A HEURISTIC FOR THE ASSIGNMENT PROBLEM AND RELATED BOUNDS

By

© Cheong Wai Lai

Submitted to the Faculty of

Graduate Studies and Research

in partial fulfillment of the requirements

for the degree of

Master of Science

McGill University

1981

ABSTRACT

We present a heuristic to solve the $m \times m$ assignment problem in $O(m^2)$ time. Let the assignment problem be formulated by a complete bipartite graph $G = (S,T,E)$, $|S| = |T| = m$. The main procedure in the heuristic is to construct a graph $G_d = (S,T,E_d) \subseteq G$, $|E_d| = 4dn$, $n = \lceil m/2 \rceil$, such that we can find a perfect matching in G_d with probability tending to 1 when $n \geq d$. It is shown that, when $d \geq 5$, the probability of a perfect matching in G_d is greater than $1 - \frac{1}{6}(\frac{d}{n})^{d^2-4d-1}$, which tends to 1 as $m \to \infty$. An $O(|S||E|)$ exact algorithm is used to find a minimum weight matching M in G_d. Any unmatched vertices relative to M will be matched by a greedy algorithm. The expected value of the total cost of the matching found by the heuristic is shown to be less than 6 if the costs are independent and identically uniformly distributed on [0.1].

ABSTRAIT

Nous présentons une méthode intuitive pour résoudre le problème d'assignation de $m \times m$ dans un temps $O(m^2)$. Le problème d'assignation est formulé par un graphe biparti complet $G = (S,T,E)$, $|S| = |T| = m$. La principale procédure dans la méthode intuitive est de construire un graphe $G_d = (S,T,E_d) \subseteq G$, $|E_d| = 4dn$, $n = \lceil m/2 \rceil$, de façon à trouver un couplage parfait dans G_d avec une probabilité qui tend vers 1 lorsque $n \geq d$. Il est prouvé que lorsque $d \geq 5$, la probabilité d'un couplage parfait G_d est plus grande que $1 - \frac{1}{6} (\frac{d}{n})^{d^2-4d-1}$, ce qui se dirige vers 1 lorsque $m \to \infty$. Un algorithme exact, le temps $O(|S||E|)$, est utilisé pour trouver un couplage M le poids minimum dans G_d. Chacun des sommits non couplés relatifs à M est couplé par un algorithme "greedy". Il est prouvé que l'espérance mathématique du coût total du couplage par la méthode intuitive est moins que 6 si les coûts sont indépendents et distribués indentiquement et uniformément dans l'intervalle [0,1].

ACKNOWLEDGEMENTS

I should like to express my appreciation to Professor David Avis who served as the advisor for this study. Furthermore, I am also grateful to him for the suggestion of the main idea of the heuristic discussed in this thesis, many helpful comments and thorough reading of the manuscript which resulted in numberless improvements and clarifications.

To My Mother and the Memory of My Father

TABLE OF CONTENTS

Chapter

CHAPTER I

Introduction

There are m men to be assigned to m jobs. The cost of assigning the i^{th} man to the j^{th} job is c_{ij}. The assignment problem is to find the assignment that assigns every job to a man and minimizes the total cost of the assignment. In practice, there are many problems that belong to this class of assignment problems. For examples, contracts are assigned to contractors, machine tools are assigned locations, goods are stored in assigned warehouse locations and so on.

This chapter will give a brief summary of major algorithms available in the literature to solve the assignment problem. In the later chapters, we will develop a new heuristic for the assignment problem and present some related bounds for the heuristic.

I.1 Preliminary definitions

This introductory section contains some elementary definitions in graph theory and probability theory which will be used throughout the remaining chapters. Additional definitions and concepts will be defined when they are needed in later chapters. Further explanation in these terms will be found in the standard texts in graph theory [4,7,8,10,14,32] and probability theory [25,26], although not all of the terminology is completely standardized.

A *graph* G is a pair (V,E), where V is a finite non-empty set of elements called *vertices*, and E is a finite set of distinct *unordered* pairs of distinct elements of V called *edges*. If $(v_i,v_j) \in E$, $v_i,v_j \in V$, then v_i and v_j are said to be *adjacent*.

A graph is said to be *directed* if the vertex pair $<v_i, v_j>$, $v_i, v_j \in V$, associated with each edge in E is an *ordered* pair; the edge $<v_i, v_j>$ is said to be directed from vertex v_i to vertex v_j, or *incident from* v_i and *incident to* v_j; v_i and v_j are said to be *adjacent*.

The *order* of a graph G is the number of vertices of G, and the *size* of a graph G is the number of edges of G.

A *bipartite graph* $G = (V,E)$ is a graph whose vertices can be partitioned into two vertex sets S and T such that $S \cap T = \phi$, $V = S \cup T$, and no two vertices are adjacent in the same set. The bipartite graph may be written as $G = (S,T,E)$. A *complete bipartite graph* K_{pq} is a bipartite graph $G = (S,T,E)$ such that $|S| = p$, $|T| = q$ and $|E| = pq$.

A graph $G' = (V',E')$ is a *subgraph* of a graph $G = (V,E)$ if $V' \subseteq V$ and $E' \subseteq E$.

The *degree* $d_G(v)$ of a vertex v is the number of edges incident to v, if G is undirected. For directed graph, the *out-degree* $d_G^+(v)$ of v is the number of edges incident from the vertex v, and the *in-degree* $d_G^-(v)$ of v is the number of edges incident to the vertex v. The degree $d_G(v)$ of v is $d_G(v) = d_G^+(v) + d_G^-(v)$.

A sequence of edges of the form (v_0,v_1) ; (v_1,v_2), ..., (v_{r-1},v_r) or abbreviated as $v_0,v_1,v_2,\ldots,v_{r-1},v_r$, is called a *walk* of length r from the *initial vertex* v_0 to the *terminal vertex* v_r. If the vertices v_0,v_1, $v_2,\ldots,v_{r-1},v_r$ are distinct, the walk is called a path. If the edges are distinct, the walk is called a trail. A walk or trail is said to be *closed* if $v_0 = v_r$. A closed path is called a *circuit* or *cycle*.

A *weighted graph* is a graph in which there is a number c_{ij} associated with each edge (i,j). The numbers c_{ij} are called *weights* or *costs*.

A *matching* M of a graph G = (V,E) is a set of edges in E such that no two edges in M are incident to the same vertex.

For a given matching M, a vertex is said to be *matched* or *covered* if there exists an *edge* in M incident to (from) it. An *unmatched* or *exposed* vertex is the vertex not covered by an edge in M.

An *alternating path* is a path whose edges are alternating in M and E-M. An *augmenting path* is an alternating path whose initial and terminal vertices are exposed.

A *perfect matching* is a matching where every vertex in G is matched.

An *alternating tree* relative to a matching M for an undirected bipartite graph is a tree whose *root* is an exposed vertex and all paths starting at the root are alternating paths.

A *hungarian tree* is an alternating tree in which there does not exist an edge not in the tree such that when the edge is added to the alternating tree it will form an augmenting path.

One notion following Berge [4] is that all graphs are directed but sometimes the direction need not be specified.

Suppose that the elements of a set $\Omega = \{w_1, w_2, \ldots\}$ corresponds to all the possible outcomes of some experiments such that when the experiment is performed the outcome will be identified by an unique element w_i. We call the set Ω a *sample space* for the experiment, and any subset of Ω is said to be an *event*. We associate with Ω a *probability measure* which assigns to certain events $A_i \subset \Omega$ a number $P(A_i)$ called the *probability* of the event A_i. $P(A_i)$ satisfies the following three axioms:

(1) $0 \leq P(A_i) \leq 1$, $A_i \subset \Omega$ and $\bigcup_{i=1}^{\infty} A_i = \Omega$,

(2) $P(\Omega) = 1$,

and (3) $P(\bigcup_{i=1}^{k} A_i) = \sum_{i=1}^{k} P(A_i)$, $\bigcap_{i=1}^{k} A_i = \phi$.

A *probability space* is a triple (Ω,Φ,P) where Φ is a nonempty subset of the power set of Ω which is closed under union and complementation. Let $\mathbb{R}$ be the set of real numbers. A *random variable* (r.v.) is a real valued point function $X : \Omega \to \mathbb{R}$ defined on the probability space (Ω,Φ,P) if the set $\{w|X(w) \le x\} \in \Phi$ for any $x \in \mathbb{R}$. Let w be the event that the random variable X equal to x and f(x) be the probability that w occurs, then f(x) is denoted by

$$P(X = x) = P(X^{-1}(x)) = f(x).$$

A *distribution function* is a point function:

$$F(x) = P(X \le x) = P(X^{-1}(-\infty,x)).$$

The *conditional probability* of A given B, $A,B \subset \Omega$, and $P(B) \neq 0$, is denoted by $P(A|B)$ and is defined by

$$P(A|B) = P(A \cap B)/P(B).$$

An event B is said to be statistically independent of event A if $P(A|B) = P(A)$ or $P(B)=0$.

The mathematical *expectation* of a random variable X is defined by

$$E(X) = \Sigma\, x \cdot P(X = x) ,$$

for the discrete case.

The *variance* of X is defined by

$$Var(X) = E[(X - E(X))^2].$$

The standard deviation of X is $\sqrt{Var(X)}$. The *coefficient of correlation* of random variable X and Y is

$$P(X,Y) = \frac{E[(X - E(X))(Y - E(Y))]}{(Var(X)\ Var(Y))^{1/2}} .$$

If $|P(X,Y)| \to 1$, X and Y are correlated.

The *coefficient of variation* of a random variable X is

$$\gamma(X) = \frac{\sqrt{Var(X)}}{E(X)} \times 100\% .$$

If $\gamma(X)$ is small when compared to 100%, the variation of X is small.

We define some notations for the asymptotic behavior of functions. Let f(n) and g(n) be two functions, we say that

$$f(n) = O(g(n)) \quad \text{iff } f(n)/g(n) \leq c$$

for some constant c and n_o such that $n > n_o$. Moreover, we say

$$f(n) = o(g(n)) \quad \text{iff } \lim_{n\to\infty} \frac{g(n)}{f(n)} = c \neq o .$$

Finally, a *heuristic* or *heuristic algorithm* or *approximate algorithm* is defined as the algorithm which will usually find good, but not necessarily optimum, solutions within an acceptable amount of time. An *exact* algorithm is an algorithm which gives an optimum solution.

I.2 Previous Work

I.2.1 Assignment problem formulation

Given a $m \times m$ matrix $C = (c_{ij})$, $c_{ij} \geq 0$, we have to find a permutation matrix $\underline{x} = (x_{ij})$ of intergers 0 and 1 that minimize the expression

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij} .$$

The value of Z will be called the total cost of the assignment or assignment sum.

The assignment problem is equivalent to the problem of finding a perfect bipartite matching of a weighted bipartite graph such that the sum of the weights of the matching is minimum.

In linear programming formulation, the assignment problem is

$$\left.\begin{array}{ll} \min & \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij} \\ \text{subject to} & \sum_{i=1}^{m} x_{ij} = 1 , \; j = 1,2,\ldots,m , \\ & \sum_{j=1}^{m} x_{ij} = 1 , \; i = 1,2,\ldots,m , \\ & x_{ij} \geq 0 . \end{array}\right\} \quad \text{(PP)}$$

The dual of this problem is

$$\left.\begin{array}{ll} \max & \left\{ \sum_{i=1}^{m} u_i + \sum_{j=1}^{m} v_j \right\} \\ \text{subject to} & u_i + v_j \le c_{ij} \ , \\ & u_i \ge 0 \ , \ i = 1,2,\ldots,m \ , \\ & v_j \ge 0 \ , \ j = 1,2,\ldots,m \ . \end{array}\right\} \quad \text{(DP)}$$

Orthogonality conditions which are necessary and sufficient for optimality of primal and dual solutions are [42]

$$\left.\begin{array}{l} x_{ij} > 0 \Rightarrow u_i + u_j = c_{ij} \ , \\ u_i > 0 \Rightarrow \sum_{j=1}^{m} x_{ij} = 1 \ , \\ v_j > 0 \Rightarrow \sum_{i=1}^{m} x_{ij} = 1 \ . \end{array}\right\} \quad \text{(OC)}$$

There are exact algorithms to solve the assignment problems in $O(m^3)$ time [12,14,38,42,50]. Some of the algorithms make use of the augmenting path theorem [10,42] which states that a matching M is a maximum cardinality matching if and only if there exists no augmenting path in G relative to M. Some exact algorithms and approximate algorithms will be reviewed in the following sections.

I.2.2 Exact Algorithms

I.2.2.1 Hungarian Primal-Dual Algorithm [12,14,20,42]

In this algorithm, the dual feasible vectors, $\underline{U} = (u_i)$ and $\underline{V} = (v_i)$, and the corresponding infeasible primal vector $\underline{x}$ orthogonal to $(\underline{U},\underline{V})$ are calculated at each step of the procedure. The algorithm will terminate the calculation when all the orthogonality conditions (OC) are satisfied. The algorithm in figure 1 will successively compute the matching M_k^*, $|M_k^*| = k$, $k = 1,2,\ldots,m$, which is the minimum weight matching relative to the other matchings of the same cardinality. The required number of steps to find the minimum weight augmenting path is $O(|E|)$. A change in dual variables will increase the cardinality of the current matching by 1. Hence, the overall time complexity for the hungarian primal-dual algorithm is $O(|V||E|)$.

```
PROCEDURE HPD(G,M)
     Input:  Graph G = (S,T,E) with costs c_ij
     Output:  A minimum weight matching M.
BEGIN
     M ← ∅ ; δ ← 0 ; δ* ← 1 ;
     u_i ← min {c_ij | ∀i,j } ; v_j ← 0, i ∈ S, j ∈ T;
     WHILE δ* > δ  DO
          Construct G' = (S',T',E'), S' = S, T' = T, E' ⊆ E such that
               (i,j) ∈ E'  iff  u_i + v_j = c_ij .
          Hungarian ← False.
          WHILE Hungarian = False DO
               IF ∃ an alternating tree X in G' THEN
                  Find one whose root is in S'.
               IF  X is augmenting  THEN
                  P = minimum weight augmenting path in X
                  M ← (P ∪ M) - (P ∩ M)
               ELSE  !  X is  a hungarian tree  !
                    Hungarian ← True.
          END
          δ* ← min {u_i | i ∈ S}
          δ ← min {c_ij - u_i - v_j | c_ij - u_i - v_j ≠ 0} .
          u_i ← u_i - δ  if i ∈ S and i ∈ X.
          v_j ← v_j + δ  if j ∈ T and j ∈ X.
     END
END.
```

Fig 1. Hungarian Primal-Dual Algorithm [42].

I.2.2.2 Primal Algorithm [42]

This algorithm (figure 2) involves no dual variables or other considerations of duality. The time complexity for the minimum weight augmenting path is $O(|S||E|)$, and the number of augmentations required is $O(|S|)$. The overall time complexity is $O(|S|^2|E|)$.

I.2.2.3 Other improvements on the original Hungarian Algorithm

Tomazawa [50] modifies the original assignment network by removing the negative cost arcs so that the Dijkstra's shortest path algorithm [1,42] can be applied. A potential is defined for each vertex in the modified network in order to achieve that. A minimum augmenting path is found by looking for the minimum cost path between two exposed vertices in $G^* = (S^*, T^*, E^*)$ in which $c^*_{ij} \leftarrow -c_{ij}$ if $(i,j) \in M$ and $c^*_{ij} \leftarrow c_{ij}$ if $(i,j) \notin M$.

```
PROCEDURE PM(G,M)

    Input:  G = (S,T,E) with costs c_ij, (i,j) ∈ E.
    Output:  Minimum weight matching M.

BEGIN

    M ← φ
    WHILE |M| < |S|  DO
        Construct G' = (S',T',E') with costs c'_ij such
            that S' = S, T' = T, E' = E, c'_ij ← - c_ij  if
            (i,j) ∉ M and c'_ij ← c_ij  if  (i,j) ∈ M.
        Find a minimum cost augmenting path P in G'.
        M ← (M ∪ P) - (M ∩ P)

    END

END.
```

Fig. 2. Primal Method for Assignment Problem.

Karp [38] gives a method to compute the minimum cost path by using priority queues. If the costs are uniformly distributed, the expected time complexity of the Karp's algorithm is $O(|S||T| \log|T|)$.

Let $C = (c_{ij})$ be the initial cost matrix and $C^* = (c^*_{ij})$ be the reduced cost matrix of C such that $c^*_{ij} = c_{ij} - x^*_i - y^*_j$, where $y^*_j = \min\{c_{ij} | \forall i\}$ and $x_i = \min \{c_{ij} - y^*_j | \forall j\}$. The initial solution for the hungarian method is the maximum cardinality matching M in which the edges $(i,j) \in M$ iff $c^*_{ij} = 0$. Carpeneto and Toth [12] propose this modification and show that the computing time of this modified algorithm is faster than that of the original hungarian algorithm for dense cost matrix.

Balinski and Gomory [3] give a primal method which gives the same time complexity as the hungarian method. In their algorithm, the feasible vector $\underline{x}$ and corresponding orthogonal vector $(\underline{U},\underline{V})$ are calculated at each step.

