Average case performance of heuristics for multi-dimensional assignment problems

Alan Frieze∗ Gregory B. Sorkin†

April 27, 2010

Abstract

We consider multi-dimensional assignment problems in a probabilistic setting. Our main results are: (i) A new efficient algorithm for the 3-dimensional planar problem, based on enumerating and selecting from a set of “alternating-path trees”; (ii) A new efficient matching-based algorithm for the 3-dimensional axial problem.

1 Introduction

A (two-dimensional) assignment can be viewed as a set of pairs \( P = \{ (i_t, j_t), t = 1, 2, \ldots, n \} \) such that

\[
\{ i_1, i_2, \ldots, i_n \} = \{ j_1, j_2, \ldots, j_n \} = [n],
\]

i.e., each row appears once in \( P \), as does each column (and without loss of generality we may take \((i_1, \ldots, i_n) = (1, \ldots, n)\)). Given an \( n \times n \) matrix of costs \( C = [C_{i,j}] \), the aim is to compute \( P \) that minimises \( C(P) = \sum_{(i,j) \in P} C_{i,j} \). This is a well-studied problem from the worst-case as well as the probabilistic point of view and it is solvable in polynomial time.

In the standard probabilistic model for the assignment problem each entry \( C_{i,j} \) is independently distributed as the exponential random variable with mean one, viz., \( \text{Exp}(1) \). There are numerous results related to this model: If \( Z_n \) is the minimum value of \( C(P) \) then we have the remarkable result, conjectured by Parisi [13],

\[
\mathbb{E}(Z_n) = \sum_{i=1}^{n} \frac{1}{i^2}.
\]

This was proved by Linusson and Wästlund [11] and Nair, Prabhakar and Sharma [12]. See Wästlund [14] for a remarkably short proof of this. Earlier work proving that \( \lim_{n \to \infty} \mathbb{E}(Z_n) = \zeta(2) \) was done by Aldous [1], [2].

In this paper we are concerned with the probabilistic analysis of multi-dimensional generalisations of this problem. We consider two versions, Planar and Axial. Let us first consider the \( d \)-dimensional Planar model. Here we are given an \( n \times \cdots \times n \) \( d \)-dimensional matrix (tensor) \( C = [C_{i_1,i_2,\ldots,i_d}] \), i.e., a map \( C : [n]^d \to \mathbb{R} \). An assignment is a set of \( n \) \( d \)-tuples \( T = \{ (i_{t}^1, i_{t}^2, \ldots, i_{t}^d), t = 1, 2, \ldots, n \} \) such that, in analogy with (1), for each dimension \( r \in [d] \), \( \{ i_{1}^r, i_{2}^r, \ldots, i_{n}^r \} = [n] \). Geometrically, for \( r \in [d] \), let an \( r \)-plane be a set of \( d \)-tuples of the form \([n]^{r-1} \times \{x\} \times [n]^{d-r} \) for some \( x \in [n] \). In the case of \( d = 2 \) a plane corresponds to a row or column of matrix \( C \). An assignment

∗Research supported by NSF grant DMS-6721878, Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh PA15213, e-mail alan@random.math.cmu.edu
†Department of Mathematical Sciences, IBM T.J. Watson Research Center, Yorktown Heights NY 10598, e-mail sorkin@watson.ibm.com
T is then a collection of d-tuples such that each of the dn planes contain exactly one d-tuple from T.

The optimisation problem here is to compute an assignment $T$ that minimises $C(T) = \sum_{(i_1, i_2, \ldots, i_d) \in T} C_{i_1, i_2, \ldots, i_d}$. This problem is NP-hard for $d \geq 3$ and the case of $d = 3$ is one of the original problems listed in Karp [10]. Some of its characteristics and applications are discussed in a recent book by Burkard, Dell’Amico and Martello [3]. Very little is known about the probabilistic behavior of the minimum $Z_{d,n}^P$ of $C(T)$ for $d \geq 3$. Grundel, Oliveira, Pasiliao and Pardalos [8] show that $Z_{d,n}^P \to 0$ whp in this case. At this point we can give some easy results on $Z_{d,n}^P$ which we state as

**Theorem 1**

$$\Omega \left( \frac{1}{n^{d-2}} \right) \leq Z_{d,n}^P \leq O \left( \frac{\log n}{n^{d-2}} \right).$$

(All proofs are given in the body of the paper.) The upper bound in this theorem is non-constructive, relying on recent work of Johansson, Kahn and Vu [9]. Our main result concerns a Bounded Depth Tree Search algorithm $\text{BDTS}(k)$. Here $k$ is a parameter that refers to the number of levels of search. It is unfortunate, but our approach only seems to give something interesting for $d = 3$.

**Theorem 2** Suppose that $1 \leq k \leq \gamma \log_2 \log n$ where $\gamma$ is any constant strictly less than 1/2. Then, whp

(a) Algorithm $\text{BDTS}(k)$ runs in time $O(n^{2k+2})$.

(b) The cost of the set of triples $T$ output by $\text{BDTS}(k)$ satisfies

$$C(T) = O(2^k n^{-1+\theta_k} \log n)$$

where $\theta_k = \frac{1}{2^{k+1}-1}$.

