for functions in the generalized problem (1.2).

Next, we introduce our main tool.

Definition 3.1.3. Let $f : S_n \longrightarrow \mathbb{R}$ be a function. Let us define function $g : S_n \longrightarrow \mathbb{R}$ by

$$g(\sigma) = \frac{1}{n!} \sum_{\omega \in S_n} f(\omega^{-1} \sigma \omega).$$

We call g the *central projection* of f.

It turns out that the central projection captures some important information regarding the distribution of values of a function.

Lemma 3.1.4. *Let $f : S_n \longrightarrow \mathbb{R}$ be a function and let g be the central projection of f. Then*

i. The averages of f and g over the k-th ring $U(\varepsilon, k)$ around the identity permutation coincide:

$$\frac{1}{|U(\varepsilon,k)|}\sum_{\sigma\in U(\varepsilon,k)} f(\sigma) = \frac{1}{|U(\varepsilon,k)|}\sum_{\sigma\in U(\varepsilon,k)} g(\sigma);$$

ii. The average values of f and g on the symmetric group coincide: $\overline{f} = \overline{g}$;

iii. Suppose that $f(\varepsilon) \geq f(\sigma)$ for all $\sigma \in S_n$. Then $g(\varepsilon) \geq g(\sigma)$ for all $\sigma \in S_n$.

Proof. We observe that $\sigma \in U(\varepsilon, k)$ if and only if σ has exactly $n - k$ fixed points. Hence for any fixed $\omega \in S_n$, the permutation $\omega^{-1}\sigma\omega$ ranges over $U(\varepsilon, k)$ as σ ranges over $U(\varepsilon, k)$. Hence

$$\begin{aligned}\frac{1}{|U(\varepsilon,k)|}\sum_{\sigma\in U(\varepsilon,k)} g(\sigma) &= \frac{1}{|U(\varepsilon,k)|}\sum_{\sigma\in U(\varepsilon,k)}\left(\frac{1}{n!}\sum_{\omega\in S_n} f(\omega^{-1}\sigma\omega)\right)\\ &= \frac{1}{n!}\sum_{\omega\in S_n}\left(\frac{1}{|U(\varepsilon,k)|}\sum_{\sigma\in U(\varepsilon,k)} f(\omega^{-1}\sigma\omega)\right) = \frac{1}{|U(\varepsilon,k)|}\sum_{\sigma\in U(\varepsilon,k)} f(\sigma)\end{aligned}$$

and (i) is proven. Part (ii) follows from (i). To prove (iii), we note that $\omega^{-1}\varepsilon\omega = \varepsilon$ for all $\omega \in S_n$ and hence $g(\varepsilon) = f(\varepsilon)$. Moreover, for any $\sigma \in S_n$

$$g(\sigma) = \frac{1}{n!}\sum_{\omega\in S_n} f(\omega^{-1}\sigma\omega) \leq \frac{1}{n!}\sum_{\omega\in S_n} f(\varepsilon) = g(\varepsilon).$$

$\square$

3.2 Facts from Representation Theory

We aim to understand the `QAP` by studying the action of S_n by conjugation on the matrix $A = (a_{ij})$. This action is well studied in representation theory of finite groups (see, for example, [FH91] or [JK81]). The crucial observation for our approach is that the central projections g of functions f defined by (1.1) or (1.2) must have a relatively simple structure. In particular, g must lie in a 4-, 3-, or 2- dimensional vector space, depending on whether we consider the general case, the cases of Sections 2.2 and 2.3 or the special case of Section 2.1. If we require, additionally, that $\overline{f} = 0$ then the dimensions drop by 1 to 3, 2 and 1, respectively. In this section, we introduce some facts about the symmetric group and its representation theory.

3.2.1 The Conjugacy Classes of S_n

Let us fix a permutation $\rho \in S_n$. As ω ranges over the symmetric group S_n, the permutation $\omega^{-1}\rho\omega$ ranges over the conjugacy class $X(\rho)$ of ρ, that is the set of permutations that have the same cycle structure as ρ. In particular we have:

Remark 3.2.1. If $f : S_n \longrightarrow \mathbb{R}$ is a function and $g : S_n \longrightarrow \mathbb{R}$ its central projection, then

$$g(\rho) = \frac{1}{|X(\rho)|} \sum_{\sigma \in X(\rho)} f(\sigma).$$

If $X \subset S_n$ is a set which splits into a union of conjugacy classes $X(\rho_i) : i \in I$, and for each such a class we have

$$\frac{1}{|X(\rho_i)|} \sum_{\sigma \in X(\rho_i)} f(\sigma) \geq \alpha$$

for some number α, then

$$\frac{1}{|X|} \sum_{\sigma \in X} f(\sigma) \geq \alpha.$$

Definition 3.2.2. A (linear) *representation* of a group G is a group homomorphism $\rho : G \to GL(V)$ where $GL(V)$ is the group of invertible linear transformations of a suitable vector space V. The *degree* of the representation is the dimension of V. Equivalently, the representation can be viewed as the G-module given by the action of G on V.

In the `QAP`, we are interested in the action of S_n on the set of $n \times n$ matrices Mat_n by permuting rows and columns. This is a representation of degree n^2, since it is a linear action on $V = \mathbb{R}^{n^2}$, rearranging the n^2 matrix elements.

The building blocks of representation theory are *irreducible* representations, that is representations which contain no non-trivial invariant submodules (subspaces). A basic fact about the representation theory of finite groups, is that the representation of any finite group (over a field of characteristic 0) decomposes as a finite direct sum of irreducible representations. This decomposition is not, in general, unique, and there may be multiple irreducible modules corresponding to a given irreducible representation. The irreducible modules corresponding to an irreducible representation are however, isomorphic, and their multiplicity in any decomposition is the same. Further, the submodule that is the sum of all the (isomorphic) submodules associated to a given irreducible representation is unique. This is called the *isotypical component* of the representation. We refer to [FH91] for further details.

We aim to understand the conjugation representation by decomposing it into its isotypical components. In particular, this decomposition will allow us to determine the central projection.

We describe the invariant subspaces of the action of S_n in the space of $n \times n$ matrices Mat_n by simultaneous permutations of rows and columns. For $n \geq 4$, there are seven such invariant subspaces, and we have grouped isomorphic subspaces together to give the four isotypical components of the representation. Our notation is inspired by the generally accepted notation of representation theory of S_n, where irreducible representations and their associated subspaces are indexed by partitions of the integer n.

Subspace L_n

Let L_n^1 be the space of constant matrices A:

$$a_{ij} = \alpha \quad \text{for some} \quad \alpha \quad \text{and all} \quad 1 \leq i, j \leq n.$$

Let L_n^2 be the subspace of scalar matrices A:

$$a_{ij} = \begin{cases} \alpha & \text{if } \ i = j \\ 0 & \text{if } \ i \neq j \end{cases} \quad \text{for some} \quad \alpha.$$

Finally, Let $L_n = L_n^1 + L_n^2$. One can observe that $\dim L_n = 2$ and that L_n is the subspace of all matrices that remain fixed under the action of S_n.

Subspace $L_{n-1,1}$

Let $L_{n-1,1}^1$ be the subspace of matrices with identical rows and such that the sum of entries in each row is 0:

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \dots & \dots & \dots & \dots \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \end{pmatrix}, \quad \text{where} \quad \alpha_1 + \ldots + \alpha_n = 0.$$

Similarly, let $L^2_{n-1,1}$ be the subspace of matrices with identical columns and such that the sum of entries in each column is 0:

$$A = \begin{pmatrix} \alpha_1 & \alpha_1 & \dots & \alpha_1 \\ \alpha_2 & \alpha_2 & \dots & \alpha_2 \\ \dots & \dots & \dots & \dots \\ \alpha_n & \alpha_n & \dots & \alpha_n \end{pmatrix}, \quad \text{where} \quad \alpha_1 + \ldots + \alpha_n = 0.$$

Finally, let $L^3_{n-1,1}$ be the subspace of diagonal matrices whose diagonal entries sum to zero:

$$A = \begin{pmatrix} \alpha_1 & 0 & \dots & 0 & 0 \\ 0 & \alpha_2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & \alpha_n \end{pmatrix}, \quad \text{where} \quad \alpha_1 + \ldots + \alpha_n = 0.$$

Let $L_{n-1,1} = L^1_{n-1,1} + L^2_{n-1,1} + L^3_{n-1,1}$. One can check that the dimension of each of $L^1_{n-1,1}$, $L^2_{n-1,1}$ and $L^3_{n-1,1}$ is $n-1$ and that $\dim L_{n-1,1} = 3n-3$. Moreover, the subspaces $L^1_{n-1,1}$, $L^2_{n-1,1}$ and $L^3_{n-1,1}$ do not contain non-trivial invariant subspaces. The action of S_n in $L_{n-1,1}$, although non-trivial, is not very complicated.

Subspace $L_{n-2,2}$

Let us define $L_{n-2,2}$ as the subspace of all *symmetric* matrices A with row and column sums equal to 0 and zero diagonal:

$$\begin{aligned} & a_{ij} = a_{ji} \quad \text{for all} \quad 1 \leq i,j \leq n; \\ & \sum_{i=1}^n a_{ij} = 0 \quad \text{for all} \quad j = 1, \ldots, n; \\ & \sum_{j=1}^n a_{ij} = 0 \quad \text{for all} \quad i = 1, \ldots, n \quad \text{and} \\ & a_{ii} = 0 \quad \text{for all} \quad i = 1, \ldots, n. \end{aligned}$$

One can check that $L_{n-2,2}$ is an invariant subspace, that $\dim L_{n-2,2} = (n^2 - 3n)/2$, and that $L_{n-2,2}$ contains no non-trivial invariant subspaces.

Subspace $L_{n-2,1,1}$

Let us define $L_{n-2,1,1}$ as the subset of all *skew symmetric* matrices A with row and column sums equal to 0:

$$\begin{aligned} &a_{ij} = -a_{ji} \quad \text{for all} \quad 1 \le i, j \le n; \\ &\sum_{i=1}^{n} a_{ij} = 0 \quad \text{for all} \quad j = 1, \ldots, n \quad \text{and} \\ &\sum_{j=1}^{n} a_{ij} = 0 \quad \text{for all} \quad i = 1, \ldots, n. \end{aligned}$$

One can check that $L_{n-2,1,1}$ is an invariant subspace and that $\dim L_{n-2,1,1} = (n^2 - 3n)/2 + 1$. Similarly, $L_{n-2,1,1}$ contains no non-trivial invariant subspaces.

By checking dimension and independence, we see that $\mathrm{Mat}_n = L_n + L_{n-1,1} + L_{n-2,2} + L_{n-2,1,1}$. The decomposition of Mat_n under conjugation into irreducible components is described in Section 2.9 of [JK81]. They use the fact that the conjugation action on matrices can be viewed as the tensor product of two copies of the natural permutation action on $\mathbb{R}^n$.

In the generalized problem (1.2), we look at the action of S_n on the 4-dimensional array $C = (c^{ij}_{kl})$ given by $\sigma(c^{ij}_{kl}) := c^{\sigma^{-1}(i)\sigma^{-1}(j)}_{kl}$. This decomposes as a tensor product $\mathrm{Mat}_n \otimes \mathrm{Mat}_n$, where the action on the first component is the conjugation action on Mat_n described above, and the action on the second component is trivial. Then the isotypical components of this action are simply given by $L_n \otimes \mathrm{Mat}_n$, $L_{n-1,1} \otimes \mathrm{Mat}_n$, $L_{n-2,2} \otimes \mathrm{Mat}_n$ and $L_{n-2,1,1} \otimes \mathrm{Mat}_n$.

We now introduce a key object in representation theory.

Definition 3.2.3. The *character* of a representation ρ is the function $\chi : G \to \mathbb{C}$

given by $\chi(\sigma) = \mathrm{trace}(\rho(\sigma))$.

The character is well-defined, since trace is invariant under change of basis.

Of particular interest are the characters of the irreducible representations associated to the isotypical subspaces described above. It turns out that we can give formulas for these irreducible characters in terms of the functions $p(\sigma)$ (number of fixed points) and $t(\sigma)$ (number of transpositions) on permutations (Definition 1.1.9). The characters are:

$$\chi_n(\sigma) = 1 \quad \text{for all} \quad \sigma \in S_n;$$

$$\chi_{n-1,1}(\sigma) = p(\sigma) - 1 \quad \text{for all} \quad \sigma \in S_n;$$

$$\chi_{n-2,2}(\sigma) = t(\sigma) + \frac{1}{2}p^2(\sigma) - \frac{3}{2}p(\sigma) \quad \text{for all} \quad \sigma \in S_n;$$

$$\chi_{n-2,1,1}(\sigma) = \frac{1}{2}p^2(\sigma) - \frac{3}{2}p(\sigma) - t(\sigma) + 1 \quad \text{for all} \quad \sigma \in S_n.$$

These character functions are computed explicitly in [Mur38] Chapter 5, Section 2. They can also be computed from the Murnaghan-Nakayama rule of representation theory (see for example Section 4.3 of [FH91]). This rule gives the character value at a permutation σ as the number of ways of filling a partition diagram with integers prescribed by the cycle structure of σ. This is one of the advantages of indexing the irreducible representations by partitions of n. In the case of the conjugation action, only the above four relatively simple characters occur.

It is often useful to view the characters as elements in the vector space of *class functions*, that is functions which are constant on equivalence classes of permutations. There is an important inner product on this space:

$$\langle \chi, \theta \rangle := \frac{1}{|G|} \sum_{\sigma \in G} \chi(\sigma)\theta(\sigma)$$

If the characters are allowed to be complex valued, the $\chi(\sigma)$ factors in the above sum would be conjugated, but that is not relevant here. It turns out the the irreducible characters form an orthonormal basis for the space of class functions. We refer again to [FH91] for further details.

In particular, we will use that

$$\sum_{\sigma \in S_n} \chi_{n-1,1}(\sigma) = \sum_{\sigma \in S_n} \chi_{n-2,2}(\sigma) = \sum_{\sigma \in S_n} \chi_{n-2,1,1}(\sigma) = 0,$$

hence the average value of all but the trivial character χ_n is 0.

To state the main result of this section, we recall the definitions of the central projection (see Definition 3.1.3).

Proposition 3.2.4. *For $n \times n$ matrices A and B, where $n \geq 4$, let $f : S_n \longrightarrow \mathbb{R}$ be the function defined by (1.1) and let $g : S_n \longrightarrow \mathbb{R}$ be the central projection of f.*

i. If $A \in L_n$ then g is a scalar multiple of the constant function $\chi_n(\sigma)$.

ii. If $A \in L_{n-1,1}$ then g is a scalar multiple of the function $\chi_{n-1,1}(\sigma)$.

iii. If $A \in L_{n-2,2}$ then g is a scalar multiple of the function $\chi_{n-2,2}(\sigma)$.

iv. If $A \in L_{n-2,1,1}$ then g is a scalar multiple of the function $\chi_{n-2,1,1}(\sigma)$.

Analogously, for the generalized function (1.2), if the tensor $C = (c_{kl}^{ij})$ lies in a given isotypical component of $\mathbb{R}^4$, then g is a scalar multiple of the corresponding character.

Proposition 3.2.4 follows from the representation theory of the symmetric group (see, for example, Part 1 of [FH91]). The set of all functions $f : S_n \longrightarrow \mathbb{R}$ is identified with the (real) group algebra of the symmetric group. The center of the group algebra is spanned by the characters of the irreducible representations of S_n. The basic fact that we are using here is that if f is a matrix element in an irreducible representation of the group then the central projection (Definition 3.1.3) must be a scalar multiple of the character of that representation.

Remark 3.2.5. We note that the functions p, p^2 and t are objective functions of type (1.2) in some generalized `QAP`. Indeed, to obtain p we choose $c_{ii}^{ii} = 1$ for all $i = 1, \ldots, n$ to be the only non-zero entries of C. To obtain p^2, we choose $c_{ij}^{ij} = 1$ for $1 \leq i, j \leq n$ to be the only non-zero entries of C. To obtain t, we choose $c_{ji}^{ij} = 1$ for all $1 \leq i < j \leq n$ to be the only non-zero entries of C. Consequently, the characters χ_n, $\chi_{n-1,1}$, $\chi_{n-2,2}$ and $\chi_{n-2,1,1}$ are objective functions of type (1.2).

3.2.2 Computing the Central Projection

Given a function f of type (1.1) or (1.2), we can compute the central projection g around the identity ε explicitly. From the linearity and orthonormality of the characters and Proposition 3.2.4, g can be written:

$$g = \gamma_n \chi_n + \gamma_{n-1,1} \chi_{n-1,1} + \gamma_{n-2,2} \chi_{n-2,2} + \gamma_{n-2,1,1} \chi_{n-2,1,1}$$

for some real coefficients $\gamma_n, \gamma_{n-1,1}, \gamma_{n-2,2}$ and $\gamma_{n-2,1,1}$. In particular, we are interested in cases where the some of the coefficients are zero, since conditions of this type distinguish the special cases of Sections 2.1, 2.2 and 2.3.

We will work with the generalized function f of type (1.2), for some tensor $C = (c_{kl}^{ij})$. To compute a particular γ_p where p is a partition of n (in this case necessarily one of $\{n\}, \{n-1,1\}, \{n-2,2\}$ or $\{n-2,1,1\}$), we project the tensor C into the isotypical component associated with p, and then compute the central projection of the part of f attributable to this component. For our problem, we will denote by $P_p(C)$ the projection of C into $L_p \otimes \mathrm{Mat}_n$, and define the function f_p on to $P_p(C)$ by:

$$f_p(\sigma) = \sum_{i,j=1}^{n} (P_p(C))_{\sigma(i)\sigma(j)}^{ij}$$

We denote by g_p the central projection of f_p:

$$g_p(\sigma) = \frac{1}{n!} \sum_{\omega \in S_n} f_p(\omega^{-1}\sigma\omega).$$

By Proposition 3.2.4 and linearity, g_p is the component of g parallel to the (orthonormal) basis element χ_p, and so $g_p = \gamma_p \chi_p$. Then we compute:

$$\gamma_p = \frac{g_p(\varepsilon)}{\chi_p(\varepsilon)} \tag{3.2}$$

where ε is the identity permutation.

Consider the case of the coefficient γ_n of the trivial character χ_n. The projection $P_n(C)$ of C onto the isotypical component $L_n \otimes \mathrm{Mat}_n$ is given by:

$$P_n(C)^{ij}_{kl} = \begin{cases} \dfrac{1}{n} \displaystyle\sum_{1 \le i' \le n} c^{i'i'}_{kk} & \text{if } i = j \\ \dfrac{1}{n(n-1)} \displaystyle\sum_{1 \le i' \ne j' \le n} c^{i'j'}_{kl} & \text{if } i \ne j \end{cases}$$

Noting that $\chi_n(\epsilon) = 1$, the coefficient γ_n is given by:

$$\gamma_n = \frac{g_n(\varepsilon)}{\chi_n(\varepsilon)} = \frac{1}{n(n-1)} \sum_{1 \le i \ne j \le n} \sum_{1 \le k \ne l \le n} c^{ij}_{kl} + \frac{1}{n} \sum_{1 \le i,k \le n} c^{ii}_{kk}$$

This is exactly the average value $(\overline{f})$ of f calculated in Lemma 3.1.1. So the coefficient of χ_n in g is zero exactly when the average value of f is zero. If we use the modified function f_0 in place of f, then the coefficient of χ_n in g will always be zero.

We can obtain formulas for the projections into the remaining isotypical components; we find these for $P_{n-1,1}(C)$ and $P_{n-2,1,1}(C)$ in Sections 3.2.3 and 3.2.4.