I.2.3 Approximate Algorithms

Donath [18] gives an $O(m^2 \ln m)$ approximate algorithm to solve the assignment problem (figure 3). The algorithm involves successive investigation of the next smallest element in the columns being scanned if the current element being scanned is in a row which was assigned to another column in the previous steps. Hence, the later steps of the algorithm will depend on the earlier steps of the algorithm.

Kurtzberg [41] proposes three approximate algorithms, namely, the row/column scan method (figure 4), the matrix scan method (figure 5), and the divide-and-conquer method (figure 6) which partition the original cost matrix into k^2 $\ell \times \ell$ submatrices. He shows that the row/column scan method and the matrix scan method are better than the divide-and-conquer method. The time complexity for the row/column scan method, matrix scan method and the divide-and-conquer method are $O(m^2)$, $O(m^3)$ and $O(m^3)$ respectively.

Figure 3. Multi-column Scan Algorithm.

```
PROCEDURE MCS(C,M)
BEGIN !  Given a m x m cost matrix C = (c_ij)  !
    SmapT_1(i) ← φ; TmapS_1(j) ← φ ; 1 ≤ i,j ≤ m
    FOR j = 1,2,...,m DO
        Find c_ij = min {c_kj | 1 ≤ k ≤ m}
        IF SmapT_j(i) = φ THEN BEGIN
            SmapT_j(i) ← j
            TmapS_j(j) ← i
        END
        ELSE BEGIN
            secondary ← {j, SmapT_j(i)}
            TmapS_j(j) ← i
            finish ← false
          WHILE   finish = false  DO BEGIN
                R ← {(r,ℓ) | c_rℓ = min {c_kw | 1 ≤ k ≤ m,  c_kw ≥ c_TmapS_j(w), w
                               k ≠ TmapS_j(w)}, w ∈ secondary} .
            IF SmapT_j(r) = φ, (r,ℓ) ∈ R THEN BEGIN
               finish ← true
               IF ℓ = j  THEN BEGIN
                  SmapT_j(r) ← ℓ
                  TmapS_j(ℓ) ← r

               END
```

```
                    ELSE BEGIN
                        SmapT_j(TmapS_j-1(ℓ)) ← j
                        FOR (r,ℓ) ∈ R,  ℓ ≠ j  DO
                            SmapT_j(r) ← SmapT_j-1(r)
                            TmapS_j(ℓ) ← TmapS_j-1(ℓ)
                        END
                    END
                END
                ELSE BEGIN
                    For each (r,ℓ) ∈ R  DO
                        TmapS_j(ℓ) ← r
                    END
                    secondary ← secondary ∪ {SmapT_j(TmapS_j(j))}
                END ! IF !
            END ! WHILE !
        END ! IF !
        M ← {(i, SmapT(i))|  ∀i }
    END ! LOOP !
END.
```

Figure 4. Row/column Scan Algorithm.

```
PROCEDURE RCS(C,M)
BEGIN   ! Given a m x m cost matrix C = (c_ij) !
     SmapT(i) ← ∅ ; TmapS(i) ← ∅ ; i = 1,2,...,m.
     For i = 1,2,...,m  DO

          Find c_ij = min {c_ik | 1 ≤ k ≤ m and TmapS(k) = ∅}
          TmapS(j) ← i ;
          Find c_ji = min {c_ki | 1 ≤ k ≤ m and SmapT(k) = ∅}
          SmapT(i) ← j

     END
```

$$Z_1 \leftarrow \sum_{i=1}^{m} c_{i,SmapT(i)}$$

$$Z_2 \leftarrow \sum_{j=1}^{m} c_{TmapS(j),j}$$

```
     IF  Z_1 ≤ Z_2  THEN

          M ← {(i,SmapT(i)) |  ∀i}

     ELSE M ← {(TmapS(j),j) |  ∀j}

END
```

Figure 5. Matrix Scan Method.

```
PROCEDURE MSM(C,M)
BEGIN

    SmapT(i) ← ϕ ; 1 ≤ i ≤ m

    WHILE  SmapT(i) = ϕ , 1 ≤ i ≤ m  DO

        Find c_ij = min {c_kℓ | 1 ≤ k,ℓ ≤ m and SmapT(k) = ϕ}

        SmapT(i) ← j

    END

    M ← {(i, SmapT(i)) | 1 ≤ i ≤ m}

END
```

Figure 6. Divide-and-Conquer Method (divide into k^2 $\ell \times \ell$ submatrices).

PROCEDURE PAR(G, M^*)

BEGIN

Let C be the cost matrix to represent G.

Partition C into k^2 $\ell \times \ell$ submatrices, R_{ij}, $1 \le i,j \le k$, $m = \ell k$.

Apply an exact algorithm to solve R_{ij}. Let Y_{ij} and a_{ij} be the corresponding permutation matrix and the assignment sum for R_{ij}.

Define a new $k \times k$ matrix $A = (a_{ij})$, $1 \le i,j \le k$. Again, apply an exact algorithm to solve A. Let $Q = (q_{ij})$, $1 \le i,j \le k$, be the corresponding permutation matrix for A.

The permutation matrix for the assignment problem is

$M^* = (\Phi_{ij})$, $1 \le i,j \le k$, where

$$\Phi_{ij} = \begin{cases} (0) & \text{if } q_{ij} = 0 \\ Y_{ij} & \text{if } q_{ij} \neq 0 \end{cases}$$

END.

I.2.4 Expected value of the assignment sum

Walkup [53,54] uses a nonconstructive method to show that the expected sum of the assignment problem is less than 3 if the costs are independent and uniformly distributed in [0,1].

Kurtzberg [41] shows that the expected sum found by RCS, MSM or PAR is less than or equal to ℓn m if the costs are uniform on [0,1].

Donath [18] uses a heuristic argument to show that the expected sum is less than or equal to 2.37··· if the costs are uniform on [0,1]. However, his argument is not rigorous because, as pointed out by Walkup [54], the later steps of the algorithm are conditioned by the ealier steps of the algorithm.

I.3 Summary of thesis

An $O(m^2)$ heuristic to solve the assignment problem is proposed. Let $G = (S,T,E)$ be the complete bipartite graph for the assignment problem. The heuristic consists of two main steps:

(1) Choose $O(m)$ edges from G to construct $G_d = (S,T,E_d)$ such that the degree of each vertex in G_d is greater than d, $d \geq 5$.

(2) Apply an $O(|S||E|)$ exact algorithm to G_d. Let M_d be the minimum weight matching of G_d. If $|M_d| = |S| = m$, stop. Otherwise, match those exposed vertices relative to M by the row scan method.

If the costs are independent and identically uniformly distributed in [0,1], the expected sum of the assignment problem given by the heuristic is less than 6. This bound can be used for the branch and bound algorithms to solve the travelling salesman problems [37].

CHAPTER II

A heuristic for the assignment problem

The heuristic is motivated by the proof of the expected assignment sum of a random graph $G = (S \cup T,E)$ given by Walkup [54]. In his proof, he considers the sum θ_d of an arbitrary matching of a regular directed bipartite graph $G_d = (S,T,E_d)$ in which the out-degree of each vertex is d, $d \geq 1$. The expected assignment sum is bounded above by the expected value of θ_d. Let $Y = (y_{ij})$ and $Z = (z_{ij})$ be two $m \times m$ matrices such that $\{y_{ij} \mid 1 \leq i,j \leq m\}$ and $\{z_{ij} \mid 1 \leq i,j \leq m\}$ are i.i.d. with common distribution function and the cost matrix $C = (c_{ij})$ is defined by $c_{ij} = \min\{y_{ij},z_{ij}\}$. The edge $<i,j>$, $i \in S$, $j \in T$, is in G_d if y_{ij} is one of the d smallest elements in $\{y_{i1},y_{i2},\ldots,y_{im}\}$, and the edge $<j,i>$, $j \in T$, $i \in S$, is in G_d if z_{ij} is one of the d smallest elements in $\{z_{1j},z_{2j},\ldots,z_{mj}\}$. Since Y and Z are two different sets of random variables, the selection of edges incident from vertices in S to vertices in T is independent of the selection of edges incident from vertices in T to vertices in S. However, we have the cost matrix only in practice. To construct G_d by a practical algorithm as the way given in the proof, we have to set $Y = Z = C$. Thus, the edges in G_d in this case will be chosen from the same set of random variables. This may introduce a dependence between the way to choose edges incident from vertices in S to vertices in T and the way to choose edges incident from vertices in T to vertices in S. For example, if c_{ij}, $i \in S$, $j \in T$, is one of the d smallest elements in $\{c_{i1},c_{i2},\ldots,c_{im}\}$ and $\{c_{1j},c_{2j},\ldots,c_{mj}\}$, the edges $<i,j>$ and $<j,i>$ will be selected. We will present an algorithm to construct G_d in such a way that the above dependence is avoided (figure 7.3 and figure 8).

The heuristic for the assignment problem to be investigated is shown in figure 7.1. If the original graph $G_o = (S_o,T_o,E_o)$ has odd number of vertices in S, a dummy vertex is added to S_o and T_o, and $2|S_o| - 1$ dummy edges are added to E_o (figure 7.2). The resulting graph $G = (S,T,E)$ will be used to construct the subgraph $G_d = (S,T,E_d)$ as shown in figure 7.3 and figure 8. Note that the selection of edges $<s,t>$, $s \in S$, $t \in T$, is independent of the selection of edges $<t^*,s^*>$, $s^* \in S$, $t^* \in T$, since $<s,t>$ and $<t^*,s^*>$ are selected from different sets of random variables. Procedure MATCH (figure 7.4) is to find a minimum weight matching in G_d. One possible exact algorithm to be used is the hungarian primal-dual algorithm. Procedure FILLUP (figure 7.5) will match those exposed vertices (if any) in G_d.

Let $HEUR_{d,OPT}$ denote the heuristic. The subscript d of $HEUR_{d,OPT}$ is the parameter used in the construction of the subgraph $G_d = (S,T,E_d)$; the subscript OPT is the name of the exact algorithm to solve the assignment problem.

<u>Figure 7.1. The heuristic $HEUR_{d,OPT}$</u> .

PROCEDURE $HEUR_{d,OPT}$ $[G_o, M_d, d]$

Input: A complete bipartite graph $G_o = (S_o, T_o, E_o)$, $|S_o| \geq d \geq 5$.

Output: An approximate minimum weight matching M_d of G_o.

BEGIN

DUMMY $[G_o, G]$! figure 7.2 !

CONSTRUCT $[G, G_d, d]$! figure 7.3 !

MATCH $[G_d, M_d]$! figure 7.4 !

FILLUP $[G, M_d]$! figure 7.5 !

END.

Figure 7.2. Algorithm DUMMY .

```
PROCEDURE  DUMMY [G_o,G]

    Input:  a weighted bipartite graph G_o = (S_o,T_o,E_o) with costs

            c_s,t ,  s ∈ S_o  and  t ∈ T_o .

    Output: a weighted bipartite graph G = (S,T,E) such that |S| = |T| = 2n.

BEGIN

   IF  |S_o| is even  THEN

        G ← G_o

   ELSE BEGIN

    ! Construct a dummy vertex in each vertex set and 4n-1 dummy edges !

    S ← S_o ∪ s ;  T ← T_o ∪ t

    c_s,t ← 0

    c_i,t ← ∞ ,  i ∈ S_o

    c_s,j ← ∞ ,  j ∈ T_o

    E ← E_o ∪ {(s,j)} ∪ {(i,t)} ∪ {(s,t)}, i ∈ S_o, j ∈ T_o

   END

END.
```

Figure 7.3. Algorithm to construct G_d.

```
PROCEDURE CONSTRUCT [G,G_d,d]
    Input: A graph G = (S,T,E), |S| = |T| = 2n, cost c_ij, i ∈ S, j ∈ T.
    Output: A graph G_d = (S,T,E_d), |E_d| = 4dn.
BEGIN
    S_1 ∪ S_2 = S, T_1 ∪ T_2 = T, |S_1| = |S_2| = |T_1| = |T_2| = n.
    E_d ← ∅
    FOR i = 1,2  DO
        FOR each vertex v ∈ S_i  DO
            FOR r = 1,2,...,d  DO
                Find a vertex v_r ∈ T_i such that
                    c_{v,v_r} = r^th smallest element in {c_{v,j} | j ∈ T_i}
                E_d ← E_d ∪ (v,v_r)
            END
        END
        FOR each vertex v ∈ T_i  DO
            FOR r = 1,2,...,d  DO
                Find a vertex v_r ∈ S-S_i such that
                    c_{v_r,v} = r^th smallest element in {c_{k,v} | k ∈ S-S_i}
                E_d ← E_d ∪ (v_r,v)
            END
        END
    END
END
```

Figure 7.4. Algorithm to find a minimum weight matching in G_d.

PROCEDURE MATCH $[G_d, M_d]$

Input: A bipartite graph $G_d = (S,T,E_d)$.

Output: A matching M_d.

BEGIN

Apply an $O(|S||E|)$ exact algorithm OPT, such as the hungarian primal-dual algorithm, to G_d.

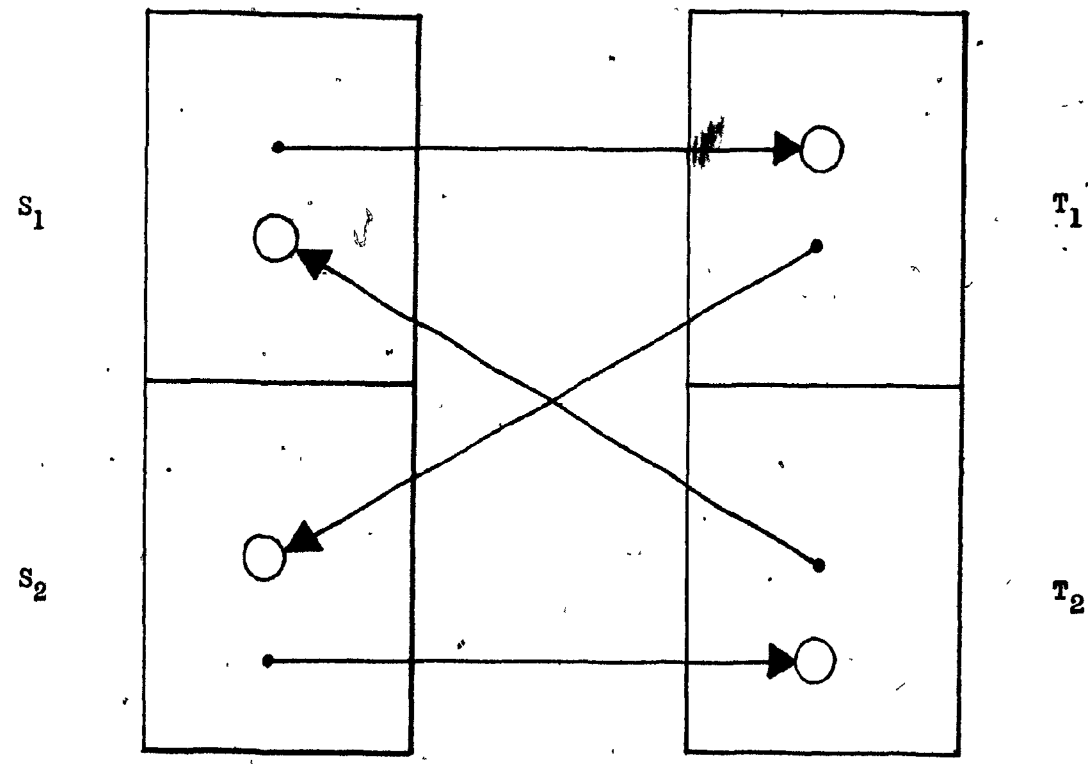
Let M_d be the minimum weight matching of G_d.

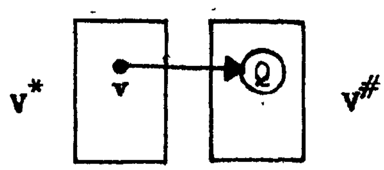
END.

Figure 7.5. Algorithm to match those exposed vertices in G relative to M_d.

```
PROCEDURE  FILLUP [G,M_d]

     Input:  Graph G = (S,T,E) and matching M_d.

     Output:  A complete matching in G.

BEGIN

     IF |M_d| = 2n  THEN RETURN

     ELSE BEGIN

          ! Row scan method !

          S' ← {exposed vertices in S relative to M_d}

          T' ← {exposed vertices in T relative to M_d}

          WHILE |S'| ≥ 1  DO

               Choose a vertex s ∈ S' and an edge (s,t) ∈ E,

                    t ∈ T such that

                    c_st = min {c_sj | j ∈ T'} .

               S' ← S' - s

               T' ← T' - t

               M_d ← M_d ∪ (s,t)

          END

     END

END.
```

Figure 8. Construction of $G_d = (S_1 \cup S_2, T_1 \cup T_2, E_d)$ from $G = (S, T, E)$, where $S_1 \cup S_2 = S$, $T_1 \cup T_2 = T$, and $|S_1| = |S_2| = |T_1| = |T_2|$.





