1994

Asymptotically Optimal Heuristics for Bottleneck and Capacity Optimization Problems

Wojciech Szpankowski
Purdue University, spa@cs.purdue.edu

Report Number:
94-069

ASYMPTOTICALLY OPTIMAL HEURISTICS FOR BOTTLENECK AND CAPACITY OPTIMIZATION PROBLEMS

Wojciech Szpankowski

Purdue University
Computer Sciences Department
West Lafayette, IN 47907

CSD-TR-94-069
October 1994

ASYMPTOTICALLY OPTIMAL HEURISTICS FOR BOTTLENECK AND CAPACITY OPTIMIZATION PROBLEMS

March 5, 1994

Wojciech Szpankowski*
Department of Computer Science
Purdue University
W. Lafayette, IN 47907
U.S.A.

Abstract

Our goal is to design simple and asymptotically optimal heuristic algorithms for a class of bottleneck and capacity optimization problems. Our approach is applied to a wide variety of bottleneck problems including vehicle routing problems, location problems, and communication network problems. In particular, we present simple and asymptotically optimal heuristic algorithms that solve the bottleneck assignment problem, the bottleneck spanning tree problem and the directed bottleneck traveling salesman problem in $\mathcal{O}(n^2)$ with high probability (our algorithm runs in $\mathcal{O}(n^{3+\varepsilon})$ for the undirected version of the bottleneck traveling salesman problem). In addition, we extend our results to the d-th best solution for some bottleneck optimization problems.

*This research was supported in part by the NSF grants CCR-8900305 (its extension CCR-9201078), and NCR-9206315 and INT-8912631, in part by AFOSR grant 90-0107, and by NATO Collaborative Grant 0057/89, and by Grant R01 LM05118 from the National Library of Medicine. This paper was revised while the author was visiting INRIA, Rocquencourt, France, and he wishes to thank INRIA (projects ALGO, MEVAL and REFLECS) for a generous support.

1. INTRODUCTION

In this paper, we investigate bottleneck and capacity optimization problems in a probabilistic framework. A *bottleneck problem* can be formulated as follows: for a given integer n minimize the objective function $Z(\alpha) = \max_{i \in S_n(\alpha)}\{w_i(\alpha)\}$ (and for *capacity problem* maximize $V(\alpha) = \min_{i \in S_n(\alpha)}\{w_i(\alpha)\}$) over a set B_n of all feasible solutions, where $S_n(\alpha)$ is the set of all objects belonging to a feasible solution $\alpha \in B_n$, and $w_i(\alpha)$ is the weight assigned to the i-th object. In our probabilistic framework, weights are drawn independently from a common distribution function $F(\cdot)$. We do not impose any special restriction on the class of distributions $F(\cdot)$ except continuity of $F(\cdot)$.

Our interest is twofold. First of all, we study asymptotic behaviors of the best solution $Z_{\min} = \min_{\alpha \in B_n} Z(\alpha)$ and the d-th best solution $Z_{(d)}$ of our bottleneck and capacity optimization problems, where $Z_{\min} = Z_{(1)} \leq Z_{(2)} \leq \cdots \leq Z_{\max}$. Secondly, using these probabilistic findings we build heuristic algorithms that asymptotically performed as good as the optimal algorithm. More precisely, the relative error between the value output by our heuristic algorithm and the optimal value $Z_{\min}$ of the optimization problem, tends to zero as the size of the problem increases. Needless to say, our heuristics are much cheaper (in terms of time and space complexities) than the optimal algorithm.

To motivate our study, we discuss three examples, namely the bottleneck traveling salesman problem, the bottleneck k-clique problem, and the bottleneck k-th center problem (cf. [16]). In the *bottleneck traveling salesman problem* (BTSP) a salesperson wishes to choose a route that minimizes the travel time on the longest day of traveling [14, 3]. For the *bottleneck k-clique problem* one wishes to partition n cities into k cliques such that the longest distance within a clique is minimized [18]. Finally, in the *bottleneck k-th center problem* one is asked to choose k cities among n such that the city furthest from a k center is as closed as possible [16].

The problems just mentioned belong to three general classes of optimization problems [16], namely *communication network problems, weighted center problems* and *vehicle routing problems.* The first class contains – besides the k-center problem – spanning tree problem, k clustering problem, k switching network problem, and so forth. In the second class, besides the k clique problem, one can also include the k supplier problem, weighted k center problem, etc. Finally, the last class contains the traveling salesman problem, k path vehicle routing, repeated city TSP, and so on (for more details see [16]). For solving each of these problems, we search for a subgraph of a given complete graph satisfying certain constrains such that the weight of the longest (shortest) edge including in the subgraph is minimized (maximized).

We establish in this paper two types of results. The first one is of a probabilistic nature, and deals with the typical behavior of the optimal solution $Z_{\min}$ and/or the dth best solution $Z_{(d)}$. In particular, for the bottleneck assignment problem (BAP) and the bottleneck traveling salesman problem (BTSP) we prove that $Z_{\min} \sim F^{-1}(\log n/n)$ *in probability* (pr.), where $F^{-1}(\cdot)$ is the inverse function of the distribution function $F(\cdot)$. This result should be interpreted as follows: for every $\varepsilon > 0$ the probability $\Pr\{|Z_{\min}/F^{-1}(\log n/n) - 1| > \varepsilon\}$ tends to zero as $n \to \infty$. Roughly speaking, this means that it is very unlikely that the optimal value $Z_{\min}$ differs from $F^{-1}(\log n/n)$ by more than $\varepsilon F^{-1}(\log n/n)$, whatever the ε is selected. In addition, for bounded d the d-th best solution $Z_{(d)}$ behaves asymptotically in a similar manner. Furthermore, for the bottleneck spanning tree problem we show that $Z_{\min} \sim F^{-1}(1/n^{1+1/n})$ (pr.), and in the case of the bottleneck k clique problem $Z_{\min} \sim F^{-1}(n^{-2/(k-1)})$ (pr.). Finally, in the bottleneck k center problem we have $Z_{\min} \sim F^{-1}(1 - \log n/n)$ (pr.).

Our second goal is to design a simple and efficient heuristic that with high probability outputs asymptotically the same solution as the optimal (but more expensive) algorithm. We repeatedly use a variation of the following scheme: After sorting all weights in an increasing order, we find such a number of edges (elements) m^* that a graph built from those m^* edges contains *almost surely* a given subgraphs (e.g., a Hamiltonian path, a matching, a clique, etc.).

