1990

Another Unified Approach to Some Bottleneck and Capacity Optimization Problems

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

Report Number:
90-1022

ANOTHER UNIFIED APPROACH TO SOME BOTTLENECK AND CAPACITY OPTIMIZATION PROBLEMS

Wojciech Szpankowski

CSD-TR-1022
September 1990

ANOTHER UNIFIED APPROACH TO SOME BOTTLENECK AND CAPACITY OPTIMIZATION PROBLEMS

September 17, 1990

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

Abstract

The purpose of this paper is twofold. Namely, to present a probabilistic analysis of a class of bottleneck (capacity) optimization problems, and to design simple and *efficient* heuristic algorithms guaranteed to be asymptotically optimal. Our unified approach is applied to a wide variety of bottleneck problems including vehicle routing problems, location problems, and communication network problems. In particular, we present a 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 $O(n^2)$ time-complexity steps (our algorithm runs in $O(n^{3+\epsilon})$ for the undirected version of the bottleneck traveling salesman problem). We also discuss polynomial heuristic algorithms for the bottleneck k-clique problem and the bottleneck k-location problem. We prove – using our probabilistic analysis – that these algorithms with high probability (whp) produce the optimal solution. Furthermore, we extend our results to the d-th best solution for some bottleneck optimization problems.

*This research was supported by AFOSR Grant 90-0107, and in part by the NSF Grant CCR-8900305, and by Grant R01 LM05118 from the National Library of Medicine.

1. INTRODUCTION

In this paper we investigate a general 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 a minor requirement of continuity for $F(\cdot)$. Our interest is twofold. First of all, we study the asymptotic behavior 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 $Z_{\min} = Z_{(1)} \leq Z_{(2)} \leq \ldots \leq Z_{\max}$. 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 of our objective function $Z(\alpha)$ evaluated for a heuristic solution α and the optimal value $Z_{\min}$ (found in our probabilistic analysis) tends to zero as the size of the problem becomes larger and larger. 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 some examples (cf. [HoS86]) that show the range of applications for our methodology. 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 [GaG78, AnV79]. 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 [LUK81]. Finally, the last example deals with the *bottleneck k-th center problem* that belongs to the location theory. In this case, one is asked to choose k cities among n such that the city furthest from a k center is as closed as possible [HoS86].

The problems just mentioned belong to three general classes of optimization problems, namely *communication network problems*, *weighted center problems* and *vehicle routing problems* [HoS86]. 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 [HoS86]). For each of these problems, we search for a subgraph of the complete graph satisfying certain constrains such

that the weight of the longest (shortest) edge including in the subgraph is minimized (maximized); such problems are – as discussed above – called *bottleneck* (*capacity*) problems. It is sometimes more convenient to transform these problems into similar problems on matrices with random weights. If possible, we shall reason in terms of such matrices.

We establish in this paper two types of results. The first one is of probabilistic nature, and deal with the typical behavior of the optimal solution $Z_{\min}$ and/or the d-th 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 ϵ the probability $\Pr\{|Z_{\min}/F^{-1}(\log n/n) - 1| > \epsilon\}$ 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 ϵ, whatever the ϵ is selected. Moreover, for any bounded d the d-th best solution $Z_{(d)}$ behaves asymptotically in a similar manner. 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.). All of these results are derived in a uniform manner, and more importantly they have a simple algorithmic interpretation. Namely, in the course of obtaining them we repeatedly use the following algorithm. After sorting all weights in an increasing order, we find such a number of edges (elements) m^* that *almost surely* the (random) graph built from these m^* edges contains a given subgraphs (e.g., a hamiltonian path, a matching, a clique, etc.). It is now a question of choosing an appropriate algorithm that constructs this subgraph, and to show that the algorithm outputs the optimal value most of the time, but the latter issue was already investigated in our probabilistic part of the paper.