The set of edges $\langle v,q \rangle$, $q \in Q$, from vertex v to the vertex set Q such that $|Q| = d$, and $\{ c_{vq} \mid q \in Q \}$ are the d smallest elements in $\{ c_{vz} \mid z \in V^{\#} \}$.

CHAPTER III

Probabilistic analysis of the heuristic $HEUR_{d,OPT}$

In this chapter, the upper bound on the probability of no perfect matching in G_d and an upper bound on the expected value of the total costs of the minimum weighted matching in G_d will be derived. In finding the upper bound on the probability of incomplete matching in G_d, the König-Hall theorem for the bipartite matching is used. This upper bound is then used to find the upper bound on the expected assignment sum when the costs are independent and identically uniformly distributed.

The bound on the expected assignment sum when the costs are not uniform on [0,1] can also be found by using the same procedure of analysis. The condition for the analysis holds is when the costs are independent and identically distributed.

III.1.1 Definition of blocking k-pair

For a directed bipartite graph $G = (S,T,E)$, the ordered pair (A,B) is a blocking k-pair in G iff the following hold (figure 9):

(1) $A \subseteq T$, and $B \subseteq S$,

(2) $|B| = k$, and $|A| = k - 1$, $2 \leq k \leq |S| - 1$,

and (3) $\Gamma^+(B) = A$ and $\Gamma^+(T-A) = S - B$.

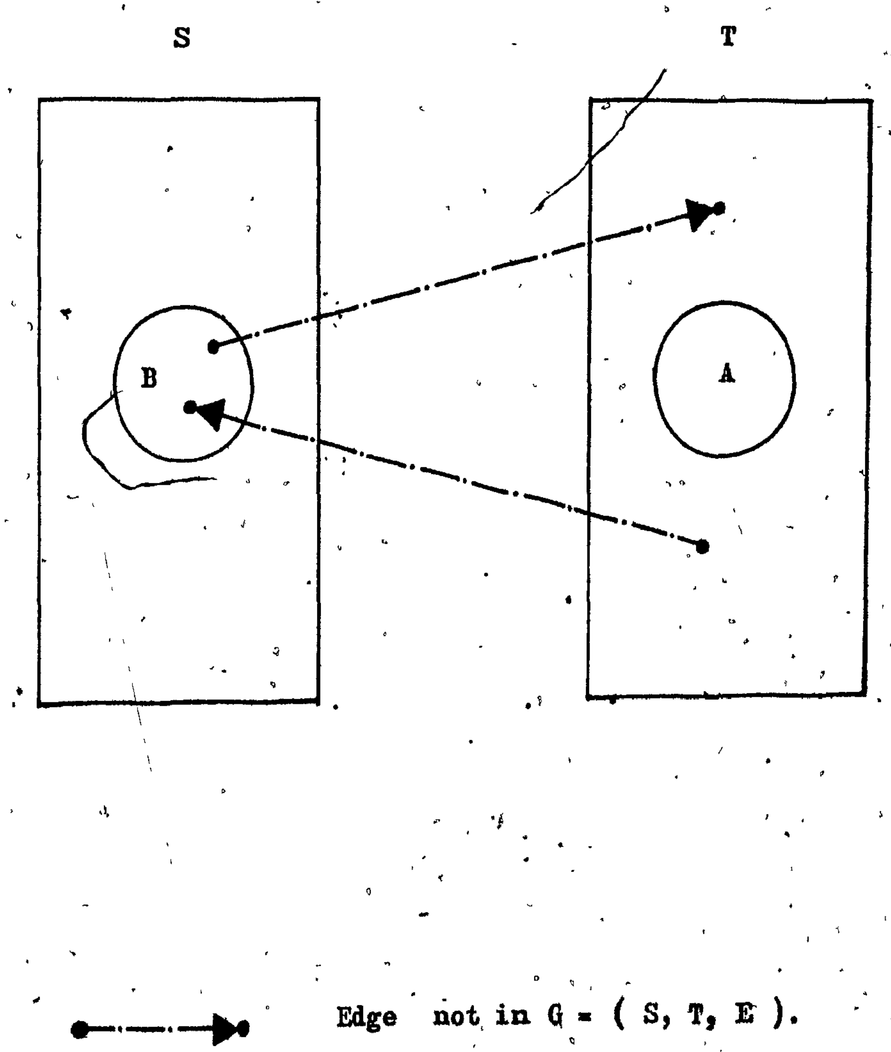
III.1.2 König-Hall theorem [7,8,14,42]

In a bipartite graph $G = (S,T,E)$, S can be matched into T if and only if for every $X \subset S$,

$$|\Gamma(X)| \geq |X|.$$

Hence, there exists a perfect matching in G if and only if there is no blocking k-pair in G.

Figure 9. Edges that cause a blocking k-pair (A,B) in G.



III.2 Some Inequalities

The proofs of the following lemmas are given in appendix A.

Lemma 1.

(1.1) If k and m are positive integers and $0 < k \le m$, then

$$\binom{m}{k} \le \frac{1}{\sqrt{2\pi k}} \left(\frac{em}{k}\right)^k .$$

(1.2) If j, p, and q are positive integers, then

$$\binom{p}{j}\binom{q}{j}^{-1} \le \left(\frac{p}{q}\right)^j , \; 0 \le j \le p \le q.$$

Lemma 2.

If $a_1 + a_2 = k-1$, $b_1 + b_2 = k$, $2 \le d + 1 \le k \le n$, $a_1 \ge 0$, $a_2 \ge 0$, $b_1 \ge 0$, $b_2 \ge 0$, and $d \ge 1$, then

$$\left(\frac{a_1}{n}\right)^{b_1} \left(\frac{a_2}{n}\right)^{b_2} \le \left(\frac{k-1}{n}\right)^k < \left(\frac{k}{n}\right)^k .$$

Lemma 3.

If $a_1 + a_2 = k-1$, $b_1 + b_2 = k$, $d \le a_1, a_2 \le k-1-d$, $0 \le b_1, b_2 \le k$, $d + 1 \le k \le n$, and $d \ge 1$, then

$$\left(\frac{n-b_1}{n}\right)^{n-a_2} \left(\frac{n-b_2}{n}\right)^{n-a_1} \le \left(\frac{n-k+d}{n}\right)^{n-k+d} \left(\frac{n-d}{n}\right)^{n-d}$$

Corollary 1.

If $a_1 + a_2 = k-1$, $b_1 + b_2 = k$, $0 \le a_1, a_2 \le k-1$, $0 \le b_1, b_2 \le k$, $d + 1 \le k \le n$, and $d \ge 1$, then

$$\left(\frac{n-b_1}{n}\right)^{n-a_2} \left(\frac{n-b_2}{n}\right)^{n-a_1} \le \left(\frac{n-k}{n}\right)^{n-k} .$$

Lemma 4.

If $a_1 + a_2 = k-1$, $b_1 + b_2 = k$, $0 \le a_1, a_2 \le k-1$, $0 \le b_1, b_2 \le k$, and if (1) $a_1 \ne 0, a_2 \ne 0$ and $d+1 \le k \le n$, or (2) $a_1 = 0$ or $a_2 = 0$, and $d+1 \le k \le n-d$, then

$$\alpha_d = \left(\frac{a_1}{n}\right)^{b_1} \left(\frac{a_2}{n}\right)^{b_2} \left(\frac{n-b_1}{n}\right)^{n-a_2} \left(\frac{n-b_2}{n}\right)^{n-a_1} \le \left(\frac{d}{n}\right)^{d}, \ d \ge 1.$$

Lemma 5.

If $k - 1 \ge a \ge k/2$, $k > b \ge k/2$, $a + b > k$, $b > a$, $n \ge k \ge n/2$, $n \ge d + 1$ and $d \ge 1$, then

$$f(a,b,k) = a^{b-a-1} (k-1-a)^{a-b+1} (n-b)^{a+b-k} (n-k+b)^{k-(a+b)} \le 1 .$$

III.3 Probability of a perfect matching in G_d

Consider the procedure CONSTRUCT to construct the subgraph $G_d = (S_1 \cup S_2,\ T_1 \cup T_2,\ E_d)$ from $G = (S,T,E)$. Let $A = A_1 \cup A_2$, $A_1 \subseteq T_1$, $A_2 \subseteq T_2$, $|A_1| = a_1$, $|A_2| = a_2$, $B = B_1 \cup B_2$, $B_1 \subseteq S_1$, $B_2 \subseteq S_2$, $|B_1| = b_1$, $|B_2| = b_2$, $|A| = k-1$, $|B| = k$, $0 \le b_1, b_2 \le k$, $0 \le a_1, a_2 \le k-1$ and $d + 1 \le k \le 2n-d$. (A,B) is a blocking k-pair in G_d if the following conditions hold (figure 10):

(1) $\Gamma^+(B_1) = A_1$,

(2) $\Gamma^+(B_2) = A_2$,

(3) $\Gamma^+(T_1 - A_1) = S_2 - B_2$,

and (4) $\Gamma^+(T_2 - A_2) = S_1 - B_1$.

Clearly, there does not exist a blocking k-pair in G_d if $d \ge \lceil n/2 \rceil$, where $n = \lceil |S|/2 \rceil$. This implies that the probability of a perfect matching in G_d is equal to 1 if $d \ge \lceil n/2 \rceil$. We record the obvious relations:

(1) $a_i = 0, d, d+1, \cdots, k-d-2, k-d-1, k-1, i = 1,2,$

(2) $b_i = 0, 1, 2, \cdots, k-1, k, i = 1,2,$

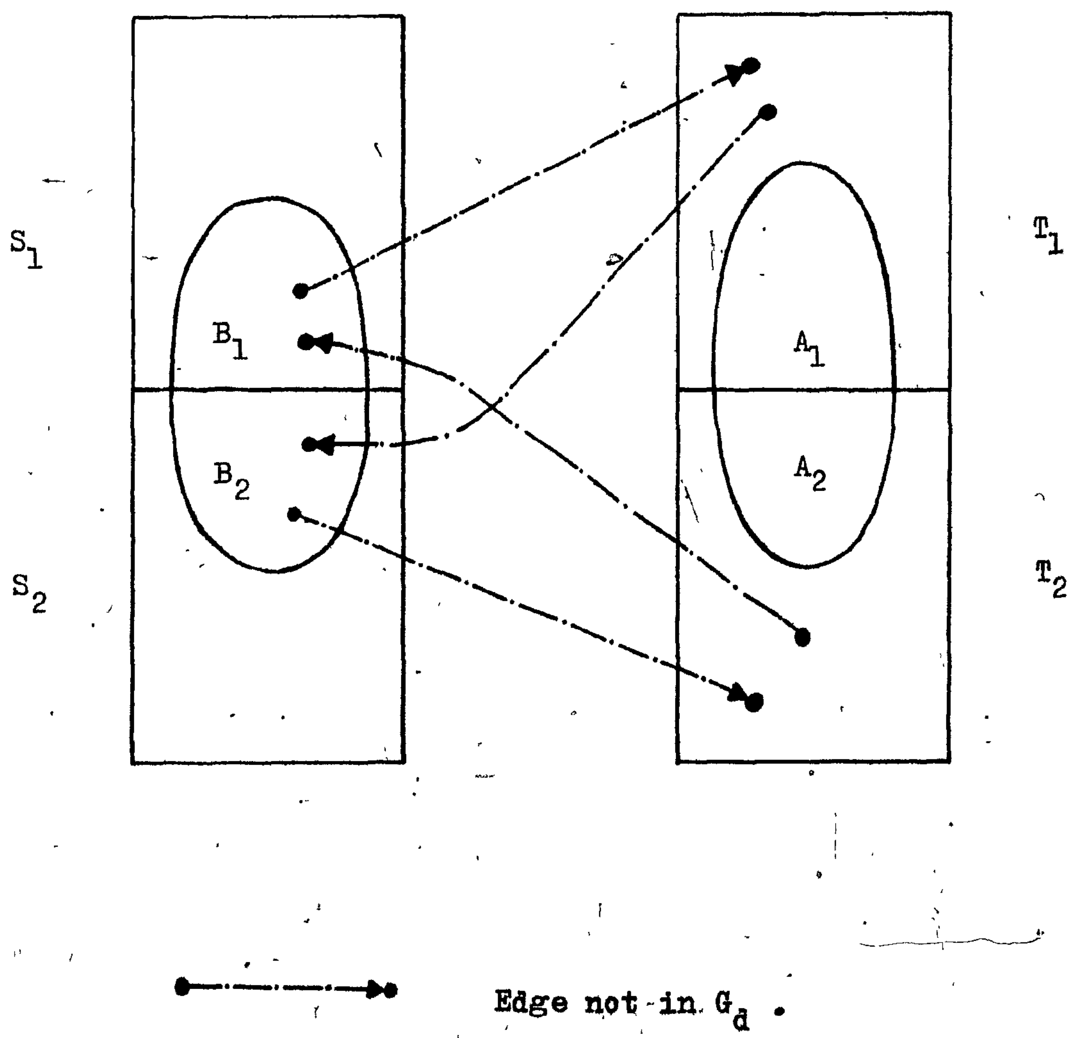
(3) $a_1 + a_2 = k-1$ and $b_1 + b_2 = k$,

(4) either (i) $a_1 \ne 0$, $a_2 \ne 0$ and $d + 1 \le k \le 2n-d$,

or (ii) $a_1 = 0$ or $a_2 = 0$, and $d + 1 \le k \le n-d$,

and (5) $d \ge 1$.

Figure 10. Edges not in $G_d = (S_1 \cup S_2 , T_1 \cup T_2 , E_d)$ that give rise to the blocking k-pair $(A_1 \cup A_2 , B_1 \cup B_2)$ in G_d, where $|A_1 \cup A_2| = |B_1 \cup B_2| - 1$.



The probability that (A,B) is a blocking k-pair in G_d is

$P_{A,B}^{(d)}(k)$

$$= P(\Gamma^+(B_1) = A_1)P(\Gamma^+(B_2) = A_2)P(\Gamma^+(T_1-A_1) = S_2-B_2)P(\Gamma^+(T_2-A_2) = S_1-B_1)$$

$$= \left[\frac{\binom{a_1}{d}}{\binom{n}{d}}\right]^{b_1} \left[\frac{\binom{a_2}{d}}{\binom{n}{d}}\right]^{b_2} \left[\frac{\binom{n-b_2}{d}}{\binom{n}{d}}\right]^{n-a_1} \left[\frac{\binom{n-b_1}{d}}{\binom{n}{d}}\right]^{n-a_2}$$

$$\leq \left[\frac{a_1^{\,b_1}\ a_2^{\,b_2}\ (n-b_2)^{n-a_1}\ (n-b_1)^{n-a_2}}{n^{2n+1}}\right]^d \text{, by lemma 1.2.} \qquad (*1)$$

The expected number of blocking k-pairs (A,B) in G_d is

$$\beta_{A,B}^{(d)}(k) = \sum_{b_1+b_2=k}\ \sum_{a_1+a_2=k-1} \binom{n}{a_1}\binom{n}{a_2}\binom{n}{b_1}\binom{n}{b_2} P_{A,B}^{(d)}(k). \qquad (*2)$$

Let $\alpha_d = \dfrac{a_1^{\,b_1}\ a_2^{\,b_2}\ (n-b_2)^{n-a_1}\ (n-b_1)^{n-a_2}}{n^{2n+1}}$,

then (*1) becomes

$$P_{A,B}^{(d)}(k) \leq (\alpha_d)^d , \qquad (*3)$$

and (*2) becomes

$$\beta_{A,B}^{(d)}(k) \leq \sum_{b_1+b_2=k}\ \sum_{a_1+a_2=k-1} \binom{n}{a_1}\binom{n}{a_2}\binom{n}{b_1}\binom{n}{b_2} (\alpha_d)^d . \qquad (*4)$$

From lemma 4, $\alpha_d \le (d/n)^d$, so (*4) becomes

$$\beta^{(d)}_{A,B}(k) \le \left(\frac{d}{n}\right)^{di} \sum_{b_1+b_2=k} \sum_{a_1+a_2=k-1} \binom{n}{a_1}\binom{n}{a_2}\binom{n}{b_1}\binom{n}{b_2} \alpha_d^{d-i}, \quad (*5)$$

for $i = 0,1,2,\cdots$.