Note that for $k$ such as $\frac{1}{3} \log_2 \log n$ this is a “mildly exponential” running time, $n^{O(\log n)}$, yielding a solution which is an $O(n^{1/\log n})$ approximation to the optimum. After dealing with the Planar version, we will turn to the Axial version. Here we are again given an $n \times \cdots \times n$ d-dimensional matrix $C$. Geometrically, let a line be a set of d-tuples of the form $\{i_1\} \times \cdots \{i_r\} \times [n] \times \{i_{r+2}\} \times \cdots \{i_d\}$ for some $r$ and $i_1, \ldots, i_r, i_{r+2}, \ldots, i_d$. In the case of $d = 2$ a line corresponds to a row or column of matrix $C$. An assignment $T$ is then a collection of $n^{d-1}$ d-tuples such that each of the $dn^{d-1}$ lines contains exactly one d-tuple from $T$.

The optimisation problem here is to compute an assignment $T$ that minimises $C(T) = \sum_{(i_1, i_2, \ldots, i_d) \in T} C_{i_1, i_2, \ldots, i_d}$. This problem is NP-hard for $d \geq 3$ and this was proved in Frieze [6]. We will prove

**Theorem 3** The optimal solution value $Z_{d,n}^A$ satisfies the following:

(a) $Z_{d,n}^A = \Omega(n^{d-2})$ whp for $d \geq 3$.

(b) When $d = 3$ there is a polynomial time algorithm that finds a solution with cost $Z$ where $Z = O(n \log n)$ whp.

This leaves the following open questions:

P1 What are the growth rates of $E[Z_{d,n}^P]$ and $E[Z_{d,n}^A]$ for $d \geq 3$?
P2 Are there asymptotically optimal, polynomial time algorithms for solving these problems when $d \geq 3$.

P3 Frieze [5] gave a bilinear programming formulation of the 3-dimensional planar problem. There is a natural heuristic associated with this formulation (see appendix). What are its asymptotic properties?

1.1 Structure of the paper

We deal with the Planar version in Section 2. We start with the proof of Theorem 1 in Section 2.1. Our next task is to analyse BDTS. We will analyse a three level version in Section 2.2. This provides intuition for the general case, analysed in Section 2.3, completing the proof of Theorem 2. The Axial problem is considered in Section 3. The lower bound in Theorem 3 is proved in Section 3.1 and the upper bound in Section 3.2.

2 Multi-Dimensional Planar Version

2.1 Proof of Theorem 1

Clearly

$$Z_{d,n}^P \geq \sum_{i_1=1}^{n} \min_{i_2, \ldots, i_d} C_{i_1, \ldots, i_d}.$$

Each term in the above sum is distributed as $\text{Exp}(n^{d-1})$ and so has expectation $1/n^{d-1}$ and variance $1/n^{2d-2}$. The Chebyshev inequality implies that the sum is concentrated around the mean.

For the upper bound we use a recent result of Johansson, Kahn and Vu [9]. This implies that whp there is a solution that only uses $d$-tuples of weight at most $K\frac{\log n}{n^{d-1}}$. The upper bound follows immediately. It should be noted that their proof is non-constructive.

2.2 Two Level Version of BDTS

In this section we consider a two level version of the algorithm BDTS. In this way we hope that to make it easier to understand the general version that is described in Section 2.3. With reference to Theorem 2, the two-level version means taking $k = 3$, $\theta = \theta_3 = 1/7$.

The heuristic has three phases:

2.2.1 Greedy Phase

The first phase is a simple greedy procedure.

Greedy Phase

1. Let $n_1 = n - n^{1-\theta}$, $J = K = [n]$, and $T = \emptyset$.

2. For $i = 1, \ldots, n_1$ do the following:

   • Let $C_{i,j,k} = \min \{C_{i,j',k'} : j' \in J, k' \in K\}$.
   • Add $(i, j, k)$ to $T$ and remove $j$ from $J$ and $k$ from $K$.

\[^{1}\text{We will often pretend that some expressions are integer. Formally, we should round up or down but it will not matter.}\]
At the end of this procedure the triples in $T$ provide a partial assignment. Let

$$Z_1 = \sum_{(i,j,k) \in T} C_{i,j,k}.$$  

**Lemma 4**

$$Z_1 \leq \frac{2}{n^{1-\vartheta}} \text{ whp.}$$

**Proof.** We observe that if $(i, j, k) \in I$ then $C_{i,j,k}$ is the minimum of $(n-i+1)^2$ independent copies of Exp(1) and is therefore distributed as Exp$((n-i+1)^2)$. Furthermore, the random variables $C_{i,j,k}, (i, j, k) \in T$ are independent. Using the facts that an Exp $\lambda$ random variable has mean $1/\lambda$ and variance $1/\lambda^2$,

$$\mathbb{E}(Z_1) = \sum_{i=1}^{n^1} \frac{1}{(n-i+1)^2} \leq \int_{x=1}^{n^1+1} dx (n-x+1)^2 \leq \frac{1}{n^{1-\vartheta}}.$$  