Our heuristics with guaranteed performance compare favorable with all known deterministic solutions to these problems. In particular, we construct $O(n^2)$ algorithms that solves exactly with very high probability 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 $O(n^{3+\epsilon}$ steps). These algorithms beat the best deterministic solutions obtained by Garfinkel and Gilbert [GaG78], and recently improved by Gabow and Tarjan [GaT88] ($O(n^{2.5}\sqrt{\log n})$ for BAP and $O(n^2 \log^* n)$ for BSTP in complete graphs; see also Frieze [FRI88] 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 have also polynomial algorithms for

the other two bottleneck problems, namely the k center problem and the k clique problem (cf. [HoS86, FeL88]).

We view this paper as a contribution to the probabilistic analysis of algorithms that – not unexpectedly – leads to some new algorithmic insights. Our unified approach to bottleneck optimization problems seems to be new, and has only something in common with the work of Weide [WEI80] and Lueker [LUE81] (cf. also Frieze [FRI89a, FRI89b]). But in contrast to Weide's and Lueker's works our approach is *algorithmically constructive*, and – more importantly – we use some simple but powerful results from the *order statistics*. It turns out that application of the order statistics to optimization problems is not restricted to bottleneck and capacity problems, and might lead to a unified approach to a large class of optimization problems (see [SZP89] for some preliminary results).

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 lightest weight in a feasible solution, that is, the formulation (2.1) becomes

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

To avoid repetition we shall further reason in terms of the bottleneck problem. We analyze it in a probabilistic framework that is summarized in the following two assumptions:

(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 $F(\cdot)$ that is *strictly continuous (increasing)* function.

We restrict our attention to problems on graphs and matrices, so any object is either a vertex (edge) 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 *directed* or *undirected*, and respectively the matrix $\mathbf{W}$ can be *asymmetric* or *symmetric*. In the latter case, assumption (B) cannot hold as it is stated since $w_{ij} = w_{ji}$, but this –in most case - 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 – each from one class – which are next rigorously 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 $G_{n,m}$. Of course, the cardinality L of the set of feasible solutions B_n is either $n!$ or $(n-1)!$ 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 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 $C_n^k = \binom{n}{k}$. On the other hand, the cardinality N of the set $S_n(cl)$ is C_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)

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

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

In fact, in many applications – most notably molecular biology and pattern recognition – one is not only interested in the best possible solution, but also in the d-th best solution, that is, the d-th order statistic of the objective function Z. We denote the d-th best solution as $Z_{(d)}$, and naturally $Z_{\min} = Z_{(1)} \leq Z_{(2)} \leq \ldots \leq Z_{\max}$. 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 [ZUK89]). 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., undisturbed) total free energy. In fact, even when all weights are exactly known, we still might want to produce, say, the first hundredth best solutions so, say, a biologist can decide which ones bear some biological meanings. Having this in mind, we also present some results for the d-th best solution $Z_{(d)}$.

Luckly enough, for most of the bottleneck optimization problems we can present fairly general algorithm. This algorithm works as follows (cf. [HoS86]).

`Algorithm BOTTLE`

begin
sort weights such that $w_{(1)} \leq w_{(2)} \leq \ldots \leq w_{(n^2)}$
$i = 0$
repeat
begin
$i = i + 1$
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

Of course, this algorithm always produces a correct answer. The question is how expensive it is. It is easy to notice that the cost (time complexity) of the algorithm depends on the sorting procedure (cf. second line of the BOTTLE), and the number of iterations of the loop **repeat-until**. In each iteration we must check whether a feasible solution exists or not (this might be even NP-complete; e.g., hamiltonian path for BTSP). The sorting part is bounded from the below by $O(n^2)$ (since every element out of n^2 has to be touched at least once). The rest depends on an optimization problem. Let the number of the loop iterations be denoted by m^*. In the worst case $m^* \sim n^2$, but a *typical* time necessary to complete this loop is much smaller. Such a typical value of m^* can be interpreted as the number of iterations needed *almost surely* to produce a feasible solution. In other words, m^* can be seen as the number of elements that one needs to (randomly) select from a matrix W to construct *almost surely* a feasible solution (e.g., a subgraph such as clique, hamiltonian path, etc.). The second factor that determines the complexity is the the feasibility test. Let C_{test} be the time required to perform the test. Then, the ultimate complexity of the algorithm is $O(\max\{n^2, m^* C_{test}\})$. In passing we note that the complexity C_{test} may not necessary be the worst-time complexity; especially if m^* is interpreted in a probabilistic manner.