Our heuristics with guaranteed performance compare favorably with all known deterministic solutions to these problems. In particular, we design algorithms that with high probability in $\mathcal{O}(n^2)$ steps solve such problems as the Bottleneck Assignment Problem (BAP), the Bottleneck Spanning Tree Problem (BSTP) and the Bottleneck (directed) Traveling Salesman Problem (BTSP) (the undirected version of BTSP we can solve in $\mathcal{O}(n^{3+\varepsilon})$ steps). These algorithms should be compared with best deterministic solutions obtained by Garfinkel and Gilbert [GaG78], and recently improved by Gabow and Tarjan [12] ($O(n^{2.5}\sqrt{\log n})$ for BAP and $O(n^2 \log^* n)$ for BSTP in complete graphs; see also Frieze [9] for similar solution to ours for BTSP. Our heuristic algorithms are easy to implement, and they run well in practice due to the fact that the constants in our complexity results are small (see Section 4 for some computer experiments). Finally, we indicate that there exist polynomial algorithms for the other two bottleneck problems, namely the k center problem and the k clique problem (cf. [16, 8])

Our approach to bottleneck optimization problems seems to be new, and has only something in common with the work of Weide [21] and Lueker [18] (cf. also Frieze [10, 11]). But in contrast to Weide's and Lueker's works our approach is *algorithmically constructive*, and – more importantly – we use some simple results from *order statistics.* It turns out that order statistics can be applied to solve some other optimization problems, and might lead to a unified approach to a large class of optimization problems (see [20] for some preliminary results). Some of our probabilistic findings are also simple consequences of known results on random graphs.

2. MAIN RESULTS

Our objective is to compute the optimal value $Z_{\min}$ defined as follows

$$Z_{\min} = \min_{\alpha \in B_n} \{ \max_{i \in S_n(\alpha)} w_i(\alpha) \} , \tag{2.1}$$

where B_n is the set of all feasible solutions, $S_n(\alpha)$ is the set of all objects belonging to the α-th feasible solution, and $w_i(\alpha)$ is the weight assigned to the i-th object. This problem is a bottleneck optimization problem since it minimizes the largest weight in a feasible solution. In another formulation, called capacity optimization problem, we ask to maximize the smallest weight in a feasible solution, that is,

$$V_{\max} = \max_{\alpha \in B_n} \{ \min_{i \in S_n(\alpha)} w_i(\alpha) \} .$$

To avoid repetitions we shall further reason in terms of the bottleneck problem.

We analyze all our optimization problems in a probabilistic framework that is summarized in the following two assumptions:[1]

(A) The cardinality $|B_n|$ of B_n is fixed and equal to L. The cardinality $|S_n(\alpha)|$ does *not* depend on $\alpha \in B_n$ and for all α it is equal to N, i.e., $|S_n(\alpha)| = N$.

(B) For all $\alpha \in B_n$ and $i \in S_n(\alpha)$ the weights $w_i(\alpha)$ are identically and independently distributed (i.i.d.) random variables with common distribution function $F(\cdot)$ which is a *strictly continuous increasing* function.

We restrict our attention to problems on graphs and matrices, so any object is either a vertex (edge) in a graph or an element of a matrix, and we denote by w_{ij} the weight assigned to the (i,j)-th edge in a graph or the (i,j)-th element of a matrix. We denote by $G_{n,m}$ a graph spanned on n vertices with m edges. By $\mathbf{W} = \{w_{ij}\}_{i,j=1}^{n}$ we define the matrix of weights. If possible, we shall reason in terms of the matrix $\mathbf{W}$. A graph $G_{n,m}$ can be

[1]These assumptions can be somewhat relaxed for the price of more subtle analysis. For example, some of our results hold even if in (B) we assume non-identical and weakly dependent weights.

directed or *undirected*, and respectively the matrix $\mathbf{W}$ can be *asymmetric* or *symmetric*. In the latter case, assumption (B) cannot hold as stated since $w_{ij} = w_{ji}$, but this usually causes only minor problems. To avoid this difficulty we modify the assumption (B) for the symmetric case such that independence is applied only to w_{ij} with $i \geq j$.

In the Introduction we have identified three classes of bottleneck optimization problems. Now, we present detailed definitions of three problems, one from each class, that are next investigated in our probabilistic framework. We formulate them for asymmetric (directed) matrices (graphs):

- *Bottleneck Assignment Problem* (BAP)

$$Z_{\min} = \min_{\sigma \in B_n} \{ \max_{1 \leq i \leq n} w_{i,\sigma(i)} \} , \tag{2.2a}$$

where $\sigma(\cdot)$ is a permutation of $\mathcal{M} = \{1, 2, ..., n\}$. For bipartite graphs the permutation $\sigma(\cdot)$ becomes a perfect matching. In the *Bottleneck Traveling Salesman Problem* (BTSP) the permutation $\sigma(\cdot)$ becomes a Hamiltonian cycle in a graph $G_{n,m}$. The cardinality L of the set of feasible solutions B_n is either $n!$ or (for complete graphs) $(n-1)!$ respectively.

- *Bottleneck k Clique Problem* (BkCP)

$$Z_{\min} = \min_{cl \in B_n} \{ \max_{i,j \in cl} w_{i,j} \} , \tag{2.2b}$$

where a clique cl is a complete subgraph spanned on k vertices in $G_{n,m}$. In terms of matrices, a clique cl can be defined as a set of k ordered pairs of indices from $\mathcal{M}$, namely $cl = \{(c_1, c_1), (c_2, c_2), ..., (c_k, c_k)\} \in \mathcal{M} \times \mathcal{M}$. Note also that the cardinality L of B_n is $L = C_n^k$ where $C_n^k = \binom{n}{k}$, while cardinality of $S_n(cl)$ is $N = 2C_k^2$.

- *Bottleneck k Location Problem* (BkLP)

$$Z_{\min} = \min_{\mathbf{c} \in B_n} \{ \max_{j \in \mathcal{M} - \mathbf{c}} w_{c_i,j} \} , \tag{2.2c}$$

where $\mathbf{c} = \{c_1, c_2, .., c_k\}$ and $c_i \in \mathcal{M}$. Note that $L = C_n^k$ and $N = n - k$.