Now

$$\text{Var}(Z_1) = \sum_{i=1}^{n^1} \frac{1}{(n-i+1)^4} \leq \frac{3}{n^{3(1-\vartheta)}} = o(\mathbb{E}(Z_1)^2)$$

and the lemma follows from the Chebyshev inequality. \hfill \square

### 2.2.2 Main Phase

The aim of this phase is to increase the size of the partial assignment defined by $T$ to $n-O(1)$. Let $I = I(T)$ be the set of first coordinates assigned in $T$, i.e., $I = I(T) = \{i : \exists j, k \text{ s.t. } (i, j, k) \in T\}$. Relabeling if necessary, without loss of generality we may assume that $I = |T|$. This phase will be split into rounds. We choose a small constant $0 < \alpha < 1$ and let $\beta = 1 - \alpha$. The aim of a round is to reduce the size of the set of unmatched first coordinates $X(T) = [n] \setminus I(T)$ by a factor $\beta$ while increasing the total cost of the matching only by an acceptably small amount. Thus we let $x_1 = n - n_1$ and $x_t = \beta^{t-1} x_1$ for $t \geq 2$. The aim of round $t$ is to reduce $|X(T)|$ from $x_t$ to $x_{t+1}$. We continue this for $t_0 = \log_{1/\beta}(x_1/L)$ rounds where $L$ is a large positive constant. Thus at the end of the Main Phase, if successful, we will have a partial assignment of size at least $n - 2L$.

So suppose now that we are at the start of a round and that $|X(T)| = x_t$. This is true for $t = 1$. Next let $w_0 = 2n^{-12/7} \log n$ and

$$w_t = 2n^{-6/7} x_t^{-8/7} \log^{1/7} n \quad \text{ for } t \geq 1.$$  

At the start of each round we will refresh the array $C$ with independent exponentials, at some cost. By this we mean that we replace $C$ by a new array $C'$ where $C'_{i,j,k} \leq C_{i,j,k} + w_{t-1}$ and the entries of $C'$ are i.i.d. Exp(1) random variables. More precisely, suppose that during the previous round we determined the precise values for all $C_{i,j,k} \leq w_{t-1}$ and left our state of knowledge for the other $C_{i,j,k}$ as being at least $w_{t-1}$. Then the memoryless property of exponentials means that

$$C'_{i,j,k} = \begin{cases} 
C_{i,j,k} - w_{t-1} & \text{ when } C_{i,j,k} > w_{t-1} \\
\text{fresh } X_{i,j,k} \sim \text{Exp}(1) & \text{ otherwise}
\end{cases}$$

has the claimed property. Thus we can start a round with a fresh matrix of independent exponentials at the expense of adding another $w_{t-1}$ to each cost. We note also that we can whp carry out the Greedy Phase only looking at those $C_{i,j,k}$ of value less than $w_0$. 

4
Let $T_t$ denote the value of $T$ at the start of round $t$ and let $I_t = I(T_t)$, $X_t = X(I_t)$. In round $t$ we will add $A_t = [n - x_t + 1, n - x_{t+1}]$ to $I_t$. By relabeling if necessary we will assume that at the start of round $t$ we have $T = \{(i, i, i) : 1 \leq i \leq n - x_t\}$. To add $i \in A_t$ to $I_t$ we find distinct indices $j, k, p, q, r, s \in I_t$ (distinctness is not strictly necessary) and replace 6 of the triples in $I_t$ by 7 new triples:

$$+ (i, j, k) - (j, j, j) - (k, k, k) + (j, p, q) + (k, r, s) - (p, p, p) - (q, q, q) - (r, r, r) - (s, s, s) + (p, \xi_1, \xi_2) + (q, \xi_3, \xi_4) + (r, \xi_5, \xi_6) + (s, \xi_7, \xi_8),$$

where $\xi_1, \ldots, \xi_8$ are distinct members of $X_t$, and each of the triples added in (2) is required to have (refreshed) cost at most $w_t$. Roughly, we are assigning a new 1-coordinate $i$, this collides with previously used 2-coordinate $j$ and 3-coordinate $k$, so the $(j, j, j)$ and $(k, k, k)$ elements are removed from the existing assignment, 1-coordinates $j$ and $k$ are re-added as $(j, p, q)$ and $(k, r, s)$ thus colliding with the previous assignment elements $(p, p, p)$, $(q, q, q)$, $(r, r, r)$, and $(s, s, s)$, and finally 1-coordinates $p, q, r, s$ are re-added as $(p, \xi_1, \xi_2)$ etc., where the $\xi_i$ are the elements not previously assigned. One may think of (2) as a binary tree version of an alternating-path construction; we will control the cost despite the tree’s expansion.

Putting $W_t = w_0 + w_1 + \cdots + w_t$ we see that if we can add one element to $T$ at a cost of at most $w_t$ in refreshed costs, then in reality it costs us at most $W_t$; step (2) increases the cost by $\leq 7W_t$. Success in a round means doing this $x_t - x_{t+1}$ times, in which case the additional cost of the Main Phase will be at most 7 times

$$\sum_{t=1}^{t_0} (x_t - x_{t+1})W_t \leq x_1(w_0 + w_1) + \sum_{t=2}^{t_0} x_tw_t$$

$$\leq 3n^{-6/7} \log n + 2x_1^{-1/7} n^{-6/7} \log^{1/7} n \sum_{t=2}^{t_0} \beta^{-t/7} \leq 4n^{-6/7} \log n. \quad (3)$$

We must now show that whp it is possible to add $x_t - x_{t+1} = \alpha x_t$ triples in round $t$ with a (refreshed) cost of at most $7w_t$ per triple. For this we fix $t$ and drop the suffix $t$ from all quantities that use it. We will treat refreshed costs as actual costs and drop the word “refreshed”.