In our probabilistic framework it is natural to consider algorithms from a typical view point. Then, the complexity should be investigated from this view point too. This will lead to several mutations of our basic algorithm which we further refer as a heuristic. A general scheme for such a heuristic algorithm is presented below (for a change we phrase it in terms of graphs).

`Algorithm HEURISTIC`

begin

sort weights such that $w_{(1)} \leq w_{(2)} \leq \ldots \leq w_{(n(n-1)/2)}$

Set $m = m^*$

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

output α or NOT FOUND

end

In the above m^* should assure with probability one the existence of a feasible solution. In practice, instead of $m = m^*$ we run the algorithm for a couple of iterations from $m = m^* - O(1)$ to $m = m^* + O(1)$. The algorithm needs some simple modifications to output the d-th best solution, namely we have to run the feasibility test at most d times.

A natural question is how good is this heuristic, that is, how fast is the algorithm, *and* how close is the value Z_{heu} found by the heuristic to the optimal value $Z_{\min}$. The latter problem is of prime importance for any approximate algorithm. In fact, a quality of an heuristic can be measure by the relative error $e_n = (Z_{\min} - Z_{heu})/Z_{\min}$, and one accepts an approximate algorithm if e_n tends to zero (fast enough!) as $n \to \infty$ in some probability sense. This condition can be verified once we know the optimal value $Z_{\min}$ which is of its own interest. The four theorems below – our main (probabilistic) results – present the limiting value of $Z_{\min}$ for the four bottleneck problems discussed above. We assume that our two basic assumptions (A) and (B) always hold. Proofs are delayed to 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 tends to infinity provided d is bounded with respect to 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 to the distribution $F(\cdot)$. For the bottleneck capacity assignment problem the following hold

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

Our approximate algorithm HEURISTIC runs in $O(n^2)$ steps and outputs (asymptotically) the optimal value (cf. (2.3)) with very high probability (whp).

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

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

where $\epsilon > 0$. 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

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

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

Finally, we comment on specific algorithms that implement our approximate algorithm HEURISTIC. We start with the asymmetric BAP. Our analysis from Section 3 (see also [ErR64]) will indicate that selecting $m^* = n(\log n + \omega_n)$ ($\omega_n \to \infty$) elements from a matrix assures with probability one the existence of a permutation. To construct such a permutation we transform the problem to another one on bipartite graphs. Namely, as easy to see a permutation in a matrix can be viewed as a perfect matching in a bipartite graph G_{n,m^*} with m^* vertices. Then, applying $O(n^{1/2}m)$ Micali-Vazirani algorithm [MiV80] for finding the maximum matching in such a general graph, the algorithm HEURISTIC becomes

`Algorithm ASYMMETRIC BAP`

Set $m^* = n(\log n + \omega_n)$
begin
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 $O(n^2)$ steps since feasibility tests costs only $O(n^{3/2}\log n)$. 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 [FRI88] to solve the directed version of the problem, and $O(n^{3+\epsilon})$ algorithm to the solve the undirected version of the problem (cf. [BOL85, FRI89b]). From our analysis in Section 3 (cf. [BOL85, FRI88]) it will be clear that $m^* = n(\log n + \log\log n + \omega_n)$ edges is enough to have *almost surely* a hamiltonian path in a directed graph , and $m^* = n/2(\log n + \log\log n + \omega_n)$ is the "magic" number for an undirected graph [FRI88]. The HEURISTIC algorithm modifies to the following

Algorithm DIRECTED BTSP

Set $m^* = n(\log n + \log\log n + \omega_n)$

begin

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

end.

The bottleneck spanning tree problem is easier to tackle. From Erdös and Rényi [ErR60] one concludes that $m^* = n^{1-1/(n-1)+\epsilon}$. Hence, the dominating factor in the complexity issue becomes the sorting part, and hence the algorithm HEURISTIC runs in $O(n^2)$ steps.