III.3.1 <u>The expected number of blocking k-pairs for $d+1 \le k \le n/2$</u>

Consider the case when $d-i = 4$, and the term

$$\sum_{b_1+b_2=k} \sum_{a_1+a_2=k-1} \binom{n}{a_1}\binom{n}{a_2}\binom{n}{b_1}\binom{n}{b_2} \alpha_d^4$$

$$= \sum_{b_1+b_2=k} \sum_{a_1+a_2=k-1} \binom{n}{a_1}\binom{n}{a_2}\binom{n}{b_1}\binom{n}{b_2} \left[\left(\frac{a_1}{n}\right)^{b_1} \left(\frac{a_2}{n}\right)^{b_2}\right]^4 \left[\left(\frac{n-b_2}{n}\right)^{n-a_1} \left(\frac{n-b_1}{n}\right)^{n-a_2}\right]^4$$

$$\le \left(\frac{k}{n}\right)^{4k} \left(\frac{n-k}{n}\right)^{4(n-k)} \sum_{b_1+b_2=k} \sum_{a_1+a_2=k-1} \binom{n}{a_1}\binom{n}{a_2}\binom{n}{b_1}\binom{n}{b_2} \text{, from lemma 2 and 3,}$$

$$\le \left(\frac{k}{n}\right)^{4k} \left(\frac{n-k}{n}\right)^{4(n-k)} \binom{2n}{k}^2 \left(\frac{k}{2n-k+1}\right)$$

$$\le \left(\frac{k}{n}\right)^{4k} \left(\frac{n-k}{n}\right)^{4(n-k)} \left[\frac{1}{\sqrt{2\pi k}} \left(\frac{2en}{k}\right)^k\right]^2 \quad (1) ,$$

from lemma 1.

$$\leq \left(\frac{k}{n}\right)^{2k} \left(1-\frac{k}{n}\right)^{4(n-n/2)} \frac{(2e)^{2k}}{2\pi k}$$

$$\leq \left(\frac{n/2}{n}\right)^{2k} e^{-2k} \frac{(2e)^{2k}}{2\pi k}$$

$$= \frac{1}{2\pi k} \quad .$$

The inequality (*5) becomes

$$\beta_{A,B}^{(d)}(k) \leq \left(\frac{d}{n}\right)^{d(d-4)} \left(\frac{1}{2\pi k}\right) \quad , \quad d \geq 5 \; .$$

The expected number of blocking k-pairs for $d+1 \leq k \leq n/2$ and $d \geq 5$ is

$$\beta_1^{(d)} = \sum_{k=d+1}^{n/2} \beta_{A,B}^{(d)}(k)$$

$$\leq \left(\frac{d}{n}\right)^{d(d-4)} \frac{1}{2\pi} \sum_{k=d+1}^{n/2} \frac{1}{k}$$

$$\leq \left(\frac{d}{n}\right)^{d(d-4)} \frac{n}{8\pi d}$$

$$\leq \frac{1}{8\pi} \left(\frac{d}{n}\right)^{d^2-4d-1} , \quad d \geq 5.$$

III.3.2 Expected number of blocking k-pairs for $n/2 \le k \le n$.

We will use the following inequalities [46] for m and r are positive integers:

$$\binom{m}{r} \le \begin{cases} \dfrac{m^m}{(1+\frac{1}{r})^r\, r^r (m-r)^{m-r}} & r = 1,2,3\cdots \\[2ex] \dfrac{m^m}{r^r\,(m-r)^{m-r}} & r = 0 \text{ and } 0^0 = 1. \end{cases} \qquad (\dagger\dagger)$$

Since

$$(1+\frac{1}{r})^r = 1 + \binom{r}{1}\frac{1}{r} + \binom{r}{2}\left(\frac{1}{r}\right)^2 + \cdots > 2 ,$$

for $r = 1,2,3,\cdots$, $(\dagger\dagger)$ becomes:

$$\binom{m}{r} \le \begin{cases} \dfrac{m^m}{2r^r (m-r)^{m-r}} & r = 1,2,3,\cdots \\[2ex] \dfrac{m^m}{r^r (m-r)^{m-r}} & r = 0 \text{ and } 0^0 = 1. \end{cases}$$

Consider the term in (*5) when (d-1) = 2 :

$$\binom{n}{a_1}\binom{n}{a_2}\binom{n}{b_1}\binom{n}{b_2}\alpha_d^2 . \qquad (*6)$$

We have

$$\binom{n}{a_1}\binom{n}{a_2} \le \frac{n^n}{2a_1{}^{a_1}(n-a_1)^{n-a_1}} \cdot \frac{n^n}{2a_2{}^{a_2}(n-a_2)^{n-a_2}}$$

$$= \frac{n^{2n}}{4a_1{}^{a_1}(n-a_1)^{n-a_1}\, a_2{}^{a_2}(n-a_2)^{n-a_2}} ,$$

if a_1 and a_2 are nonzero positive integers. For the case b_1 and b_2, we cannot have b_1 and b_2 both equal to zero at the same time. We have,

$$\binom{n}{b_1}\binom{n}{b_2} \le \begin{cases} \left(\frac{n^n}{2b_1{}^{b_1}(n-b_1)^{n-b_1}}\right)\left(\frac{n^n}{2b_2{}^{b_2}(n-b_2)^{n-b_2}}\right) & b_1 \ge 1,\ b_2 \ge 1 \\ \left(\frac{n^n}{b_1{}^{b_1}(n-b_1)^{n-b_1}}\right)\left(\frac{n^n}{2b_2{}^{b_2}(n-b_2)^{n-b_2}}\right) & b_1 = 0,\ b_2 \ge 1 \\ \left(\frac{n^n}{2b_1{}^{b_1}(n-b_1)^{n-b_1}}\right)\left(\frac{n^n}{b_2{}^{b_2}(n-b_2)^{n-b_2}}\right) & b_1 \ge 1,\ b_2 = 0 \end{cases} .$$

Therefore, the following inequality holds:

$$\binom{n}{b_1}\binom{n}{b_2} \le \frac{n^{2n}}{2b_1{}^{b_1}(n-b_1)^{n-b_1}\, b_2{}^{b_2}(n-b_2)^{n-b_2}} .$$

Similarly,

$$\binom{n}{a_1}\binom{n}{a_2} \le \frac{n^{2n}}{2a_1{}^{a_1}(n-a_1)^{n-a_1}\, a_2{}^{a_2}(n-a_2)^{n-a_2}} .$$

Let $f_1 = \dfrac{a_1^{b_1} a_2^{b_2} (n-b_2)^{n-a_1} (n-b_1)^{n-a_2}}{b_1^{b_1} b_2^{b_2} (n-a_1)^{n-a_1} (n-a_2)^{n-a_2}}$,

and $f_2 = \dfrac{a_1^{b_1} a_2^{b_2} (n-b_2)^{n-a_1} (n-b_1)^{n-a_2}}{a_1^{a_1} a_2^{a_2} (n-b_2)^{n-b_2} (n-b_1)^{n-b_1}} \cdot \dfrac{1}{n^2}$.

The expression (*6) becomes:

$$\binom{n}{a_1}\binom{n}{a_2}\binom{n}{b_1}\binom{n}{b_2} \alpha_d^2 \le \frac{1}{4} f_1 f_2 . \qquad (*7)$$

<u>III.3.2.1</u> The bound for f_1 is obtained by simply applying the inequality $(1 + y/x)^x < e^y$ to the expression. Hence, if a_1, a_2, b_1 and b_2 are non-zero positive integers, then

$$f_1 = \left(1 - \frac{b_1-a_1}{b_1}\right)^{b_1} \left(1 - \frac{b_2-a_2}{b_2}\right)^{b_2} \left(1 - \frac{b_2-a_1}{n-a_1}\right)^{n-a_1} \left(1 - \frac{b_1-a_2}{n-a_2}\right)^{n-a_2}$$

$$\le e^{-2} ,$$

as $b_1 + b_2 = k$ and $a_1 + a_2 = k-1$.

If $b_1 = 0$, then $b_2 = k$. Therefore, the bound for f_1 when $b_1 = 0$, $b_2 = k$ and $0 \le a_1, a_2 \le k-1$ is

$$f_1 = \frac{a_1^0}{0^0} \cdot \frac{a_2^k}{k^k} \cdot \frac{(n-k)^{n-a_1}}{(n-a_1)^{n-a_1}} \cdot \frac{(n-0)^{n-a_2}}{(n-a_2)^{n-a_2}}$$

$$= \left(1 - \frac{k-a_2}{k}\right)^k \left(1 - \frac{k-a_1}{n-a_1}\right)^{n-a_1} \left(1 + \frac{a_2}{n-a_2}\right)^{n-a_2}$$

$$\leq e^{-k + (a_1+a_2) - k + a_2}$$

$$< e^{-k + (k-1) - k + (k-1)} \text{, since } a_1+a_2 = k-1 \text{ and } a_2 \leq k-1,$$

$$< e^{-2} .$$

Similarly, if $b_2 = 0$ then $b_1 = k$. The bound for f_1 when $b_1 = k$ and $b_2 = 0$ is

$$f_1 = \frac{a_1^k}{k^k} \frac{a_2^0}{0^0} \left(\frac{n-0}{n-a_1}\right)^{n-a_1} \left(\frac{n-k}{n-a_2}\right)^{n-a_2}$$

$$\leq \left(1 - \frac{k-a_1}{k}\right)^k (1)\left(1 + \frac{a_1}{n-a_1}\right)^{n-a_1} \left(1 - \frac{k-a_2}{n-a_2}\right)^{n-a_2}$$

$$< e^{-2} ,$$

since $a_1 + a_2 = k-1$ and $a_1 \leq k-1$.

Therefore, for $b_1 + b_2 = k$, $a_1 + a_2 = k-1$, and $n/2 \leq k \leq n$, we have :

$$f_1 \leq e^{-2} .$$

III.3.2.2

The upper bound for f_2 is more lengthy. In the following sections, we have to consider every combination of the value of a_1, a_2, b_1 and b_2 in order to show that $f_2 \le 1$. If we substitute $a_2 = k-1-a_1$ and $b_2 = k-b_1$ into f_2, the expression f_2 is

$$f_2 = \left(\frac{k-1-a_1}{a_1}\right)^{a_1-b_1+1} \left(\frac{n-b_1}{n-k+b_1}\right)^{a_1+b_1-k} \frac{(a_1)(n-b_1)}{n^2} \; ; \qquad (*8)$$

$$f_2 \le \left(\frac{k-1-a_1}{a_1}\right)^{a_1-b_1+1} \left(\frac{n-b_1}{n-k+b_1}\right)^{a_1+b_1-k} . \qquad (*9)$$

There are five cases to consider, namely,

[case i] $a_1 = b_1$,

[case ii] $a_1 < b_1$, $(a_1 + b_1) > k$,

[case iii] $a_1 < b_1$, $(a_1 + b_1) < k$,

[case iv] $a_1 > b_1$, $(a_1 + b_1) > k$,

[case v] $a_1 > b_1$, $(a_1 + b_1) \le k$.

III.3.2.2.1 [case i]

(a) If $a_1 = b_1 = (k-1)/2$, then, from (*8),

$$f_2 = \left(\frac{k-1}{2}\right)^0 \left(\frac{k-1}{2}\right)^1 \left(n - \frac{k-1}{2}\right)^0 \left(n - \frac{k+1}{2}\right)^1 \frac{1}{n^2} \le 1 .$$

(b) If $a_1 = b_1 < (k-1)/2$, from (*8),

$$f_2 = a_1^{0} \, (k-1-a_1)^{1} \, (n-b_1)^{2a_1-k+1} \, (n-k+b_1)^{k-2a_1} \, n^{-2} \le 1 .$$

since $k-1-a_1 \le n$, $n-k+b_1 \le n-b_1$, $n-b_1 \le n$ and $2a_1 \le k$.

(c) If $a_1 = b_1 > (k-1)/2$, then

$$f_2 = \left(\frac{k-1-a_1}{1}\right)\left(\frac{n-b_1}{n-k+b_1}\right)^{2a_1-k} \frac{n-b_1}{n^2} \le 1 ,$$

since $k-1-a_1 \le n$, $n-b_1 \le n-k+b_1$, and $n-b_1 \le n$.

Hence, (a), (b) and (c) give $f_2 \le 1$, for $a_1 = b_1$.

<u>III.3.2.2.2</u> [case ii] $b_1 > a_1$ and $(a_1+b_1) > k$.

(a) If $a_1 > (k-1)/2$ and $b_1 \le k/2$, then it is a contradiction to the assumption that $b_1 > a_1$.

(b) If $a_1 < (k-1)/2$ and $b_1 \ge k/2$, then $f_2 \le 1$ holds clearly.

(c) If $a_1 < (k-1)/2$ and $b_2 \le k/2$, then $(a_1+b_1) < k$ is a contradiction to the assumption that $(a_1+b_1) > k$.

(d) If $a_1 > (k-1)/2$ and $b_1 \ge k/2$, let $a = a_1$ and $b = b_1$, from lemma 5, we have $f_2 \le 1$.

(e) If $a_1 = (k-1)/2$ then $b_1 > (k+1)/2$ and

$$f_2 \le (1)\left(\frac{n-b_1}{n-k+b_1}\right)^{a_1+b_1-k} \le 1 .$$

(f) If $b_1 = k/2$ then $a_1 \ge k/2$, a contradiction to $b_1 > a_1$.

III.3.2.2.3 [case iii] $b_1 > a_1$ and $(a_1 + b_1) < k$.

(a) For $a_1 > (k-1)/2$ and $b_1 \le k/2$, a contradiction to the assumption that $a_1 < b_1$.

(b) For $a_1 < (k-1)/2$ and $b_1 \le k/2$, $f_2 \le 1$ holds clearly.

(c) For $a_1 > (k-1)/2$ and $b_1 \ge k/2$, a contradiction to the assumption that $(a_1+b_1) < k$.

(d) For $a_1 < (k-1)/2$, $b_1 \ge k/2$, we have

$$(b_1 - a_1 - 1) \ge k - (a + b)$$

so

$$f_2 \le \left[\left(\frac{a_1}{k-1-a_1}\right)\left(\frac{n-k+b_1}{n-b_1}\right)\right]^{k-(a+b)}$$

but, $(a_1+b_1) < k$ and $b_1 - a_1 \le k-2a_1-1$, then,

$$-(n-k+b_1)a_1 + (k-1-a_1)(n-b_1)$$

$$= (k-2a_1-1)n + b_1 - (b_1-a_1)k \ge 0.$$

So, $f_2 \le 1$.

(e) For $a_1 = (k-1)/2$, $b_1 \le k/2$, we have $f_2 \le 1$.

(f) For $b_1 = k/2$ and $a_1 \le (k-1)/2$, we have $f_2 \le 1$.

III.3.2.2.4 [case iv] $a_1 > b_1$ and $(a_1 + b_1) > k$.

(a) $a_1 > (k-1)/2$ and $b_1 > k/2 \Rightarrow f_2 \le 1$.

(b) $a_1 < (k-1)/2$ and $b_1 > k/2 \Rightarrow b_1 > a_1$, a contradiction.

(c) $a_1 < (k-1)/2$ and $b_1 < k/2 \Rightarrow (a_1+b_1) < k$, a contradiction.

(d) $a_1 > (k-1)/2$ and $b_1 < k/2 \Rightarrow a_1-b_1+1 \geq a_1+b_1-k.$

i.e. $f_2 \leq [(\frac{k-1-a_1}{a_1})(\frac{n-b_1}{n-k+b_1})]^{a_1-b_1+1}$,

but, $(n-k+b_1)a_1 - (n-b_1)(k-1-a_1)$

$= (2a_1-k)n + (n-b_1) - (a_1-b_1)k > 0,$

since $(a_1 + b_1) > k \Rightarrow 2a_1 - k > a_1 - b_1$. Thus,

$f_2 \leq 1.$

(e) $a_1 = (k-1)/2 \Rightarrow b_1 > (k+1)/2$, a contradiction

to $a_1 > b_1$.

(f) $b_1 = k/2 \Rightarrow a_1 \geq k/2$,

$$f_2 \leq (\frac{k-1-a_1}{a_1})^{a_1-b_1+1}(1) \leq 1 \quad .$$

<u>III.3.2.2.5</u> [Case V] $a_1 > b_1$ and $(a_1 + b_1) < k$.

(a) $a_1 > (k-1)/2$ and $b_1 > k/2 \Rightarrow a_1 + b_1 \geq k$, a contradiction.

(b) $a_1 > (k-1)/2$ and $b_1 < k/2 \Rightarrow f_2 \leq 1$.

(c) $a_1 < (k-1)/2$ and $b_1 > k/2 \Rightarrow b_1 > a_1$, a contradiction.

(d) $a_1 < (k-1)/2$ and $b_1 < k/2$. Let $a = k-1-a_1$ $b = k-b_1$,

so $(a+b) > k$ and $b \geq a$. Applying lemma 5, we have $f_2 \leq 1$.

(e) $a_1 = (k-1)/2 \Rightarrow b_1 < (k+1)/2$. Then

$$f_2 \leq (1)\left(\frac{n+b_1-k}{n-b_1}\right)^{k-(a_1+b_1)} \leq 1 .$$

(f) $b_1 = k/2 \Rightarrow a_1 < k/2 \Rightarrow a_1 < b_1$, a contradiction.