We start by estimating the number of choices for assigning $p$. Ignoring other indices, the number of choices is distributed as the binomial $\text{Bin}(\nu, 1 - e^{-wx^2}) = \text{Bin}(\nu, (1 - o(1))wx^2)$ where $\nu = n - x$.

Here $1 - e^{-wx^2}$ is the probability that for a given $p$, there exist $\xi_1, \xi_2$ such that $C_{p, \xi_1, \xi_2} \leq w$. Note that

$$wx^2 = 2(x/n)^6/7 \log^{1/7} n = o(1) \text{ and that } wnx^2 \gg \log n$$

and so the Chernoff bounds imply that, qs, we can choose a set $P$ of size exactly $wnx^2/2 = o(n)$, such that for each $p \in P$ there is at least one choice $\xi_1, \xi_2 \in X$ such that the triple $(p, \xi_1, \xi_2)$ is good, i.e., $C_{p, \xi_1, \xi_2} \leq w$. Given this set of choices $P$ we find that the number of choices for $q \notin P$ is distributed as the binomial $\text{Bin}(\nu - |P|, 1 - e^{-wx^2})$ and we can once again qs choose a set $Q$, disjoint from $P$ such that $|Q| = wnx^2/2$ and each $q \in Q$ is in some good triple $(q, \xi_3, \xi_4)$ where $\xi_3, \xi_4 \in X$. Similarly, we can choose sets $R, S$ of choices for $r, s$, of size $wnx^2/2$, such that $P, Q, R, S$ are pairwise disjoint.

**Observation 5** Each $\xi \in X$ is in $\text{Bin}(nxw, 1 - e^{-w})$ good triples of the form $(p \in P, \xi', \xi'')$ and so qs it is in at most

$$2wnx = \frac{4n^{1/7} \log^{1/7} n}{x^{1/7}}$$

---

*A sequence of events $E_n, n \geq 0$ are said to occur quite surely, qs, if $\Pr(E_n) = 1 - O(n^{-K})$ for any constant $K > 0$. 

5
such triples.

We now discuss our choices for \( j \) and \( k \). For a fixed \( j \) there are \( w^2 n^2 x^4 / 4 \) pairs in \( P \times Q \) and each has a probability \( 1 - e^{-w} \) of forming a good triple \((j, p, q)\). Let \( j \) be *useful* if there is such a pair and *useless* otherwise. Then

\[
\Pr(\text{\( j \) is useless}) \leq \exp\left\{ -\frac{w^3 n^2 x^4}{4} \right\} \leq 1 - \frac{w^3 n^2 x^4}{5}.
\]

It follows that the number of useful \( j \not\in Y = P \cup Q \cup R \cup S \) dominates \( \text{Bin}(n - o(n), w^3 n^2 x^4 / 5) \) and so \( qs \) we can choose a set \( J \) of useful \( j \not\in Y \) of size

\[
\frac{w^3 n^3 x^4}{6} = \frac{4n^{3/7} x^{4/7} \log^{3/7} n}{3} = o(n).
\]

We can by a similar argument choose a set \( K \) of useful \( k \) of this size disjoint from \( J \) and \( Y \).

**Observation 6** A fixed \( p \) is in at most \( \text{Bin}(wn^2 x^2 / 2, 1 - e^{-w}) \) good triples \((j, p, q)\) where \((j, q) \in J \times Q\) and so \( qs \) every \( p \) is in at most \( w^2 n^2 x^2 \) such triples.

Suppose then that in the middle of a round we have added \( y < \alpha x \) triples to \( T \). The number of \( \xi \in X \) that can be used in a good triple \((p, \xi, \eta)\) will have been reduced by \( y \). The number of \( \eta \) will have been reduced by the same amount. It follows from Observation 5 that the number of choices for \( p \) will have been reduced by at most \( 2\alpha x \times 2wnx \). By Observation 6 this reduces the number of choices for \( j \) by at most \( 2\alpha x \times 2wnx \times w^2 n^2 x^2 + 7\alpha x \ll |J| = w^3 n^3 x^4 / 6 \). The additional term \(+7\alpha x \) accounts for the choices we lose because they have previously been used in this round. So our next \( i \) will get a choice of at least \( \text{Bin}((w^3 n^3 x^4 / 7)^2, 1 - e^{-w}) \) choices for a good triple \((i, j, k)\).

So the expected number of choices is at least \( w^{7} n^{6} x^{8}/49 = (2^7/49) \log n \) and then the probability there is no choice is \( o(n^{-1}) \). This is sufficient to ensure that \text{whp} there is always at least one choice for every \( i \).

### 2.2.3 Final Phase

We now have to add only \( O(1) \) indices to \( I \). At this point there is a problem with the bottom-up approach of the previous phase if \( x < 8 \), clearest in the case \( x = 1 \), say the single element \( n \), when each of \( \xi_1, \ldots, \xi_8 \) would have to be \( n \), leading to an illegal assignment. Thus instead we will work top down. The details of this will cause more conditioning of the matrix, and therefore we refresh \( C \) after each increase in \( I \), at an extra cost of \( w = Kn^{-6/7} \log^{1/7} n \). So, if successful, the cost of this round is \( O(W_{t_0} + w) = O(n^{-6/7} \log^{1/7} n) \).