In addition, we consider explicitly one more problem that belongs to the first category, but its importance justifies to pay some additional attention to it.

- *Bottleneck Spanning Tree* (BST)

$$Z_{\min} = \min_{sp \in B_n} \{ \max_{i,j \in sp} w_{i,j} \} , \tag{2.2d}$$

where sp is a spanning tree of a graph $G_{n,m}$. Naturally, for complete graphs $L = |B_n| = n^{n-2}$ and $N = n - 1$.

In fact, in many applications – most notably molecular biology – one is not only interested in the best possible solution, but also in the d-th best one. We denote the d-th best solution as $Z_{(d)}$, and thus $Z_{\min} = Z_{(1)} \le Z_{(2)} \le \ldots \le Z_{\max}$. We observe that $Z_{(d)}$ is the dth order statistic of the objective function $Z(\alpha)$.

As a motivating example for such a study, consider a problem in which weights are known only approximately (e.g., Gibbs energy in RNA, DNA or protein foldings [22]). Then, the best solution in terms of these approximate energy values does not necessary produce the optimal structure in terms of the true energy values. However, if the problem is not too sensitive to small perturbation in weights, then one may expect that the second, the third, or the tenth best solution is the one that minimizes the total true (i.e., unperturbed) total free energy. In fact, even when all weights are known exactly, we still might want to produce, say, the first hundredth best solutions so, say a biologist, can decide which ones bear some biological meaning. Having this in mind, we also present some results for the dth best solution $Z_{(d)}$.

Luckily enough, for most of the bottleneck optimization problems we can present a fairly general algorithm which is presented below (cf. also [16]).

`Algorithm BOTTLE`

begin
  build a min-heap from n^2 weights $\{w_{ij}\}_{ij=1}^{n}$
  $i = 0$
  repeat
    begin
    $i = i + 1$
    remove the ith smallest weight from the min-heap
    add $w_{(i)}$ element to the structure built so far
    build a partial solution β (not necessary a feasible solution)
    end
  until β becomes a feasible solution α
output α
end

Clearly, the algorithm BOTTLE always gives a correct answer. Its complexity depends on the min-heap construction algorithm (cf. second line of the BOTTLE), and the number of iterations in the loop **repeat-until**. In each iteration one must check whether a feasible solution exists or not (this might be even NP-complete; e.g., Hamiltonian path for BTSP).

A min-heap of n^2 weights can be built in $O(n^2)$ steps in the worst-case (cf. [1]). Let m^* be the number of iterations in the loop **repeat-until.** In the worst case, $m^* \sim n^2$, but *typically* ("on average") it takes much less time to complete this loop. Such a typical value of m^* can be interpreted as the number of iterations needed to produce a feasible solution *almost surely* (a.s.) or *with high probability* (whp). In other words, m^* can be seen as the number of elements that one needs to select from the matrix $\mathbf{W}$ in order to construct almost surely or with high probability a feasible solution (e.g., a subgraph such as clique, Hamiltonian path, etc.).

Clearly, removing m^* elements from the min-heap costs at most $O(m^* \log n)$. Let $\mathcal{C}_{test}$ be the time required to perform the feasibility test in BOTTLE. Then, the overall complexity $\mathcal{C}_n$ of the algorithm is $O(\max\{n^2, m^* \log n, m^* \mathcal{C}_{test}\})$.

The time-complexity $\mathcal{C}_n$ can be interpreted in the worst-case scenario or in a probabilistic framework. In the former case, $m^* = n^2$ and $\mathcal{C}_{test}$ is the worst case complexity of the feasibility test. In the later case, both m^* and $\mathcal{C}_{test}$ can be much smaller, and then naturally the overall complexity $\mathcal{C}_n$ must be understood probabilistically. At least two interpretations are possible: Namely, we write $C_n = \mathcal{O}(f(n))$ *in probability* (pr.) or *with high probability* (whp) if there exists a constant A such that for any $\varepsilon > 0$

$$\lim_{n\to\infty} \Pr\{\mathcal{C}_n < (1+\varepsilon)Af(n)\} = 1 \ .$$

We also say $\mathcal{C}_n = \mathcal{O}(f(n)$ *almost surely* (a.s.) if the above condition is replaced by a stronger one, namely

$$\lim_{n\to\infty} \Pr\{\max_{k\geq n}\{\mathcal{C}_k/(Af(k)) - 1\} < \varepsilon\} = 1 \ .$$

To avoid any confusion, we use $O(\cdot)$ to denote the worst-case (deterministic) complexity, while we write $\mathcal{O}(\cdot)$ for either in probability or almost sure complexity.

Our main goal is to study simple heuristics for BOTTLE algorithm that are efficient in probabilistic sense. Clearly, its complexity depends on the probabilistic behavior of m^* and $\mathcal{C}_{test}$. In addition, it is sensitive to the heuristic itself. In this paper, we consider a class of heuristic algorithms with the following general paradigm:

`Algorithm HEURISTIC`

% Set m^* to be the smallest m that assures with high probability the existence of a feasible solution %

begin

build a min-heap

remove first m^* best weights from the min-heap, i.e., $w_{(1)} \leq w_{(2)} \leq \dots \leq w_{(m^*)}$

apply feasibility test to G_{n,m^*}

output α or NOT FOUND

end

In practice, instead of $m = m^*$ we run the algorithm for a couple of iterations from $m = m^* - O(1)$ to $m = m^* + O(1)$. If one is interested in the dth best solutions, then the feasibility test must be run at most d times. In any case, the complexity of this heuristic is $O(\max\{n^2, m^* \log n,\ C_{test}\})$ or $\mathcal{O}(\max\{n^2, m^* \log n,\ C_{test}\})$ which is an improvement over the exact algorithm BOTTLE.

How good is HEURISTIC? We investigate the value Z_{heu} output by the heuristic, and compare it to the optimal value $Z_{\min}$ (or the dth best solution $Z_{(d)}$) of the original optimization problem. We measure the relative error $e_n = (Z_{\min} - Z_{heu})/Z_{\min}$, and for asymptotically optimal heuristics one expects that $e_n \to 0$ as $n \to \infty$.

The next four theorems present asymptotic behaviors of $Z_{\min}$ for the four bottleneck problems discussed above under our two basic assumptions (A) and (B). Proofs are delayed till the next section. Some additional algorithmic consequences of these findings are discussed below.