III.3.2.2.6

(a) If $(a_1+b_1) = k$, then,

$$f_2 \leq \left(\frac{a_1}{k-1-a_1}\right)^{b_1-a_1-1} (1) \leq 1 .$$

(b) If $b_1-a_1-1 = 0$, then

$$f_2 \leq (1)\left(\frac{n-b_1}{n-k+b_1}\right)^{a_1+b_1-k}$$

$$= \left(\frac{n-b_1}{n-k+b_1}\right)^{2b_1-k-1}$$

$$\leq 1 .$$

III.3.2.3

From section III.3.2.1 and section III.3.2.2, we have shown that $f_1 \leq e^{-2}$ and $f_2 \leq 1$ respectively. The inequality (*7) becomes

$$\binom{n}{a_1}\binom{n}{a_2}\binom{n}{b_1}\binom{n}{b_2} \alpha_d^2 \leq \frac{1}{4e^2} . \qquad (*10)$$

Substitute d-1 = 2 and i = d-2 to inequality (*5), we obtain the expected number of blocking k-pair for fixed size of A and B, and n/2 ≤ k ≤ n ;

$$\beta^{(d)}_{A,B}(k) \le \left(\frac{d}{n}\right)^{d(d-2)} \sum_{b_1+b_2=k} \sum_{a_1+a_2=k-1} \frac{1}{4e^2} \quad , d \ge 5.$$

$$\therefore \quad \beta^{(d)}_{A,B}(k) \le \frac{k^2}{4e^2} \left(\frac{d}{n}\right)^{d(d-2)} \quad . \qquad (*11)$$

The expected number of blocking k-pairs for n/2 ≤ |B| ≤ n is

$$\beta^{(d)}_2 = \sum_{k=n/2}^{n} \beta^{(d)}_{A,B}(k)$$

$$\le \left(\frac{d}{n}\right)^{d(d-2)} \frac{1}{4e^2} \cdot \frac{n^3}{2}$$

$$\le \left(\frac{d}{n}\right)^{d^2-3d-3} \frac{n^3}{2} \left(\frac{d}{n}\right)^3 \left(\frac{n/2}{n}\right)^d \frac{1}{4e^2} \quad , \quad d \le n/2$$

$$\le \left(\frac{d}{n}\right)^{d^2-3d-3} \frac{d^3}{2^{d+2}} \cdot \frac{1}{2e^2} \quad .$$

$$\therefore \quad \beta^{(d)}_2 \le \frac{1}{2e^2} \left(\frac{d}{n}\right)^{d^2-3d-3} \quad , \quad d \ge 5 \quad .$$

III.3.2.4

The expected number of blocking k-pairs (A,B) for d+1 ≤ |B| ≤ 2n- d is

$$\beta^{(d)} = 2\{\beta^{(d)}_1 + \beta^{(d)}_2\} \quad . \qquad (*12)$$

The constant factor in (*12) is due to the fact that $\beta^{(d)}$ is symmetric in $k = |B|$ about the mid-range $(2n+1)/2$. Let (A,B) be the blocking k-pair, the pair (S-B, T-A) is also a blocking k-pair, by applying the definition of blocking k-pairs to the pair (S-B, T-A). Substituting the upper bounds for $\beta_1^{(d)}$ and $\beta_2^{(d)}$ in (*12), we have

$$\beta^{(d)} \le 2\left\{\frac{1}{8\pi}\left(\frac{d}{n}\right)^{d^2-4d-1} + \frac{1}{2e^2}\left(\frac{d}{n}\right)^{d^2-3d-3}\right\}$$

$$\le 2\left\{\frac{1}{8\pi} + \frac{1}{4e^2}\right\}\left(\frac{d}{n}\right)^{d^2-4d-1} .$$

So,

$$\beta^{(d)} < \frac{1}{6}\left(\frac{d}{n}\right)^{d^2-4d-1} , \quad d \ge 5 . \qquad (*13)$$

III.3.3 Probability of a perfect matching in G_d

By the Konig-Hall's theorem, there exists a perfect matching in G_d iff there does not exist a blocking k-pair in G_d, for $2d < k \le 2n - d$. The probability of a perfect matching in G_d is equivalent to find the probability of no blocking k-pairs in G_d. We have the following theorem.

Theorem 1

Let $p_{2n,d}$ be the probability of a perfect matching in G_d, the following inequalities hold:

$$1-p_{2n,d} \begin{cases} < \frac{1}{6}\left(\frac{d}{n}\right)^{d^2-4d-1} , & 5 \le d \le n/2 \\ = 0 , & d > n/2 . \end{cases}$$

III.4 Expected value of the assignment sum

Let $y_{r,n}^{(i)}$ be the r^{th} smallest element in the set

$$Y_i = \{y_1^{(i)}, y_2^{(i)}, \ldots, y_n^{(i)}\} \quad , i \in S \cup T ,$$

such that

$$y_{1,n}^{(i)} \le y_{2,n}^{(i)} \le \ldots \le y_{r,n}^{(i)} \le \ldots \le y_{n,n}^{(i)} \quad .$$

If $y_j^{(i)}$, $j = 1,2,\ldots,n$, are independent and identically uniformly distributed, the expected value of the random variable $y_{r,n}^{(i)}$ is

$$E\,(y_{r,n}^{(i)}) = \frac{r}{n+1} \quad .$$

Now, consider the set which consists of the d smallest elements in Y_i, say,

$$D_i = \{y_{K,n}^{(i)} \mid 1 \le K \le d, \; y_{K,n}^{(i)} \in Y_i\} \quad .$$

The expected value of the rank K of the element $y_{K,n}^{(i)}$ in D_i is

$$E(K \mid y_{K,n}^{(i)} \in D_i) = \sum_{K=1}^{d} K \quad \frac{1}{d} = \frac{1}{d}\,\frac{d(d+1)}{2} = \frac{d+1}{2} \quad ,$$

since all the elements in D_i have the same probability $\frac{1}{d}$, $d \ge 5$. The expected value of $y_{K,n}^{(i)}$ in D_i is

$$E(y_{K,n}^{(i)} \mid y_{K,n}^{(i)} \in D_i)$$

$$= \frac{1}{n+1}\, E(K \mid y_{K,n}^{(i)} \in D_i)$$

$$= \frac{d+1}{2(n+1)} .$$

Let $G^{(M)}$ be the set of all subgraphs of G that contain at least one perfect matching. Let θ_d be the sum of an arbitrary matching M_d in the random directed bipartite graph $G_d = (S,T,E_d)$ if $G_d \in G^{(M)}$. Let Y_i be the set of random variables for vertex i. Define

$$\theta_d = \begin{cases} \sum_{i=1}^{2n} y_{K,n}^{(i)} & \text{if } G_d \in G(M) \\ 2n & \text{if } G_d \notin G^{(M)} . \end{cases}$$

If the costs are independent and identically uniform on [0,1], the expected value of the arbitrary weight matching M_d given that $G_d \in G^{(M)}$ is

$$E(\theta_d \mid G_d \in G^{(M)})$$

$$= E(\sum_{i=1}^{2n} y_{K,n}^{(i)} \mid y_{K,n}^{(i)} \in D_i)$$

$$= \sum_{i=1}^{2n} E(y_{K,n}^{(i)} \mid y_{K,n}^{(i)} \in D_i)$$

$$= \sum_{i=1}^{2n} \frac{d+1}{2(n+1)}$$

$$= \frac{(d+1)n}{n+1} . \qquad (*14)$$

Moreover, $E(\theta_d \mid G_d \notin G^{(M)}) = E(2n) = 2n$.

The graph G_d, $d \geq 1$, clearly satisfies the inclusion property:

$$G_1 \subset G_2 \subset \ldots \subset G_{d-1} \subset G_d \subset \ldots \subset G_n = G .$$

It follows that if $G_d \notin G^{(M)}$ then $G_{d'} \notin G^{(M)}$, for $1 \le d' \le d$. The expected rank K of the variable $y^{(i)}_{K,n}$ given that $y^{(i)}_{K,n} \in D_i$, $G_{d'} \notin G^{(M)}$ and $G_d \in G^{(M)}$ is

$$E(K \mid G_{d'} \notin G^{(M)},\ G_d \in G^{(M)} \text{ and } y^{(i)}_{K,n} \in D_i)$$

$$= \sum_{r=d'+1}^{d} (r)\left(\frac{1}{d-d'}\right)$$

$$= \frac{1}{d-d'}\left(\frac{(d-d')(d+d'+1)}{2}\right)$$

$$= \frac{d+d'+1}{2} \quad .$$

Therefore, the expected sum of the matching M_d such that $G_d \in G^{(M)}$ and $G_{d'} \notin G^{(M)}$, $d' < d$, is

$$E(\theta_d \mid G_{d'} \notin G^{(M)} \text{ and } G_d \in G^{(M)})$$

$$= E\left(\sum_{i=1}^{2n} y^{(i)}_{K,n} \mid G_{d'} \notin G^{(M)},\ G_d \in G^{(M)},\ y^{(i)}_{K,n} \in D_i\right)$$

$$= \sum_{i=1}^{2n} E(y^{(i)}_{K,n} \mid G_{d'} \notin G^{(M)},\ G_d \in G^{(M)},\ y^{(i)}_{K,n} \in D_i)$$

$$= \sum_{i=1}^{2n} \frac{1}{n+1} E(K \mid G_{d'} \notin G^{(M)},\ G_d \in G^{(M)},\ y^{(i)}_{K,n} \in D_i)$$

$$= \left(\frac{1}{n+1}\right)\left(\frac{d+d'+1}{2}\right)(2n)$$

$$= \frac{(d+d'+1)(n)}{n+1} \quad . \qquad (*15)$$

III.4.1

Let $m_d(\underline{x})$ be the approximate solution to the assignment problem (minimum weight matching problem) found by the heuristic $HEUR_{d,OPT}$, $5 \le d \le n$. Recall that

$$P(G_d \notin G^{(M)}) = 1-p_{2n,d} \begin{cases} \le \frac{1}{6}\left(\frac{d}{n}\right)^{d^2-4d-1} & , 5 \le d \le n/2 \ , \\ = 0 & , n/2 < d \le n \ . \end{cases} \qquad (*16)$$

The upper bound of $E(m_d(\underline{x}))$, $5 \le d \le n$, will be shown to be less than 6. There are two cases to be considered, the case when d=5 and the case when $6 \le d \le n$.

III.4.1.1

For d=5, the expected value of $m_5(\underline{x})$ when $n \ge 2d$ is,

$$\begin{aligned} &E(m_5(\underline{x})) \\ &\le E(\theta_5 \mid G_5 \in G^{(M)})P(G_5 \in G^{(M)}) + E(\theta_5 \mid G_5 \notin G^{(M)})P(G_5 \notin G^{(M)}) \\ &\le \frac{(5+1)n}{n+1} + (2n)\left(\frac{1}{6}\left(\frac{5}{n}\right)^4\right) \\ &< 6 \ . \end{aligned}$$

Similarly, the expected value of $m_5(\underline{x})$ when $d > n/2$ is

$$\begin{aligned} E(m_5(\underline{x})) &\le E(\theta_5 \mid G_5 \in G^{(M)})(1) + E(\theta_5 \mid G_5 \notin G^{(M)})(0) \\ &< 6 \ , \end{aligned}$$

since $P(G_5 \notin G^{(M)}) = 0$ when $d > n/2$. Hence, the expected value of $m_5(\underline{x})$ for $n \ge 6$ is less than 6.

III.4.1.2

For $n \geq d \geq 6$, the expected value of $m_d(\underset{\sim}{x})$ is

$$
\begin{aligned}
&E(m_d(\underset{\sim}{x})) \\
&\leq E(\theta_5 \mid G_5 \in G^{(M)}) \;\; P(G_5 \in G^{(M)}) \\
&\quad + E(\theta_6 \mid G_5 \notin G^{(M)} \text{ and } G_6 \in G^{(M)}) \;\; P(G_5 \notin G^{(M)}) \\
&\quad + E(\theta_d \mid G_d \in G^{(M)} \text{ and } G_6 \notin G^{(M)}) \;\; P(G_6 \notin G^{(M)}) \\
&\quad + E(\theta_d \mid G_d \notin G^{(M)}) \;\; P(G_d \notin G^{(M)}) \,. \qquad (*17)
\end{aligned}
$$

(a) $6 \leq d \leq n/2$

Substituting inequalities (*14), (*15) and (*16) into (*17), we have,

$$
\begin{aligned}
&E(m_d(\underset{\sim}{x})) \\
&\leq \frac{6n}{n+1} + \frac{12}{n+1}\left(\frac{1}{6}\left(\frac{5}{12}\right)^4\right) + \frac{(d+7)n}{(n+1)}\left(\frac{1}{6}\left(\frac{6}{n}\right)^{11}\right) + (2n)\frac{1}{6}\left(\frac{d}{n}\right)^{d^2-4d-1} \\
&\leq \frac{6n}{n+1} + \frac{1}{16(n+1)} + \frac{1}{71(n+1)} + \frac{1}{3}\,\frac{d}{2^d}\,\left(\frac{d}{n}\right)^{d^2-5d-2} .
\end{aligned}
$$

Because $d/(2^d)$ and $(d/n)^{d^2-5d-2}$ are nonincreasing w.r.t. d when $6 \leq d \leq n/2$, substitute $d = 6$ and $n = 12$ into the expression and we obtain

$$
\begin{aligned}
&E(m_d(\underset{\sim}{x})) \\
&\leq \frac{1}{n+1}\left(6n + \frac{1}{16} + \frac{1}{71} + \frac{13}{512}\right) \\
&< 6 \,.
\end{aligned}
$$

(b) $n/2 < d \leq n$

When $d > n/2$, $P(G_d \not\subset G^{(M)}) = 0$. The last term in (*17) is equal to zero. Using the same derivation of $E(m_d(\underline{x}))$ for $6 \leq d \leq n/2$, except by setting $P(G_d \not\subset G^{(M)}) = 0$, the bound for $E(m_d(\underline{x}))$ is

$$E(m_d(\underline{x})) \leq \frac{1}{n+1} (6n + \frac{1}{16} + \frac{1}{71}) < 6 \quad .$$

III.4.1.3

From the results derived in section III.4.1 and III.4.2, we have the following theorem.

Theorem 2

If the costs are independent and indentically uniformly distributed in [0,1], the expected assignment sum $E(m_d(\underline{x}))$ is less than 6 for $n \geq d \geq 5$.

CHAPTER IV

Computational Results

This chapter will discuss the theoretical and practical time and space complexity of the heuristic $HEUR_{d,OPT}$. A comparison of the running time and the solution calculated by the heuristic and an exact algorithm is made.

IV.1 Worst case time complexity of $HEUR_{d,OPT}$

The heuristic $HEUR_{d,OPT}$ consists of four main procedures, namely, DUMMY, CONSTRUCT, MATCH and FILLUP (figure 7). Clearly, the worst case time complexity for DUMMY is $O(|S|^2)$. For the procedure CONSTRUCT, at most we have to check $d|S|^2$ elements in $G = (S,T,E)$; the time complexity for CONSTRUCT is $O(|S|^2)$. Because G_d has $2d|S|$ edges, an $O(|S|\,|E|)$ exact algorithm to solve G_d would give $O(|S|^2)$ time. Thus, MATCH has $O(|S|^2)$ time complexity. The row scan method used in FILLUP requires at most $\binom{S}{2}$ steps. The worst case time complexity of FILLUP is $O(|S|^2)$. Therefore, the overall time complexity for $HEUR_{d,OPT}$ is $O(|S|^2)$.

IV.2 Space Complexity of $HEUR_{d,OPT}$

The computer memory space required to store the graph $G = (S,T,E)$, the subgraph $G_d = (S,T,E_d)$ and the matching M_d are $O(|E|)$, $O(|E_d|)$ and $O(|S|)$ respectively. Moreover, the memory space required to do the computation for most of the exact algorithms described in section I.2.1 is $O(|S|)$ in general. Thus, the overall space complexity of $HEUR_{d,OPT}$ is $O(|E|)$.

IV.3 Experimental Results

The heuristic $HEUR_{5,HPD}$ and the hungarian primal-dual algorithm HPD are compared for some random instances. The costs are generated by a random number generator which produces random numbers uniform on [0,1].

Let t_o be the running time in computer service time units for the hungarian primal-dual algorithm, $t_a(d)$ be the running time in service units for the heuristic $HEUR_{d,OPT}$, OPT(G) be the solution (assignment sum) given by an exact algorithm OPT, $HEUR_{d,OPT}(G)$ be the solution given by $HEUR_{d,OPT}$, and $\xi = (HEUR_{d,OPT}(G) - OPT(G))/OPT(G)$.

Table 1 shows the means and standard deviations of t_o and $t_a(5)$ for different orders of graphs G. Since the computing speed varies from computer to computer, the ratio $t_o/t_a(5)$ is also calculated (table 2). By using simple curve fitting technique, the equations for $\bar{t}_o$, $\bar{t}_a(5)$ and $\overline{t_o/t_a(5)}$ were determined (table 3). The coefficients of correlations $P(|S|^3, \bar{t}_o)$, $P(|S|^2, \bar{t}_a(5))$ and $P(|S|, \overline{t_o/t_a(5)})$ were almost equal to 1.000. These show that the equations are quite appropriate. Furthermore, the equations also confirm the theoretical results that the time complexity of HPD is $O(|S|^3)$ and $HEUR_{d,HPD}$ is $O(|S|^2)$. The coefficients of variations $\gamma(t_o)$, $\gamma(t_a(5))$ and $\gamma(t_o/t_a(5))$ were less than 6%, which show that the values of t_o, $t_a(5)$, $t_o/t_a(5)$ are almost constant for fixed $|S|$. The case when $t_a(5) \geq t_o$ was when $|S| \leq 22$. For $|S| = 22$, $t_a(5) \leq t_o$. For $|S| = 20$, all three cases: $t_a(5) < t_o$, $t_a(5) = t_o$ and $t_a(5) > t_o$ occurred quite evenly. For $|S| = 18$, $t_a(5) > t_o$ happened on every occasion.