Once we have the projections $P_p(C)$, it is easy to test whether the coefficient γ_p is zero in the central projection of f. However, the theorems of Chapter II concern the distribution of values around the maximum, which may not be located at the identity, but instead at a permutation τ that we do not know in advance. Given τ, we could replace f by the shifted function f^τ from Remark 3.1.2, whose maximum lies at the identity. This shift is accomplished by using $\tau(C)$ in place of C. We then compute the projection $P_p(\tau(C))$ and test if the coefficient γ_p^τ of χ_p in the central projection of f_p^τ is zero.

Since there may be no efficient way of finding τ, we will have to settle for finding conditions such that $\gamma_p^\tau = 0$ for all $\tau \in S_n$. We use this to derive the relaxed conditions for Theorems 2.1.1, 2.2.1 and 2.3.1.

We begin by noting that the function f_p^τ from $P_p(\tau(C))$ is given by:

$$f_p^\tau(\sigma) := \sum_{i,j=1}^{n} (P_p(\tau(C)))^{ij}_{\sigma(i)\sigma(j)} = \sum_{i,j=1}^{n} (P_p(C))^{ij}_{\sigma\tau(i)\sigma\tau(j))} = f_p(\sigma\tau)$$

From (3.2), we see that the coefficient γ_p^τ is zero when the central projection g_p^τ of f_p^τ is zero at the identity ε. Observe that:

$$g_p^\tau(\varepsilon) = f_p^\tau(\varepsilon) = f_p(\tau)$$

So we will look for conditions so that $f_p(\tau) = 0$ for all $\tau \in S_n$.

In the case of the trivial character, χ_n, this is an easy exercise. The function f_n given by the tensor $P_n(C)$ is constant on all permutations with value $\overline{f}$. So the condition we arrive at is $\overline{f} = 0$. For the remaining characters, things are not quite as simple. In the next two sections, we find conditions so that the functions $f_{n-1,1}$ and $f_{n-2,1,1}$ are zero. These two conditions determine the special cases of Sections 2.2 and 2.3.

3.2.3 Conditions Determining the Pure Special Case

Consider now the character $\chi_{n-1,1}$. We begin by calculating $P_{n-1,1}(C)$, the projection of C into $L_{n-1,1} \otimes \mathrm{Mat}_n$.

To do this, we break $L_{n-1,1}$ into its constituent subspaces, $L^1_{n-1,1}$, $L^2_{n-1,1}$ and $L^3_{n-1,1}$. Any $W = (w_{ij}) \in L_{n-1,1}$ is the sum of a matrix X with identical rows whose entries sum to zero, a matrix Y with identical columns whose entries sum to zero, and a diagonal matrix Z with the diagonal summing to zero. If the diagonals of X, Y and Z are given by vectors $x = (x_i)$, $y = (y_i)$ and $z = (z_i)$ respectively, then we can write:

$$W = \begin{pmatrix} x_1 + y_1 + z_1 & x_2 + y_1 & \dots & x_n + y_1 \\ x_1 + y_2 & x_2 + y_2 + z_2 & \dots & x_n + y_2 \\ \dots & \dots & \dots & \dots \\ x_1 + y_n & x_2 + y_n & \dots & x_n + y_n + z_n \end{pmatrix},$$

$$\text{That is:} \qquad w_{ij} = \begin{cases} x_i + y_i + z_i & \text{if } i = j \\ x_j + y_i & \text{if } i \neq j \end{cases}$$

To decompose W into X, Y and Z, we solve for x_i, y_i and z_i in terms of the w_{ij}. Using the facts that $\sum_{i=1}^{n} x_i = 0 = \sum_{i=1}^{n} y_i$, we find that:

$$w_{ii} = x_i + y_i + z_i \qquad \sum_{j=1}^{n} w_{ij} = nx_i + z_i \qquad \text{and} \qquad \sum_{j=1}^{n} w_{ji} = ny_i + z_i$$

Then we get:

$$x_i = \frac{1}{n(n-2)} \left((n-1) \sum_{j=1}^{n} w_{ij} + \sum_{j=1}^{n} w_{ji} - nw_{ii} \right)$$

$$y_i = \frac{1}{n(n-2)} \left((n-1) \sum_{j=1}^{n} w_{ji} + \sum_{j=1}^{n} w_{ij} - nw_{ii} \right)$$

$$z_i = \frac{1}{(n-2)} \left(nw_{ii} - \sum_{j=1}^{n} w_{ji} - \sum_{j=1}^{n} w_{ij} \right)$$

We use these formulas to express the projection $P_{n-1,1}(A)$ of a matrix A into $L_{n-1,1}$ as the sum of row, column and diagonal matrices whose entries sum to zero. Define the following operators on Mat_n:

$$Q^1(A)_{ij} := \frac{1}{n(n-2)}\left((n-1)\sum_{j'=1}^{n} a_{ij'} + \sum_{j'=1}^{n} a_{j'i} - \sum_{i',j'=1}^{n} a_{i'j'} - na_{ii} + \sum_{i'=1}^{n} a_{i'i'}\right)$$
$$Q^2(A)_{ij} := \frac{1}{n(n-2)}\left((n-1)\sum_{i'=1}^{n} a_{i'j} + \sum_{i'=1}^{n} a_{ji'} - \sum_{i',j'=1}^{n} a_{i'j'} - na_{jj} + \sum_{i'=1}^{n} a_{i'i'}\right)$$
$$Q^3(A)_{ij} \begin{cases} \frac{1}{n-2}\left(na_{ii} - \sum_{i'=1}^{n} a_{i'i'} - \sum_{j'=1}^{n} a_{ij'} - \sum_{j'=1}^{n} a_{j'i} + \frac{2}{n}\sum_{i',j'=1}^{n} a_{i'j'}\right) & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases}$$

Then $P_{n-1,1} = Q^1 + Q^2 + Q^3$ since this operator is linear, idempotent and has range $L_{n-1,1}$. Similarly, we can describe the projection from $\mathrm{Mat}_n \otimes \mathrm{Mat}_n$ to $L_{n-1,1} \otimes \mathrm{Mat}_n$ as the sum of three operators $Q^1(C)$, $Q^2(C)$ and $Q^3(C)$. We use the shorthand $C^A_{kl} = \sum_{i',j'=1}^{n} c^{i'j'}_{kl}$ for the sum of all entries of (c^{ij}_{kl}) given fixed k,l and $C^D_{kl} = \sum_{i'}^{n} c^{i'i'}_{kl}$ for the sum of the diagonal entries of (c^{ij}_{kl}) given fixed k,l:

$$Q^1(C)^{ij}_{kl} := \frac{1}{n(n-2)}\left((n-1)\sum_{j'=1}^{n} c^{ij'}_{kl} + \sum_{j'=1}^{n} c^{j'i}_{kl} - nc^{ii}_{kl} - C^A_{kl} + C^D_{kl}\right)$$
$$Q^2(C)^{ij}_{kl} := \frac{1}{n(n-2)}\left((n-1)\sum_{i'=1}^{n} c^{i'j}_{kl} + \sum_{i'=1}^{n} c^{ji'}_{kl} - nc^{jj}_{kl} - C^A_{kl} + C^D_{kl}\right)$$
$$Q^3(C)^{ij}_{kl} \begin{cases} \frac{1}{n-2}\left(nc^{ii}_{kl} - \sum_{j'=1}^{n} c^{ij'}_{kl} - \sum_{j'=1}^{n} c^{j'i}_{kl} - C^D_{kl} + \frac{2}{n}C^A_{kl}\right) & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases}$$

Since $f_{n-1,1}$ is linear, we can break it up as a sum of functions on the three pieces of $P_{n-1,1}(C)$:

$$f_{n-1,1} = f^1_{n-1,1} + f^2_{n-1,1} + f^3_{n-1,1}$$

where we have:

$$f^1_{n-1,1}(\sigma) :=$$
$$\sum_{i,j=1}^{n} \frac{1}{n(n-2)} \left((n-1) \sum_{j'=1}^{n} c^{ij'}_{\sigma(i)\sigma(j)} + \sum_{j'=1}^{n} c^{j'i}_{\sigma(i)\sigma(j)} - nc^{ii}_{\sigma(i)\sigma(j)} - C^A_{\sigma(i)\sigma(j)} + C^D_{\sigma(i)\sigma(j)} \right)$$
$$= \sum_{i=1}^{n} \frac{1}{n(n-2)} \left((n-1) \sum_{j,j'=1}^{n} c^{ij'}_{\sigma(i)j} + \sum_{j,j'=1}^{n} c^{j'i}_{\sigma(i)j} - n \sum_{j=1}^{n} c^{ii}_{\sigma(i)j} - \sum_{j=1}^{n} C^A_{ij} + \sum_{j=1}^{n} C^D_{ij} \right)$$

$$f^2_{n-1,1}(\sigma) :=$$
$$\sum_{i,j=1}^{n} \frac{1}{n(n-2)} \left((n-1) \sum_{i'=1}^{n} c^{i'j}_{\sigma(i)\sigma(j)} + \sum_{i'=1}^{n} c^{ji'}_{\sigma(i)\sigma(j)} - nc^{jj}_{\sigma(i)\sigma(j)} - C^A_{\sigma(i)\sigma(j)} + C^D_{\sigma(i)\sigma(j)} \right)$$
$$= \sum_{j=1}^{n} \frac{1}{n(n-2)} \left((n-1) \sum_{i,i'=1}^{n} c^{i'j}_{i\sigma(j)} + \sum_{i,i'=1}^{n} c^{ji'}_{i\sigma(j)} - n \sum_{i=1}^{n} c^{jj}_{i\sigma(j)} - \sum_{i=1}^{n} C^A_{ij} + \sum_{i=1}^{n} C^D_{ij} \right)$$

$$f^3_{n-1,1}(\sigma) :=$$
$$\sum_{i=1}^{n} \frac{1}{(n-2)} \left(nc^{ii}_{\sigma(i)\sigma(i)} - \sum_{j'=1}^{n} c^{ij'}_{\sigma(i)\sigma(i)} - \sum_{j'=1}^{n} c^{j'i}_{\sigma(i)\sigma(i)} - C^D_{\sigma(i)\sigma(i)} + \frac{2}{n} C^A_{\sigma(i)\sigma(i)} \right)$$

Note that in the expressions for $f^1_{n-1,1}$ and $f^2_{n-1,1}$ we are able to re-index j and i respectively since the sum is over the entire set $\{1, ..., n\}$. Similarly, we note that $\sum_{i,j=1}^{n} C^A_{\sigma(i)\sigma(j)}$ and $\sum_{i,j=1}^{n} C^D_{\sigma(i)\sigma(j)}$ do not depend on σ.

Now consider the matrix $D = (d_{ik})$ given by:

$$\begin{aligned} d_{ik} &:= \frac{1}{n(n-2)} \Big((n-1) \sum_{j,l=1}^{n} c^{ij}_{kl} + \sum_{j,l=1}^{n} c^{ji}_{kl} - n \sum_{l=1}^{n} c^{ii}_{kl} + (n-1) \sum_{j,l=1}^{n} c^{ji}_{lk} \\ &\qquad + \sum_{j,l=1}^{n} c^{ij}_{lk} - n \sum_{l=1}^{n} c^{ii}_{lk} + n^2 c^{ii}_{kk} - n \sum_{j=1}^{n} c^{ij}_{kk} - n \sum_{j=1}^{n} c^{ji}_{kk} \Big) \\ &= \frac{1}{n(n-2)} \Big((n^2-2n) c^{ii}_{kk} + \sum_{j \neq i} \sum_{l \neq k} \Big((n-1)(c^{ij}_{kl} + c^{ji}_{lk}) + c^{ij}_{lk} + c^{ji}_{kl} \Big) \Big) \end{aligned}$$

Then D defines a linear assignment problem (see Definition 1.1.4) given by the function:

$$h(\sigma) = \sum_{i=1}^{n} d_{i\sigma(i)} \tag{3.3}$$

We check that if we set:

$$\begin{aligned} d &= \frac{1}{n}\sum_{i,k=1}^{n} d_{ik} = \frac{1}{n(n-2)}\left(\sum_{i,j=1}^{n}(2C_{ij}^{D} - 2C_{ij}^{A}) + \sum_{i=1}^{n}(2C_{ij}^{A} - nC_{ij}^{D})\right) \\ &= \frac{1}{n(n-2)}\sum_{j\neq i}\sum_{l\neq k} c_{kl}^{ij} - \frac{1}{n}\sum_{i,k=1}^{n} c_{kk}^{ii} \end{aligned}$$

then we have:

$$h(\sigma) = f_{n-1,1}^{1}(\sigma) + f_{n-1,1}^{2}(\sigma) + f_{n-1,1}^{3}(\sigma) + d = f_{n-1,1}(\sigma) + d \qquad \text{for all } \sigma \in S_n$$

So the problem of checking whether $f_{n-1,1}(\sigma) = 0$ everywhere reduces to the problem of checking whether (3.3) is constant. We remark that this is a problem of computing the central projection of a `LAP` (see the discussion in Section 4.1.1).

We can state fairly simple necessary and sufficient conditions for $h(\sigma)$ to be constant for all $\sigma \in S_n$ in terms of the matrix D. They are exactly that D can be written a sum of a row matrix and a column matrix, and has entries that sum to zero. That is for some pair of vectors $a, b \in \mathbb{R}^n$ such that $\sum_{i=1}^{n}(a_i + b_i) = 0$, we have:

$$D = \begin{pmatrix} a_1 + b_1 & a_1 + b_2 & \dots & a_1 + b_n \\ a_2 + b_1 & a_2 + b_2 & \dots & a_2 + b_n \\ \dots & \dots & \dots & \dots \\ a_n + b_1 & a_n + b_2 & \dots & a_n + b_n \end{pmatrix}, \qquad d_{ij} = (a_i + b_j)$$

If D has this form, then it is clear that the objective value will be the constant $\sum_{i=1}^{n}(a_i + b_i)$ for all $\sigma \in S_n$. Any two vectors a and b define such a D (and, without loss of generality, we can choose $a_1 = 0$).

Consider for $i, k \geq 2$ two permutations, with σ_1 satisfying $\sigma_1(1) = 1$ and $\sigma_1(i) = k$, σ_2 satisfying $\sigma_2(1) = k$ and $\sigma_2(i) = 1$, and $\sigma_1(j) = \sigma_2(j)$ for all $j \neq 1, i$. Then:

$$h(\sigma_1) - d_{11} - d_{ik} = h(\sigma_2) - d_{1k} - d_{i1}$$

For h to be constant on all permutations, we have $h(\sigma_1) = h(\sigma_2)$, hence:

$$d_{11} + d_{ik} = d_{1k} + d_{i1} \qquad \text{for all } i, k \geq 2 \tag{3.4}$$

These $(n-1)^2$ conditions determine D given its initial row and column (ie. vectors a and b above), so any D yielding a constant h must have the above form.

To summarize, the conditions (3.4) where D is obtained from the tensor C via (3.3) guarantee that the coefficient of $\chi_{n-1,1}$ in the central projection of f^τ is 0. This is the natural condition for a `QAP` (1.2) to be of the "pure" type of Section 2.2. Expanding and simplifying (3.4) in terms of C gives the conditions (2.2). If we assume that C is symmetric, then (2.2) simplifies further to (2.1).

3.2.4 Conditions Determining the Symmetric Special Case

We can perform a similar analysis for the character $\chi_{n-2,1,1}$. We calculate the projection $P_{n-2,1,1}(C)$ of C into $L^2_{n-2,1,1} \otimes \mathrm{Mat}_n$:

$$P_{n-2,1,1}(C)^{ij}_{kl} = \frac{1}{2}\left(c^{ij}_{kl} - c^{ji}_{kl} - \frac{1}{n}\sum_{i'=1}^{n} c^{i'j}_{kl} + \frac{1}{n}\sum_{i'=1}^{n} c^{ji'}_{kl} - \frac{1}{n}\sum_{j'=1}^{n} c^{ij'}_{kl} + \frac{1}{n}\sum_{j'=1}^{n} c^{j'i}_{kl}\right)$$

Some simple sufficient conditions arise from forcing the projection $P_{n-2,1,1}(C)$ of C into $L^2_{n-2,1,1} \otimes \mathrm{Mat}_n$ to be zero, that is setting:

$$\frac{1}{2}\left(c^{ij}_{kl} - c^{ji}_{kl} - \frac{1}{n}\sum_{i'=1}^{n} c^{i'j}_{kl} + \frac{1}{n}\sum_{i'=1}^{n} c^{ji'}_{kl} - \frac{1}{n}\sum_{j'=1}^{n} c^{ij'}_{kl} + \frac{1}{n}\sum_{j'=1}^{n} c^{j'i}_{kl}\right) = 0 \qquad \text{for all } k, l$$

The resulting condition is that for all k, l the matrix $A = (a_{ij})$ given by $a_{ij} = c^{ij}_{kl}$ is the sum of a symmetric matrix, a matrix with constant columns, and a matrix with constant rows.

We can use the approach of Section 3.2.3 to get weaker sufficient conditions. Define the tensor $D = (d_{kl}^{ij})$ by:

$$d_{kl}^{ij} = c_{kl}^{ij} - c_{kl}^{ji} - \frac{1}{n}\sum_{i'=1}^{n} c_{kl}^{i'j} + \frac{1}{n}\sum_{i'=1}^{n} c_{kl}^{ji'} - \frac{1}{n}\sum_{j'=1}^{n} c_{kl}^{ij'} + \frac{1}{n}\sum_{j'=1}^{n} c_{kl}^{j'i}$$

Note that $d_{kl}^{ij} = -d_{kl}^{ji}$ for all i, j, k, l, and in particular that $d_{kl}^{ii} = 0$. Note also that $\sum_{j=1}^{n} d_{kl}^{ij} = 0$ for all i, k, l. We want conditions so that:

$$h(\sigma) = \sum_{i,j=1}^{n} d_{\sigma(i)\sigma(j)}^{ij} = 0 \qquad \text{for all } \sigma \in S_n$$

Since $d_{kl}^{ii} = 0$ for all i, k, l, the terms in the above sum are non-zero only if $i \neq j$. Using the approach of Section 3.2.3, we can reduce the above expression to checking the following linear conditions on D:

$$d_{12}^{12} + d_{kl}^{ij} = d_{kl}^{12} + d_{12}^{ij} \qquad \text{for all } i \neq j, k \neq l \tag{3.5}$$

By subtracting the equation with term d_{kl}^{ji} from the equation with term d_{kl}^{ij} and using the fact that $d_{kl}^{ij} = -d_{kl}^{ji}$, we find:

$$(d_{12}^{12} + d_{kl}^{ij}) - (d_{12}^{12} + d_{kl}^{ji}) = (d_{kl}^{12} + d_{12}^{ij}) - (d_{kl}^{12} + d_{12}^{ji})$$

$$\Leftrightarrow \qquad d_{kl}^{ij} - d_{kl}^{ji} = d_{12}^{ij} d_{12}^{ji} \qquad \Leftrightarrow \qquad d_{kl}^{ij} = d_{12}^{ij}$$

It is clear that if $d_{kl}^{ij} = d_{12}^{ij}$ for all $i \neq j, k \neq l$ then (3.5) is satisfied. Since $d_{kl}^{ij} = -d_{kl}^{ji}$ for all i, j, k, l and $\sum_{j=1}^{n} d_{kl}^{ij} = 0$ for all i, k, l, it suffices to check this for $1 \leq i < j < n$. This gives the $n(n-1)\binom{n-1}{2} = \frac{1}{2}(n^4 - 4n^3 + 5n - 2n)$ linear equations (2.3) determining the symmetric case of Section 2.3.