Let us now replace the notation of (2) by

\[
+(i, j, k) - (j_1, j_3) - (k_1, k_2, k) + (j_1, p, q) + (k_1, p, r) - (p_1, p, p_3) - (q_1, q_2, q) - (r_1, r, r_3) - (s_1, s_2, s) + (p_1, i_2, p_3) + (q_1, q_2, j_3) + (r_1, s_2, i_3) + (s_1, k_2, r_3),
\]

where any subtracted triple such as \((j_1, j_3)\) denotes a previous match (we are no longer assuming the convention that such a triple would be \((j, j, j)\)), and where \(i_2, i_3\) are unused 2- and 3-coordinates respectively.

Fix \( j \) (and thus its previously matched companion indices \( j_1, j_3 \)) and let \( Z_j \) be the number of choices for \( p, q \) (with their previously matched companion indices \( p_1, p_3, q_1, q_2 \)) such that \( C(j_1, p, q), C(p_1, i_2, p_3), C(q_1, q_2, j_3) \leq w \). This has the distribution \( B_1(B_2(n, w)B_3(n, w), w) \) where
2.3 General 3-Dimensional Version

We follow the same three phase strategy. \( k \) is a positive integer, \( 2 \leq k \leq \gamma \log \log n \).

2.3.1 Greedy Phase

This is much as before. Proceed as in Section 2.2.1 but taking \( \theta = \theta_k \) (recall \( \theta \)’s definition from Theorem 2) and defining \( n_1 \) accordingly. Lemma 4 continues to hold.

2.3.2 Main Phase

Let

\[ \alpha = 2^{-2k-2} \left(1 - \sqrt{2/3}\right) \]

and let \( \beta, t_0 \) and \( x_t, t = 1, \ldots, t_0 \) be defined as in Section 2.2.2. Let \( I_t, X_t, A_t \) have the same meaning as well. Now let \( w_0 = 2n^{-2(1-\theta_k)} \log n \) and

\[ w_t = 2x_t^{-1-\theta_k}n^{-1} \log_{\theta_k} n, \quad \text{for } t \geq 1 \]

and

\[ W_t = w_0 + w_1 + \cdots + w_t = O \left( \frac{\log_{\theta_k} n}{n^{1-\theta_k}} \right). \]

The aim of round \( t \) is once again to add \( x_t - x_{t+1} \) new indices to \( I_t \) using triples with (refreshed) cost at most \( w_t \). We will assume that at the start of round \( t \) we have \( T = \{(i, i, i) : 1 \leq i \leq n - x_t\} \).

In analogy with (2), to add \( i \in A_t \) to \( I_t \) we will add \( 2^{k+1} - 1 \) triples to \( T \) and remove \( 2^{k+1} - 2 \) triples, in which case the additional cost of the Main Phase will be at most \( 2^{k+1} - 1 \) times

\[ \sum_{t=1}^{t_0} (x_t - x_{t+1})W_t \leq x_1(w_0 + w_1) + \sum_{t=2}^{t_0} x_t w_t \]

\[ \leq 3n^{\theta_k-1} \log n + 2x_1^{-\theta_k}n^{-1} \log_{\theta_k} n \sum_{t=2}^{t_0} \beta^{-\theta_k t} \leq 4n^{\theta_k-1} \log n. \]

The notation used in (2) is obviously insufficient. We imagine a rooted tree \( \Gamma \) of triples. The root will be \( r = (i_0, j_0, k_0) \) where \( i_0 \) is the index to be added to \( I_t \). The root is at level zero. The triples at odd levels are to be deleted from \( T \) and the vertices at even levels are to be added to \( T \). Every triple at an odd level \( 2l - 1 \) will therefore have the form \((p, p, p)\) where \( p \in I_t \). This triple will have one child \((p, a, b)\) which will replace the parent triple in 1-plane \( p \). If \( l < k \) then \( a,b \in I_t \) and if \( l = k \) then \( a,b \in X_t \). A triple \( u = (p, a, b) \) at an even level will have two children. By construction, \( u \) will be the unique triple in 1-plane \( p \), but now we will have two triples in 2-plane \( a \) and 3-plane \( b \). Thus the children of \( u \) are \((a,a,a)\) and \((b,b,b)\). This defines a tree corresponding to adding \( 2^{k+1} - 1 \) and
removing $2^{k+1} - 2$ triples from $T$. We ensure that if $u = (p, a, b)$ is a triple at an even level, then $p, a, b$ do not appear anywhere else in the tree, except at the child of $u$ as previously described. We do this so that additions in one part of the tree do not clash with additions in another part and then the additions and deletions give rise to a partial assignment. We also insist that if $u = (p, a, b)$ is a triple at an even level then $C_{p, a, b} \leq w$. We call such a tree feasible. We considered each level of $\Gamma$ to be ordered so it makes sense to talk of the $r$th vertex of level $2l$ where $1 \leq r \leq 2^l$.