The experimental results for HPD(G) and $HEUR_{5,HPD}(G)$ are shown in table 4. As a whole, $\overline{HPD(G)} < 1.5846$, $\overline{HEUR_{5,HPD}(G)} < 1.5909$, and the standard deviations of HPD(G) and $HEUR_{5,HPD}(G)$ were 0.2204 and 0.2207 respectively. $HEUR_{5,HPD}(G)$ and HPD(G) were quite close to each other (table 5 and table 6). The probability of the event $\xi = 0$ to occur was greater than $\frac{2}{3}$ at worst in the experiment. Further, $\max\{\xi\} < 0.048$, $\min\{\xi \mid \xi \neq 0\} > 0.0009$, and $\max\{HEUR_{5,HPD}(G) - HPD(G)\} < 0.08$ were observed. It is observed that if $|S|$ increases, $|HEUR_{5,HPD}(G) - HPD(G)|$ decreases for $|S| \geq 50$.

Table 1. Comparison of the average running time of HPD and $HEUR_{d,HPD}$.

$\|S\|$	Number of trials	$\bar{t}_o$	$\sigma(t_o)$	$\gamma(t_o) = \frac{\sigma(t_o)}{\bar{t}_o}$
22	25	0.234	0.0129	0.0552
50	25	2.1364	0.0849	0.0398
100	25	15.1576	0.3775	0.0249
150	10	49.4760	1.1904	0.0241

$\|S\|$	$\overline{t_a(5)}$	$\sigma(t_a(5))$	$\gamma(t_a(5))$
22	0.2184	0.0125	0.0571
50	1.0204	0.0335	0.0328
100	4.0300	0.1258	0.0312
150	9.0550	0.2507	0.0277

Table 2. means and standard deviations of $t_o/t_a(5)$.

$\|S\|$	$\overline{t_o/t_a(5)}$	$\sigma(t_o/t_a(5))$	$\gamma(t_o/t_a(5))$
22	1.07259	0.04884	0.04554
50	2.09361	0.04729	0.02259
100	3.76267	0.08097	0.02152
150	5.46462	0.05300	0.00970

Table 3. Equations for $t_o, t_a(5)$, and $t_o/t_a(5)$.

y	x	$y = bx + c$		$P(x,y)$
		b	c	
$\overline{t_o}$	$\|S\|^3$	1.4597×10^{-5}	0.290128	0.999996
$\overline{t_a(5)}$	$\|S\|^2$	4.01495×10^{-4}	1.9286×10^{-2}	0.999999
$\overline{t_o/t_a(5)}$	$\|S\|$	3.41483×10^{-2}	0.349453	0.999898

Table 4. Experimental results for HPD(G) and $HEUR_{5,HPD}(G)$.

Let opt = HPD(G) and app = $HEUR_{5,HPD}(G)$.

S	$\overline{opt}$	σ(opt)	min {opt}	max {opt}
22	1.5446	0.3159	0.9292	2.3250
50	1.5468	0.1788	1.0977	1.9195
100	1.6487	0.1661	1.4008	2.0564
150	1.6186	0.1367	1.3932	1.8849

|S|	$\overline{app}$	σ(app)	min {app}	max {app}
22	1.5452	0.3153	0.9292	2.3250
50	1.5516	0.1791	1.0977	1.9195
100	1.6644	0.1491	1.4008	2.0741
150	1.6195	0.1370	1.3932	1.8849

Table 5. Experimental results for $\xi = \dfrac{HPD(G) - HEUR_{5,HPD}(G)}{HPD(G)}$.

$\|S\|$	$\bar{\xi}$	$\sigma(\xi)$	max $\{\xi\}$	min $\{\xi\}$
22	0.000500	0.002498	0.012489	0.012448
50	0.003106	0.009907	0.047968	0.000898
100	0.002214	0.004267	0.015606	0.001379
150	0.000553	0.001287	0.003923	0.001602

$\|S\|$	$\bar{\xi}$ given that $\xi \neq 0$	$\sigma(\xi\|\xi \neq 0)$	frequency of $\xi \neq 0$	size of sample
22	0.012489	-	1	25
50	0.015531	0.018646	5	25
100	0.006919	0.005021	8	25
150	0.002763	0.001641	2	10

Table 6. Experiment results for δ = HPD(G) − $\text{HEUR}_{5,\text{HPD}}$(G).

$\lvert S \rvert$	$\bar{\delta}$	$\sigma(\delta)$	min $\{\delta \mid \delta \neq 0\}$	max $\{\delta\}$
22	0.000637	0.003186	0.015930	0.015930
50	0.004707	0.015109	0.001633	0.073014
100	0.003376	0.007277	0.002396	0.026840
150	0.000913	0.002102	0.002762	0.006360

$\lvert S \rvert$	$\bar{\delta}$ given that $\delta \neq 0$	$\sigma(\delta \mid \delta \neq 0)$
22	0.015930	-
50	0.023535	0.028563
100	0.011740	0.008630
150	0.004563	0.002546

IV.4 Bad instances for the heuristic $HEUR_{d,OPT}$.

Consider the graph $G_d = (S_1 \cup S_2, T_1 \cup T_2, E_d)$, where $|S_1 \cup S_2| = |T_1 \cup T_2| = m = 2n$; the number of exposed vertices in G_d relative to the matching $M_d \in G_d$ is maximum if and only if (figure 11)

(a) $\Gamma^+(S_i) = T_i^*,\ T_i^* \subseteq T_i,\ |T_i^*| = d,\ i = 1,2,$

and (b) $\Gamma^+(T_i) = S_j^*,\ S_j^* \subseteq S_j,\ |S_j^*| = d,\ j = 3-i,\ i = 1,2.$

Machol and Wien [44] give an instance, G^*, which force the hungarian method to the worst case time bound. A modification of the cost matrix $C^* = (c_{ij}^*)$ that they have given is defined as follow:

$$c_{ij}^* = \begin{cases} 0 & i = 1 \text{ or } j = 1 \\ 2(i-1)(2j-i)/N & 1 < i \le j \le m \\ \{2(j-1)(2i-j)-1\}/N & 1 < j < i \le m \end{cases}$$

where $N = 2(m)(m-1)$. The modification converts the original costs to the range of 0 to 1. This instance will cause $HEUR_{d,OPT}$ to give the maximum number of exposed vertices in G_d (figure 12). Machol and Wien show that the optimal assignment sum is $OPT(G^*) = \{1 + m(m^2 - 2m - 2)/4\}/N$, and the permutation matrix $\underline{x} = (x_{ij})$ is defined by $x_{ij} = 1$ if $i+j = m+1$, and $x_{ij} = 0$ otherwise (figure 13). The possible entries selected by $HEUR_{d,OPT}$ is shown in figure 12 and 13. It can be easily shown that

$$HEUR_{d,OPT}(G^*) - OPT(G^*) \le \frac{5m+12}{24}.$$

In fact, the graph G' in which the costs have following relationships:

(a) $c_{ij} > c_{i,j-1}$,

and (b) $c_{ij} > c_{i-1,j}$,

may give the results shown in figure 12 and 13. For example, the graph G'' in which the costs c''_{ij} are defined as follows:

$$c''_{ij} = \begin{cases} i \cdot j/m^m & 1 \le i,j \le m \text{ but } i = j \ne m \\ 1 . & i = j = m \end{cases}$$

is such a graph. The graph G'' may force $HEUR_{d,OPT}$ to choose the edge (m,m) if $4d < m$. The optimal assignment sum for G'' is $(m(m+1)(m+2))/(6m^m)$. But $HEUR_{d,OPT}(G'') = c_{mm}$ + positive terms > 1. Therefore, the ratio

$$\frac{HEUR_{d,OPT}(G'') - OPT(G'')}{OPT(G'')} > 6m^{m-4} - 1 .$$

Figure 11. A bad instance for $HEUR_{d,OPT}$.

Graph $G_d = (S_1 \cup S_2, T_1 \cup T_2, E_d)$:

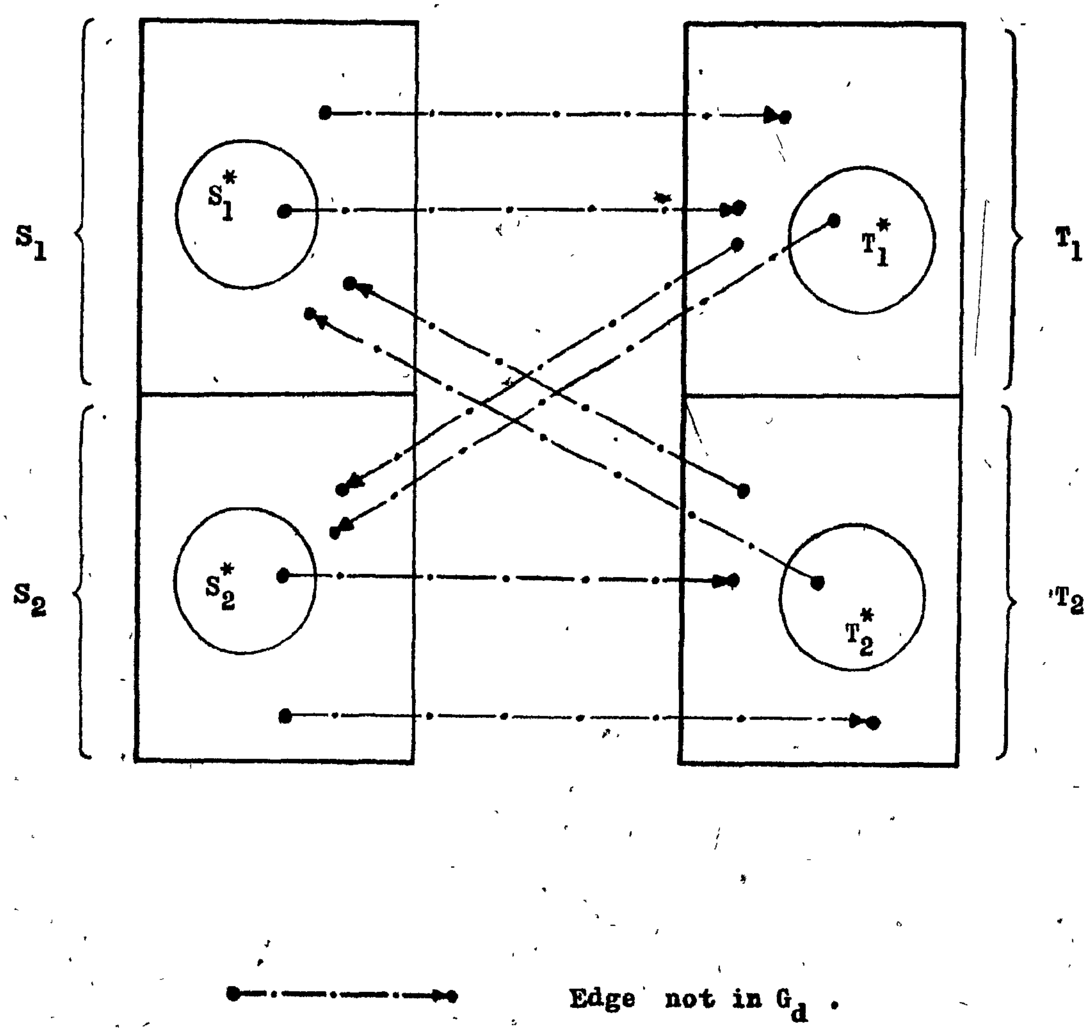
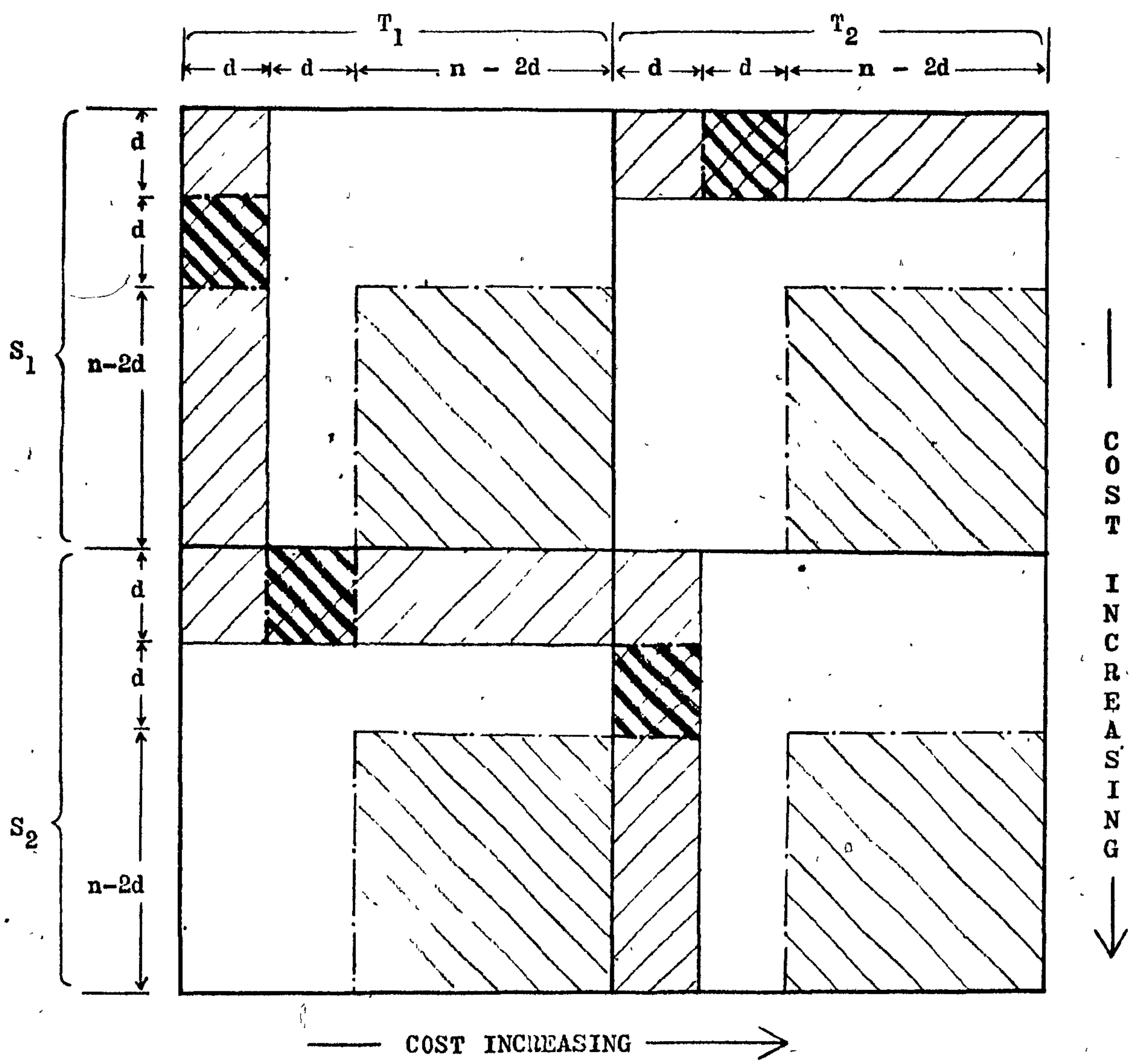


Figure 12. Possible entries (edges) in the cost matrix of the graph to be selected by $HEUR_{d,OPT}$.