Theorem 1. `Bottleneck and Capacity Assignment Problems`

(i) For symmetric and asymmetric BAP the d-th best solution $Z_{(d)}$ converges in probability to $F^{-1}(\log n/n)$ as $n \to \infty$ provided $d = o(\log n/\log\log n)$, that is,

$$\lim_{n\to\infty} \frac{Z_{(d)}}{F^{-1}(\log n/n)} = 1 \qquad (pr.) \tag{2.3a}$$

where $F^{-1}(\cdot)$ denotes the inverse function of the distribution $F(\cdot)$. For the bottleneck capacity assignment problem the following holds

$$\lim_{n\to\infty} \frac{V_{(d)}}{F^{-1}(1-\log n/n)} = 1 \qquad (pr.) \tag{2.3b}$$

Our HEURISTIC runs in $\mathcal{O}(n^2)$ steps and outputs (asymptotically) optimal value (as shown in (2.3a) and (2.3b)) with very high probability.

(ii) For the bottleneck and capacity traveling salesman problem (2.3a) and (2.3b) hold too. Our HEURISTIC runs in $\mathcal{O}(n^2)$ steps for the directed version of BSTP, and in $\mathcal{O}(n^{3+\varepsilon})$ for the undirected version of BTSP. ∎

Theorem 2. `Bottleneck Spanning Tree Problem`
Asymptotically the optimal solution for BSTP becomes

$$\lim_{n\to\infty} \frac{Z_{\min}}{F^{-1}(n^{-1-1/n})} = 1 \qquad (pr.) \tag{2.4}$$

Our HEURISTIC runs in $\mathcal{O}(n^2)$ *steps and gives the optimal value (2.4) (whp).*

Theorem 3. `Bottleneck k Clique Problem`
For large n*, and* k *bounded with respect to* n*, the optimal solution for the* k *clique problem satisfies*

$$\lim_{n\to\infty} \frac{Z_{\min}}{F^{-1}(n^{-2/(k-1)+\varepsilon})} = 1 \qquad (pr.) \tag{2.5}$$

where $\varepsilon > 0$*. There exists a polynomial version of our algorithm HEURISTIC.* ■

Theorem 4. `Bottleneck k Center Problem`
For large n*, and* k *bounded with respect to* n*, the optimal solution for the* k *center problem becomes*

$$\lim_{n\to\infty} \frac{Z_{\min}}{F^{-1}(n^{-1/(n-k)+\varepsilon})} = 1 \qquad (pr.) \tag{2.6}$$

where $\varepsilon > 0$*. There exists a polynomial version of our algorithm HEURISTIC.* ■

We now can comment on specific implementations of HEURISTIC. We start with the asymmetric BAP. Our analysis from Section 3 (see also [7]) shows that $m^* = n(\log n + \omega_n)$ ($\omega_n \to \infty$) elements selected from matrix $\mathbf{W}$ almost surely constitute a permutation. To construct such a permutation we transform the problem to one on bipartite graphs G_{n,m^*}. Applying $O(n^{1/2}m^*)$ Micali-Vazirani algorithm [19] for finding the maximum matching in such a general graph, the algorithm HEURISTIC becomes:

`Algorithm ASYMMETRIC BAP`

begin

select $m^* = n(\log n + \omega_n)$ smallest weights

apply Micali-Vazirani algorithm to G_{n,m^*}

end.

For symmetric BAP one needs to set $m^* = \frac{n}{2}(\log n + \omega_n)$. Naturally, these algorithms run in $\mathcal{O}(n^2)$ steps since the feasibility test costs only $\mathcal{O}(n^{3/2}\log n)$.

For the bottleneck traveling salesman problem (BTSP) the challenge is how to find efficiently a Hamiltonian path. We shall use here $O(n^{1.5})$ (Las Vegas) algorithm of Frieze [9] to solve the directed version of the problem, and $\mathcal{O}(n^{3+\varepsilon})$ algorithm to solve the undirected version of the problem (cf. also [4, 11]). From our analysis in Section 3 (cf. [4, 9]) it will be clear that $m^* = n(\log n + \log\log n + \omega_n)$ edges is enough to assure *almost surely* a Hamiltonian path in a directed graph, and $m^* = \frac{n}{2}(\log n + \log\log n + \omega_n)$ is the "magic" number for an undirected graph [9]. Then, HEURISTIC looks as follows:

```
Algorithm DIRECTED BTSP
```

begin

select $m^* = n(\log n + \log\log n + \omega_n)$ smallest weights

apply Frieze's algorithm DHAM to G_{n,m^*}

end.

The bottleneck spanning tree problem is easier to tackle. From Erdös and Rényi [6] one concludes that $m^* = n^{1-1/(n-1)+\varepsilon}$ (whp). The dominating part in HEURESTIC is the heap construction, and thus the algorithm runs in $\mathcal{O}(n^2)$ steps.

Finally, we show that $m^* = n^{2(1-1/(k-1))}$ (cf. [4, 18]) for the k clique problem, and $m^* = n^{2-1/n}$ for the k center problem. This implies, unfortunately, that almost all n^2 weights have to be inspected, and saving in time is very limited. If k is bounded in n (but might be large) there is, of course, a polynomial algorithm to build a feasible solution. Moreover, even in the case of unbounded k, Fellows and Langston [8] proved that there exists a polynomial algorithm for constructing feasible solutions for these problems.

In passing, we note that capacity problems require only minor changes. In fact, in BOTTLE and HEURISTIC algorithms one needs to build a max-heap instead of a min-heap. In particular, in Theorem 1 (cf. (2.3b)) we pointed out how to obtain our main result for the capacity assignment problem. The rest is left to the interested reader.

3. ANALYSIS THROUGH ORDER STATISTICS

In this section we prove Theorems 1 to 4. The most interesting is to obtain the asymptotics for the d-th best solution $Z_{(d)}$ of Theorem 1 (cf. Section 3.1). The other results are simple extensions of some known threshold results on random graphs. For the reader convenience, however, we briefly discuss them in Section 3.2.