3.3 The Central Cone

We have established that the central projection of a quadratic objective function f lies in a 4-dimensional vector space. If we take the projection of the normalized function f_0 (computed via Lemma 3.1.1), the coefficient of trivial character

χ_n will become 0, and the projection g of f_0 lies in the the 3-dimensional vector space spanned by $\chi_{n-1,1}, \chi_{n-2,2}$ and $\chi_{n-2,1,1}$. Let us chose a convenient basis in span$\{\chi_{n-1,1}, \chi_{n-2,2}, \chi_{n-2,1,1}\}$:

$$g_1 = \chi_{n-1,1} = p-1, \quad g_2 = \chi_{n-2,2} + \chi_{n-2,1,1} + 3\chi_{n-1,1} = p^2 - 2 \quad \text{and}$$

$$g_3 = \chi_{n-2,1,1} - \chi_{n-2,2} = 1 - 2t.$$

For the purposes of understanding the distribution, we can also assume that maximum of f lies at the the identity permutation ε (Remark 3.1.2). By Lemma 3.1.4, the maximum of g also lies at ε.

For this reason, we restrict our attention to the part of the 3-dimensional space of normalized central projections g that are maximized at ε. The condition that g is maximized at ε can be expressed as $n! - 1$ linear homogeneous equations:

$$g(\varepsilon) \geq g(\sigma) \text{ for all } \varepsilon \neq \sigma \in S_n$$

These define a convex polyhedral cone K in $\mathbb{R}^3$, which we call the *central cone.*

It turns out that K has a reasonably simple structure, being spanned by only 4 (if n is even) or 5 (if n is odd) extreme rays. The condition $g(\varepsilon) = 1$ defines a plane H in $\mathbb{R}^3$ and the intersection $B = H \cap K$ is a *base* of K, that is, a polygon such that every $g \in K$ can be uniquely represented in the form $g = \lambda h$ for some $h \in B$.

Lemma 3.3.1 (Description of Central Cone). *Let us define functions*

$$\begin{aligned}
r_1 &= \frac{-np + n + p^2 - 2}{n-2},\\
r_2 &= 1 - 2t,\\
r_3 &= \frac{2np - 3p - 2n - p^2 - 2t + 6}{n^2 - 5n + 6},\\
r_4 &= \frac{p + 2t - 2}{n-2} \qquad\qquad \textit{and}\\
r_{5o} &= \frac{-2np + 3p^2 - 3p + 2tn + n - 3}{n^2 - 2n - 3}.
\end{aligned}$$

Then

i. *If* $\varepsilon \in S_n$ *is the identity, then*

$$r_1(\varepsilon) = r_2(\varepsilon) = r_3(\varepsilon) = r_4(\varepsilon) = r_{5o}(\varepsilon) = 1;$$

ii. *If* n *is even then* r_1, r_2, r_3 *and* r_4 *are the vertices (in consecutive order) of the planar quadrilateral* $B = \operatorname{conv}\{r_1, r_2, r_3, r_4\}$ *that is a base of the central cone* K*;*

iii. *If* n *is odd then* r_1, r_2, r_3, r_4 *and* r_{5o} *are the vertices (in consecutive order) of the planar pentagon* $B = \operatorname{conv}\{r_1, r_2, r_3, r_4, r_{5o}\}$ *that is a base of the central cone* K*.*

The base of the central cone is illustrated in Figure 3.1.

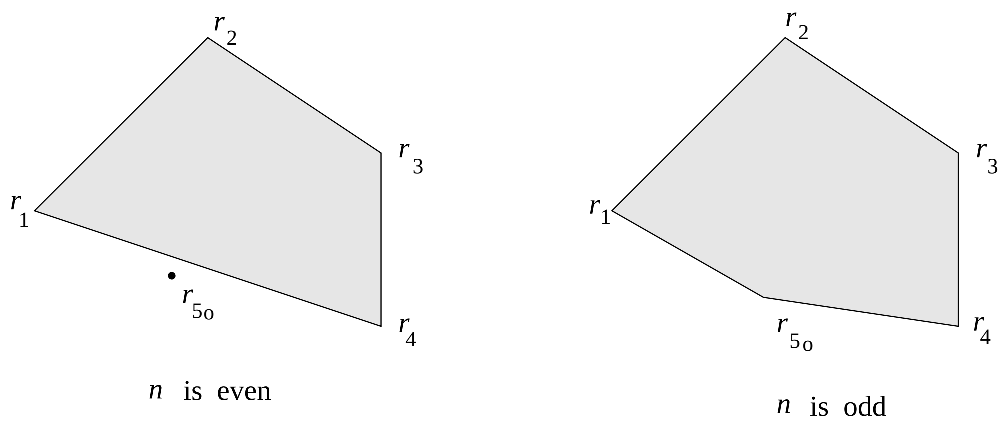

Figure 3.1: The Base of the Central Cone

Proof. A function $g \in \operatorname{span}\{g_1, g_2, g_3\}$ can be written as a linear combination $g = \alpha_1 g_1 + \alpha_2 g_2 + \alpha_3 g_3$. Then $g(\varepsilon) = \alpha_1(n-1) + \alpha_2(n^2-2) - \alpha_3$ and the conditions $g(\varepsilon) \geq g(\sigma)$ are written as

$$\alpha_1(n-1) + \alpha_2(n^2-2) + \alpha_3 \geq \alpha_1\big(p(\sigma)-1\big) + \alpha_2\big(p^2(\sigma)-2\big) + \alpha_3\big(1-2t(\sigma)\big),$$

which for $\sigma \neq \varepsilon$ are equivalent to

$$\alpha_1 + \alpha_2\big(n + p(\sigma)\big) + \alpha_3\frac{2t(\sigma)}{n - p(\sigma)} \geq 0.$$

We now need to find out which of the possible choices of $p(\sigma), t(\sigma)$ yield equations of facets (faces of dimension 2), and which are redundant and can be eliminated. Equivalently, we can adopt the dual perspective, and find the extreme rays of the dual cone. This is a technical computation, which we have left for the end of the section (Lemma 3.3.3).

Using this result, in the case of even n, the system reduces to

$$\begin{aligned}
&\alpha_1 + n\alpha_2 \geq 0\\
&\alpha_1 + (2n-3)\alpha_2 \geq 0\\
&\alpha_1 + (2n-2)\alpha_2 + \alpha_3 \geq 0\\
&\alpha_1 + n\alpha_2 + \alpha_3 \geq 0
\end{aligned} \tag{3.6}$$

whereas for odd n, the system is equivalent to

$$\begin{aligned}
&\alpha_1 + n\alpha_2 \geq 0\\
&\alpha_1 + (2n-3)\alpha_2 \geq 0\\
&\alpha_1 + (2n-2)\alpha_2 + \alpha_3 \geq 0\\
&\alpha_1 + (n+1)\alpha_2 + \alpha_3 \geq 0\\
&n\alpha_1 + n^2\alpha_2 + (n-3)\alpha_3 \geq 0.
\end{aligned} \tag{3.7}$$

The set of all feasible 3-tuples $(\alpha_1, \alpha_2, \alpha_3)$ is a polyhedral cone, which, for even n, has at most 4 extreme rays and for odd n has at most 5 extreme rays. We call an inequality of (3.6)–(3.7) *active* on a particular tuple if it holds with equality.

It is readily verified that for even n the following tuples span the extreme rays of the set of solutions to (3.6):

$$\begin{array}{ll}
\begin{pmatrix}-n, & 1, & 0\end{pmatrix} & \text{4th and 1st inequalities are active} \\
\begin{pmatrix}0, & 0, & 1\end{pmatrix} & \text{1st and 2nd inequalities are active} \\
\begin{pmatrix}2n-3, & -1, & 1\end{pmatrix} & \text{2nd and 3d inequalities are active} \\
\begin{pmatrix}1, & 0, & -1\end{pmatrix} & \text{3d and 4th inequalities are active}
\end{array}$$

and that for odd n the following tuples span the extreme rays of the set of solutions to (3.7):

$$\begin{array}{ll}
\begin{pmatrix}-n, & 1, & 0\end{pmatrix} & \text{5th and 1st inequalities are active} \\
\begin{pmatrix}0, & 0, & 1\end{pmatrix} & \text{1st and 2nd inequalities are active} \\
\begin{pmatrix}2n-3, & -1, & 1\end{pmatrix} & \text{2nd and 3d inequalities are active} \\
\begin{pmatrix}1, & 0, & -1\end{pmatrix} & \text{3d and 4th inequalities are active} \\
\begin{pmatrix}-2n-3, & 3, & -n\end{pmatrix} & \text{4th and 5th inequalities are active}
\end{array}$$

We obtain r_1, r_2, r_3, r_4 and r_{5o} by scaling the corresponding linear combinations $\alpha_1 g_1 + \alpha_2 g_2 + \alpha_3 g_3$ so that the value at the identity is equal to 1 and hence r_1, r_2, r_3, r_4 and r_{5o} lie on the same plane in $\mathrm{span}\{g_1, g_2, g_3\}$. $\square$

Remark 3.3.2 (Generators of the Central Cone). We observe that r_1 and r_{5o} have spike distributions (in particular, r_1 has the "sharp spike" distribution of Section 2.4.1) corresponding to the cases of $\gamma_1 = 1$ and $\gamma_5 = 1$ respectively in Theorem 2.4.1, that r_2 has the damped oscillator distribution corresponding to the case of $\gamma_2 = 1$ and that r_3 and r_4 have bullseye distributions corresponding to the cases of $\gamma_3 = 1$ and $\gamma_4 = 1$ respectively. If n is even then $r_{5o} \notin K$, for if σ is a product of $n/2$ commuting 2-cycles, so that $p(\sigma) = 0$ and $t(\sigma) = n/2$, then $r_{5o}(\sigma) = (n^2+n-3)/(n^2-2n-3) > 1 = r_{5o}(\varepsilon)$.

3.3.1 Asymptotic Geometry of the Central Cone

We make some brief observations on the structure of the central cone. The geometry of the cone depends on the basis chosen for $\mathbb{R}^3$. A natural choice for a basis is the three characters $\chi_{n-1,1}, \chi_{n-2,2}$ and $\chi_{n-2,1,1}$. The coordinates we found in the proof of Lemma 3.3.1 are with respect to the basis g_1, g_2, g_3. Changing basis to $\chi_{n-1,1}, \chi_{n-2,2}, \chi_{n-2,1,1}$ we get the following generators:

$$\begin{aligned}
r_1 &: \quad \frac{1}{n-2}\big(-n+3, \quad 1, \quad 1\big) \\
r_2 &: \quad \big(0, \quad -1, \quad 1\big) \\
r_3 &: \quad \frac{1}{n^2-5n+6}\big(2n-6, \quad -2, \quad 0\big) \\
r_4 &: \quad \frac{1}{n-2}\big(1, \quad 1, \quad -1\big) \\
r_{5o} &: \quad \frac{1}{n^2-2n-3}\big(-2n-6, \quad n+3, \quad -n+3\big)
\end{aligned}$$

From this data we can compute the angles between the generators with respect to the basis $\chi_{n-1,1}, \chi_{n-2,2}, \chi_{n-2,1,1}$. Two facts are worth noting. First, the angle between the (opposite) extreme rays r_1 and r_3 is asymptotically π, so the cone becomes very wide. Second, the function r_{5o} approaches the plane defined by r_1 and r_4 as $n \to \infty$. In fact the distance between the point of r_{5o} on the unit sphere and this plane is $O(n^{-2})$. So r_{5o} is "almost" a linear combination of r_1 and r_4 for large n.

We conclude this section with the proof of technical lemma used in Lemma 3.3.1.

Lemma 3.3.3. *For a permutation $\sigma \in S_n$, $\sigma \neq \varepsilon$, let $a_\sigma \in \mathbb{R}^2$ be the point*

$$a_\sigma = \left(p(\sigma), \quad \frac{2t(\sigma)}{n-p(\sigma)}\right).$$

Let $P = \operatorname{conv}\{a_\sigma : \sigma \neq \varepsilon\}$ be the convex hull of all such points a_σ.

If n is even, the extreme points of P are

$$(0,0), \quad (n-3,0), \quad (n-2,1) \quad \text{and} \quad (0,1).$$

If n is odd, the extreme points of P are

$$(0,0), \quad (n-3,0), \quad (n-2,1), \quad \big(0,(n-3)/n\big) \quad \textit{and} \qquad (1,1).$$

Proof. The set of all possible values $\big(p(\sigma),t(\sigma)\big)$, where $\sigma \neq \varepsilon$, consists of all pairs of non-negative integers (p,t) such that $p \leq n-2$, $2t \leq n$ and, additionally, $p+2t \leq n-3$ or $p+2t=n$. To find the extreme points of the set of feasible points $\big(p, 2t/(n-p)\big)$, we choose a generic vector (γ_1,γ_2) and investigate for which values of p and t the maximum of

$$\gamma_1 p + \gamma_2 \frac{2t}{n-p}$$

is attained.

Clearly, we can assume that $\gamma_2 \neq 0$. If $\gamma_2 < 0$ then we should choose the smallest possible t which would be $t=0$ unless $p=n-2$ when we have to choose $t=1$. Depending on the sign of γ_1, this produces the following pairs

$$(p,t) = \Big\{(0,0), \quad (n-3,0), \quad (n-2,1)\Big\}.$$

If $\gamma_2 > 0$ then the largest possible value of $2t/(n-p)$ is 1. If $\gamma_1 > 0$ this produces the (already included) point

$$(p,t) = \big(n-2,1\big).$$

If $\gamma_1 < 0$ we get

$$(p,t) = (0, n/2) \quad \text{for even } n$$

and

$$(p,t) = \Big\{\big(0,(n-3)/2\big), \big(1,(n-1)/2\big)\Big\} \quad \text{for odd} \quad n.$$

Summarizing, the extreme points of P are

$$(0,0), \quad (n-3,0), \quad (n-2,1), \quad (0,1) \qquad \text{for even } n$$

and

$$(0,0), \quad (n-3,0), \quad (n-2,1), \quad \big(0,(n-3)/n\big), \quad (1,1) \qquad \text{for odd } n$$

as claimed. □

3.4 Some Estimates

In this section we make some estimates of the number of permutations satisfying certain conditions. We need these estimates to prove the distributional results in the following sections.

We begin with a Markov type estimate, which asserts, roughly, that a function with a sufficiently large average takes sufficiently large values sufficiently often.

Lemma 3.4.1. *Let X be a finite set and let $f : X \longrightarrow \mathbb{R}$ be a function. Suppose that $f(x) \leq 1$ for all $x \in X$ and that*

$$\frac{1}{|X|}\sum_{x\in X} f(x) \geq \beta \quad \textit{for some} \quad \beta > 0.$$

Then for any $0 < \gamma < 1$ we have

$$\big|\big\{x \in X : f(x) \geq \beta\gamma\big\}\big| \geq \beta(1-\gamma)|X|.$$

Proof. We have

$$\begin{aligned}\beta &\leq \frac{1}{|X|}\sum_{x\in X} f(x) = \frac{1}{|X|}\sum_{x: f(x)<\beta\gamma} f(x) + \frac{1}{|X|}\sum_{x: f(x)\geq\beta\gamma} f(x) \\ &\leq \beta\gamma + \frac{\big|\big\{x : f(x) \geq \beta\gamma\big\}\big|}{|X|}.\end{aligned}$$

Hence

$$\big|\{x : f(x) \geq \beta\gamma\}\big| \geq \beta(1-\gamma)|X|.$$

□

We next estimate the fraction of permutations that have no fixed points and no 2-cycles.

Lemma 3.4.2 (Permutations with no Fixed Points and 2-cycles). *The fraction of permutations $\sigma \in S_n$ without fixed points is $e^{-1}(1+o(1))$. More precisely, it is equal to d_n, see Definition 1.1.8. Similarly, the fraction of permutations without fixed points and 2-cycles is $e^{-3/2}(1+o(1))$.*

Proof. We use an exponential generating function for this problem as described in, for example, pp. 170–172 of [GJ83]. The *cycle index* of a permutation σ is:

$$Z(\sigma) = \prod x_i^{|\text{cycles of length } i \text{ in } \sigma|}$$

Then we can define:

$$Z(S_n) = \frac{1}{n!} \sum_{\sigma \in S_n} Z(\sigma)$$

Now [GJ83] shows that the exponential generating function for all permutations is:

$$\sum_{n \geq 0} Z(S_n) = \exp\left(\sum_{i \geq 1} \frac{x_i}{i}\right)$$

Then the exponential generating function for the cycle indices of all permutations with no fixed points is given by:

$$\exp\left(\sum_{i \geq 2} \frac{x_i}{i}\right)$$

We count the number of permutations by weighting the cycles by their length, via substituting z^i for x_i. This gives an exponential generating series in z for the number of permutations of size n with no fixed points. Extracting the coefficient of $z^n/n!$, we get:

$$\begin{aligned}\left[\frac{z^n}{n!}\right] \exp\left(\sum_{i \geq 2} \frac{z^i}{i}\right) &= n![z^n] \exp\left(\log\left(\frac{1}{1-z}\right) - z\right) \\ &= n![z^n] \frac{1}{1-z} \exp(-z) = n! d_n\end{aligned}$$

The final step uses the Taylor expansions for $1/(1-z)$ and $\exp(-z)$:

$$\frac{1}{1-z} = \sum_{i\geq 0} z^i \quad \text{and} \quad \exp(-z) = \sum_{i\geq 0} \frac{(-1)^i}{i!} z^i$$

We can observe that $d_i \geq \frac{1}{3}$ for $i \geq 2$.

Similarly, the exponential generating series for the number of permutations with no fixed points or 2-cycles is:

$$\frac{1}{1-z} \exp(-z) \exp\left(\frac{-z^2}{2}\right)$$

We can expand this into Taylor series as:

$$\sum_{i\geq 0} z^i \sum_{j\geq 0} \frac{(-1)^j z^j}{j!} \sum_{k\geq 0} \frac{(-1)^k z^k}{2^k k!}$$

Taking the coefficient of z^n and rearranging terms, we get the number of permutations with no fixed points or 2-cycles:

$$\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k}{k!2^k} \sum_{j=0}^{n-2k} \frac{(-1)^j}{j!} = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k}{k!2^k} d_{n-2k}$$

Since $0 \leq d_i \leq 1$, and $d_i \to e^{-1}$ as $i \to \infty$, the sum converges to $e^{-1.5}$ as $n \to \infty$, and in particular is at least $\frac{1}{5}$ for $n \geq 3$. □

Recall the Definition (1.1.9) of $t(\sigma)$.

Lemma 3.4.3. *We have*

$$\frac{1}{|U(\varepsilon, k)|} \sum_{\sigma \in U(\varepsilon,k)} t(\sigma) = \frac{1}{2}\nu(n-k),$$

where ν is the function of Definition 1.1.8 and ε is the identity permutation.