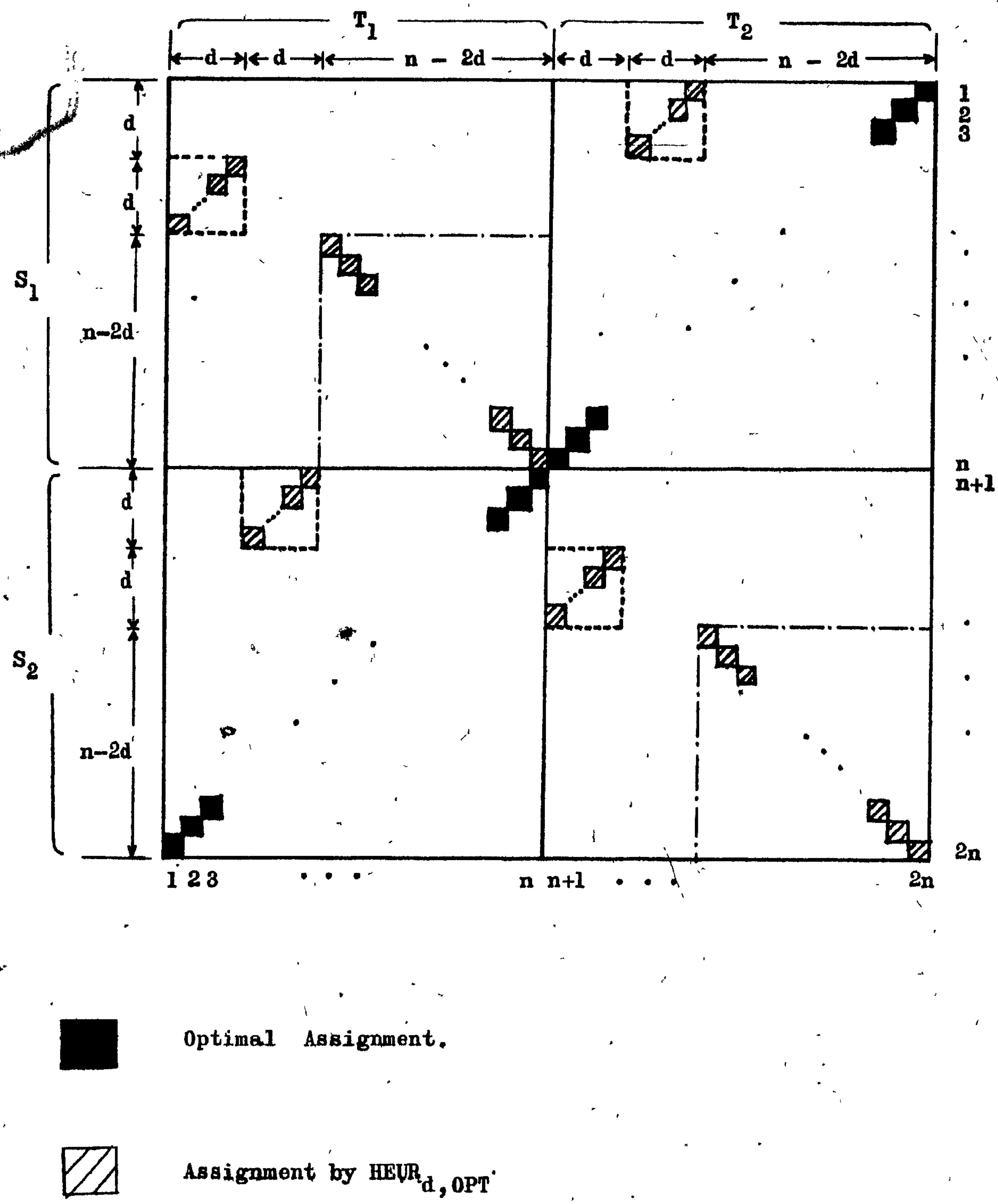


Edges in G_d .

Set of edges in G_d containing the matching M_d of G_d .

Edges incident to the exposed vertices relative to M_d in G_d

Figure 13. **Entries (edges) in the cost matrix of the instance, G^* or G'', to be selected by the exact algorithm and $HEUR_{d,OPT}$.**



CHAPTER V

Applications and related problems

V.1

One of the applications of the assignment problem is the marriage problem; Dantzig [16] shows that monogany is the best type of marriage. An important application of the assignment problem is that it can be used directly in the development of the solution algorithms for the well known travelling salesman problem TSP [13,14,15,35,36,43]. The travelling salesman problem is defined as follows: given a graph G = (V,E) with nonnegative edge weights, we have to find a minimum cost cycle (a closed path) which passes each vertex in G exactly once. A symmetric TSP is when the costs $c_{ij} = c_{ji}$, $i,j \in V$, $i \neq j$. A nonsymmetric TSP is when the costs c_{ij} and c_{ji}, $i \neq j$, $i,j \in V$, may be different.

Karp [37] proposes an $O(n^3)$ approximate algorithm for the nonsymmetric travelling salesman problem, where n is the number of cities. His algorithm first solves the problem as an assignment problem and then patches together the cycles of the resulting permutation to form a tour. He shows that the ratio of the length of the tour found by his algorithm to the optimal length of the tour is less than $1 + \epsilon(n)$ with probability tending to 1, where $\epsilon(n) \to \infty$ as $n \to \infty$ if all the edge weights are independent uniform on [0,1]. His proof involves the upper bound of 3 on the expected value of the random assignment problem given by Walkup [54]. The computational time of Karp's algorithm can be improved by replacing the exact assignment algorithm by $HEUR_{d,OPT}$. The theoretical result of the ratio, that is, the length of the tour by the approximate algorithm to the optimal tour, is unchanged if we modify the proof in [37] as follows:

(a) the exact algorithm is replaced by $HEUR_{d,OPT}$,

(b) the upper bound of the expected value of the assignment is changed to 6,

and (c) the factor, $\sqrt{3n\ell n4} + 1$, in the definition of the *bad matrix* is replaced by the factor $\sqrt{3n\ell n7} + 1$.

V.2

The time complexity of $HEUR_{d,OPT}$ is bounded below by the size of the graph G. Clearly, $HEUR_{d,OPT_1}$ computes faster than $HEUR_{d,OPT_2}$ does if OPT_1 computes faster than OPT_2. Some exact algorithms compute faster in solving sparse assignment problems [5,42], while some exact algorithms do not [12]. Since G_d is a sparse graph, $d < n/2$, we need an exact algorithm which is designed to solve sparse assignment problems.

A variant of $HEUR_{d,OPT}$ is shown in figure 14. The procedure to find an augmenting path P in G_d is $O(|E_d|)$. Thus, the algorithm $VARN_d$ (figure 14) has $O(|S|^2)$ time complexity. $VARN_d$ only chooses a random matching M_d^* in G_d; so, the sum of the costs in M_d^* is greater than or equal to $OPT(G_d)$. Hence, $VARN_d(G) \geq HEUR_{d,OPT}(G)$ holds. However, $VARN_d$ is easier to implement than $HEUR_{d,OPT}$ because it is not necessary to find the minimum weight augmenting path in G_d. The upper bound for the expected value of $VARN_d(G)$ is the same as that of $HEUR_{d,OPT}(G)$.

Another way to improve the computational time of $HEUR_{d,OPT}$ is to find an efficient algorithm to find the d smallest elements in a set of unordered numbers. A method to select the d smallest elements is by means of an algorithm to select the d^{th} smallest element $y_{d,n}$ from an unordered set of numbers. After $y_{d,n}$ is found, we have to select the d-1 elements in the set of numbers which are smaller than $y_{d,n}$. There are fast algorithms to select the d^{th} smallest element from an unordered set of numbers [1,6,27,33, 40,47,49,56].

Figure 14. Algorithm $VARN_d$.

```
PROCEDURE  VARN_d[G_o,M]
     Input:  Graph G_o = (S,T,E_o).
     Output:  A matching M.
BEGIN
     DUMMY [G_o,G]
     d* ← 0 ; M ← ∅ ; G_o ← G ; E_1 ← ∅
     G*_d ← (S,T, {∅})
     WHILE  d* ≤ d and |M| < |S|  DO
          d* ← d* + 1 ; G ← G - E_1
          CONSTRUCT [G,G_1,1] ; where G_1 = (S,T,E_1).
          E_d* ← E_d* ∪ E_1
          G_d* ← (S,T,E_d*)
          WHILE there exists an augmenting path P in G_d* relative
               to M  DO
               M ← (M ∪ P) - (M ∩ P)
          END
     END
     IF |M| < |S| THEN FILLUP[G_o,M]
END.
```

CHAPTER VI

Conclusion

The heuristic $HEUR_{d,OPT}$, $d \geq 5$, has been shown to give optimal or near optimal solution to the assignment problem in which the costs are independent and identically uniformly distributed on [0,1] in $O(|S|^2)$ time complexity. Theoretically, the expected value of the assignment sum is less than 6. Experimentally, it is bout 1.6. Comparison of algorithms for the assignment problem is shown in table 7.

Table 7. Comparison of algorithms for the assignment problem given that the costs are uniform on (0,1).

Algorithm	Upper bound for the expected value of the assignment sum.	Worst case time Complexity
Hungarian Primal-Dual Algorithm	3	$O(\|S\|^3)$
Multiple-Column Scan Method	2.37	$O(\|S\|^2 \ln \|S\|)$
Row/Column Scan Method	$\ln(\|S\|)$	$O(\|S\|^2)$
$HEUR_{d,OPT}$, $d \geq 5$	6	$O(\|S\|^2)$

APPENDIX A

Proofs of Lemmas

Lemma 1

(1.1) $\binom{m}{k} \le \frac{1}{\sqrt{2\pi k}}\left(\frac{em}{k}\right)^k$, $0 < k \le m$.

(1.2) $\binom{p}{j}\binom{q}{j}^{-1} \le \left(\frac{p}{q}\right)^j$, $0 \le j \le p \le q$.

Proof:

(1.1) $\binom{m}{k} = \frac{m(m-1)\ldots(m-k+1)}{k!}$

$\le m^k \{k^{k+1/2} e^{-k} (2\pi)^{1/2}\}^{-1}$, by Stirling's inequality,

$\le \frac{1}{\sqrt{2\pi k}} \left(\frac{em}{k}\right)^k$. Q.E.D.

(1.2) It is trivial since $\frac{p}{q} \ge \frac{p-i}{q-i}$ for $p \le q$ and $0 \le i \le p$.

Lemma 2

If $a_1+a_2 = k-1$, $b_1+b_2 = k$, $2 \le d+1 \le k \le n$, and a_1, a_2, b_1, $b_2 \ge 0$, then

$$\left(\frac{a_1}{n}\right)^{b_1} \left(\frac{a_2}{n}\right)^{b_2} \le \left(\frac{k-1}{n}\right)^k .$$

Proof:

By the binomial theorem,

$$\left(\frac{k-1}{n}\right)^k = \sum_{i=0}^{k} \binom{k}{i} \left(\frac{a_1}{n}\right)^i \left(\frac{a_2}{n}\right)^{k-i} .$$

Hence, the lemma holds.

Lemma 3

If $a_1 + a_2 = k-1$, $b_1 + b_2 = k$, $d \le a_1, a_2 \le k-1-d$, $0 \le b_1, b_2 \le k$, $2d + 1 \le k \le n$, and $d \ge 1$, then

$$\left(\frac{n-b_1}{n}\right)^{n-a_2} \left(\frac{n-b_2}{n}\right)^{n-a_1} \le \left(\frac{n-k+d}{n}\right)^{n-k+d} \left(\frac{n-d}{n}\right)^{n-d} .$$

Proof:

Let $a = a_2$, $b = b_1$, $a_1 = k-a-1$ and $b_2 = k-b$, then

$$\left(\frac{n-b_1}{n}\right)^{n-a_2} \left(\frac{n-b_2}{n}\right)^{n-a_1}$$

$$\le \frac{(n-b)^{n-a} \; (n-k+b)^{n-k+a}}{n^{2n-k}} , \quad \text{as } n \ge n-k+b .$$

[Case 1]

Let $y(a,b) = (n-b)^{n-a} \; (n-k+b)^{n-k+a}$, partial differentiating $y(a,b)$ w.r.t. b, we obtain

$$\frac{\partial}{\partial b} y(a,b) = (n-b)^{(n-a)-1} (n-k+b)^{(n-k+a)-1} [(n-b)(n-k+a) - (n-a)(n-k+b)] .$$

[case 1.1] If $(n-b) = 0$ or $(n-k+b) = 0$, then $y(a,b) = 0$.

[case 1.2]

$$\frac{\partial}{\partial b} y(a,b) \begin{cases} > 0 & \text{if } b < a \\ = 0 & \text{if } b = a \\ < 0 & \text{if } b > a \end{cases}$$

That is to say, $y(a,a) \geq y(a,b)$.

[case 2]

Differentiating y(a,a) w.r.t. a, we obtain the following:

$$\frac{\partial}{\partial a} y(a,a) = (n-a)^{n-a} \ (n-k+a)^{n-k+a} \ \ln(\frac{n-k+a}{n-a}) \ .$$

Thus,

$$\frac{\partial}{\partial a} y(a,a) \begin{cases} > 0 & \text{if } a > k/2 \\ = 0 & \text{if } a = k/2 \\ < 0 & \text{if } a < k/2 \ . \end{cases}$$

This implies that $y(k/2, k/2) \le y(a,a)$ for all a,

$y(a^-,a^-) < y(a,a)$ for $k/2 \le a^- < a$, and $y(a,a) > y(a^+,a^+)$ for $a < a^+ \le k/2$.

But $d \le a \le k-1-d < k-d$ and $y(d,d) = y(k-d,k-d)$, the following inequalities hold:

$$y(a,b) \le y(d,d) = (n-d)^{n-d} \ (n-k+d)^{n-k+d} \ ,$$

or,

$$\frac{y(a,b)}{n^{2n-k}} \le (\frac{n-d}{n})^{n-d} \ (\frac{n-k+d}{n})^{n-k+d} \ .$$

Since $(\frac{n-b_1}{n})^{n-a_2} \ (\frac{n-b_2}{n})^{n-a_1} \le \frac{y(a,b)}{n^{2n-k}}$,

we have,

$$(\frac{n-b_1}{n})^{n-a_2} \ (\frac{n-b_2}{n})^{n-a_1} \le (\frac{n-k+d}{n})^{n-k+d} \ (\frac{n-d}{n})^{n-d} \ . \qquad \text{Q.E.D.}$$

Corollary 1

If $a_1 + a_2 = k-1$, $b_1 + b_2 = k$, $0 \le a_1, a_2 \le k-1$, $0 \le b_1, b_2 \le k$, $d + 1 \le k \le n$, and $d \ge 1$, then

$$\left(\frac{n-b_1}{n}\right)^{n-a_2} \left(\frac{n-b_2}{n}\right)^{n-a_1} \le \left(\frac{n-k}{n}\right)^{n-k}.$$

Proof:

Using the notations given in lemma 3, we have the following inequalities:

$$y(k,k) > y(k-d,\ k-d) = y(d,d)\ ,$$

$$\frac{y(d,d)}{n^{2n-k}} < \frac{y(k,k)}{n^{2n-k}} = \left(\frac{n-k}{n}\right)^{n-k}\ ,$$

and $$\left(\frac{n-b_1}{n}\right)^{n-a_2} \left(\frac{n-b_2}{n}\right)^{n-a_1} < \frac{y(d,d)}{n^{2n-k}}$$, from lemma 3,

Therefore, the inequality holds.

Lemma 4

Let $a_1 + a_2 = k-1$, $b_1 + b_2 = k$, $0 \le a_1, a_2 \le k-1-d$; $0 \le b_1, b_2 \le k$, and if (1) $a_1 \ne 0$, $a_2 \ne 0$, and $d+1 \le k \le n$, or (2) $a_1 = 0$ or $a_2 = 0$, and $d+1 \le k \le n-d$, then

$$\alpha_d = \left(\frac{a_1}{n}\right)^{b_1} \left(\frac{a_2}{n}\right)^{b_2} \left(\frac{n-b_1}{n}\right)^{n-a_2} \left(\frac{n-b_2}{n}\right)^{n-a_1} \le \left(\frac{d}{n}\right)^{d}, \quad d \ge 1 .$$

Proof:

Case 1. ($a_1 \ne 0$, $a_2 \ne 0$ and $d+1 \le k \le n$)

$$\alpha_d \le \left(\frac{k-1}{n}\right)^{k} \left(\frac{n-k+d}{n}\right)^{n-k+d} \le \left(\frac{k}{n}\right)^{k} \left(\frac{n-k+d}{n}\right)^{n-k+d}, \text{ from lemma 2 and 3.}$$

Let $y(k) = k^k (n-k+d)^{n-k+d}$, differentiating $y(k)$ w.r.t. k, we have

$$\frac{\partial}{\partial k} y(k) = k^k (n-k+d)^{n-k+d} \ln\left(\frac{k}{n-k+d}\right)$$

or

$$\frac{\partial}{\partial k} y(k) \begin{cases} > 0 & \text{if } k > (n+d)/2 \\ = 0 & \text{if } k = (n+d)/2 \\ < 0 & \text{if } k < (n+d)/2 . \end{cases}$$

This implies that $y(k)$ is a concave (upward) function with minimum point at $((n+d)/2, y((n+d)/2))$. Since $y(d) = y(n)$, we have $y(k) \le y(n)$ for $d+1 \le k \le n$. Thus

$$\alpha_d \le \frac{y(k)}{n^{n+d}} \le \frac{y(n)}{n^{n+d}} = \frac{n^n d^d}{n^{n+d}} = \left(\frac{d}{n}\right)^{d} .$$

<u>Case 2.</u> ($a_1 = 0$ or $a_2 = 0$, and $d + 1 \le k \le n - d$)

If $a_1 = 0$ and $b_1 \ne 0$, or $a_2 = 0$ and $b_2 \ne 0$, then $\alpha_d = 0$. Consider the case when $a_1 = 0$ and $b_1 = 0$, or $a_2 = 0$ and $b_2 = 0$. By lemma 2 and corollary 1, we have

$$\alpha_d \le \left(\frac{k-1}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k} \le \frac{k^k (n-k)^{n-k}}{n^n} .$$