The bottleneck optimization problem (2.1), is *ranking-dependent*, that is, the optimal solution depends only on the rank of the weights $w_i(\alpha)$ but not on specific values of $w_i(\alpha)$. More formally, the bottleneck optimization problem possesses the following property:

$$f(Z_{\min}) = \min_{\alpha\in B_n}\{\max_{i\in S_n(\alpha)} f(w_i(\alpha))\} \tag{3.1}$$

for every increasing function $f(\cdot)$. Since $F(\cdot)$ is strictly increasing function by assumption (B), to analyze $Z_{\min}$ we may choose any suitable distribution (say uniform), and then transform the results by $F^{-1}(\cdot)$ back to the original problem. We shall proceed below along these lines.

For simplicity, we shall only reason in terms of the matrix optimization problem (2.1) (e.g., BAP). Clearly, the BOTTLE algorithm finds the optimal solution after inspecting m^* weights, where m^* is the smallest number of elements of $\mathbf{W}$ that assures the existence of a feasible solution. Thus, $Z_{\min} = w_{(m^*)}$ where $w_{(m^*)}$ is the m^*th order statistics of n^2 weights from $\mathbf{W}$.

In our probabilistic framework, we first assume that the weights W_i (capital letters denote random variables) for $1 \le i \le n^2$ are uniformly distributed i.i.d. random variables, and we denote by M_n^* the minimum number of elements that must be inspected in order to construct *almost surely* a feasible solution. Let $M_n^* \sim m_n^*$ (a.s.) where m_n^* is a function of n. We prove below that the M_n^*-th order statistic of W_i is $W_{(M_n^*)} \sim m_n^*/n^2$ with high probability (provided $m_n^* \to \infty$ as $n \to \infty$), that is, for any $\varepsilon > 0$ we have $\lim_{n\to\infty} \Pr\{|W_{M_n^*} - m_n^*/n^2| > \varepsilon \cdot m_n^*/n^2\} = 0$. This is a simple consequence of the following lemma (cf. see also [13]).

Lemma 5. *Let $U_{(1)} \le \ldots \le U_{(m)}$ be order statistics of m uniformly distributed i.i.d. random variables. Let also R_m (rank) be a random variable such that $R_m/r \to 1$ (pr.) for some $r \le m$. Then for any $\varepsilon > 0$*

$$\lim_{m\to\infty} \Pr\{(1-\varepsilon)EU_{(r)} \le U_{(R_m)} \le (1+\varepsilon)EU_{(r)}\} = 1 \tag{3.2}$$

provided $r \to \infty$ as $m \to \infty$. In other words, $U_{(R_m)}/EU_{(r)} \to 1$ (pr.) .

Proof. We have (cf. [13])

$$EU_{(r)} = \frac{r}{m+1} \qquad ; \qquad var\ U_{(r)} = \frac{r(m-r+1)}{(m+1)^2(m+2)} .$$

Thus, by Chebyshev's inequality we conclude that

$$\Pr\{|U_{(r)}/EU_{(r)} - 1| \ge \varepsilon\} \le \frac{1}{\varepsilon^2 r} .$$

Now we proceed as follows. From the assumption concerning R_m we can choose δ as small as we want such that $\Pr\{(1-\delta)r \le R_m \le (1+\delta)r\} = 1 - o(1)$ as $m \to \infty$. Then,

$$\begin{aligned}\Pr\{|U_{(R_m)}/EU_{(r)} - 1| \ge \varepsilon\} &= \sum_{k \le r(1-\delta)} \Pr\{|U_{(R_m)}/EU_{(r)} - 1| \ge \varepsilon,\ R_m = k\} \\ &+ \sum_{k \ge r(1+\delta)} \Pr\{|U_{(R_m)}/EU_{(r)} - 1| \ge \varepsilon,\ R_m = k\}\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{r(1-\delta)\le k\le r(1+\delta)} \Pr\{|U_{(R_m)}/EU_{(r)} - 1| \ge \varepsilon,\ R_m = k\} \\
&= 2\cdot o(1) + \frac{2\delta}{\varepsilon^2(1-\delta)} \,.
\end{aligned}
$$

The last expression can be made as small as we wish due to arbitrariness of $0 < \delta \ll 1$. This completes the proof. ■

Corollary. *Under the hypotheses of Lemma 5 and assumption (B), we have for any $\varepsilon > 0$*

$$
\lim_{n\to\infty} \Pr\{(1-\varepsilon)F^{-1}(m_n^*/n^2) \le Z_{\min} \le (1+\varepsilon)F^{-1}(m_n^*/n^2)\} = 1 \tag{3.3}
$$

provided $m_n^* \to \infty$.

Proof. Observe that by Lemma 5 for uniformly distributed weights $Z_{\min} = W_{M_n^*} \sim m_n^*/n^2$ (pr.) provided $m_n^* \to \infty$. Now use (3.1) to prove (3.3) for any distribution function $F(\cdot)$ that satisfies (B). ■

From the above, in particular from (3.3), one concludes that for proving our results concerning the optimal values $Z_{\min}$ we need only to evaluate m_n^* (a.s.). In the case of d-th best solution a little more intricate analysis is necessary. We give more details in the next section, when the bottleneck assignment problem is discussed.

3.1 Bottleneck Assignment and Traveling Salesman Problems

Let us start with the optimal value $Z_{\min}$. We estimate m_n^* (a.s.). Consider first the asymmetric BAP problem. In this case, a feasible solution is a permutation $\sigma(\cdot)$ (cf. (2.2a)), that is, in a feasible solution no two elements share a column and/or a row. To compute m_n^* (a.s.) we select randomly elements from a $n \times n$ matrix $\mathbf{W}$, and stop when for the first time every column and every row contains at least one element.

For BSTP we should construct a Hamiltonian cycle. It turns out, as proved by Frieze [9], that the same condition as for BAP (i.e., every vertex has degree at least one) guarantees that a directed graph with weights from $\mathbf{W}$ possesses almost surely a Hamiltonian cycle. However, for the symmetric BAP and undirected BSTP we have a little different situation. It is proved [4] that an undirected (random) graph is Hamiltonian (a.s.) when the minimum degree of this graph is at least two. In terms of the weight matrix $\mathbf{W}$, this means that one should select at least two elements in every row (column) of $\mathbf{W}$.