We now have to show that whp there is always at least one such tree $\Gamma$ for each $i \in A_t$. We take the same bottom-up approach that we did in Section 2.2. We fix $t$ and drop the suffix $t$ from all quantities that use it. We start by estimating the number of choices for a $p$ that can be in a triple $(p, x, y)$ at level $2k$. Ignoring other indices, the number of choices is again distributed as the binomial $\text{Bin}(n, 1 - e^{-wx^2}) = \text{Bin}(\nu, (1 - o(1))nx^2)$ where $\nu = n - x = n - o(n)$. Note that $\nu x^2 = K(x/n)^{1-\theta_k} \log \theta_k n = o(1)$ and that $\nu x^2 = \Omega(n^{\theta_k}) \gg \log n$. (Here our notation $f(n) \gg g(n)$ means that $f(n)/g(n) \to \infty$ with $n$). So the Chernoff bounds imply that we can choose a set $P$ of size exactly $\nu x^2/2 = o(n)$, such that for each $p \in P$ there is at least one choice $\xi_1, \xi_2$ such that the triple $(p, \xi_1, \xi_2)$ is good, i.e., $C_{p, \xi_1, \xi_2} \leq w$. We will in fact be able to choose $2^k$ disjoint sets $P_{i,k}, 1 \leq l \leq 2^k$ since replacing $n$ by $n - 2^k \nu x^2/2$ will not significantly change the above calculations. (Here $2^k \nu x^2 = O(n^{1-\theta_k + \theta_k^2 \log \theta_k + \gamma} n) = o(n)$).

**Observation 7** Each $\xi \in X$ is in $\text{Bin}(x, 1 - e^{-w})$ good triples of the form $(p \in P_{i,k}, \xi, \ldots)$ and so $qs$ is in at most $2\nu x^2$ such triples. (Here $\nu x^2 = 2\left(\frac{n \log n}{x}\right)^{\theta_k} \gg \log n$).

Let

$$\nu_0 = \nu x^2/2 \text{ and } \nu_{l+1} = \nu \nu_l^2/2 \text{ for } 0 \leq l < k.$$  

The solution to this recurrence is

$$\nu_l = \left(\frac{\nu}{2}\right)^{2^{l+1}-1} x^{2^{l+1}} = (n \log n)^{(2^{l+1}-1)\theta_k} x^{(2^{l+1}-2^{l+1})\theta_k}.$$  

Observe that $\nu_l$ increases with $l$. Note also that if $l \leq k - 2$ then

$$w\nu_l^2 \leq w\nu_{k-2}^2 = 2\left(\frac{x}{n}\right)^{2^k \theta_k} \log (2^{k-1})\theta_k n = o(1),$$  

$$w\nu_l \nu_0 = w^2 \nu_l^2 x^2 = 2\left(\frac{n \log n}{x}\right)^{2\theta_k} \gg \log n.$$  

We now have the basis for an inductive claim that $qs$ if $l \leq k - 1$ and $u = (p, a, b)$ is a triple at an even level $2(k - l)$ then there are at least $\nu_l$ choices for $p$ such that there exists a triple $u = (p, a, b)$ with $C_u \leq w$ and a feasible tree $\Gamma_u$ with $u$ as root and depth $2l + 1$. Our analysis above has proved the base case of $l = 0$. Imagine now that we are filling in the possibilities for the $r$th triple $(p, a, b)$ at level $k - l$. We fill in these possibilities level by level starting at level $2k$. Imagine also that we have identified $\nu_{l-1}$ choices for each of $a, b$. This can be an inductive assumption, so for example $a$ will have to be a possible selection for the first component of the $(2r - 1)$st triple at level $2(k - (l - 1))$.

For a fixed $p$, conditional on our having selected exactly $\nu_{l-1}$ choices $A, B$ for $a, b$, let $p$ be useful if there is a pair $(a, b) \in A \times B$ with $C_{p, a, b} \leq w$ and useless otherwise. Then, using (7),

$$\Pr(p \text{ is useless}) \leq \exp \left\{ -w\nu_{l-1}^2 \right\} \leq 1 - \frac{2w\nu_{l-1}^2}{3}.$$
It follows that the number of useful \( p \) that have not been previously selected dominates \( \text{Bin}(n - o(n), 2wv_{l-1}^2/3) \). Here \( o(n) = \sum_{s \leq l} 2^{k-s}wv_s^2 \) bound the number of forbidden \( p \)'s. It follows that \( \text{qs} \) we can choose a set of useful \( p \)'s of size \( wv_{l-1}^2/2 = o(n) \). We can do this so that each node of \( \Gamma \) gets distinct choices.

**Observation 8** A fixed \( a \) is in at most \( \text{Bin}(nv_{l-1}/2, 1 - e^{-w}) \) good triples \((p, a, b)\) feasible for level \( 2(k-l) \) and so \( \text{qs} \) every \( a \) is in at most \( wnv_{l-1}^2 \) such triples, see (8).

This completes our induction. We now apply the above to show that round \( t \) succeeds \( \text{whp} \).