Finally, we show that $m^* = n^{2(1-1/(k-1))}$ [BOL85, LUK81] 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 unboundness of k Fellows and Langston [FeL88] 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 the BOTTLE and the HEURISTIC algorithms one needs to sort in a decreasing order instead of increasing order. In particular, in Theorem 1 (cf. (2.3b)) we pointed out how to construct our main result for the capacity assignment problem. The rest is left to an interesting reader.

3. ANALYSIS THROUGH ORDER STATISTICS

In this section we prove our main results stated in Theorems 1 to 4. As we shall see, the most difficult to handle is the asymptotics for the d-th best solution $Z_{(d)}$. We have one such a result for BAP and CAP (cf. Theorem 1), and we treat this case with a special attention. Fortunately, a probabilistic analysis of $Z_{\min}$ can be handled in a uniform manner. Basic points of such an analysis are presented below.

The bottleneck optimization problem (2.1) possesses very special feature, namely the one that we call *ranking-dependent*. By this we mean that the optimal solution depends only on the rank of the weights $w_i(\alpha)$ and not on the concrete value of $w_i(\alpha)$. In other words, if one transforms all weights according to any increasing function $f(\cdot)$, then the rank of the optimal solution remains the same. More formally, the bottleneck optimization problem is ranking-dependent since the following holds

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

for every increasing function $f(\cdot)$. As a simple consequence of this, one notes that probabilistic behavior of $Z_{\min}$ under our assumption (B) (i.e., distribution $F(\cdot)$ is a strictly increasing function) can be proven for one selected distribution, say uniform $U(0,1)$, and then transform to any other distribution $F(\cdot)$ by the transform $F^{-1}(\cdot)$. Indeed, this simply follows from the fact that $X = F^{-1}(U)$ where $F(\cdot)$ is the distribution function of the random variable X, and U represents uniformly distributed random variable on the interval $[0,1]$.

To prove our main results, we proceed as follows. Let $w_{(1)} \leq w_{(2)} \leq ... \leq w_{(n^2)}$ denote all n^2 weights ordered in an increasing sequence; for simplicity we shall only reason in terms of the matrix optimization problem (2.1)(e.g., BAP). Then, according to the algorithm BOTTLE the optimal solution to a bottleneck optimization problem is found after inspecting m^* elements (edges). Naturally, $Z_{\min} = w_{(m^*)}$. In our probabilistic framework, we assume that the weights W_i (capital letters denote random variables) for $1 \leq i \leq n^2$ are uniformly distributed i.i.d. (independently, identically distributed) random variables, and we denote by M_n^* the number of elements necessary to inspect in order to construct *almost surely* a feasible solution. Note that the M_n^*-th order statistic of, say, n^2 uniformly distributed random variables W_i is $W_{(M_n^*)} = M_n^*/(n^2+1) + o(1)$ (pr.), that is, for any ϵ the following holds $\lim_{n\to\infty} \Pr\{|W_{M_n^*} - M_n^*/(n^2+1)| > \epsilon\} = 0$. This is a simple consequence of the following two elementary facts [GAL87]:

- The r-th order statistic $U_{(r)}$ of m uniformly distributed i.i.d. random variables satisfies

$$EU_{(r)} = \frac{r}{m+1} \quad ; \quad varU_{(r)} = \frac{r(m-r+1)}{(m+1)^2(m+2)} , \tag{3.2}$$

  hence by Chebyshev's inequality $U_{(r)} \to EU_{(r)}$ (pr.) as $m \to \infty$. In fact, by Borel-Cantelli lemma the convergence *in probability* may be replaced by *almost sure* convergence provided $r = o(m)$

- The above property holds even if r is replaced by a random variable R_m such that $R_m/r \to 1$ (a.s.)

Then, our claim follows from the Chebyshev's inequality, and therefore $Z_{\min} \sim W_{(M_n^*)}$ (pr.). In summary, we just proved that as $n \to \infty$

$$Z_{\min} \to \frac{M_n^*}{n^2} \quad (pr.) , \tag{3.3a}$$

for uniformly distributed weights W_i. This and (3.1) further imply that for any other distribution function $F(\cdot)$

$$Z_{\min} \to F^{-1}(M_n^*/n^2) \quad (pr.) \tag{3.3b}$$

provided $F^{-1}(\cdot)$ is strictly continuous (cf. [ChT78, p.68]).