We first reduce the evaluation of m_n^* to an urn-and-ball problem. In such a model, n balls are thrown randomly and independently into n urns. To treat uniformly the above two cases (i.e., symmetric and asymmetric) we define m_n^* as the first time until every urn has at least $K \geq 1$ balls. How to compute such a quantity? Holst [15] proved that $m_n^* = n \cdot (\log n + (K-1)\log\log n + \omega_n)$ (a.s.) where $\omega_n \to \infty$ as $n \to \infty$.

It turns out that a simple technique called *poissonization* can be used to establish Holst's result (cf. [2, 15]). The poissonization approach is useful in some other problems, hence we present here heuristic arguments of Aldous [2]. The key idea is to assume that the arrival times of balls into urns is a Poisson process with parameter 1. We denote by $POIS(\lambda)$ a Poisson process with parameter λ. Then, every box receives a Poisson process with parameter $1/n$, and by superexponentiality property of a Poisson distribution we have

$$\Pr\{a \text{ } box \text{ } contains \text{ } at \text{ } least \text{ } K \text{ } balls \text{ } at \text{ } time \text{ } t\} \approx e^{-t/n}(t/n)^{K-1}/(K-1)! \, .$$

But, by poissonization, the input processes to urns are independent, so the number of urns with at least K balls is distributed as $POIS(ne^{-t/n}(t/n)^{K-1}/(K-1)!)$. Note that the event $\{M_n^* \leq t\}$ is equivalent to the event that *the number of boxes with at least K balls is equal to zero at time t*. Then, immediately

$$\Pr\{M_n^* \leq t\} \approx \exp\left(-ne^{-t/n}(t/n)^{K-1}/(K-1)!\right) \, . \tag{3.4a}$$

This further implies that

$$\Pr\{M_n^* \leq n \cdot (\log n + (K-1)\log\log n + \omega_n)\} \sim \exp(-e^{-\omega_n}) \, , \tag{3.4b}$$

and the latter probability tends to one whenever $\omega_n \to \infty$. Hence, we just shown that $M_n^* = m_n^* = n \cdot (\log n + (K-1)\log\log n + \omega_n)$ (a.s.). (The difficulty of this analysis – not shown here – is to prove depoissonization, that is, to conclude the final result regarding the original model from the Poisson one; for more details see [2, 15]).

Part of Theorem 1 regarding the optimal value $Z_{\min}$ follows immediately from the above, Lemma 5, and (3.3). However, for the d-th best solution $Z_{(d)}$ we need a little more elaborate approach. We prove Theorem 1 by establishing an upper bound and a lower bound.

Let $M_{n,d}^*$ be the minimum number of randomly selected elements from $\mathbf{W}$ that assures with high probability existence of at most d permutations (feasible solutions) in BAP problem. It is easy to see that $M_{n,d}^*$ is almost surely equal to our previous estimate m_n^* (cf. (3.4)) with $K = d$. Thus, for $d = o(\log n/\log\log n)$ we have $M_{n,d}^* = m_n^* \sim n\log n$ (a.s.).

Now, we are ready to establish an upper bound for $Z_{(d)}$. We use slightly modified BOTTLE to find first d best solutions. Repeating our previous arguments, one easily sees that for uniformly distributed weights

$$Z_{(d)} \leq W_{M_{n,d}^*} \sim \frac{m_n^*}{n^2} \sim \frac{\log n}{n} \, . \tag{3.5}$$

To obtain a lower bound, one may observe that $Z_{(d)} \geq Z_{\min} \sim m_n^*/n^2$, but we choose to take a little more longer proof to illustrate applications of order statistics to the performance evaluation of combinatorial optimization problems. Moreover, this longer "tour" can be further refined to extend some of our results, but we do not elaborate more on this in the current paper. We first prove the following simple result.

Lemma 6 *The dth best solution $Z_{(d)}$ can be bounded from the below as follows*

$$Z_{(d)} \geq \max_{1\leq j\leq n}\{\min_{1\leq i\leq n} W_{ij}\} . \tag{3.6}$$

Proof. Observe that $Z_{(d)} \geq Z_{(1)} = Z_{\min}$. Then, for every permutation $\sigma(\cdot)$ we have

$$\max\{w_{i,\sigma(i)}\} \geq \max_{1\leq j\leq n}\{\min_{1\leq i\leq n} w_{ij}\} .$$

which suffices to prove the lemma. To prove the above, let i^* and j^* be the row and the column of $\mathbf{W}$ for which the right-hand side (RHS) of the above inequality is satisfied. Consider any permutation $\sigma(\cdot)$, and let i' be such an index that $\sigma(i') = j^*$. Clearly, $1 \leq i' \leq n$. If $i^* = i'$, then $w_{i,\sigma(i)} \geq w_{i^*,j^*}$ for all $1 \leq i \leq n$. If $i' \neq i^*$, then by definition $\max\{w_{i,\sigma(i)}\} \geq w_{i',j^*} \geq w_{i^*,j^*}$ Again, the inequality holds, and this completes the proof. ■

The next lemma deals with order statistics, and before we proceed, let us denote by $W_{(d),(n-d+1)}$ an element of $\mathbf{W}$ selected according to the following procedure: We first find the d-th smallest weight in, say j-th column. Denote such a value as $W_{(d),j}$. Next, we find the $(n-d+1)$-st largest value in the sequence $W_{(d),1}, W_{(d),2}, \ldots, W_{(d),n}$. Observe that by Lemma 6 we know that $Z_{(d)} \geq W_{(1),(n)}$.

To discover asymptotic behavior of $W_{(d),(n-d+1)}$ we use the following lemma that is of its own interest and finds many other applications in combinatorial optimization (cf. [2, 17, 13, 20]).

Lemma 7. `Order Statistics.`

Let X_1, X_2 ,..., X_n be identically exchangeable (i.e., any joint distribution depends only on the number of variables involved, not indexes of the variables) nonnegative random variables with common distribution function $F(x)$, and let $G(x) = 1 - F(x)$ be defined on half real line $(0, \infty)$. Denote $F_r(x) = \Pr\{X_1 < x, ..., X_r < x\}$ and $G_r(x) = \Pr\{X_1 > x, ..., X_r > x\}$ for any $1 \leq r \leq n$. Let also $Z_{(r)}$ be the r-th order statistic of the sequence X_1, X_2 ,..., X_n.