Proof. For a pair of indices $1 \leq i < j \leq n$, let

$$t_{ij}(\sigma) = \begin{cases} 1 & \text{if } \sigma(i) = j \quad \text{and} \quad \sigma(j) = i \\ 0 & \text{otherwise.} \end{cases}$$

Then $t(\sigma) = \sum_{i<j} t_{ij}(\sigma)$. Let us compute the average of $t_{ij}(\sigma)$ over $U(\varepsilon, k)$. To choose a permutation $\sigma \in U(\varepsilon, k)$, one has to choose k fixed points in $\binom{n}{k}$ ways and then a permutation without fixed points on the remaining $n-k$ symbols in $d_{n-k}(n-k)!$ ways. Hence $|U(\varepsilon, k)| = d_{n-k}n!/k!$. To choose a permutation $\sigma \in U(\varepsilon, k)$ where (ij) is a 2-cycle, one has to choose k fixed points in $\binom{n-2}{k}$ ways and then a permutation without fixed points on $n-k-2$ symbols. Hence the total number of permutations $\sigma \in U(\varepsilon, k)$ with $t_{ij}(\sigma) = 1$ is

$$\binom{n-2}{k} d_{n-k-2}(n-k-2)! = \frac{d_{n-k-2}(n-2)!}{k!}.$$

Thus for all pairs $i < j$ we have

$$\frac{1}{|U(\varepsilon, k)|} \sum_{\sigma \in U(\varepsilon, k)} t_{ij}(\sigma) = \frac{(n-2)! d_{n-k-2}}{n! d_{n-k}} = \frac{\nu(n-k)}{n(n-1)}$$

and

$$\frac{1}{|U(\varepsilon, k)|} \sum_{\sigma \in U(\varepsilon, k)} t(\sigma) = \sum_{i<j} \frac{1}{|U(\varepsilon, k)|} \sum_{\sigma \in U(\varepsilon, k)} t_{ij}(\sigma) = \binom{n}{2} \frac{\nu(n-k)}{n(n-1)} = \frac{\nu(n-k)}{2}.$$

□

3.5 Proof of Bullseye Special Case

We are now ready to prove our main results. We begin with Theorem 2.1.1. The proof is based on the observation that A satisfies the conditions of Section 2.1 if and only if $A \in L_n + L_{n-2,2}$ (see Section 3.2).

Proof of Theorem 2.1.1. Without loss of generality, we may assume that the maximum of $f_0(\sigma)$ is attained at the identity permutation ε (see Remark 3.1.2). Excluding the non-interesting case of $f_0 \equiv 0$, by scaling f, if necessary, we can assume that $f_0(\varepsilon) = 1$. Let g be the central projection of f_0. Then by Parts (2) and (3) of Lemma 3.1.4, we have $\overline{g} = 0$ and $1 = g(\varepsilon) \geq g(\sigma)$ for all $\sigma \in S_n$. Moreover, since

$A \in L_n + L_{n-2,2}$, by Parts (1) and (3) of Proposition 3.2.4, g must be a linear combination of the constant function χ_n and $\chi_{n-2,2}$. Since $\overline{g} = 0$, g should be proportional to $\chi_{n-2,2}$ and since $g(\varepsilon) = 1$, we have

$$g = \frac{2}{n^2 - 3n}\chi_{n-2,2} = \frac{2t + p^2 - 3p}{n^2 - 3n}.$$

Now $\sigma \in U(\varepsilon, k)$ if and only if $p(\sigma) = k$. Applying Part (1) of Lemma 3.1.4 and Lemma 3.4.3 we get

$$\frac{1}{|U(\varepsilon,k)|}\sum_{\sigma \in U(\varepsilon,k)} f_0(\sigma) = \frac{1}{|U(\varepsilon,k)|}\sum_{\sigma \in U(\varepsilon,k)} g(\sigma) = \frac{k^2 - 3k + \nu(n-k)}{n^2 - 3n}$$

and the proof follows. □

Proof of Theorem 2.1.2. As in the proof of Theorem 2.1.1, we assume that the maximum value of f_0 is equal to 1.

Let us estimate the cardinality $|U(\tau, k)| = |U(\varepsilon, k)|$. Since $\sigma \in U(\varepsilon, k)$ if and only if σ has k fixed points, to choose a $\sigma \in U(\varepsilon, k)$ one has to choose k points in $\binom{n}{k}$ ways and then choose a permutation of the remaining $n - k$ points without fixed points. Using Lemma 3.4.2, we get

$$|U(\tau, k)| \geq \binom{n}{k}(n-k)!/3 = \frac{n!}{3k!}.$$

Applying Lemma 3.4.1 with $\beta = \beta(n, k)$ and $X = U(\tau, k)$, from Theorem 2.1.1 we conclude that

$$\begin{aligned} \mathbf{P}\Big\{\sigma \in S_n : f_0(\sigma) \geq \gamma\beta(n,k)\Big\} &\geq \frac{(1-\gamma)\beta(n,k)|U(\tau,k)|}{n!} \\ &\geq \frac{(1-\gamma)\beta(n,k)}{3k!}. \end{aligned}$$

□

3.6 Proof of Pure Special Case

In this section, we prove Theorems 2.2.1 and 2.2.2. We observe that A satisfies the conditions of Section 2.2 if and only if $A \in L_n + L_{n-2,1,1} + L_{n-2,2}$ (see Section 3.2). As in Section 3.5, the L_n component contributes just a constant to f. Since the $L_{n-1,1}$ component attributed to the Linear Assignment Problem is absent, we call this case "pure".

In the proof of Theorem 2.1.1, the conditions on A restricted the central projection g of f_0 to the one-dimensional subspace spanned by $\chi_{n-1,1}$. With the added conditions that $g(\varepsilon) \geq g(\sigma)$ for all permutations σ, g must lie on a ray in the interior of the central cone, K. By fixing the value of g at the identity, $g(\varepsilon) = 1$, we determine g completely.

In Theorem 2.2.1, the conditions on A are relaxed so that the central projection g of f_0 lies in the two-dimensional subspace spanned by $\chi_{n-2,1,1}$ and $\chi_{n-2,2}$. Hence with the added conditions $g(\varepsilon) \geq g(\sigma)$ for all permutations σ, g must lie in a two-dimensional slice of the central cone. We call this slice K_p, the *pure central cone.* Since it is a two-dimensional convex cone, it must be spanned by two extreme rays. We compute the extreme rays of K_p by intersecting the subspace spanned by $\chi_{n-2,1,1}$ and $\chi_{n-2,2}$ with the the central cone. Conveniently, one of the extreme rays of the central cone is also an extreme ray of K_p, namely r_2. We check that $r_2 = 1 - 2t$ is in the subspace spanned by $\chi_{n-2,2}$ and $\chi_{n-2,1,1}$:

$$\chi_{n-2,1,1} - \chi_{n-2,2} = (\frac{1}{2}p^2 - \frac{3}{2}p - t + 1) - (t + \frac{1}{2}p^2 - \frac{3}{2}p) = 1 - 2t$$

Then r_2 is extreme for K_p, since it is extreme for K.

The second extreme ray lies between r_1 and r_4 if n is even, and between r_{5o} and

r_4 if n is odd. By computing the intersection, we get the second extreme ray:

$$\begin{aligned} r_{6e} &= \frac{p^2 - 3p - n - 6t + 2tn + 4}{n^2 - 4n + 4} \qquad \text{if } n \text{ is even} \\ &= \frac{n-3}{n-2} r_4 + \frac{1}{n-2} r_1 = \frac{1}{n-2}\chi_{n-2,2} + \frac{4-n}{(n-2)^2}\chi_{n-2,1,1} \end{aligned}$$

$$\begin{aligned} r_{6o} &= \frac{p^2 - 3p - n - 4t + 2tn + 3}{n^2 - 4n + 3} \qquad \text{if } n \text{ is odd} \\ &= \frac{2n-4}{3n-3} r_4 + \frac{n+1}{3n-3} r_{5o} = \frac{-1}{n-1}\chi_{n-2,2} + \frac{1}{n-3}\chi_{n-2,1,1} \end{aligned}$$

We have scaled the extreme rays so that the values at the identity ε are $r_2(\varepsilon) = r_{6e}(\varepsilon) = r_{6o}(\varepsilon) = 1$.

Remark 3.6.1. We observe that r_2 has the damped oscillator distribution corresponding to the case of $\gamma_1 = 1$ in Theorem 2.2.1, whereas r_{6e} and r_{6o} both have the bullseye distribution corresponding to the case of $\gamma_2 = 1$ in Theorem 2.2.1. If n is even, then $r_{6o} \notin K_p$. Indeed, if σ is a product of $n/2$ commuting 2-cycles, so that $p(\sigma) = 0$ and $t(\sigma) = n/2$, then $r_{6o}(\sigma) = (n^2 - 3n + 3)/(n^2 - 4n + 3) > 1 = r_{2o}(\varepsilon)$.

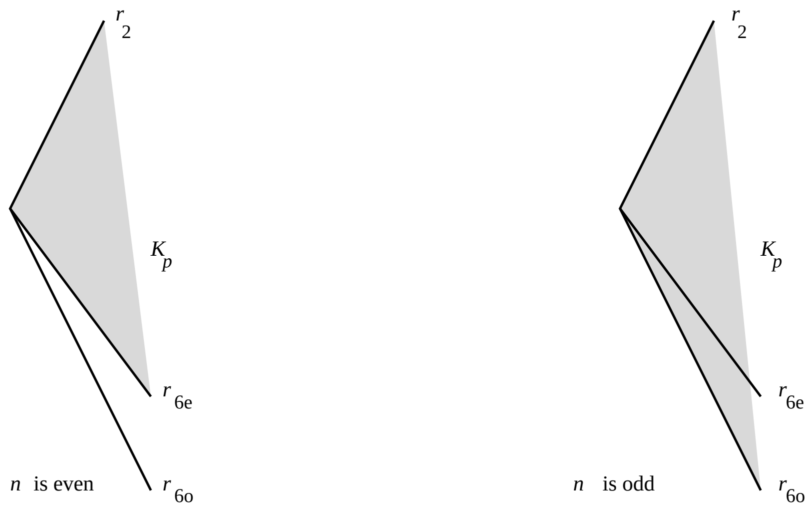

Figure 3.2: The Central (Pure) Cone

Proof of Theorem 2.2.1. We proceed as in the proof of Theorem 2.1.1 (Section 3.5) with some modifications. Without loss of generality, we assume that the maximum of $f_0(\sigma)$ is attained on the identity permutation ε and that $f_0(\varepsilon) = 1$. Let g be the central projection of f_0. Since $A \in L_n + L_{n-2,2} + L_{n-2,1,1}$, by Parts (1), (3) and (4) of Proposition 3.2.4, g is a linear combination of χ_n, $\chi_{n-2,2}$ and $\chi_{n-2,1,1}$. By Part (2) of Lemma 3.1.4, we have $\overline{g} = \overline{f_0} = 0$, so g is a linear combination of $\chi_{n-2,2}$ and $\chi_{n-2,1,1}$ alone. Moreover, by Part (3) of Lemma 3.1.4, we have $1 = g(\varepsilon) \geq g(\sigma)$ for all $\sigma \in S_n$. Hence g lies in the central cone K_p. From our description of the cone above, we conclude that g must be a convex combination of r_1 and r_{2e} for n even and a convex combination of r_1 and r_{2o} for n odd. Applying Part (1) of Lemma 3.1.4, we can replace the average of f_0 over the set $U(\varepsilon, k)$ by the average of g over $U(\varepsilon, k)$. The proof now follows by Lemma 3.4.3 and the observation that $\sigma \in U(\varepsilon, k)$ if and only if $p(\sigma) = k$. $\square$

To prove Theorem 2.2.2, we need need one preliminary result.

Lemma 3.6.2. *Let g be a linear combination of $g_1 = \chi_{n-2,2} + \chi_{n-2,1,1} = p^2 - 3p + 1$ and $g_2 = \chi_{n-2,1,1} - \chi_{n-2,2} = 1 - 2t$ such that $g(\varepsilon) = 1$. For a $3 \leq k \leq n-1$, let σ_k be a permutation such that $p(\sigma_k) = k$ and $t(\sigma_k) = 0$ and let θ_k be a permutation such that $p(\theta_k) = k$ and $t(\theta_k) = 1$. Then*

$$\max\{g(\sigma_k), g(\theta_k)\} \geq \frac{k^2 - 3k + 1}{n^2 - 3n + 1}.$$

Proof. Since $g(\varepsilon) = 1$, $g_1(\varepsilon) = n^2 - 3n + 1$ and $g_2(\varepsilon) = 1$, we can write

$$g = \alpha_1 \frac{p^2 - 3p + 1}{n^2 - 3n + 1} + \alpha_2(1 - 2t)$$

for some α_1 and α_2 such that $\alpha_1 + \alpha_2 = 1$. Then

$$g(\sigma_k) = \alpha_1 \frac{k^2 - 3k + 1}{n^2 - 3n + 1} + \alpha_2 \quad \text{and}$$

$$g(\theta_k) = \alpha_1 \frac{k^2 - 3k + 1}{n^2 - 3n + 1} - \alpha_2$$

We observe that $g(\sigma_k)$ and $g(\theta_k)$ are linear functions of α_1 and α_2 and that for

$$\alpha_1 = 1 \quad \text{and} \quad \alpha_2 = 0$$

we have

$$g(\sigma_k) = g(\theta_k) = \frac{k^2 - 3k + 1}{n^2 - 3n + 1} \tag{3.8}$$

Let

$$\lambda_1 = \frac{n^2 - 3n + 1}{2(k^2 - 3k + 1)} + \frac{1}{2} \quad \text{and} \quad \lambda_2 = \frac{n^2 - 3n + 1}{2(k^2 - 3k + 1)} - \frac{1}{2}.$$

Then $\lambda_1, \lambda_2 > 0$ and

$$\lambda_1 g(\sigma_k) + \lambda_2 g(\theta_k) = \alpha_1 + \alpha_2 = 1.$$

Comparing this with (3.8) we conclude that there are no values α_1 and α_2 such that $\alpha_1 + \alpha_2 = 1$ and

$$g(\sigma_k), g(\theta_k) < \frac{k^2 - 3k + 1}{n^2 - 3n + 1}.$$

$\square$

Proof of Theorem 2.2.2. Without loss of generality, we may assume that the maximum value of f_0 is attained at the identity permutation ε (see Remark 3.1.2). Excluding an obvious case of $f_0 \equiv 0$, by scaling f, if necessary, we may assume that $f_0(\varepsilon) = 1$. Let g be the central projection of f_0. As in the proof of Theorem 2.2.1, we deduce that g is a linear combination of $\chi_{n-2,2}$ and $\chi_{n-2,1,1}$ and that $g(\varepsilon) = 1$.

Let us choose a $3 \leq k \leq n - 3$ and let X_k be the set of permutations σ such that $p(\sigma) = k$ and $t(\sigma) = 0$ and let Y_k be the set of permutations θ such that $p(\theta) = k$ and $t(\theta) = 1$. To choose a permutation $\sigma \in X_k$, one has to choose k fixed points in $\binom{n}{k}$ ways and then a permutation without fixed points or 2-cycles on the remaining $(n - k)$ points. Then, by Lemma 3.4.2

$$|X_k| \geq \frac{1}{5}\binom{n}{k}(n-k)! = \frac{1}{5}\frac{n!}{k!}.$$

Similarly, to choose a permutation $\theta \in Y_k$, one has to choose a 2-cycle in $\binom{n}{2}$ ways, k fixed points in $\binom{n-2}{k}$ ways and a permutation without fixed points or 2-cycles on the remaining $(n-k-2)$ points. Then, by Lemma 3.4.2

$$|Y_k| \geq \frac{1}{5}\binom{n}{2}\binom{n-2}{k}(n-k-2)! = \frac{n!}{10k!}.$$

Let us choose a permutation $\sigma \in X_k$ and a permutation $\theta \in Y_k$ and let $Z = X_k$ if $g(\sigma) \geq g(\theta)$ and $Z = Y_k$ otherwise. Then

$$|Z| \geq \frac{n!}{10k!}$$

and by Lemma 3.6.2,

$$g(\sigma) \geq \frac{k^2 - 3k + 1}{n^2 - 3n + 1} \quad \text{for all} \quad \sigma \in Z.$$

The set Z is a disjoint union of some conjugacy classes $X(\rho)$ and for each $X(\rho)$ by Remark 3.2.1, we have

$$g(\rho) = \frac{1}{|X(\rho)|} \sum_{\sigma \in X(\rho)} f_0(\sigma) \geq \frac{k^2 - 3k + 1}{n^2 - 3n + 1}$$

and hence

$$\frac{1}{|Z|} \sum_{\sigma \in X(\rho)} f_0(\sigma) \geq \frac{k^2 - 3k + 1}{n^2 - 3n + 1}.$$

Applying Lemma 3.4.1 with $X = Z$ and $\beta = \beta(n, k)$, we get that

$$\mathbf{P}\Big\{\sigma \in S_n : f_0(\sigma) \geq \gamma\beta(n,k)\Big\} \geq \frac{(1-\gamma)\beta(n,k)}{10k!}.$$

$\square$

Remark 3.6.3. It follows from the proof that we are able to choose the required number of "good" permutations among the permutations whose distance to the optimal permutation τ is $n - k$.

3.7 Proof of Symmetric Case

The proofs of Theorems 2.3.1 and 2.3.2 are essentially the same as the proof in the previous section. In this case, A satisfies the conditions of Section 2.3 if and only if $A \in L_n + L_{n-1,1} + L_{n-2,2}$. Again, the L_n component contributes just a constant to f.

We then have the central projection g of f_0 lying in the two-dimensional subspace spanned by $\chi_{n-1,1}$ and $\chi_{n-2,2}$. Hence with the added conditions $g(\varepsilon) \geq g(\sigma)$ for all permutations σ, g must lie in a two-dimensional slice of the central cone. We call this slice K_s, the *symmetric central cone.* Since it is a two-dimensional convex cone, it must be spanned by two extreme rays. We compute the extreme rays K_s by intersecting the subspace with the central cone. It turns out that one of the extreme rays of the central cone K is also an extreme ray of K_s, namely:

$$r_3 = \frac{2np - 3p - 2n - p^2 - 2t + 6}{n^2 - 5n + 6}$$

We check that r_3 is in the subspace spanned by $\chi_{n-1,1}$ and $\chi_{n-2,2}$:

$$r_3 = \frac{2}{n-2}\chi_{n-1,1} - \frac{2}{n^2 - 5n + 6}\chi_{n-2,2}$$

The second extreme ray lies between r_1 and r_4 if n is even, and between r_1 and r_{5o} if n is odd:

$$\begin{aligned} r_{7e} &= \frac{-np + n + p^2 + p + 2t - 4}{2n - 4} \qquad \text{if } n \text{ is even} \\ &= \frac{1}{2}r_1 + \frac{1}{2}r_4 = \frac{4-n}{2n-4}\chi_{n-1,1} + \frac{1}{n-2}\chi_{n-2,2} \end{aligned}$$

$$\begin{aligned} r_{7o} &= \frac{-n^2p + np^2 + n^2 + np + 2nt - 4n - 3p + 3}{2n^2 - 7n + 3} \qquad \text{if } n \text{ is odd} \\ &= \frac{n-2}{2n-1}r_1 + \frac{n+1}{2n-1}r_{5o} = \frac{1-n}{2n-1}\chi_{n-1,1} + \frac{2n}{2n^2 - 7n + 3}\chi_{n-2,2} \end{aligned}$$

We have scaled the extreme rays so that the values at the identity ε are $r_3(\varepsilon) = r_{7e}(\varepsilon) = r_{7o}(\varepsilon) = 1$.

Remark 3.7.1. We observe that r_3 has a bullseye distribution corresponding to the case of $\gamma_1 = 1$ in Theorem 2.1.1, whereas r_{7e} and r_{7o} both have spike-type distributions corresponding to the case of $\gamma_2 = 1$ in Theorem 2.2.1. If n is even, then $r_{7o} \notin K_s$. Indeed, if σ is a product of $n/2$ commuting 2-cycles, so that $p(\sigma) = 0$ and $t(\sigma) = n/2$, then $r_{7o}(\sigma) = (2n^2 - 4n + 3)/(2n^2 - 7n + 3) > 1 = r_{7o}(\varepsilon)$. Geometrically, the symmetric cone K_s looks exactly like the pure cone K_p, Figure 3.3.