Suppose that in the middle of a round we have added \( y < \alpha x \) triples to \( T \). The number of \( \xi \in X \) that can be used in a good triple \((p, \xi, \eta)\) at level \( 2k \) will have been reduced by \( y \). Thus the number of choices for \( p \) in any triple in this level will have been reduced by at most \( 2^k \times 2 \times \alpha x \times 2woutx \), see Observation 7. This reduces the number of choices for \( p \) in a triple at level \( 2(k-l) \) by at most \( 2^{k+2^k}\alpha woutx^2 \times wv_{l-1}^2 = 2^{k+3}\alpha wv_{l-1}^2 \), see Observation 8. So let \( \mu_t \) denote the number of choices for \( p \) in triples \((p, \cdot, \cdot)\) at level \( 2(k-l) \) that are forbidden by choices further down the tree. We have just argued that \( \mu_1 \leq 2^{k+3}\alpha wv_{l-1}^2 \). In general we can use Observation 8 to conservatively argue that

\[
\mu_t \leq wnv_{l-1}(\mu_{t-1} + 2^{k+1}\alpha x).
\]

It follows that for \( l \geq 2 \) we have

\[
\frac{\mu_l}{\nu_l} \leq 2^{k+2}\frac{\mu_{l-1}}{\nu_{l-1}} + 2^{k+2}\alpha \left( \frac{x}{n \log n} \right)^{2^l} \leq 2^{k+1}\frac{\mu_{l-1}}{\nu_{l-1}} + 2^{k+2}\alpha \left( \frac{x}{n \log n} \right)^{2^l}.
\]

It follows that

\[
\frac{\mu_k}{\nu_k} \leq 2^{k-2}\frac{\mu_1}{\nu_1} + 2^{k+1}\alpha \left( \frac{x}{n \log n} \right)^{2^k} \leq 2^{2k+2}\alpha.
\]

We see that at the root there will still be at least \((1 - 2^{2k+2}\alpha)\nu_{k-1} \) choices for \( j_0, k_0 \). So \( i_0 \) will get a choice of at least \( \text{Bin}(1 - 2^{2k+2}\alpha)^2\nu_{k-1}^2, 1 - e^{-w}) \) choices for a good triple \((i_0, j_0, k_0)\). So the expected number of choices is at least \( 2wv_{l-1}^2/3 \), our choice of \( \alpha \) implies this. Now \( wv_{l-1}^2 = 2\log n \) and this is sufficient to ensure that \( \text{whp} \) there is always at least one choice for every \( i_0 \).

### 2.3.3 Final Phase

We can execute the Main Phase so long as \( x \geq 2^k \). Now assume that \( 1 \leq x < 2^k \). We now have to add only \( O(1) \) indices to \( I \). This time we refresh \( C \) an \( O(2^k) \) number of times at an extra cost of \( w = \frac{\log \theta_k n}{n^{1-\theta_k}} \) each time we add an index. So, if successful, the cost of this round is \( O(W_0 + w) = O(\log \frac{nk}{n^{1-\theta_k}}) \).

We first make an inductive assumption: We have a partial assignment \( I \) where \(|I| \leq n - 2 \). (The reader might think that we should assume \(|I| \leq n - 1 \), but here we use the induction hypothesis after one more index has temporarily been deleted from \( I \), prior to a replacement). Assume that the matrix \( C \) is unconditioned and \( i \notin I \): Then we can in \( O(n^2) \) time \( \text{whp} \) find a set \( P \) of size \( \nu_{l-1} \) (with \( x = 1 \) in definition (6)) and a collection \( Q_{p, p'} \in P \) of sets of size \( \nu_{l-1} \) such that for each \((p, q \in P_{p'})\) there is an assignment \( P' \) with \((i, p, q) \in P' \) and \( I(P') \supseteq I(P) \) and \( C(P') = C(P) + C(i, p, q) + O(w) \). This is true for \( \ell = 1 \) since we can make the changes

\[
+(i, p, q) - (p_1, p, p_3) - (q_1, q_2, q) + (p_1, i_2, p_3) + (q_1, q_2, i_3)
\]
where $i_2, i_3$ are unused 2- and 3-coordinates respectively. The number of choices for $p, q$ are independent $\text{Bin}(n, w)$.

For the inductive step, we first refresh the matrix $C$. Then for each $p \in [n]$ we let $I' = I - \{p\}$ and apply the induction hypothesis to generate $\nu_{k-2}^2$ choices of assignment that add back $p_1$ to $I'$. We find that $\text{whp}$ at least $w\nu_{k-2}^2/2 = \nu_{k-1}$ of these have $C(p_1, \ldots) \leq w$. Let this set be $P$. Now refresh $C$ again and apply the same argument for each $p \in P$ to generate choices $Q_p$ for $p$. This completes the induction.

Now let $\ell = k$ and refresh $C$ one more time. Let $P, Q_p, p \in P$ be the sets of size $\nu_{k-1}$ promised by the above argument. We have $\text{Bin}(\nu_{k-1}^2, w)$ choices of $j, k$ which can be used to add $i \notin I$ to $I$ at a cost of $O(w)$. In expectation this is $2\log n$ and so we succeed $\text{whp}$.

For the execution time of the algorithm we simply bound the number of possible trees $\Gamma$. This completes the proof of Theorem 2.

3 Multi-Dimensional Axial Version

We turn to the proof of Theorem 3.

3.1 Lower bound

It is clear that $Z_{d,n}^A \geq Z_1 + Z_2 + \cdots + Z_{n-d-2}$ where $Z_i$ is the minimum cost of the 2-dimensional assignment with cost matrix $A_{j,k} = C_{i_1, \ldots, i_{d-2}, j, k}$. We know that $Z_j \geq (1 - o(1))\zeta(2)$ $\text{whp}$ and the $Z_i$’s are independent. It follows that $\text{whp}$ $Z_{d,n}^A \geq (1 - o(1))n^{d-2}\zeta(2) > 3n^{d-2}/2$.