(i) *If for every $c > 1$*

$$\lim_{x\to\infty} \frac{G_{n-r+1}(cx)}{G_{n-r+1}(x)} = 0 \tag{3.7}$$

(i.e., $G_{n-r+1}(x)$ has exponential tail), then

$$Z_{(r)} \leq a_n^{(r)} \qquad (pr.) \tag{3.8}$$

where $a_n^{(r)}$ is the smallest solution of the following equation

$$\binom{n}{n-r+1} G_{n-r+1}(a_n^{(r)}) = 1 \ . \tag{3.9}$$

(ii) *If X_1, X_2 ,..., X_n are independently distributed, then*

$$\Pr\{Z_{(r)} > x\} = \sum_{i=0}^{r-1} \binom{n}{i} F^i(x)[1-F(x)]^{n-i} \ . \tag{3.10}$$

If, in addition, (3.7) holds and $n-r$ is bounded with respect to n, then

$$Z_{(r)} \sim a_n^{(r)} \qquad (pr.) \tag{3.11}$$

Proof. By definition of the r-th order statistic we have

$$\Pr\{Z_{(r)} > x\} = \Pr\{ \bigcup_{j_1,\dots,j_{n-r+1}} \bigcap_{i=1}^{n-r+1} (X_{j_i} > x)\} \tag{3.12}$$

for all distinct $j_1, ..., j_{n-r+1} \in \{1, ..., n\}$. For (i) we apply Boole's inequality to the above and set $x = (1+\varepsilon)a_n^{(r)}$. Then, using (3.7) and (3.9)

$$\begin{aligned} \Pr\{Z_{(r)} > (1+\varepsilon)a_n^{(r)}\} &\leq \binom{n}{n-r+1} G_{n-r+1}((1+\varepsilon)a_n^{(r)}) \\ &= \binom{n}{n-r+1} G_{n-r+1}(a_n^{(r)}) o(1) = o(1) \ , \end{aligned}$$

and this proves (3.8). Then, (3.10) is a simple consequence of (3.12) and the independence assumption. Finally, (3.11) follows from (3.10) after some simple algebra that is left to the reader (cf. [13] p. 247). ∎

Now, we are ready to prove a lower bound for $Z_{(d)}$. We use (3.6) and Lemma 7, however, to show the power of Lemma 7 we determine the asymptotic behavior of $W_{(d),(n-d+1)}$ for any d. We set later $d = 1$ to complete the proof of Theorem 1 for $Z_{(d)}$.

From (3.10) we compute the distribution function for $W_{(d),j}$ as the d-th order statistic of $W_{1,j}, ..., W_{n,j}$. Then, using (3.9) we estimate the $n-d+1$-st order statistic for $W_{(d),1}, ..., W_{(d),n}$. But, due to ranking-dependent property of bottleneck problems we are free to select a distribution of the weights. Since we plan to apply Lemma 7 we need a distribution satisfying (3.7). The best seems to be an exponential distribution, so we assume

$F(x) = 1 - e^{-x}$. Then, by (3.11) $Z_{(d)} \sim a_n$ (for simplicity we drop the upper index in the notation of a_n) where a_n solves asymptotically the following equation (cf. (3.9))

$$\binom{n}{d} (G_d(a_n))^d = 1 \ . \tag{3.13a}$$

where (cf. (3.10))

$$G_d(x) = e^{-nx} \sum_{i=0}^{d-1} \binom{n}{i} (e^x - 1)^i \ . \tag{3.13b}$$

But, for bounded d we can reduce (3.13a) to

$$\frac{n}{\sqrt[d]{d!}} G_d(a_n) = 1 \ , \tag{3.14}$$

which possesses the following asymptotic solution for bounded d

$$a_n \sim \frac{\log(n\beta \log^{d-1}(n\beta))}{n} \tag{3.15}$$

where $\beta = \sqrt[d]{d!}/(d-1)!$. Indeed, the above can be shown by inspection. Using $e^x - 1 \sim x$ for $x \to 0$, the LHS of (3.14) becomes

$$\frac{n}{\sqrt[d]{d!}} G_d(a_n) = \sum_{i=1}^{d-1} \frac{(d-1)!}{i!} \frac{(1 + \frac{\log\log^{(d-1)} n\beta}{\log n\beta})^i}{\log^{d-1-i} n\beta} \to 1$$

where the last implication holds, for example, for d bounded. Set now $d = 1$ to get $a_n \sim \log n/n$. To complete the proof one needs only to translate this result to the uniform distribution. But, $1 - e^{-a_n} \sim a_n$ for $a_n \to 0$. This observation completes the proof of the lower bound, and hence Theorem 1.

3.2 Remaining Proofs

For the remaining of the bottleneck problems (cf. Theorem 2 to 4) we only provide proofs for the optimal value $Z_{\min}$. Extension to d-th best solution is possible, and details of appropriate statement formulations and proofs are left for the reader.

By Lemma 5 and its Corollary, proofs of Theorem 2 to 4 reduce to finding the value of m_n^* which can be obtained as a by-product of some threshold results in random graphs. In the case of spanning tree and k clique problem, we immediately obtain from [6] (cf.Theorem 1, Corollary 1), and [18] (cf. Theorem 2)

$$m_n^* = n^{1-1/(n-1)} \qquad \text{and} \qquad m_n^* = n^{2-2/(k-1)+\varepsilon}$$

respectively. This and (3.3) complete the proof of Theorem 2 and 3.

To prove Theorem 4 for the k center problem we first note that a feasible solution in this case (cf. (2.2c)) consists of all $k(n-k)$ elements of the weight matrix $\mathbf{W}$. This simply represents all edges connecting the k centers with all other vertices. A simple combinatorial enumeration, as the one in Erdös and Rényi [6] implies that

$$\Pr\{a\ feasible\ solution\ exists\ in\ a\ matrix\ with\ m\ selected\ elements\} \approx O\left(\frac{m^{k(n-k)}}{n^{2k(n-k)-k}}\right),$$

hence $M_n^* \sim n^{2-1/(n-k)+\varepsilon}$, as needed for the proof of Theorem 4.

4. COMPUTER EXPERIMENTS AND CONCLUDING REMARKS

In order to visualize and verify our theoretical results we have programmed our algorithms BOTTLE and HEURISTIC for the BAP problem. In BOTTLE we used an improved *Hungarian Method* to check whether a perfect matching exists. In both algorithms BOTTLE and HEURISTIC we build a heap to sort efficiently (in $O(n^2)$ steps) weights w_{ij}. Finally, we implemented in HEURISTIC a simple and effective (time-complexity of $O(n^2)$) subalgorithm to inspect whether the selected weights cover the whole matrix $\mathbf{W}$, that is, whether there is at least one weight in every column and every row.