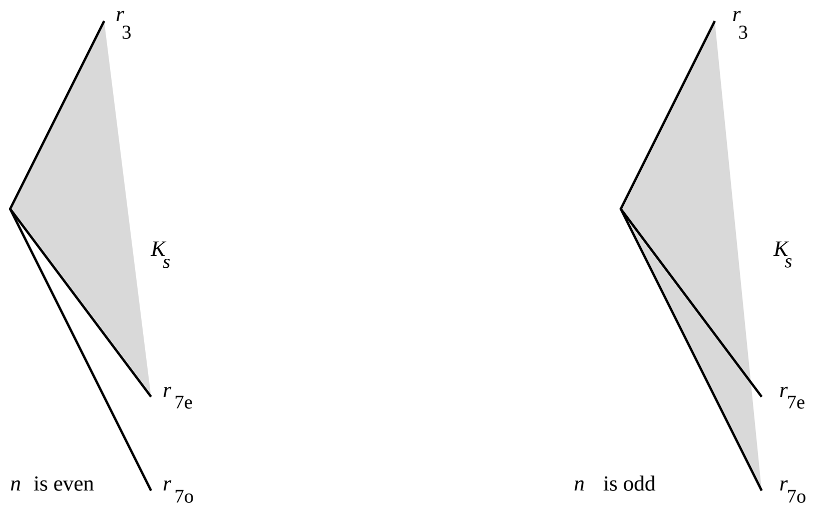

Figure 3.3: The Central (Symmetric) Cone

Proof of Theorem 2.3.1. The proof of Theorem 2.3.1 is completely similar to the proof of Theorem 2.1.1, using the extreme rays r_3, r_{7e} and r_{7o} in place of r_2, r_{6e} and r_{6o}. $\square$

To prove Theorem 2.3.2, we make a computation similar to Lemma 3.6.2.

Lemma 3.7.2. *Let g be a linear combination of $g_1 = \chi_{n-1,1} = p - 1$ and $g_2 = 2\chi_{n-2,2} + 3\chi_{n-1,1} = p^2 + 2t - 3$ such that $g(\varepsilon) = 1$. For a $3 \leq k \leq n-3$, let σ_k be* *a permutation such that* $p(\sigma_k) = k$ *and* $t(\sigma_k) = 0$ *and let* η_k *be a permutation such that* $p(\eta_k) = 0$ *and* $t(\eta_k) = k$. *Then*

$$\max\{g(\sigma_k), g(\eta_k)\} \geq \frac{3k-5}{n^2 - kn + k + 2n - 5}.$$

Proof. We can write

$$g = \alpha_1 \frac{p-1}{n-1} + \alpha_2 \frac{p^2 + 2t - 3}{n^2 - 3}$$

for some α_1 and α_2 such that $\alpha_1 + \alpha_2 = 1$. Then

$$g(\sigma_k) = \alpha_1 \frac{k-1}{n-1} + \alpha_2 \frac{k^2 - 3}{n^2 - 3} \quad \text{and}$$
$$g(\eta_k) = -\alpha_1 \frac{1}{n-1} + \alpha_2 \frac{2k-3}{n^2-3}.$$

We observe $g(\sigma_k)$ and $g(\eta_k)$ are linear functions of α_1 and α_2 and that for

$$\alpha_1 = \frac{kn - k - 2n + 2}{-n^2 + kn - k - 2n + 5} \quad \text{and} \quad \alpha_2 = \frac{3 - n^2}{-n^2 + kn - k - 2n + 5}$$

we have

$$g(\sigma_k) = g(\eta_k) = \frac{3k-5}{n^2 - kn + k + 2n - 5} \quad \text{and} \quad \alpha_1 + \alpha_2 = 1. \tag{3.9}$$

Let

$$\lambda_1 = \frac{n^2 + 2kn - 3n - 2k}{k(3k-5)} \quad \text{and} \quad \lambda_2 = \frac{kn^2 - k^2 n - n^2 + k^2 + 3n - 3k}{k(3k-5)}.$$

Then $\lambda_1, \lambda_2 > 0$ and

$$\lambda_1 g(\sigma_k) + \lambda_2 g(\eta_k) = \alpha_1 + \alpha_2 = 1.$$

Comparing this with (3.9), we conclude that there are no values α_1, α_2 such that $\alpha_1 + \alpha_2 = 1$ and

$$g(\sigma_k), g(\eta_k) < \frac{3k-5}{n^2 - kn + k + 2n - 5}.$$

$\square$

Proof of theorem 2.3.2. The proof is similar to that of Theorem 2.2.2 (Section 3.6). Without loss of generality, we assume that the maximum value of f_0 is 1 and is attained at the identity permutation ε. Let g be the central projection of f_0. We deduce that g is a linear combination of $\chi_{n-1,1}$ and $\chi_{n-2,2}$ and hence a linear combination of g_1 and g_2, and that $g(\varepsilon) = 1$. Let us choose a $3 \leq k \leq n-3$ and let X_k be the set of permutations σ such that $p(\sigma) = k$ and $t(\sigma) = 0$ and let Y_k be the set of permutations η such that $p(\eta) = 0$ and $t(\eta) = k$. As in the proof of Theorem 2.3.2, we have

$$|X_k| \geq \frac{1}{5}\frac{n!}{k!}.$$

To choose a permutation $\eta \in Y_k$, one has to choose k transpositions (2-cycles) in $\dfrac{n!}{(n-2k)!k!2^k}$ ways and a permutation without fixed points or 2-cycles on the remaining $n-2k$ points. Hence, by Lemma 3.4.2

$$|Y_k| \geq \frac{1}{5}\frac{n!}{k!2^k}.$$

Let us choose a permutation $\sigma \in X_k$ and a permutation $\eta \in Y_k$ and let $Z = X_k$ if $g(\sigma) \geq g(\eta)$ and let $Z = Y_k$ otherwise. Then

$$|Z| \geq \frac{n!}{5k!2^k}$$

and by Lemma 3.7.2,

$$g(\sigma) \geq \frac{3k-5}{n^2 - kn + k + 2n - 5} \quad \text{for all} \quad \sigma \in Z.$$

The proof now proceed as in the proof of Theorem 2.2.2, Section 3.6. $\square$

Remark 3.7.3 (Scarcity of relatively good values). Let us consider the function f of Section 2.3.2. We observe that

$$f = \alpha_1 r_3 + \alpha_2 r_{7e},$$

for

$$\alpha_1 = \frac{n^2 - nm - 2n + 3m - 3}{n^2 - nm + 2n + m - 5} \quad \text{and} \quad \alpha_2 = \frac{4n - 2m - 2}{n^2 - nm + 2n + m - 5}.$$

Thus f is a convex combination of r_3 and r_{7e}, hence $1 = f(\varepsilon) \geq f(\sigma)$ for all $\sigma \in S_n$ and $\overline{f} = 0$. Remark 3.2.5 implies that f is a generalized function (1.2) of the required type.

3.8 Proof of General Case

The proof of Theorem 2.4.1 follows from our description of the central cone (Lemma 3.3.1), and the observations used in the proof of Theorem 2.2.1 in Section 3.6.

To prove Theorem 2.4.2, we need another lemma showing that the central projection has good values on at least one of several large classes of permutations.

Lemma 3.8.1. *Let g be a linear combination of $g_1 = p - 1$, $g_2 = p^2 - 2$ and $g_3 = 1 - 2t$ such that such that $g(\varepsilon) = 1$. For a $2 \leq k \leq n - 2$, let σ_k be a permutation such that $p(\sigma_k) = k$ and $t(\sigma_k) = 0$, let η be a permutation such that $p(\eta) = 0$ and $t(\eta) = 1$ and let θ be permutation such that $p(\theta) = t(\theta) = 0$. Then*

$$\max\{g(\sigma_k), g(\eta), g(\theta)\} \geq \frac{k - 2}{n^2 - kn + k - 2}.$$

Proof. We can write

$$g = \alpha_1 \frac{p - 1}{n - 1} + \alpha_2 \frac{p^2 - 2}{n^2 - 2} + \alpha_3(1 - 2t)$$

for some α_1, α_2 and α_3 such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Then

$$g(\sigma_k) = \alpha_1 \frac{k - 1}{n - 1} + \alpha_2 \frac{k^2 - 2}{n^2 - 2} + \alpha_3$$

$$g(\eta) = -\frac{\alpha_1}{n - 1} - \alpha_2 \frac{2}{n^2 - 2} - \alpha_3$$

$$g(\theta) = -\frac{\alpha_1}{n - 1} - \alpha_2 \frac{2}{n^2 - 2} + \alpha_3.$$

We observe that $g(\sigma_k)$, $g(\eta)$ and $g(\theta)$ are linear functions of α_1, α_2 and α_3 and that for

$$\alpha_1 = \frac{k(1-n)}{n^2 - nk + k - 2}, \quad \alpha_2 = \frac{n^2-2}{n^2 - nk + k - 2} \quad \text{and} \quad \alpha_3 = 0$$

we have

$$g(\sigma_k) = g(\eta) = g(\theta) = \frac{k-2}{n^2 - nk + k - 2} \quad \text{and} \quad \alpha_1 + \alpha_2 + \alpha_3 = 1. \tag{3.10}$$

Let

$$\lambda_1 = \frac{n^2 - 2n}{k^2 - 2k}, \quad \lambda_2 = \frac{n^2 - nk}{2k - 4} \quad \text{and} \quad \lambda_3 = \frac{n^2 - kn - 2n + 2k}{2k}.$$

Then $\lambda_1, \lambda_2, \lambda_3 > 0$ and

$$\lambda_1 g(\sigma_k) + \lambda_2 g(\eta) + \lambda_3 g(\theta) = \alpha_1 + \alpha_2 + \alpha_3 = 1.$$

Comparing this with (3.10), we conclude that there are no values α_1, α_2 and α_3 such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$ and

$$g(\sigma_k), g(\eta), g(\theta) < \frac{k-2}{n^2 - nk + k - 2}.$$

□

Proof of Theorem 2.4.2. The proof follows those of Theorem 2.2.2 (Section 3.6) and Theorem 2.3.2 (Section 3.7) with some modifications. Let X_k be the set of all permutations σ such that $p(\sigma) = k$ and $t(\sigma) = 0$. As in the proof of Theorem 2.2.2, we have

$$|X_k| \geq \frac{1}{5} \frac{n!}{k!}.$$

Let Y be the set of all permutations σ such that $p(\sigma) = 0$ and $t(\sigma) = 1$. To choose a permutation $\sigma \in Y$, one has to choose a 2-cycle in $\binom{n}{2}$ ways and then an arbitrary permutation of the remaining $(n-2)$ symbols without fixed points and 2-cycles.

Using Lemma 3.4.2, we estimate

$$|Y| \geq \frac{1}{5}\frac{n!}{2(n-2)!}(n-2)! = \frac{1}{10}n!.$$

Let us choose a permutation $\sigma_k \in X_k$, a permutation $\eta \in Y$ and a permutation $\theta \in X_0$. Let us choose Z to be one of X_k, X_0 and Y, depending where the maximum value of $g(\sigma_k)$, $g(\eta)$ or $g(\theta)$ is attained. Hence

$$|Z| \geq \frac{n!}{5k!}$$

and by Lemma 3.8.1,

$$g(\sigma) \geq \frac{k-2}{n^2-kn+k-2} \quad \text{for all} \quad \sigma \in Z.$$

We proceed now as in the proof of Theorem 2.2.2, Section 3.6. □

3.9 Examples of `QAP`'s and their Central Projections

We remarked in the statements of our distributional results (Theorems 2.2.1, 2.3.1 and 2.4.1), that, at least for even n, we can find a function of type (1.1) that has average value 0, maximum value 1, and attains the averages on the rings $U(\varepsilon, k)$ specified in the theorems. It is easy to build a tensor of type (1.2) that meets these conditions (even for n odd), for example by using linear combinations of the tensors introduced in Remark 3.2.5. In this section we construct examples of type (1.1) satisfying the same conditions. Stated another way, we will show that any function in the "even" central cone K of Section 3.3 (that is, the convex hull of r_1, r_2, r_3 and r_4) is the projection of some `QAP` of type (1.1) with average value 0 and maximum 1. This provides an interesting source of pathological `QAP`'s.

It is not obvious to us whether the annex to the central cone for n odd formed by adding r_{5o} to the list of generators is also the projection of a `QAP` of type (1.1) with the correct averages on the rings $U(\varepsilon, k)$ and maximum 1. It follows from Remark 3.3.2 that functions in the annex (and not on the boundary of the even cone) are not maximized at the identity for even n, since the average value of r_{5o} on the permutations which are a product of $n/2$ transpositions is too large. The results of Section 3.3.1 show that the additional part of the central cone for odd n is small and contains functions not very different from functions that lie in the central cone.

To construct our examples, we recall from Section 3.3 that the projection g of a function f with $\overline{f} = 0$ lies in a 3 dimensional vector space. Then we can determine g uniquely by determining its value on three independent coordinates. It is convenient to use the coordinates v_1, v_2, v_3 where v_1 is the value of g at the identity, ε, v_2 is the value of g on any 2-cycle, equal to the average value of f on 2-cycles, and v_3 is the value of g on any 3-cycle, equal to the average value of f on 3-cycles.

From our formulas for r_1, r_2, r_3 and r_4, it is easy to compute the values v_1, v_2, v_3 at these extreme points for the base of the central cone. We substitute $p = n$, $t = 0$ to get v_1, $p = n-2$, $t = 1$ for v_2 and $p = n-3$, $t = 0$ for v_3. Since the maximum of 1 is attained at the identity, we will have $v_1 = 1$ for each extreme point r_i. The full list of v_i's for each r_i are summarized in Table 3.1.

Value↓ Ray→	r_1	r_2	r_3	r_4
v_1	1	1	1	1
v_2	-1	-1	1	1
v_3	$\dfrac{-2n+7}{n-2}$	1	1	$\dfrac{n-3}{n-2}$

Table 3.1: Values of Extreme Rays at ε, on 2-cycles, and on 3-cycles

We now give our example, a `QAP` with data (A, B) that depends on four real parameters b, c, d, e. We will show that any point in the convex hull of r_1, r_2, r_3, r_4 is the central projection of some `QAP` defined by (A, B) for a suitable choice of the parameters. The matrix $A = (a_{ij})$ does not depend on b, c, d, e:

$$a_{ij} = \begin{cases} 1 & \text{if } \; i = 1, j = 2 \\ 0 & otherwise \end{cases} \qquad A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

The matrix $B = (b_{ij})$ is given by:

$$b_{ij} = \begin{cases} 0 & \text{if } \; i = j \\ 1 & \text{if } \; i = 1, j = 2 \\ b & \text{if } \; i = 2, j = 1 \\ c & \text{if } \; i = 1, j \geq 3 \text{ or } j = 2, i \geq 3 \\ d & \text{if } \; i = 2, j \geq 3 \text{ or } j = 1, i \geq 3 \\ e & \text{otherwise} \end{cases} \qquad B = \begin{pmatrix} 0 & 1 & c & c & c & \dots & c \\ b & 0 & d & d & d & \dots & d \\ d & c & 0 & e & e & \dots & e \\ d & c & e & 0 & e & \dots & e \\ d & c & e & e & 0 & \dots & e \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ d & c & e & e & e & \dots & 0 \end{pmatrix}$$

Then the objective function f given by (1.1) is:

$$f(\sigma) = \sum_{i,j=1}^{n} b_{\sigma(i)\sigma(j)} a_{ij} = b_{\sigma(1)\sigma(2)}$$

This problem has average value $\overline{f} = 0$ over all permutations exactly when:

$$1 + b + 2(n-2)c + 2(n-2)d + (n-2)(n-3)e = 0 \tag{3.11}$$

Clearly $f(\varepsilon) = 1$. This is a maximum of f if and only if:

$$b, c, d, e \leq 1 \tag{3.12}$$

In fact, since A is constant, the projection g of the `QAP` (A, B) is a linear function of the parameters b, c, d, e. Thus Equations 3.11 and 3.12 define a polytope in the parameter space. The central projection of the set of `QAP`'s (A, B) defined by this polytope is the set of g attainable from `QAP`'s of the above form. Since the maximum is 1 and the average is 0, this projection is a convex set contained in the base of the central cone. To show that we get the entire base (for even n), it is enough to show we get each of the extreme rays r_1, r_2, r_3, r_4. Then by scaling the matrix B, we will get the entire (even) cone.

We know that f has a maximum of $v_1 = 1$ at the identity. We compute the average of f on 2-cycles:

$$v_2 = \frac{2}{n(n-1)}\left(\frac{n^2-5n+6}{2} + 2(n-2)c + b\right)$$

and on 3-cycles:

$$v_3 = \frac{3}{n(n-1)(n-2)}\left(\frac{n^3-9n^2+26n-24}{3} + 2(n-2)(n-3)c + 2(n-2)d\right)$$

Now observe that if we take:

$$b, c, d = 1 \text{ and } e = \frac{-4n-6}{(n-2)(n-3)} \text{ then } v_2 = 1 \text{ and } v_3 = 1$$

So by our calculations in Table 3.1, the projection of f must be r_3.

If we take:

$$b, c, e = 1 \text{ and } d = \frac{-n^2+3n-4}{2(n-2)} \text{ then } v_2 = 1 \text{ and } v_3 = \frac{n-5}{n-2}$$

In this case the projection of f is r_4.

If we take:

$$b, d, e = 1 \text{ and } c = \frac{-n^2+3n-4}{2(n-2)} \text{ then } v_2 = -1 \text{ and } v_3 = \frac{-2n+7}{n-2}$$

In this case the projection of f is r_1.

And finally, if we take:

$$c, d, e = 1 \text{ and } b = -n^2+n+1 \text{ then } v_2 = -1 \text{ and } v_3 = 1$$

In this case the projection of f is r_2. This completes the proof.

Remark 3.9.1 (`QAP`*'s with few large values).* All the functions in this section attain their maximum of 1 at many points including the $(n-2)!$ permutations fixing both 1 and 2. While optimal values are abundant, we can choose B so that these are the only "large" values of the objective function. This then gives us examples where the number of near-optimal permutations is relatively small.

Consider the `QAP` given by setting the entries of B so that:

$$b = 1 \quad \text{and} \quad c = d = e = \frac{-2}{n^2 - n - 2}$$

This is a symmetric `QAP`, with average value 0 over all permutations. The maximum of 1 occurs at the $2(n-2)!$ permutations that fix the set $\{1, 2\}$. However, the remaining $(n^2 - n - 2)(n-2)!$ permutations given an objective value of $-2/(n^2 - n - 2)$, worse than the average value. This shows that the frequency estimate of δn^{-2} in Part (i) of Corollary 2.4.3 cannot be improved, even in the symmetric case.

Consider now choosing B so that:

$$b = 1, \quad c = d = \frac{-1}{n-2} \text{ and } e = \frac{2}{n^2 - 2n}$$

The central projection of the `QAP` defined by (A, B) has the bullseye distribution of Section 2.1. We can check that this has an average value of 0, and $2(n-2)!$ optimal values of 1. Most permutations will yield a positive objective value of $2/(n^2 - 2n)$, however only the optimal permutations will exceed this value. Thus the frequency estimate of δn^{-2} in Part (i) of Corollary 2.2.3 cannot be improved, even in the bullseye case.