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

3.1 Bottleneck Assignment and Traveling Salesman Problems

Let us start with the optimal value $Z_{\min}$, and therefore we concentrate on computing M_n^* (a.s.). Consider first the asymmetric BAP problem. In this case a feasible solution becomes 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 a selected element. It turns out, as proved by Frieze [FRI88], that the same condition guarantees that a directed graph with the weight matrix $\mathbf{W}$ possesses almost surely a hamiltonian cycle. However, for the symmetric BAP and undirected BSTP we have a little different situation. It is proved [BOL85] 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 can be read as a requirement that a random selection of elements from $\mathbf{W}$ stops when for the first time every row (column) contains at least two chosen elements.

We reduce the evaluation of M_n^* to an urn-and-ball problem. In such a model balls are thrown randomly and independently to n urns. We may ask questions like what is the number of balls needed until every box has at least one ball; at least two balls, etc. 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 balls. How to compute such a quantity? It turns out that a simple technique called *poissonization* may easily answer this question (cf. [ALD89, HOL86]). Holst [HOL86] proved that $M_n^* = n \cdot (\log n + (K-1)\log\log n + \omega_n)$ (a.s.) where $\omega_n \to \infty$ as $n \to \infty$.

The poissonization idea is useful in some other problems considered in this paper. Therefore, we present here heuristic arguments that lead to the evaluation of M_n^* (cf. [ALD89]). The key idea is to assume that the arrival time 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 any Poisson process we have

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

But poissonization makes boxes independent, hence the number of boxes with at least K

balls is distributed as a $POIS(ne^{-t/n}(t/n)^{K-1}/(K-1)!)$. 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

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

This further implies that

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

and the latter probability tends to one whenever $\omega_n \rightarrow \infty$. Hence, we just proved that $M_n^* = n \cdot (\log n + (K-1)\log\log n + \omega_n)$ (a.s.). (The difficulty of this analysis – not shown here – is to prove depoissonization, that is, how to conclude the final result regarding the original model from the Poisson model; for more details see [ALD89, JaS89].)

Part of Theorem 1 regarding the optimal value $Z_{\min}$ follows immediately from the above and (3.3). In particular, to verify (3.3a) we use (3.4) to show the convergence in probability.[1] However, to prove the results for the d-th best solution $Z_{(d)}$ we need a little more elaborate methodology. We prove our result (cf. Theorem 1) by showing an upper bound and a lower bound. The upper bound repeats our arguments from the prove of $Z_{\min}$ to show that for uniformly distributed weights the following holds [BOL85]

$$Z_{(d)} \leq \frac{M_n^* + d}{n^2 + 1} \sim \frac{M_n^*}{n^2} \sim \frac{\log n}{n} \ , \tag{3.5}$$

and the last implication holds as long as $d/M_n^* \rightarrow 0$ for $n \rightarrow \infty$. Therefore, in the rest of this section we deal with the lower bound for the d-th best solution.

Let us illustrate our approach to the lower bound by considering again the optimal value. We shall reason in terms of the asymmetric BAP problem. The following is easy to show

$$Z_{\min} \geq \max_{1 \leq j \leq n} \{ \min_{1 \leq i \leq n} W_{ij} \} \ . \tag{3.6a}$$

The idea behind this lower bound is as follows: first we take minimum weight in every column, and then maximum from these selected weights. Now, we try to generalize this bound for the d-th best solution $Z_{(d)}$. Fix j, and find the d-th smallest weight in the j-th column. Denote such an value as $W_{(d),j}$. Next find the $(n-d+1)$-st largest value in the sequence $W_{(d),1}, W_{(d),2}, ..., W_{(d),n}$. Call such a value as $W_{(d),(n-d+1)}$. Then, the following is an easy generalization of (3.6a)

$$Z_{(d)} \geq W_{(d),(n-d+1)} \ . \tag{3.6b}$$

[1] This analysis can be easily extended, as pointed out above, to the the *almost sure* convergence.