3.2 Upper bound for $d = 3$

For the upper bound we need a result of Dyer, Frieze and McDiarmid [4]. We will not state it in full generality, instead we will tailor its statement to precisely what is needed. Suppose that we have a linear program

$$\begin{align*}
P : \quad & \text{Minimize } c^T x \text{ subject to } Ax = b, \ x \geq 0. \\
& \text{Here } A \text{ is an } m \times n \text{ matrix and the cost vector } c = (c_1, c_2, \ldots, c_n) \text{ is a sequence of independent copies of } \text{Exp}(1). \text{ Let } Z_P \text{ denote the minimum of this linear program. Note that } Z_P \text{ is a random variable. Next let } y \text{ be any feasible solution to } P. \\
& \text{Theorem 9 ([4])} \\
& \mathbf{E}(Z_P) \leq m \max_{j=1,2,\ldots,n} y_j. \quad (9)
\end{align*}$$

Furthermore, $Z_P$ is at most $1 + o(1)$ times the RHS of (9), $\text{whp}$.

Now consider the following greedy-type algorithm. We find a minimum 2-dimensional assignment for 1-plane $i = 1$, we then find a minimum assignment for 1-plane $i = 2$, consistent with choice for 1-plane $i = 1$, and so on:

**Greedy**

1. For $i = 1, \ldots, n$ do the following:
   - Let $G = K_{n,n} \setminus (M_1 \cup M_2 \cup \cdots M_{i-1})$;
   - If $(j, k) \in E(G)$ let $A_{j,k} = C_{i,j,k}$.
   - Let $M_i$ be a minimum cost matching of $G$ using edge weights $A$. 
The output, $M_1, M_2, \ldots, M_n$ defines a set of triples $T = \{(i, j, k) : (j, k) \in M_i\}$. We claim that if $Z_i = A(M_i)$ then

$$E(Z_i) \leq \frac{2n}{n - i + 1}. \quad (10)$$

For this we apply Theorem 9 to the linear program

$$\text{Minimise } \sum_{(j,k) \in E(G)} A_{j,k} x_{j,k} \quad \text{subject to}$$

$$\sum_{k: (j,k) \in E(G)} x_{j,k} = 1, \quad j = 1, 2, \ldots, n$$

$$\sum_{j: (j,k) \in E(G)} x_{j,k} = 1, \quad k = 1, 2, \ldots, n$$

$$x_{j,k} \geq 0, \quad j, k = 1, 2, \ldots, n.$$

We note that there are $2n$ constraints and that $x_{j,k} = 1/(n - i + 1)$ is a feasible solution. With Theorem 9, this implies (10) and the upper bound in Theorem 3 for the case $d = 3$.

References

[1] D. Aldous, Asymptotics in the random assignment problem, Probability Theory and Related Fields 93 (1992) 507-534.

[2] D. Aldous, The $\zeta(2)$ limit in the random assignment problem, Random Structures and Algorithms 18 (2001) 381-418.

[3] R. Burkard, M. Dell’Amico and S. Martello, Assignment Problems, SIAM Publications 2009.

[4] M.E. Dyer, A.M. Frieze and C. McDiarmid, Linear programs with random costs, Mathematical Programming 35 (1986) 3-16.

[5] A. M. Frieze, A bilinear programming formulation of the 3-dimensional assignment problem, Mathematical Programming 7 (1974) 376-379.

[6] A. M. Frieze, Complexity of a 3-dimensional assignment problem, European Journal of Operational Research 13 (1983) 161-164.

[7] A. M. Frieze and J. Yadegar, An algorithm for solving 3-dimensional assignment problems with application to scheduling a teaching practice, Journal of the Operational Research Society 32 (1981) 989-995.

[8] D. Grundel, C. Oliveira, E. Pasiliao and E. Pardalos, Asymptotic Results for Random Multidimensional Assignment problems, Computational Optimization and Applications 30 (2005) 275-293.

[9] A. Johansson, J. Kahn and V. Vu, Factors in random graphs, Random Structures and Algorithms 33 (2008) 1-28.

[10] R.M. Karp, Reducibility among combinatorial problems, in R.E. Miller and J.W. Thatcher (Eds), Complexity of Computer Computations, Plenum Press, New York (1972) 85-103.
A Bilinear Programming Formulation

Frieze [5] re-formulated the 3-dimensional planar problem as

\[
\text{Minimise } \sum_{i,j,k=1}^{n} C_{i,j,k} y_{i,j} z_{i,k} \text{ subject to } x, y \in P_A
\]

where \( P_A \) is the bipartite matching polyhedron \( \sum_{i=1}^{n} x_{i,j} = 1 = \sum_{j=1}^{n} x_{i,j} \), for all \( 1 \leq i, j \leq n \).

Now denote the objective above by \( C(y, z) \). The following heuristic was used successfully in a practical situation [7]:

1. Choose \( y_0, z_0 \) arbitrarily; \( Z_0 = C(y_0, z_0); i = 0 \).

2. Repeat until \( Z_{i+1} = Z_i \).

   • Let \( y_{i+1} \) maximise \( C(y, z_i) \).
   • Let \( z_{i+1} \) maximise \( C(y_{i+1}, z) \).
   • \( Z_{i+1} = C(y_{i+1}, z_{i+1}) \).
   • \( i = i + 1 \).