CHAPTER IV

Concluding Remarks

In this chapter, we discuss how our methods apply to other optimization problems, and to the study of heuristics. We include a small sample of computational results that we compare to our picture of the distribution. We also show how to derandomize the algorithm of Corollary 2.2.3 in the bullseye case of Section 2.1. Finally, we state some open problems.

4.1 Related Optimization Problems

Our methods can be applied to study the distributions of other combinatorial optimization problems. In this section we discuss linear and higher dimensional assignment problems, and their special cases.

4.1.1 The Linear Assignment Problem

Using our methods to analyze the distribution of values in the Linear Assignment Problem (`LAP`) of Definition 1.1.4, we note that the central projection g of f of type (1.4) is spanned by the characters χ_n and $\chi_{n-1,1}$ from Section 3.2. So if we preprocess the function f (cf. Remark 3.1.2) so that it has average value 0 and maximum 1 at the identity ε, the central projection g of f around this maximum will be :

$$g(\sigma) = \frac{\chi_{n-1,1}(\sigma)}{n-1} = \frac{p(\sigma) - 1}{n-1}$$

where $p(\sigma)$ is the number of fixed points in σ (see Definition 1.1.9). Then g has the "bullseye"-type distribution that we suggest is characteristic of relatively easy optimization problems.

We noted in Definition 1.1.4 that if the central projection of a QAP is spanned by the characters χ_n and $\chi_{n-1,1}$, then that QAP reduces to a LAP, which is solvable in polynomial time.

4.1.2 Higher-dimensional Assignment Problems

We can also consider the higher dimensional assignment problems and their special cases. The k-dimensional assignment problem [Law63] is the problem of maximizing for some $2k$-dimensional tensor C the function:

$$f(\sigma) = \sum_{i_1,i_2,\ldots,i_k=1}^{n} c^{i_1 i_2 \ldots i_k}_{\sigma(i_1)\sigma(i_2)\ldots\sigma(i_k)} \tag{4.1}$$

Then the central projection g lies in 2^k-dimensional subspace spanned by characters that we could in principle find explicitly. The difficulty lies in working in this high dimensional space.

We cite two examples of higher dimensional assignment problems that are considered interesting optimization models.

4.1.3 The BiQuadratic Assignment Problem

The *BiQuadratic Assignment Problem* (BiQAP) is a 4-dimensional assignment problem, where the (8-dimensional) tensor C decomposes as a product of two (4-dimensional) tensors. That is, for some $A, B \in \mathbb{R}^4$, the tensor C in Equation (4.1) is:

$$c^{i_1 i_2 i_3 i_4}_{j_1 j_2 j_3 j_4} = a_{i_1 i_2 i_3 i_4} b_{j_1 j_2 j_3 j_4}$$

This problem arises in Very Large Scale Integrated circuit design (VLSI), see for example [BÇK94].

4.1.4 The Weighted Hypergraph Matching Problem

In the *Weighted Hypergraph Matching Problem* we are given a k-uniform hypergraph on a set of n vertices. The *edges* of such a hypergraph are subsets of size k; each edge e_i has an associated weight w_i. The problem is to find the maximum weight *matching*, or set of disjoint edges. Some applications of this problem are found in [Vem98].

We can reduce Weighted Hypergraph Matching to k-dimensional assignment much as we reduced `TSP` to `QAP`. We encode the incidence matrix of a maximum matching in $A = (a_{i_1 i_2 \ldots i_k})$:

$$a_{i_1 i_2 \ldots i_k} = \begin{cases} 1 & \text{if } \{i_1, \ldots, i_k\} = \{mk+1, \ldots, mk+k\} \text{ for some } 0 \leq m < \lfloor \frac{n}{k} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

and we encode the weights w_i of the edges e_i in $B = (b_{i_1 i_2 \ldots i_k})$:

$$b_{i_1 i_2 \ldots i_k} = \begin{cases} w_e / \binom{n}{k} & \text{if } \{i_1, \ldots i_k\} = \text{edge } e \text{ for some } e \in E \\ 0 & \text{otherwise} \end{cases}$$

Then this problem reduces to maximizing Equation (4.1) for the tensor C given by:

$$c^{i_1 i_2 \ldots i_k}_{j_1 j_2 \ldots j_k} = a_{i_1 i_2 \ldots i_k} b_{j_1 j_2 \ldots j_k}$$

4.2 Notes on Heuristics

As mentioned in Section 1.3.2, in light of the practical value and theoretical difficulty of the `QAP`, there has been an effort to find effective heuristics for the problem. There is a good survey of several heuristics in [BÇPP99]. In this section, we consider our results from the perspective of designing and analyzing heuristics. We are particularly interested in why, as we noted in Section 1.3, many heuristics have produced better results on `TSP` that on the general `QAP`.

Heuristic methods typically have embedded in them some type of “local search” that involves looking for new good permutations nearby known good permutations. This strategy is well suited to a “bullseye” type of distribution. Consider the extreme case of a tensor whose value is exactly:

$$2\chi_{n-2,2}(\sigma) = 2t(\sigma) + p^2(\sigma) - 3p(\sigma)$$

at any point (see Section 3.2 and Remark 3.2.5). The optimum is $n^2 - 3n$ at the identity, ε. Suppose we begin a local search at a permutation σ that agrees with the identity ε in k points. Then the objective value is $f(\sigma) = k^2 - 3k + 2t$, where t is the number of transpositions in σ. The local search procedure will consider permutations which differ from σ by a single 2-cycle.

If $t > 0$, σ transposes two elements of ε. Then replacing σ by the permutation σ' which swaps these two elements but otherwise agrees with σ, increases the number of fixed points by 2, while decreasing the number of transpositions by 1. As long as $k > 1$, this improves the objective value to $f(\sigma') = (k+2)^2 - 3(k+2) + 2(t-1) = k^2 + k + 2t - 4$, so local search will explore this direction. If $t = 0$, take σ' to differ from σ by a single transposition in such a way as to improving the number of points agreeing with ε by 1 (assuming $\sigma \neq \varepsilon$). This changes the objective value to at least $f(\sigma') \geq (k+1)^2 - 3(k+1) = k^2 - k - 2$. Then this offers an improving direction if $k > 1$ that is correct in the sense that it reduces the distance to the global optimum. Repeating this procedure (taking any improving direction at each step) finds the optimum in at most $n - k$ moves, assuming that we start with $k \geq 2$. It is not too difficult to choose a starting point which agrees with the optimum on at least 2 points.

Of course for interesting optimization problems, the objective values are not constant on rings around the optimum, improving directions can sometimes lead away from the global optimum, and the local search can get trapped at local maxima. We suggest that the case of the "spike" distribution seen in the `QAP`, but not the `TSP`, is much worse for local search – the search will *usually* move away from the global maximum, except when the permutation σ is very close to or very far away from the optimum. Since most permutations are far away, a local search will tend to begin and remain far away from the local maximum. It may be that in these cases, better search strategies rely heavily on random sampling.

The distribution of a typical function f around its maximum in the symmetric `QAP` is a mixture of the bullseye and spike distributions. If we examine the analysis in the proofs (for example, Section 3.7), we see that this type of distribution causes the most difficulty for our estimates – these are the distributions on which the class of permutations far from the optimum performs modestly, and the class of permutations that perform well near the optimum is relatively small. We will call such a distribution a "diluted spike", see Figure 4.1.

This interference of the bullseye and spike distributions (which, in some sense, are "pulling in the opposite directions"), provides in our opinion, a plausible explanation of the computational hardness of the general symmetric `QAP` even in comparison with other NP-hard problems such as `TSP`.

It seems that in practice, the most successful heuristics combine some aspects of local search and random choice. We mention two heuristics very briefly. The *Tabu search* strategy (detailed in [GL98]) is based on local search, but allows the solution to decrease at some steps. This algorithm stores information about where it has been

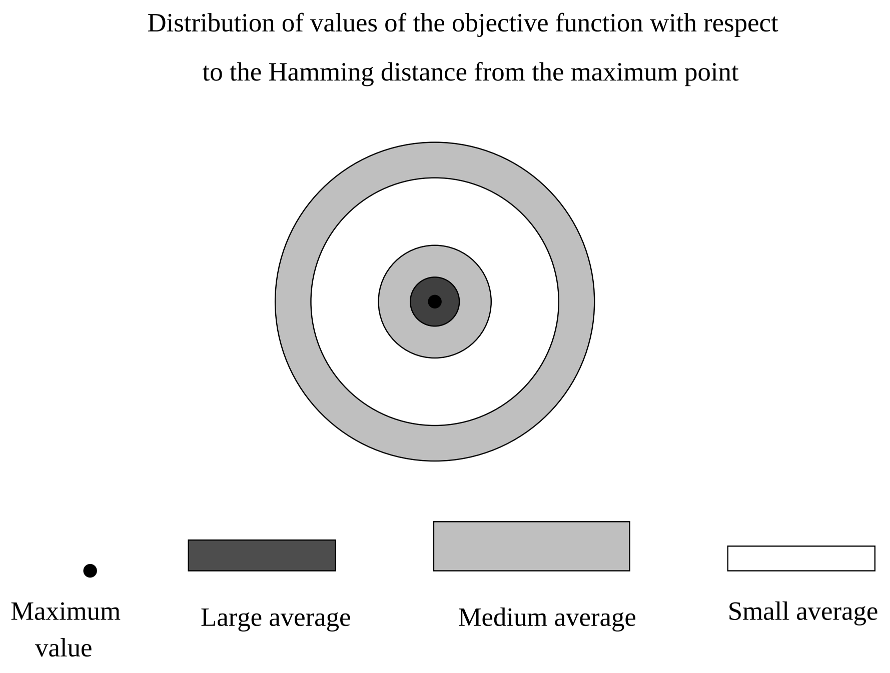

Figure 4.1: Diluted Spike Distribution

in order to prevent looping. Tabu search often makes the same decisions as a pure local search; decreasing the objective value at some steps can prevent the algorithm from getting stuck at a local maximum.

The *Greedy Randomized Adaptive Search Procedure* (GRASP) [LPR94] first constructs an initial permutation by randomly assigning some values, and then fixes the remaining ones by a greedy algorithm. Local search is then applied to the resulting permutation. This process can be repeated for many initial random assignments. The combination of random choice and local search seems to us to be a reasonable strategy for finding the optimum in a "diluted spike" distribution.

4.2.1 Proofs of Optimality

Finding the maximum of a general `QAP`, even with these heuristics is quite difficult. Finding a proof optimality is more difficult still, and typically follows the discovery of the optimum only after substantial additional time and effort (see for example

[ABGL02]). The optimality proofs usually rely on a *branch-and-bound* strategy.

The branch-and-bound strategy is to recursively partition the feasible set into "branches", and, for each branch, either search it completely, or show that the objective values in the branch are bounded above by some feasible solution. Branches that are bounded do not have to be searched, and can instead be "cut". For the `TSP` and `QAP`, the branching is typically done by fixing certain parts of the assignment. The idea is that if the fixed part of the assignment is wrong, then the branch can be cut.

Given a bullseye-type distribution, this approach can be quite successful. After finding some permutations with large objective values (ideally the optimum), it should be possible to cut many branches that lie in regions far from the optimum, and hence have a low average value.

In the situation of a spike-type distribution, it seems that it should be very difficult to implement an effective branch-and-bound procedure, since the numerous large values far from the maximum will appear in almost every branch. This leaves little room for error in the estimates of the maximum on each branch.

We remark that if we believe we have discovered the optimum, but lack an optimality proof, then, after shifting this point to the identity, we can calculate the central projection (see Section 3.9), and hence the distribution around the presumed optimum. We hope that this type of qualitative information would be useful in designing branch-and-bound algorithms.

4.3 Computational Results

In this section, we apply our methods to some examples that are considered to be of practical interest. We compute the central projection around the optimum, and sample points to gauge how well this approximates the distribution.

We restrict our attention to functions of type (1.1) for two reasons. First, as we remarked in Section 1.3, the generalized problem (1.2) is seen much less often in practice. Second, it follows from Remark 3.2.5 that any function in the central cone (Section 3.3) is actually an objective function of type (1.2). Then for objective functions of this form, the range of possible distributions is quite wide, and our estimates are tight. For functions of type (1.1) it is possible that the range of distributions is more restricted, and that our estimates could be improved. The computations in this section offer some data to fuel speculation.

A number of interesting examples of `QAP`'s of type (1.1) have been collected in the `QAPLIB` [BKR97]. These are the standard test sets for computational approaches to the `QAP`. As such, they have the advantage that many of them have been solved through years of effort. Unfortunately, it is still not feasible to solve `QAP`'s larger than about $n = 30$ to optimality. For larger problems, [LP92] suggests a method of generating `QAP`'s that have a known optimal solution, but are sufficiently generic for use in testing algorithms. We include some results on these generated problems in Section 4.3.3.

We also take examples from the Nugent, or `nug` series of problems introduced in [NVR68]. This series is well suited for our experiments, providing several related problems based on a simple structure. These problems have been among the most studied `QAP` test cases, and have now been solved [ABGL02]. They are derived from the type of "facility location" problem suggested by Koopmans-Beckmann (see Section 1.3). The problems involve placing factories on a given rectangular grid (say 5×6 for n=30) with Manhattan (L^1) distances. So the distance matrix A is symmetric, with zero diagonal, and small positive integer entries (at most 9 in the

5×6 case). The flow matrix B is also symmetric and consists of small positive integer entries. The original examples were for $n = 12, 15, 20, 30$. The remaining examples were created later by deleting rows and columns (factories) from the flow matrices, and building the distance matrices from new (sometimes non-rectangular) grids.

We classify the `nug` problems as part of the symmetric special case of Section 2.3, but, since the row and columns sums vary, not part of the pure special case of Section 2.2.

4.3.1 Central Projections for the `nug` Problems

To compute the central projection, we use the methods of Section 3.9. We begin by finding the global minimum τ, relying on the work compiled in the `QAPLIB`, and we shift the problem so that τ lies at the identity ε (see Remark 3.1.2). We are interested in finding out whether the shape of the central projection of the shifted problem is a bullseye distribution (Section 2.1), a spike distribution (Section 2.3), or a diluted spike with properties of both (Section 4.2). For this purpose, we translate the objective function f so that it has an average over all permutations of 0, and scale f so that it has a maximum (rather than a minimum) of 1 at the identity.

It is convenient to work with problems that are symmetric (such as the `nug`'s) and have even n. Then, after preprocessing, the central projection g of f must be a convex combination of the extreme functions r_3 and r_{7e} generating the even central symmetric cone K_s (see Section 3.7). That is, $g = \alpha_1 r_3 + \alpha_2 r_{7e}$, where $\alpha_1 + \alpha_2 = 1$. So we need to find α_1. Having fixed the value at the identity ε to be 1, we note that α_1 is completely determined by the value of g on any conjugacy class other than the class containing the identity. The smallest such class is the set of 2-cycles. We observe that r_3 has an average value of 1 on 2-cycles, while r_{7e} has an average of 0 on

2-cycles. So we conclude that in this case α_1 is exactly the average of g on 2-cycles.

For each even `nug` problem from `QAPLIB` (all $12 \leq n \leq 30$ except for $n = 26$), we have computed the parameter α_1 as outlined above using `MATLAB`. The results are presented in Table 4.1.

nug	12	14	16	18	20	22	24	28	30
α_1	0.694	0.733	0.774	0.802	0.828	0.844	0.850	0.874	0.882

Table 4.1: Values of Parameter α_1 on `nug` Problems

Note that the bullseye case corresponds to $\alpha_1 = 1$, and the spike case corresponds to $\alpha_1 = 0$ (see Remark 3.7.1). Thus the central projections of the `nug` problems lie in between these extremes and are a type of "diluted spike" (see Section 4.2). It turns out that they correspond very closely to the worst case for our estimates.

In the proof of the symmetric case (Section 3.7), we saw that there was a tradeoff between the strategies of picking permutations near the optimum and picking permutations far from the optimum. The most difficult case fell in the middle, where both strategies have equal (and relatively low) success. This transition point depends on the size n of the problem, and a parameter m. In Remark 3.7.3, we calculated for symmetric `QAP`'s the value of α_1 at which the transition occurs in terms of n and m:

$$\alpha_1 = \frac{n^2 - nm - 2n + 3m - 3}{n^2 - nm + 2n + m - 5}$$

From the algorithmic point of view, m is roughly a measure of the computational resources available to us. In the case of sampling, the value of m represents the number of points on which a permutation σ must agree with the optimum τ to have a sufficiently large expected value. By letting $k = m$ in Theorem 2.2.2, we get an approximation guarantee for sampling, similar to Corollary 2.4.3.

Consider fixing $\gamma = 1/2$ in Theorem 2.2.2. To ensure that the expected number of permutations meeting our guarantee is at least 1, the number of permutations we would have to sample is:

$$\frac{10m!2^m(n^2 - mn + m + 2n - 5)}{(3m - 5)}$$

This estimate holds for $m \geq 3$. At $m = 3$, the size of the sample is quite reasonable – $120n^2(1+o(1))$, but as we increase m the constant grows quickly. Given the computer power available today, we could consider using $m = 7$ (though we would not advocate doing so), where the size of the sample is $403200n^2(1+o(1))$, but probably not $m = 8$.

We calculate the α_1 value for the transition points (worst cases for Theorem 2.2.2) at both $m = 3$ and $m = 7$ in the Table 4.2. We remark that the parameter α_1 for

$n =$	12	14	16	18	20	22	24	28	30
At $m = 3$	0.692	0.733	0.765	0.790	0.810	0.826	0.840	0.862	0.871
At $m = 7$	0.628	0.688	0.730	0.763	0.788	0.809	0.825	0.851	0.862
α_1 for nug	0.694	0.733	0.774	0.802	0.828	0.844	0.850	0.874	0.882

Table 4.2: Transition Points for α_1 Compared to nug Parameters

the nug series of problems is very close to the parameter that gives the worst case central projection for our estimates.

4.3.2 Results of Sampling the nug Problems

For each nug problem we sampled 20000 points, and computed at each point the ratio of the objective value of the zero-average function f_0 to the global minimum, that is $f_0(\sigma)/f_0(\tau)$. The code we used is reproduced in Appendix A. At 20000 points the shape of the distribution does not change much if we vary the random seed (the code in Appendix A shows the seeds used in the results presented here). With this code running on math department hardware takes about 1 second to sample 500 nug points. The choice of 20000 points per problem represents the limit of our patience.