Let $g(k) = k^k (n-k)^{n-k}$. Differentiating $g(k)$ w.r.t. k, we obtain

$$\frac{\partial}{\partial k} g(k) = k^k (n-k)^{n-k} \ln\left(\frac{k}{n-k}\right) ,$$

and

$$\frac{\partial}{\partial k} g(k) \begin{cases} > 0 & \text{if } k > n/2 \\ = 0 & \text{if } k = n/2 \\ < 0 & \text{if } k < n/2 . \end{cases}$$

Therefore, $g(k)$ is concave upward and symmetric about $n/2$. Since $d + 1 \le k \le n - d$, $g(k) \le g(n-d)$. This implies that

$$\alpha_d \le \frac{g(n-d)}{n^n} = \frac{(n-d)^{n-d} d^d}{n^n} \le \left(\frac{d}{n}\right)^d . \qquad \text{Q.E.D.}$$

Lemma 5

If $k-1 \geq a \geq k/2$, $k \geq b \geq k/2$, $a + b > k$, $b > a$, $n \geq k \geq n/2$, $n \geq d + 1$ and $d \geq 1$, then

$$f(a,b,k) = a^{b-a-1}(k-a-1)^{a-b+1}(n-b)^{a+b-k}(n-k+b)^{k-a-b} \leq 1 .$$

Proof:

Case 1. ($k = n$)

Substituting k by n into f(a,b,k), we have

$$f(a,b,n) = \left(\frac{a}{n-a-1}\right)^{b-a-1} \left(\frac{n-b}{b}\right)^{a+b-n}$$

$$\leq \left[\left(\frac{a}{n-a-1}\right)\left(\frac{n-b}{b}\right)\right]^{b-a-1} \text{, since } (n-b)/b \leq 1 \text{ and } b-a-1 < a+b-n,$$

$$\leq 1, \text{ since } b > a \text{ and } n-a-1 \geq n-b.$$

Case 2. ($a = k-1$ and $k \neq n$)

If $a = k-1$ then $b = k$. So, we have

$$f(k-1,k,k) = (k-1)^0\, 0^0\, (n-k)^{k-1}\, n^{1-k} \leq 1.$$

Case 3. ($k-1 > a \geq k/2$ and $k \neq n$)

Consider the logarithmic series:

$$\ln z = 2\left\{\left(\frac{z-1}{z+1}\right) + \frac{1}{3}\left(\frac{z-1}{z+1}\right)^3 + \frac{1}{5}\left(\frac{z-1}{z+1}\right)^5 + \dots\right\}, \quad z > 0,$$

and the inequalities:

$$\ln z < 2 \left\{ \left(\frac{z-1}{z+1}\right) + \frac{1}{3}\left(\frac{\left(\frac{z-1}{z+1}\right)^3}{1 - \left(\frac{z-1}{z+1}\right)^2} \right) \right\} , \quad 0 \leq \frac{z-1}{z+1} < 1 ,$$

and

$$\ln z > 2 \left(\frac{z-1}{z+1} \right) , \qquad 0 \leq \frac{z-1}{z+1} < 1.$$

Let $g(a,k) = \dfrac{a/(k-a-1) - 1}{a/(k-a-1) + 1} = \dfrac{2a-k+1}{k-1} < 1$,

$$h(b,k) = \frac{(n-k+b)/(n-b) - 1}{(n-k+b)/(n-b) + 1} = \frac{2b-k}{2n-k} < 1, \text{ and}$$

$$\begin{aligned}\triangle &= (b-a-1)\, g(a,k) - (a+b-k)\, h(b,k) \\ &\leq (b-a-1)\left[\frac{(2a-k+1)+1}{(k-1)+1}\right] - \left[(b-a-1)+(2a-k+1)\right]\left[\frac{(2a-k+1)+2(b-a-1)+1}{2(\ 2k\ ) - k}\right] \\ &\leq - \frac{(2a-k+1)^2 + (2a-k+1) + 2(b-a-1)^2 - 2(b-a-1)}{3k} \\ &\leq - \frac{(2a-k+1)(2a-k+2)}{3k} .\end{aligned}$$

Taking logarithm on $f(a,b,k)$, we have

$$\begin{aligned}&\tfrac{1}{2} \ln f(a,b,k) \\ &= \tfrac{1}{2} \left[(b-a-1) \ln\left(\frac{a}{k-a-1}\right) - (a+b-k) \ln\left(\frac{n-k+b}{n-b}\right) \right] \\ &\leq (b-a-1)\left[g(a,k) + \frac{1}{3}\left(\frac{2a-k+1}{k-1}\right)^3 \Big/ \left(1-\left(\frac{2a-k+1}{k-1}\right)^2\right)\right] - (a+b-k)\, h(b,k)\end{aligned}$$

$$\leq \quad \Delta + \frac{(b-a-1)}{3} \cdot \frac{(2a-k+1)^2 \ (2a-k+2)}{k \ (2k-2a-2) \ (2a)}$$

$$\leq \quad \frac{(2a-k+1)(2a-k+2)}{3k} \left(-1 + \frac{(b-a-1)(2a-k+1)}{2 \ (k-a-1) \ (2a)} \right)$$

$$\leq \quad 0 \, ,$$

since $2a-k+1 \geq 1$, $k-a-1 \geq b-a-1$ and $2a > 2a-k+1$. Therefore, $f(a,b,k) \leq 1$.

APPENDIX B

Summary of notations

B.1 Arithmetic

$\binom{n}{a}$	$\frac{n(n-1)\cdots(n-a+1)}{a!}$	$a \le n$
	1	$a = 0$
	0	$a > n$

$\max\{x_i \mid i=1,\cdots,n\}$	maximum value in the set $\{x_1,x_2,\cdots,x_n\}$
$\min\{x_i \mid i=1,\cdots,n\}$	minimum value in the set $\{x_1,x_2,\cdots,x_n\}$
$\lvert x \rvert$	absolute value of x
$\lceil a \rceil$	the least integer greater than or equal to a
$\lfloor a \rfloor$	the greatest integer less than or equal to a
$\ln a$	natural logarithm of a
lim	limit operator
w.r.t.	with respect to
iff	if and only if
$\Rightarrow$	implies
$\rightarrow$	tends to or converges to
$x \leftarrow y$	x is assigned the value of y
$f(n) = O(g(n))$	$f(n)/g(n) \le c$, for some constant c, n_o, $n > n_o$
e	$\lim_{n\to\infty}(1 + 1/n)^n$

B.2 <u>Set</u>

ϕ	empty set
$A \cup B$	$\{x \mid x \in A \text{ or } x \in B\}$
$A \cap B$	$\{x \mid x \in A \text{ and } x \in B\}$
$\in$	belongs to
$A \subseteq B$	for all x, $(x \in A) \Rightarrow (x \in B)$
$A \subset B$	$A \subseteq B$, but $A \neq B$
$\lvert A \rvert$	number of elements in the set A
$\bigcup_{i=1}^{n} A_i$	$\{A_1 \cup A_2 \cup \cdots \cup A_n\}$
$\bigcap_{i=1}^{n} A_i$	$\{A_1 \cap A_2 \cap \cdots \cap A_n\}$
$[a,b]$	$\{x \mid a \le x \le b\}$
$(a,b]$	$\{x \mid a < x \le b\}$
$[a,b)$	$\{x \mid a \le x < b\}$
$A - B$	$\{x \mid x \in A \text{ and } x \notin B\}$
$\exists$	there exists
$\ni$	such that

B.3 Probability and Statistic

E(X)	expected value of the random variable X.
Var(X)	variance of the random variable X.
P(X)	probability of the random variable X.
$\overline{X}$	sample mean of X.
Cov(X,Y)	covariance of the random variables X and Y.
P(X,Y)	correlation coefficient of the random variables X and Y.
γ(X)	coefficient of variation of X.
σ(X)	standard deviation of X.
i.i.d.	independent and identically distributed.

B.4 Graph

G = (V,E)	graph with set of vertices V and edges E.
$G - E^*$	removal of all edges in E^* (where g = (V,E) and $E^* \subseteq E$) but removing no vertices.
$G - V^*$	removal of vertices in V^* and all edges adjacent to them (where G = (V,E), $V^* \subseteq V$).
$G_1 \cup G_2$	the graph defined by $(V_1 \cup V_2, E_1 \cup E_2)$, where $G_1 = (V_1,E_1)$, $G_2 = (V_2,E_2)$.
$\Gamma^+(V^*)$	$\{x \mid \langle i,x \rangle \in E,\ i \in V^*,$ x is *incident from* i$\}$, $\lvert\Gamma^+(V^*)\rvert = d_G^+(V^*)$.

$\Gamma^-(V^*)$	$\{x \mid \langle x,i\rangle \in E,\ i \in V^*,\ x$ is *incident to* $i\}$, $\lvert\Gamma^-(V^*)\rvert = d_G^-(V^*)$.
$\Gamma(V^*)$	$\Gamma^+(V^*) \cup \Gamma^-(V^*)$.
c_{ij}	cost of edge (i,j)
M	Matching
$A = (a_{ij})$	A square matrix with its elements denoted by a_{ij}.

REFERENCES

1. A.V. Aho, J.E. Hopcroft, and J.D. Ullman.
The Design and Analysis of Computer Algorithms.
Addison - Wesley, Reading, Mass., 1974.

2. S. Baase.
Computer Algorithms: Introduction to Design and Analysis.
Addison - Wesley, Reading, Mass., 1978.

3. M.L. Balinski, and R.E. Gomory.
A Primal Method for the Assignment and Transportation Problems.
Management Science, X(3): 578-593, 1964.

4. C. Berge.
Graphs and Hypergraphs.
North-Holland, London, 1973.

5. R.S. Ban, F. Glover, and D. Klingman.
The Alternating Basis Algorithm for Assignment Problem.
Mathematical Programming, XIII: 1-13, 1977.

6. M. Blum, R.W. Floyd, V.R. Pratt, R.L. Rivest, and R.E. Tarjan.
Time Bounds for Selection.
Journal of Computer and System Science VII(4): 448-461, 1972.

7. B. Bollobás.
Graph Theory: An Introductory Course.
Springer-Verlag, N.Y., 1979.

8. B. Bollobás.
External Graph Theory.
Academic Press, London, 1978.

9. B. Bollobás and S.E. Eldridge.
Maximal matching in graphs with given minimal and maximal degrees.
Math. Proc. Camb. Phil. Soc. LXXIX: 221-234, 1976.

10. J.A. Bondy, and U.S.R. Murty.
Graph Theory with Applications.
The Macmillan Press Ltd., London, 1976.

11. F. Bourgeois, and Jean-Claude Lassalle.
An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices.
CACM XIV(12): 802-806, 1971.

12. G. Carpaneto, and P. Toth.
Solution of the Assignment Problem.
ACM Transaction on Mathematical Software VI: 104-111, 1980.

13. G. Carpaneto and P. Toth.
Some New Branching and Bounding Criteria for the Asymmetric Travelling Salesman Problem.
Management Science, XXVI(7): 736-743, 1980.

14. N. Christofides.
Graph Theory: An Algorithmic Approach.
Academic Press, London, 1975.

15. N. Christofides.
The Travelling Salesman Problem.
In N. Christofides et al., Editors, *Combinatorial Optimization,* Wiley-Interscience Publication, John Wiley and Sons, London, 1979.

16. G. Danzig.
Discrete Value Extremum Problems.
Operation Research V: 266-277, 1957.

17. H.A. David.
Order Statistics.
John Wiley-Sons, Inc, N.Y. 1970.

18. W.E. Donath.
Algorithm and Average-value Bounds for Assignment Problems.
IBM J. Res Develop XIII: 380-386, 1969.

19. J. Edmords and E.L. Johnson.
Matching, Euler Tours and the Chinese Postman.
Mathematical Programming V: 88-124, 1973.

20. H.A. Eiselt and H.V. Frajer.
Operation Research Handbook-standard Algorithms and Methods with Examples.
W. de Gruyter, Berlin-N.Y., 1977.

21. P. Erdös and A. Renyi.
On Random Graph I.
Publications Mathematicae, VI: 280-297, 1959.

22. P. Erdös and A. Renyi.
On Random Matrices.
Publ. Math. Inst. Hung. Acad. Sci., 8A: 455-461, 1963.

23. P. Erdös and A. Renyi.
On the Existence of a factor of degree one of a connected Random Graph.
Acta Mathematica Academiae Scientiarum Hungaricae Tomus XVII(3-4): 359-368, 1966.

24. P. Erdös and J. Spencer.
Probabilistic Methods in Combinatorics.
Academic Press, N.Y., 1974.

25. W. Feller.
An Introduction to Probability Theory and its Applications, Volume I.
John Wiley and Sons, Inc., N.Y., 3rd Edition, 1968.

26. W. Feller.
An Introduction to Probability Theory and its Applications, Volume II.
John Wiley and Sons, Inc., N.Y., 2^{nd} Edition, 1971.

27. R.W. Floyd, and R.L. Rivest.
Expected Time Bounds for Selections.
CACM XVIII: 165-172. 1975.

28. L.R. Jr. Ford, and D.R. Fukerson.
Flows in Networks.
Princeton University Press, 1962.

29. M.R. Garey anb D.S. Johnson.
Computers and Intractability: A Guide to the Theory of NP-Completeness.
Freeman, San Francesco, 1979.

30. J.E. Graver and M.E. Watkins.
Combinatorics and Emphasis on the Theory of Graphs.
Springer-Verlag, N.Y., 1977.

31. M. Hall, Jr..
Combinatorial Theory.
Blarsdell publishing Company, Waltham, Mass., 1967.

32. F. Harary.
Graph Theory.
Addison-Wesley, Reading, Mass., 1972.

33. E. Horowitz and S. Sahni.
Fundamentals of Computer Algorithms.
Computer Science Press, Inc., Maryland, 1978.

34. T.C. Hu.
Integer Programming and Network Flows.
Addison-Wesley, Reading, Mass., 1970.

35. R. Jonker, G. Deleve, J.A. VanDer Velde, and A. Volgenant.
Rounding Symmetric Travelling Salesman Problems with an Asymmetric Assignment Problem.
Operation Research, XXVIII(3): 623-627, 1980.

36. R.M. Karp.
Probabilistic Analysis of Partitioning Algorithms for the Travelling Salesman Problem in the Plane.
Mathematical Operation Research II: 209-224, 1977.

37. R.M. Karp.
A Patching Algorithm for the Nonsymmetric Travelling Salesman Problem.
Memorandum No. UCB/ERL M78/2, Electronics Research Lab., College of Engineering, University of California, Berkeley, 1978.

38. R.M. Karp.
An Algorithm to Solve the m×n Assignment Problem in Expected Time O(m n log n).
Networks X: 143-152, 1980.

39. D.E. Knuth.
The Art of Computer Programming, Volume I: Fundamental Algorithms.
Addison-Wesley, Reading, Mass., 1968.

40. D.E. Knuth
The Art of Computer Programming, Volume III: Sorting and Searching.
Addison-Wesley, Reading, Mass., 1973.

41. J.M. Kurtzberg.
On Approximation Methods for the Assignment Problem.
JACM IX: 419-439, 1962.

42. E.L. Lawler.
Combinatorial Optimization: Networks and Matroids.
Holt, Rinehart and Winston, N.Y. 1976.

43. J.D.C. Little, K.G. Murty, D.W. Sweeney, and C. Karel.
An Algorithm for the Travelling Salesman Problem.
Operation Research XI: 972-989, 1963.

44. R.E. Machol and M. Wien.
A 'Hard' Assignment Problem.
Operation Research, XXIX: 190-192, 1976.

45. C. McMillan, Jr.
Mathematical Programming.
John Wiley and Sons, Inc., 2nd Edition, 1975.

46. D.S. Mitrinovic.
Analytic Inequalities.
Springer-Verlag, Herdelberg, 1970, p. 195.

47. J.I. Munro and M.S. Paterson.
Selection and sorting with limited storage.
IEEE FOCS, p. 253-258, 1978.

48. H.M. Salkin.
Integer Programming.
Addison-Wesley, Reading, Mass., 1975.

49. P.M. Spira.
On the Number of Comparison Necessary to Rank an Element.
In R.Rustin, Editor, Courant Computer Science Symposium 9. *Computational complexity,* Algorithmic Press, Inc., N.Y., p. 153-160, 1973.

50. N. Tomizawa.
On Some Technique Useful for the Solution of Transportation Problems.
Networks, I: 193-194, 1971.

51. J. Topping.
Errors of observation and their Treatment.
Chapman and Hall, London, 1972.

52. D.W. Walkup.
Polya sequence, binomial convolution and the union of Random sets.
J. Appl. Probability XIII:76-85, 1976.

53. D.W. Walkup.
Matching in Random Regular Bipartite Graph.
Discrete Mathematics XXXI(1): 59-64, 1980.

54. D.W. Walkup.
On the Expected Value of a Random Assignment Problem.
SIAM J. Computing VIII: 440-442, 1979.

55. B.W. Weide.
Random Graph and Graph Optimization Problems.
SIAM J. Computing IX(3): 552-557, 1980.

56. A.C. Yao and F.F. Yao.
On the Average Case Complexity of Selecting k^{th} Best.
IEEE 19^{th} Annual Symposium on FOCS. p. 280-299, 1978.