We have used three different distributions, namely normal distribution $N(0,1)$, gamma distribution $gamma(\lambda, \beta)$, and beta distribution $beta(\alpha_1, \alpha_2)$. For each distribution we evaluated the optimal value using our exact algorithm BOTTLE and compare it with the theoretical optimal value obtained from Theorem 1.

From Table 1 we observe a very good accuracy of our theoretical results even for small size of the problem ($100 \le n \le 500$). This shows a good quality of our heuristic algorithms. From the table we note that the running time for BOTTLE is approximately $121n^3$, while for HUERISTIC is only $15n^2$ which is significant time saving even for moderate values of n.

We should also point out that BOTTLE algorithm can be implemented in a more efficient manner (e.g., by adding not just one weight but a group of weights in order to construct faster a feasible solution). We did not implemented any modifications of BOTTLE since we prefered to compare our HEURISTIC to a "pure" optimal BOTTLE algorithm.

Finally, there are several directions in which one can pursue this research. First of all, it might be interesting to extend this analysis to other bottleneck optimization problems. Even more interesting is to see whether order statistic approach can be used for solving other optimization problems such as the linear assignment problem, traveling salesman problem, location problem, and so forth. Some preliminary results in this direction are reported in [20].

Table 1: Comparison of simulation and theoretical results.

DISTRIBUTION	SIZE	VALUE			TIME	
		THEORY	OPTIMAL	HEURISTIC	BOTTLE	HEURISTIC
Normal (0,1)	100	-1.500000	-1.62355	-1.48766	77	8
	200	-1.770000	-1.860940	-1.87524	736	58
	300	-1.920000	-1.957682	-1.99665	3285	117
	400	-2.020000	-2.098472	-2.05880	7758	288
	500	-2.100000	-2.161564	-2.16474	15002	566
Exponential (1)	100	0.047146	0.053723	0.05328	79	34
	200	0.026849	0.033738	0.02819	741	43
	300	0.019196	0.026014	0.02601	3527	135
	400	0.015092	0.016736	0.01559	6495	298
	500	0.012507	0.015816	0.01582	18070	411
Uniform (0,1)	100	0.046052	0.043567	0.04347	69	7
	200	0.026492	0.032597	0.03373	869	41
	300	0.019013	0.026200	0.02593	3456	139
	400	0.014979	0.018658	0.01866	7340	303
	500	0.012429	0.015084	0.01405	16085	481

ACKNOWLEDGMENT

It is a pleasure to acknowledge the excellent job done by Maryjane Scharenberg in preparing and running computer experiments. The author is also in debt to referees for very careful reading of the manuscript and pointing out several misprints and inaccuracies in the first draft of this paper.

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] Aldous, D., *Probability Approximations via the Poisson Clumping Heuristic*, Springer Verlag, New York 1989.

[3] D. Angluin and L. Valiant, Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings, *J. Computer and System Sciences*, 18, 155–193, 1979.

[4] B. Bollobas, *Random Graphs*, Academic Press, London 1985.

[5] Y. Chow and H. Teicher, *Probability Theory*, Springer-Verlag, New York 1978.

[6] P. Erdös and A. Rényi, On the Evolution of Random Graphs, *Publ. Math. Inst. Hungar. Acad. Sci.*, 5, 17-61, 1960.

[7] P. Erdös and A. Rényi, On Random Matrices, *Publ. Math. Inst. Hungar. Acad. Sci.*, 8, 455-461, 1964.

[8] M. Fellows and M. Langston, Nonconstructive Tools for Proving Polynomial-Time Decidability, *J. of the ACM*, 35, 727-739, 1988.

[9] A. Frieze, An Algorithm for Finding Hamiltonian Cycles in Random Directed Graphs, *J. Algorithms*, 9, 181-204, 1988.

[10] A. Frieze, On Matching and Hamilton Cycles in Random Graphs, in *Surveys in Combinatorics, 1989* (ed. J. Siemons), 84-114, Bambridge University Press, 1989.

[11] A. Frieze, Probabilistic Analysis of Graph Algorithms, Carnegie Mellon University, TR-88-42, 1989.

[12] H. Gabow and R. Tarjan, Algorithms for Two Bottleneck Optimization Problems, *J. Algorithms*, 411-417, 1988.

[13] J. Galambos, *The Asymptotic Theory of Extreme Order Statistics*, R.E. Krieger Publication Company, Malabar, 1987.

[14] R. Garfinkel and K. Gilbert, The Bottleneck Traveling Salesman Problem: Algorithms and Probabilistic Analysis, *J. ACM*, 25, 435–448, 1978.

[15] L. Holst, On Birthday, Collector's, Occupancy and other Classical Urn Problems, *International Statistical Review*, 54, 15-27, 1986.

[16] D. Hochbaum and D. Shmoys, A Unified Approach to Approximate Algorithms for Bottleneck Problems, *J. of the ACM*, 33, 533-550, 1986.

[17] Lai, T. and Robbins, H., A Class of Dependent Random Variables and Their Maxima, *Z. Wahrscheinhich.,* 42, pp. 89–111, 1978.

[18] Lueker, G., Optimization Problems on Graphs With Independent Random Edge Weights, *SIAM J. Computing*, 10, pp. 338–351, 1981.

[19] S. Micali and Vazirani, V., An $O(\sqrt{|V|} \cdot E)$ Algorithm for Finding Maximum Matching in General Graphs, *Proc. 21-st Annual IEEE Symp. on Found. of Comput. Sci.*, 17-27, 1980.

[20] Szpankowski, W., Combinatorial Optimization Through Order Statistics, *Proc. Second International Symposium on Algorithms*, Taipei, Taiwan, Lecture Notes in Computer Science, 557 (eds. W.L. Hsu and R.C.T. Lee), pp. 208-217, Springer-Verlag 1991.

[21] Weide, B., Random Graphs and Graph Optimization Problems, *SIAM J. Computing*, 9, pp. 552–557, 1980.

[22] M. Zuker, Computer Prediction of RNA Structure, *Methods in Enzymology*, 180, 262-288, 1989.