In Figures 4.2 and 4.3, we show histograms of the set of ratios obtained in the problems `nug12` and `nug30`. In both cases we see that the objective values are

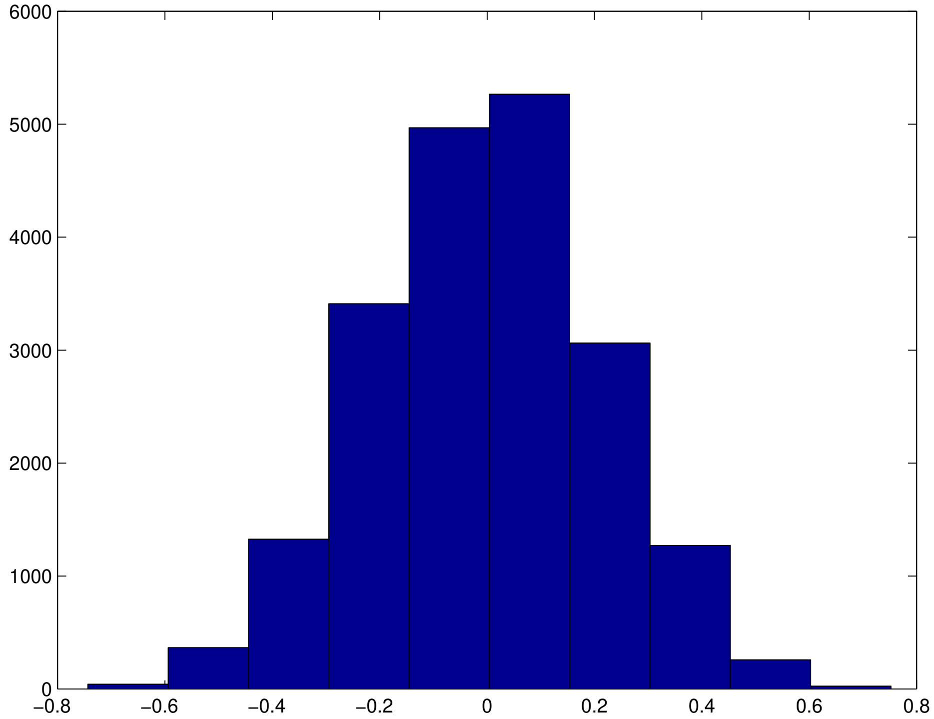

Figure 4.2: Frequency of Ratios $f_0(\sigma)/f_0(\tau)$ for `nug12`

concentrated near the average of 0, with the frequency of occurrence declining rapidly in either direction. The shapes of the distributions of `nug12` and `nug30` are similar, except that the `nug30` values are more tightly packed around the average of 0. The distribution is similar for the other `nug` problems, with the variance shrinking as n increases.

We have computed the standard deviation of the approximation ratios over all permutations in Table 4.4. This is obtained by dividing the standard deviation of the

$n =$	12	14	16	18	20	22	24	28	30
Std. Dev.	0.213	0.188	0.170	0.156	0.143	0.151	0.125	0.115	0.109

Table 4.3: Standard Deviation of Approximation Ratios for the `nug` Problems

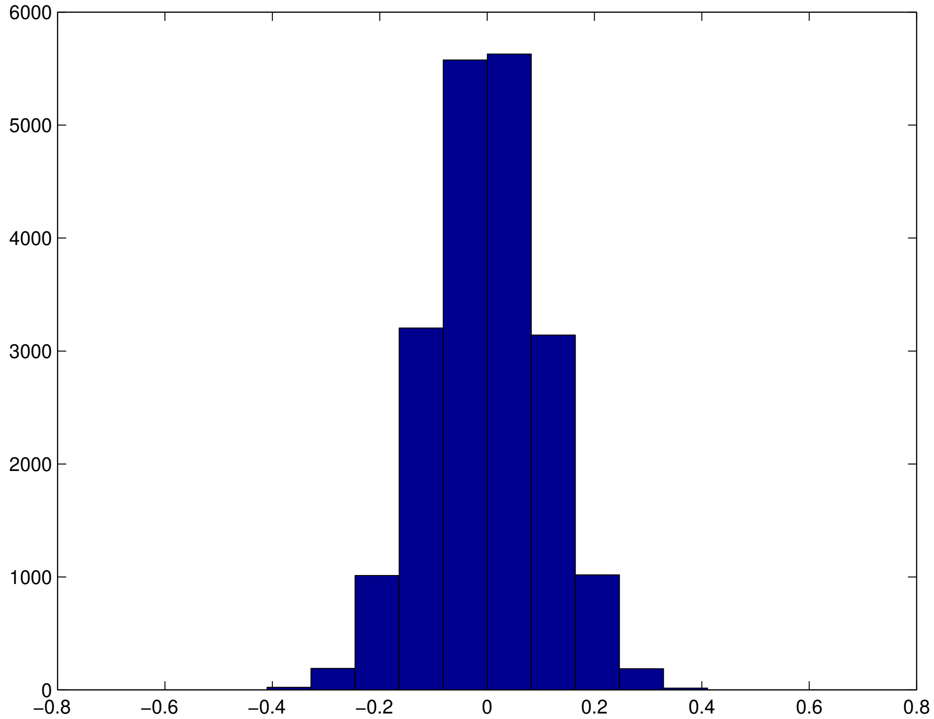

Figure 4.3: Frequency of Ratios $f_0(\sigma)/f_0(\tau)$ for `nug30`

solutions by the global maximum; a recipe for calculating the variance (and hence the standard deviation) of a `QAP` is found in [Bar].

In Table 4.3, we show the approximation ratios obtained by taking the best permutation from each sample. These approximation ratios clearly decline with n,

$n =$	12	14	16	18	20	22	24	28	30
Best Ratio	0.752	0.680	0.662	0.585	0.582	0.588	0.479	0.435	0.411

Table 4.4: Best Approximation Ratio Obtained in Samples for `nug` Problems

but are substantially better than the values guaranteed in Corollary 2.2.3.

We are interested in finding how the fraction of "good" values depends on n. To this end, we have, for each `nug` problem, computed the fraction of the 20000 sample values attaining an approximation ratio of:

$$\frac{f_0(\sigma)}{f_0(\tau)} \geq \frac{\gamma}{h(n)}$$

for $h(n) \in \{1, n, n^2, n^3\}$. The constant γ is chosen so that the fraction of `nug20` values above $\gamma/h(n)$ is constant. We took $\gamma/h(20) = 1/20$. We see that the fraction

$\frac{\gamma}{h(n)}$ ↓ n→	12	14	16	18	20	22	24	28	30
1/20	0.414	0.404	0.388	0.373	0.365	0.380	0.343	0.332	0.321
$1/n$	0.355	0.361	0.364	0.365	0.365	0.391	0.367	0.378	0.378
$20/n^2$	0.256	0.294	0.324	0.350	0.365	0.401	0.401	0.392	0.413
$400/n^3$	0.136	0.219	0.284	0.328	0.365	0.411	0.406	0.436	0.446

Table 4.5: Fraction of Permutations above $\gamma/h(n)$ for `nug` Problems

of permutations greater than a constant γ decreases as n increases, the fraction greater than γ/n stays roughly constant, and the fraction greater than γ/n^2 and γ/n^3 appears to increase towards 1/2. It appears that our estimates are fairly weak for this series of problems. It is interesting to ask if our estimates could be improved, either for all objective functions of type (1.1), or at least for classes of problems similar to the `nugs` with some modeling value.

We comment that `nug22` has more large values the one would predict based on the rest of the series. If we look at table 4.2, we can also see that `nug22` is further from the "worst case" and closer to the bullseye than is suggested by the remainder of the series.

4.3.3 Computational results for the `lipa` Problems

We repeated the computations performed on the `nug` problems on the `lipa` problems [LP92] from the `QAPLIB` [BKR97]. These problems are designed as large test cases for the `QAP`. They are constructed so as to have known optimal solution, but to otherwise resemble naturally occurring intractable `QAP`'s. There are two series of generated `lipa` problems in the QAPLIB. The problems in both series are symmetric, and have sizes $n = 20, 30, \dots, 90$. The "a" series, which we denote `lipa_a` has the entries of A drawn from $\{0, 1, 2\}$, and the entries of B drawn from $\{0, 1, \dots n\}$. For the "b" series, denoted `lipa_b`, the entries of both A and B are drawn from $\{0, 1, \dots n\}$.

In Table 4.6, we see that the parameter α_1 for the generated `lipa` problems are close matches for the "worst case" α_1 values for $m = 3$ and $m = 7$. It may be that

$n =$	20	30	40	50	60	70	80	90
At $m = 3$	0.810	0.871	0.902	0.922	0.934	0.944	0.951	0.956
At $m = 7$	0.788	0.862	0.897	0.918	0.932	0.942	0.949	0.955
α_1 for `lipa_a`	0.802	0.869	0.902	0.921	0.934	0.943	0.950	0.956
α_1 for `lipa_b`	0.808	0.870	0.903	0.922	0.934	0.944	0.951	0.956

Table 4.6: Transition Points for α_1 Compared to `lipa` Parameters

these "worst case" values are typical of generic (and presumably the most difficult) `QAP`'s. It is interesting to note that the generated `lipa` `QAP`'s and the `nug` `QAP`'s both have averages on rings around the optimum very close to the averages that are most difficult for our estimates.

In Table 4.7 we show the approximation ratio obtained by taking the best permutation from a random sample of 2000 on each `lipa` problem. These approximation

$n =$	20	30	40	50	60	70	80	90
For `lipa_a`	0.375	0.263	0.250	0.186	0.167	0.161	0.157	0.151
For `lipa_b`	0.336	0.181	0.147	0.116	0.101	0.085	0.085	0.055

Table 4.7: Best Approximation Ratio Obtained in Samples for `lipa` Problems

ratios decline with n as in the `nug` series, and are somewhat worse at comparable values.

We have also computed the variance for the `lipa` problems, and recorded the standard deviation of the approximation ratios over all permutations in Table 4.8.

We notice that in these cases, as well as the `nug` problems, the variance of the

$n =$	20	30	40	50	60	70	80	90
S.D. `lipa_a`	0.096	0.072	0.060	0.051	0.046	0.042	0.039	0.036
S.D. `lipa_b`	0.074	0.049	0.036	0.029	0.024	0.021	0.018	0.016

Table 4.8: Standard Deviation of Approximation Ratios for the `lipa` Problems

approximation ratio is decreasing with n. There is some difference in the variances between series at comparable sizes, this can likely be attributed to the structure of the problems. The standard deviations and best approximation ratios follow similar downward trends in all three sets of problems. From the point of view of sampling, it would make sense that having a small variance with respect to the maximum would make a problem difficult to approximate.

We can again test which fraction of the values attains a given approximation ratio, with the ratio possible depending on n. We use the setup of Section 4.3.2, Table 4.5, to get the `lipa` results in Table 4.9. We notice the same trends in the

$\frac{\gamma}{h(n)} \downarrow$ n$\rightarrow$	20	30	40	50	60	70	80	90
lipa_a 1/20	0.310	0.245	0.206	0.164	0.146	0.120	0.104	0.082
lipa_a $1/n$	0.310	0.324	0.344	0.348	0.365	0.364	0.377	0.377
lipa_a $20/n^2$	0.310	0.381	0.423	0.438	0.454	0.459	0.469	0.469
lipa_b 1/20	0.249	0.153	0.084	0.042	0.020	0.008	0.002	0.001
lipa_b $1/n$	0.249	0.249	0.250	0.243	0.250	0.243	0.240	0.242
lipa_b $20/n^2$	0.249	0.328	0.371	0.387	0.417	0.424	0.431	0.436

Table 4.9: Fraction of Permutations above $\gamma/h(n)$ for `lipa` Problems

`lipa` problems that we saw in the `nug` problems, with the `lipa` problems worse at comparable n, and the `lipa_b` worse than the `lipa_a`. Since the averages of the problems on rings around the optimum are very close together, the difference must come from variance within the rings. This is confirmed by comparing the standard deviations of the three series.

4.4 Derandomization

Another question is whether the approximations of Corollaries 2.2.3 and 2.4.3 can be obtained *deterministically.* In this section, we will show that in the bullseye case of Section 2.1, the algorithm of Corollary 2.4.3 (picking points at random) can be derandomized in polynomial time.

Our algorithm has two steps. The first step is to find a class of permutations with large average (expected) value. Then we apply a standard derandomization technique to find a permutation in this class which achieves this expected value. In [GY02], the authors use the second step of this algorithm on the `QAP`. They compute a permutation σ which has $f_0(\sigma) \geq 0$, and then show that if n is a prime power, the domination number of σ (see Section 1.3.1) is at least $(n-2)!$. They call the process of finding a permutation that meets the expected value on a class the "Greedy Expectation Algorithm".

We begin with some definitions. Let us take a subset S of size m. We will call a function $\sigma_S : S \to \{1, 2, \ldots n\}$ a *partial assignment* if it assigns to each $i \in S$ a distinct element of $\{1, 2, \ldots n\}$. So σ_S is the restriction of a permutation to S. We denote by $\sigma_S(S)$ the set of elements $\{\sigma_S(s) | s \in S\}$. Consider the set of all possible partial assignments of S into the set $\{1, 2, \ldots n\}$. There are $n_{(k)} := n(n-1)\ldots(n-k+1)$ possible such assignments, one of which is the restriction of the optimal permutation τ to S.

Given a partial assignment σ_S the *conditional expectation of σ given σ_S* is the average of the objective function over all σ that restrict to σ_S in their first S positions:

$$E\left[\sigma | \sigma_S\right] = \sum_{\substack{\sigma \in S_n \\ \sigma_{|S} = \sigma_S}} f(\sigma)$$

Using the observations of Lemma 3.1.1, we can compute the conditional expectation in time $O(n^2)$ for objective functions of type (1.1) and in time $O(n^4)$ for objective function of type (1.2). This calculation is also done in [GY02]. We include the formula for the expectation at the end of this section, as Lemma 4.4.1.

We now describe our algorithm. Suppose we have a bullseye `QAP`, and we want to produce a permutation σ satisfying the estimate of part (i) of Corollary 2.2.3. That is, suppose we take f_0 as in Section 2.1, where f_0 is maximized at some permutation τ not known to us, and we want to find, for some α chosen in advance, a permutation σ satisfying:

$$f_0(\sigma) \geq \frac{\alpha}{n^2} f_0(\tau)$$

We first need to identify a large class of permutations on which the average of f_0 is at least $\frac{\alpha}{n^2} f_0(\tau)$. Let us fix a $k \geq 3$ such that:

$$\frac{k^2 - 3k + \nu(n-k)}{n^2 - 3n} \geq \frac{\alpha}{n^2}$$

We can do this independently of n with $k = O\left(\sqrt{\alpha}\right)$. Then by Theorem 2.1.1, the set of permutations that agree with τ on at least k points has at least this average value.

In particular, for at least one subset S_0 of the $n_{(k)}$ subsets of n, the set of permutations agreeing with τ on S_0 has average value at least α/n^2. There are $n_{(k)}$ possible partial assignments of S_0, one of which agrees with τ on S_n. Thus we can get a partial assignment σ_{S_0} that has conditional expectation at least α/n^2 by enumerating the $n_{(k)}$ possible assignments of the $n_{(k)}$ k-subsets of $\{1, 2, \ldots n\}$.

Now we will extend the partial assignment σ_{S_0} into a permutation σ which at least meets the expectation, using the Greedy Expectation Algorithm. In this procedure, we assign the values of $\{1, 2, \ldots n\} \setminus S_0$ sequentially using conditional expectation. Let $S \supseteq S_0$ be the set of currently assigned values, and σ_S be the corresponding partial assignment. If l is the lowest remaining unassigned value (that is, $\min\{\{1,2,\ldots n\} \setminus S\}$), we compute the conditional expectations for each possible assignment of $\sigma(l)$. At least one of these must meet the overall expectation of $\sigma(S)$. We choose such a value (say, the largest), and fix this to be the value of $\sigma(l)$.

By iterating this procedure, we build a permutation with expected value at least α/n^2. The total time for this procedure is $O(n_{(k)} \cdot n_{(k)} \cdot n^2 + n^2 \cdot n \cdot n) = O(n^{2k+2})$ for objective functions of type (1.1), and $O(n^{2k+4})$ for objective functions of type (1.2). In fact, if we are lucky, it will be much faster, since we can stop search the partial assignments once we have any one that yields a conditional expectation of α/n^2.

We can use the same algorithm to obtain a permutation deterministically that meets the estimate of part (ii) of Corollary 2.2.3 in mildly exponential time $\exp\{n^\beta\}$ for some $\beta < 1$.

It does not appear to be as easy to derandomize the pure, symmetric and general cases of Section 2.2, 2.3 and 2.4. The difficulty is finding a partial assignment with large conditional expectations.

We finish by giving the formula for the conditional expectation of f given a partial assignment σ_S. This is derived in [GW70].

Lemma 4.4.1.

$$\begin{aligned}
E[\sigma|\sigma_S] &= \sum_{i\in S}\sum_{j\in S} c^{ij}_{\sigma_S(i)\sigma_S(j)} + \frac{1}{n-k}\sum_{i\in S}\sum_{j\notin S}\sum_{j'\notin\sigma_S(S)} c^{ij}_{\sigma_S(i)j'} \\
&\quad + \frac{1}{n-k}\sum_{i\notin S}\sum_{j\in S}\sum_{i'\notin\sigma_S(S)} c^{ij}_{i'\sigma_S(j)} + \frac{1}{n-k}\sum_{i\notin S}\sum_{i'\notin\sigma_S(S)} c^{ii}_{i'i'} \\
&\quad + \frac{1}{(n-k)(n-k-1)}\sum_{\substack{i,j\notin S\\ i\neq j}}\sum_{\substack{i',j'\notin\sigma_S(S)\\ i'\neq j'}} c^{ij}_{i'j'}
\end{aligned}$$

4.5 Further Questions

The estimates of Theorems 2.1.1, 2.2.1, 2.3.1 and 2.4.1 for the number of near-optimal permutations can be used to bound the optimal value by a sample optimum in branch-and-bound algorithms. Those estimates are (nearly) best possible for the generalized problem (1.2). However, it is not clear whether they can be improved in the case of standard `QAP` (1.1) or how to improve them in interesting special cases. In particular, we ask the following question:

- Let $f : S_n \longrightarrow \mathbb{R}$ be the objective function in the Traveling Salesman Problem (cf. Section 1.2), let $\overline{f}$ be the average value of f and let $f_0 = f - \overline{f}$. Let τ be an optimal permutation, so that $f_0(\tau) \geq f_0(\sigma)$ for all $\sigma \in S_n$. Is it true that for any fixed $\gamma > 0$ there is a number $\delta = \delta(\gamma) > 0$ such that the probability that a random permutation $\sigma \in S_n$ satisfies the inequality $f_0(\sigma) \geq \frac{\gamma}{n} f_0(\tau)$ is at least $n^{-\delta}$ for all sufficiently large n?

The small sample of results in Section 4.3 suggests that the answer is "yes". One way that we could try to tighten the analysis is to use information from the standard deviation and higher moments of the distribution which are computable in polynomial time.

In [Bar], there is a preliminary results in this direction, obtained by relating the L^∞ norm of a function to its L^{2k} norm (that is, its $2k$-th moment):

Theorem 4.5.1 (Barvinok). *For any $\alpha > 0$, there exists a $\mu = \mu(\alpha) > 0$ such that the fraction of permutations satisfying:*

$$|f_0(\sigma)| \geq \frac{\alpha}{n} |f_0(\tau)|$$

is at least $n^{-\mu}$. In fact, one can choose $\mu = c\alpha^2$ for some absolute constant c.

This shows that we have enough permutations with sufficiently large deviation, however it tells us nothing about the direction of the deviation. It could mean that we have a many small values (close to the minimum), rather than large values.

As well as finding improved approximation algorithms, it would be very interesting to find corresponding hardness results with an ultimate goal of proving sharp bounds. No hardness of approximation results with respect to the average are known.

We would like to be able to derandomize our algorithm in the general case of Section 2.4. In Section 4.4, we showed how we could derandomize our algorithm in the bullseye case of Section 2.1.

It appears that the problems without linear part (the “bullseye” and “pure” cases of Sections 2.1 and 2.2) are easier than the general `QAP`. A possible way to use this is to split the problem into the pure quadratic and linear parts and try to “estimate away” the linear part.

APPENDICES

APPENDIX A

MATLAB code for sampling permutations

In this appendix, we include the code used to sample permutations and calculate the central projections and standard deviations of the nug and lipa problems. The results are presented in Section 4.3. Our results should be reproducible, since we fix the random seed before each run. The code is written in MATLAB, and grouped into eight subroutines, each in separate files.

The driver is file test_shell.m.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% test_shell.m %%%%%%%%%%%%%%%%%%%%%%%%%
%
% Reads a QAP of two MxM matrices A and B from the data file
% <solved_prb_data.m>.  This also includes a vector giving the
% optimal permutation P; for now we concentrating on the properties
% of some examples that are either contrived or well studied.
%
% The loop then tests "trials" permutations at random, and returns
% statistics about the best and worst values found, as well as some
% facts about the neighbourhood of the optimum, which we believe
% helps us understand the difficulty of the problem.
% We also call a routine to compute the variance of the values.
%
%
trials= 20000;
values_vec = zeros(1,trials);   % Collect approximation ratios.

% Reset random number state to get reproducible results.
randn('state',0)

% Input data.
```

```
[M,raw_A,raw_B,P] = solved_prob_data;
conj_a = perm_2_matrix(M,P);    % Use to get min at id.
A = conj_a*raw_A*conj_a';       % Now min is at id.
B = normalize_b_fast(M,A,raw_B);% Now average is 0.
global_min = trace(A*B');       % Calculate global_min.
total_val2=0;
relax_max=relaxed_max(A,B);     % Gives an upper bound for problem.
% relax_min=relaxed_min(A,B);   % Lower bound.
% max_obj=trace(relax_min*A*relax_min'*B');
min_obj=trace(relax_max*A*relax_max'*B');

compare_at = 20;             % Test values at this problem size.
gamma = 1/20;                % Parameter.
hits = [0 0 0 0];            % Count occurrences of "good" values.
for i=1:trials
  Z2 = rand_perm(M);
  obj_val = trace(Z2*A*Z2'*B');
  values_vec(i) = obj_val;
  if obj_val < min_obj
    min_obj = obj_val;
    best_perm = Z2;
  end
  if (obj_val < global_min*gamma)
    hits(1) = hits(1) +1;
  end
  if (obj_val < global_min*gamma*compare_at/M)
    hits(2) = hits(2) +1;
  end
  if (obj_val < global_min*gamma*compare_at^2/M^2)
    hits(3) = hits(3) +1;
  end
  if (obj_val < global_min*gamma*compare_at^3/M^3)
    hits(4) = hits(4) +1;
  end
  total_val2 = total_val2 + obj_val;
end

% Print statistics
```

```
format compact

% Show best permutation obtained;
% best_perm;
% Sanity check - should average to about 0.
Average_of_random_permutations = total_val2/trials
global_min
min_for_random_permutations = min_obj
Approx_ratio_for_min = min_obj/global_min
Freq_of_good_values_vec = hits/trials
avg_on_2cy = sum_on_2cy(M,A,B)*2/M/(M-1);% Avg. f value on 2-cycles
Scaled_avg = avg_on_2cy/global_min  % shape of dist. (\alpha_1).
% hist(values_vec/global_min)
fvar = find_var_zd(M,A,B);
Calculated_std_dev = sqrt(fvar);
Normalized_csd = -Calculated_std_dev/global_min

format loose
```

File perm_2_matrix.m converts a permutation in standard notation to a permutation matix. Standard notation is used in the input file.

```
%%%%%%%%%%%%%%%%%%%%%%%%%% perm_2_matrix.m %%%%%%%%%%%%%%%%%%%%%%%%
%
function Z = perm_2_matrix(M,ord_a)
%
% Convert a permutation on M numbers from standard notation
% to a permuation matrix Z.
% This matrix acts on the left by multiplication.
%
p1 = zeros(M);
for i = 1:M
  p1(i,ord_a(1,i)) = 1;
end
Z = p1';
```

File normalize_b_fast.m calculates the average of the problem over all permuataions, and then subtracts a constant from the entries of B so that the problem now has average zero. If we use the new B in place of the old B, we “shift” the distribution so that the average value over all permutations is zero.

```
%%%%%%%%%%%%%%%%%%%%%%%%%% normalize_b_fast.m %%%%%%%%%%%%%%%%%%%%%%%%
%
function B = normalize_b_fast(M,A,raw_B)
alpha1 = 0;
beta1 = 0;
alpha2 = 0;
beta2 = 0;
for i=1:M
  alpha2 = alpha2 + A(i,i);
  beta2 = beta2 + raw_B(i,i);
  for j=1:M
    alpha1=alpha1+A(i,j);
    beta1=beta1+raw_B(i,j);
  end
end
alpha1=alpha1-alpha2;
beta1=beta1-beta2;
avg_qap_val = alpha1*beta1/M/(M-1)+alpha2*beta2/M;
avg_c = alpha1+alpha2;
if (abs(alpha1+alpha2) > 0.0001)
  normalizing_factor = avg_qap_val/avg_c;
  normalizing_b = normalizing_factor*ones(M,M);
else
  normalizing_factor = 0;
  Problem_is_prenormalized=1   % Problem may already be normalized
  normalizing_b = zeros(M,M);
end;
B=raw_B-normalizing_b;
```

File `relaxed_max.m` produces an upper bound for the problem. The lazy way to do this is just to hard code a large number, like 10^{10}.

```
%%%%%%%%%%%%%%%%%%%%%%%%%% relaxed_max.m %%%%%%%%%%%%%%%%%%%%%%%%
%
function X = relaxed_max(A,B)
%
% Given diagonalizable matrices A and B, this function should
% return orthogonal matrix X maximizing <XAX',B'>.
% It does this by transforming the problem to the diagonal version,
```

```
% and then finding a permutation which puts the elements in the same
% order.  Because diagonalizing uses orthogonal matrices, we can
% go back and construct X.
%
[L0, B1] = eig(B');
L1=L0';
[L2, A1] = eig(A);
[discard, ord_a] = sort(diag(A1));
[discard, ord_b] = sort(diag(B1));
M = size(ord_a,1);
p1 = zeros(M);
p2 = zeros(M);
for i = 1:M
  p1(i,ord_a(i,1)) = 1;
  p2(i,ord_b(i,1)) = 1;
end
Z = p1'*p2;       % So now Z'*A1*Z*B1 is maximized over perms
X = L1'*Z'*L2';
```

File rand_perm.m makes a random permutation by choosing a vector of (Gaussian) random numbers, and taking the permutation that sorts this vector into ascending order.

```
function Z = rand_perm(M)
%%%%%%%%%%%%%%%%%%%%%%%%%%% rand_perm.m %%%%%%%%%%%%%%%%%%%%%%%%
%
%
% Generates a random permutation matrix of size M.
%
rv = diag(randn(M,1));
[discard, ord_a] = sort(diag(rv));
p1 = zeros(M);
for i = 1:M
  p1(i,ord_a(i,1)) = 1;
end
Z = p1;
```

File sum_on_2cy.m sums the values of the objective function over all 2-cycle. From the discussion in Section 4.3, in the case where A is symmetric, this is sufficient to determine the central projection of the distribution around the identity.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%% sum_on_2cy.m %%%%%%%%%%%%%%%%%%%%%%%
%
function Z = sum_on_2cy(M,A,B)
%
% Sums the values of tr(X*A*X'*B') over 2-cycles (transpositions) X.
% Inputs A and B are M by M matrices, the X's are matrix
%  representations of a permutation.
%
Z = 0;                  % Sum of values of f on 2-cycles
for i=1:M
  for j=i+1:M
    Tr=ones(1,M);                % This 2-cycle.
    for k=2:M
      Tr(1,k)=k;                 % initialize
    end
    Tr(1,j)=i;
    Tr(1,i)=j;
    Trm=perm_2_matrix(M,Tr);
    Z = Z + trace(Trm*A*Trm'*B');
  end
end
```

The file `find_var_zd.m` computes the sum of the squares of the objective values over all permutations of a `QAP` (A, B). Since the problem has been normalized so that the average objective value is zero, this sum is exactly the variance.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%% find_var_zd.m %%%%%%%%%%%%%%%%%
%
% This function calculates the variance of the QAP given by (A,B)
% in n^4 time.  In this routine, we assume the diagonals of A and B
% are zero.  The routine find_var also keeps track of diagonal terms.

function cvar = find_var_zd(M,A,B)

pa1111 = 0;
pa211b = 0;
pa211c = 0;
pa211d = 0;
pa211e = 0;
```

```
pa22b = 0;
pa22c = 0;
pa4 = 0;
pb1111 = 0;
pb211b = 0;
pb211c = 0;
pb211d = 0;
pb211e = 0;
pb22b = 0;
pb22c = 0;
pb4 = 0;
for i=1:M
  for j=1:M
    if (j ~= i)
      for k=1:M
        if (k ~= i & k ~= j)
          for l=1:M
            if (l ~=i & l~=j & l~=k)
              pa1111 = pa1111 + A(i,j)*A(k,l);
              pb1111 = pb1111 + B(i,j)*B(k,l);
            end
          end
          pa211b = pa211b + A(i,j)*A(i,k);
          pa211c = pa211c + A(i,j)*A(k,i);
          pa211d = pa211d + A(j,i)*A(i,k);
          pa211e = pa211e + A(j,i)*A(k,i);
          pb211b = pb211b + B(i,j)*B(i,k);
          pb211c = pb211c + B(i,j)*B(k,i);
          pb211d = pb211d + B(j,i)*B(i,k);
          pb211e = pb211e + B(j,i)*B(k,i);
        end
      end
      pa22b = pa22b + A(i,j)*A(j,i);
      pa22c = pa22c + A(i,j)*A(i,j);
      pb22b = pb22b + B(i,j)*B(j,i);
      pb22c = pb22c + B(i,j)*B(i,j);
    end
  end
```

```
  pa4 = pa4 + A(i,i)*A(i,i);
  pb4 = pb4 + B(i,i)*B(i,i);
end
s1 = pa1111*pb1111/M/(M-1)/(M-2)/(M-3);
s2 = (pa211b*pb211b+pa211c*pb211c+pa211d*pb211d+pa211e*pb211e)/M/(M-1)/(M-2);
s3 = (pa22b*pb22b+pa22c*pb22c)/M/(M-1);
s4 = pa4*pb4;
cvar = s1+s2+s3+s4;
```

Finally we give a sample input file, `solved_prob_data.m`. This is the data for the `nug12` problem.

```
function [M,A,B,P] = solved_prob_data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% solved_prob_data_nug12.m %%%%%%%%%%%%%%%%%%%%%%
%
% Contains the data for an instance of QAP to try.
% M is the dimension of the problem, A and B are the input matrices.
% Want to minimize trace(Z*A*Z'*B') where Z is a permutation matrix.
% P is the optimal permutation, found by years of hard work by
% many researchers.
%
% This data is known as "nug12"
%
M=12;
A=[0 1 2 3 1 2 3 4 2 3 4 5;
1 0 1 2 2 1 2 3 3 2 3 4;
2 1 0 1 3 2 1 2 4 3 2 3;
3 2 1 0 4 3 2 1 5 4 3 2;
1 2 3 4 0 1 2 3 1 2 3 4;
2 1 2 3 1 0 1 2 2 1 2 3;
3 2 1 2 2 1 0 1 3 2 1 2;
4 3 2 1 3 2 1 0 4 3 2 1;
2 3 4 5 1 2 3 4 0 1 2 3;
3 2 3 4 2 1 2 3 1 0 1 2;
4 3 2 3 3 2 1 2 2 1 0 1;
5 4 3 2 4 3 2 1 3 2 1 0];
B=[0  5  2  4  1  0  0  6  2  1  1  1;
5  0  3  0  2  2  2  0  4  5  0  0;
2  3  0  0  0  0  0  5  5  2  2  2;
```

```
4  0  0  0  5  2  2 10  0  0  5  5;
1  2  0  5  0 10  0  0  0  5  1  1;
0  2  0  2 10  0  5  1  1  5  4  0;
0  2  0  2  0  5  0 10  5  2  3  3;
6  0  5 10  0  1 10  0  0  0  5  0;
2  4  5  0  0  1  5  0  0  0 10 10;
1  5  2  0  5  5  2  0  0  0  5  0;
1  0  2  5  1  4  3  5 10  5  0  2;
1  0  2  5  1  0  3  0 10  0  2  0];
P=[12,7,9,3,4,8,11,1,5,6,10,2];
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[ABCC98] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. On the solution of traveling salesman problems. In *Proceedings of the International Congress of Mathematicians, Vol. III*, pages 192–202. Documenta Mathematica, Berlin, 1998.

[ABCC01] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. `TSP` cuts which do not conform to the template paradigm. In M. Junger and D. Naddef, editors, *Computational Combinatorial Optimization*, pages 261–304. Springer, 2001.

[ABGL02] K. Anstreicher, N. Brixius, J.-P. Goux, and J. Linderoth. Solving large quadratic assignment problems on computational grids. *Math Programming B*, 91(3):563–588, 2002.

[ACG$^+$99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation. Combinatorial Optimization Problems and their Approximability Properties.* Springer-Verlag, Berlin, 1999.

[AHS00] E. Arkin, R. Hassin, and M. Sviridenko. Approximating the maximum quadratic assignment problem. *Inform. Process. Lett.*, 77(1):13–16, 2000.

[AHU74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The design and analysis of computer algorithms.* Addison-Wesley Pub. Co., Reading, Mass., 1974.

[Aro98] S. Arora. Polynomial-time approximation schemes for euclidean `TSP` and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[AZ01] E. Angel and V. Zissimopoulos. On the landscape ruggedness of the quadratic assignment problem. *Theoret. Comput. Sci.*, 263(1-2):159–172, 2001. Combinatorics and computer science (Palaiseau, 1997).

[Bar] A. Barvinok. Estimating l^∞ norms by l^{2k} norms for functions on orbits. *Foundations of Computational Mathematics*, (to appear).

[BÇK94] R. Burkard, E. Çela, and B. Klinz. On the biquadratic assignment problem. In *Quadratic Assignment and Related Problems (New Brunswick, NJ, 1993)*, pages 117–146. Amer. Math. Soc., Providence, RI, 1994.

[BÇPP99] R. Burkard, E. Çela, P. Pardalos, and L. Pitsoulis. The quadratic assignment problem. In D.-Z. Du and P.M. Pardalos, editors, *Handbook of Combinatorial Optimization vol. 3*, pages 241–337. Kluwer Academic Publishers, 1999.

[BCSS97] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation.* Springer-Verlag, New York, 1997.

[BKR97] R. Burkard, S. Karisch, and F. Rendl. `QAPLIB` - a quadratic assignment problem library. *Journal of Global Optimization*, 10:391–403, 1997. Problems and solutions maintained on-line at: `http://www.opt.math.tu-graz.ac.at/qaplib`.

[Blä02] M. Bläser. An 8/13-approximation algorithm for the asymmetric maximum `TSP`. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 64–73. Association for Computing Machinery, 2002.

[BMCP98] A. Brüngger, A. Marzetta, J. Clausen, and M. Perregaard. Solving large scale quadratic assignment problems in parallel with the search library ZRAM. *Journal of Parallel and Distributed Computing*, 50:157–66, 1998.

[CJMZ01] J. Cirasella, D. S. Johnson, L. A. McGeoch, and W. Zhang. The asymmetric traveling salesman problem: Algorithms, instance generators, and tests. In A. L. Buchsbaum and J. Snoeyink, editors, *ALENEX 2001 Proceedings*, pages 32–59. Springer Lecture Notes in Computer Science 2153, 2001.

[Dia88] P. Diaconis. *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics, Hayward, CA, 1988.

[FH91] W. Fulton and J. Harris. *Representation Theory: A First Course*. Springer-Verlag, New York, 1991.

[GJ83] I. Goulden and D. M. Jackson. *Combinatorial Enumeration*. John Wiley & Sons, Inc., New York, 1983.

[GL98] F. Glover and M. Laguna. Tabu search. In *Handbook of Combinatorial Optimization, Vol. 3*, pages 621–757. Kluwer Acad. Publ., Boston, MA, 1998.

[GW70] G.W. Graves and A. B. Whinston. An algorithm for the quadratic assignment problem. *Management Science*, 17(7):452–471, 1970.

[GY02] G. Gutin and A. Yeo. Polynomial approximation algorithms for the `TSP` and the `QAP` with a factorial domination number. *Discrete Applied Mathematics*, 119(1-2):107–116, 2002.

[HR00] R. Hassin and S. Rubinstein. Better approximations for max `TSP`. *Information Processing Letters*, 75(4):181–186, 2000.

[JK81] G. James and A. Kerber. *The Representation Theory of the Symmetric Group*. Addison-Wesley Publishing Co., Reading, Mass., 1981. With a foreword by P. M. Cohn, With an introduction by G. de B. Robinson.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103, New York, 1972. Plenum.

[KB57] T. C. Koopmans and M. J. Beckmann. Assignment problems and the location of economic activity. *Econometrica*, pages 53–76, 1957.

[Kuh55] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, 2:83–97, 1955.

[Law63] E. L. Lawler. The quadratic assignment problem. *Management Science*, 9:586–599, 1963.

[LP92] Y. Li and P. M. Pardalos. Generating quadratic assignment test problems with known optimal permutations. *Computational Optimization and Applications*, 1:163–184, 1992.

[LPR94] Y. Li, P. M. Pardalos, and M. G. C. Resende. A greedy randomized adaptive search procedure for the quadratic assignment problem. pages 237–261, 1994.

[LS] M. Lewenstein and M. Sviridenko. An 5/8-approximation algorithm for the maximum assymetric TSP. Submitted.

[MP91] D. L. Miller and J. F. Pekny. Exact solution of large asymmetric travelling salesman problems. *Science*, 231(4995):754–761, 1991.

[Mur38] F. D. Murnaghan. *The Theory of Group Representations*. Johns Hopkins Press, Baltimore, 1938. Reprinted by Dover Publications Inc., New York, 1963.

[NVR68] C. E. Nugent, T. E. Vollmann, and J. Ruml. An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16(1):150–173, 1968.

[Que86] M. Queyranne. Performance ratio of polynomial heuristics for triangle inequality quadratic assignment problems. *Oper. Res. Lett.*, 4(5):231–234, 1986.

[Ser84] A. I. Serdyukov. An algorithm with an estimate for the traveling salesman problem of the maximum. *Upravlyaemye Sistemy*, 25:80–86, 1984. (in Russian).

[Vem98] R. R. Vemuganti. Applications of set covering, set packing and set partitioning models: a survey. In *Handbook of Combinatorial Optimization, Vol. 1*, pages 573–746. Kluwer Acad. Publ., Boston, MA, 1998.

[Ye99] Y. Ye. Approximating quadratic programming with bound and quadratic constraints. *Math. Programming*, 84(2):219–226, 1999.

ABSTRACT

The Distributions of Values in Combinatorial Optimization Problems

by

Tamon Stephen

Chair: Alexandre I. Barvinok

We study the distribution of objective function values of a combinatorial optimization problem defined on a group, focusing on the Quadratic Assignment Problem (`QAP`), and its special case, the Traveling Salesman Problem (`TSP`). For these two problems, we estimate the fraction of permutations σ such that $f(\sigma)$ lies within a given neighborhood of the optimal value of f, and relate the optimal value to the average value of f over a neighborhood of the optimal permutation. We describe a natural class of `QAP` functions (which includes, for example, the objective function in the asymmetric Traveling Salesman Problem) with a relative abundance of near-optimal permutations. Also, we identify a large class of functions f with the property that permutations close to the optimal permutation in the Hamming metric of the symmetric group S_n tend to produce near optimal values of f (such is, for example, the objective function in the symmetric Traveling Salesman Problem). We show examples of `QAP's` where just the opposite happens: an average permutation in the vicinity of the optimal permutation may be much worse than an average permutation

in the whole group S_n.

We interpret our results algorithmically, obtaining guarantees for simple polynomial and non-polynomial algorithms, and in the context of heuristics. Additionally, we compare our results to distributional statistics obtained via computational experiments.