This bound should be understood in the *stochastic inequality* sense [ChT87], that is, it holds for every realization.

The problem of finding the matching lower bound for $Z_{(d)}$ reduces to a tight bound on $W_{(d),(n-d+1)}$. For this we need to study some features of order statistics. Note that $\{W_{(d),j}\}_{j=1}^{n}$ is a sequence of i.i.d. random variables. Moreover, the d-th order statistic $W_{(d),.}$ comes also from n i.i.d. weights $W_{i,j}$. We need the following lemma, that is of its own interest and finds many other applications in combinatorial optimization (cf. [ALD89, SZP89]). A simpler version of this lemma is known – but not very well known – so we present here a sketch of the proof. For some more information regarding order statistics see Aldous [ALD89] and Galambos [GAL87] (see also [LaR78]).

Lemma 5. 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 the indexes of the variables [GAL87]) nonnegative random variables with common distribution function $F(x)$, and let $G(x) = 1 - F(x)$ be defined on whole 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 $1 \le r \le n$. 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

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

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) Independent Case. *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

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

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

$$\Pr\{Z_{(r)} > (1+\epsilon)a_n^{(r)}\} \leq \binom{n}{n-r+1} G_{n-r+1}((1+\epsilon)a_n^{(r)}) = \binom{n}{n-r+1} G_{n-r+1}(a_n^{(r)})o(1) = o(1)\,,$$

and this proves (3.8). Moreover, (3.10) is a simple consequence of (3.12) and the independence assumption. Finally, (3.11) follows from (3.10) after some simple algebra (cf. [GAL87 p. 247]). ■

Now we ready to prove the lower bound for $Z_{(d)}$. We use (3.6) and Lemma 5. From (3.10) we compute the distribution function for $W_{(d),j}$ as the d-th order statistic of $W_{1,j}, \ldots, W_{n,j}$. Then, using (3.9) we estimate the $n-d+1$-st order statistic for $W_{(d),1}, \ldots, W_{(d),n}$. 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 5 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

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

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) with a_n from (3.15) 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 follows from the fact that d is bounded. This also implies that $a_n \sim \log n/n$. To complete the proof we need 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.

As discussed above, in many applications the sensitivity analysis of an optimization problem is of prime interest. We can answer some sensitivity questions using the technique developed so far. Consider the following problem. We have shown that the maximum over all smallest values of every column has asymptotically the same value as the optimal value $Z_{\min}$ (cf. (3.6a)). A natural question to ask is whether the optimal value is asymptotically preserved when the smallest value in every column is replaced by the d-th smallest value. More formally, as above define for every column j the d smallest value as $W_{(d),j}$ Then, consider $Z^{(d)}_{\min} = \max_{1 \le j \le n} W_{(d),j}$. How large can d be to assure that $Z_{\min} \sim Z^{(d)}_{\min}$? One may expect that if this holds for d large enough, then there exits a simple asymptotically optimal randomized algorithm that constructs a feasible solution for BAP. By Lemma 5, with the exponential distribution of weights, we see that $Z^{(d)}_{\min} \sim a_n$, where a_n solves $nG_d(a_n) = 1$. But this equation is almost the one we consider in (3.14). So, elementary modifications lead to

$$Z^{(d)}_{\min} \sim \frac{\log(n/(d-1)!\log^{d-1}(n/(d-1)!))}{n} . \tag{3.16}$$

Of course, if d is bounded with respect to n, then $Z_{\min} \sim Z^{(d)}_{\min} \sim \log n/n$. A more sophisticated analysis shows that this asymptotic relationship holds also for $d = o(\log n/\log\log n)$. Note that it is *not* enough to assure *almost sure* construction of a feasible solution (i.e., a permutation) for the BAP problem. However, this finding can be used to save some (running) time for algorithms solving BAP problem.

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.

As explained earlier (cf. Eqs. (3.3a)-(3.3b)), proofs of Theorem 2 to 4 reduce to finding the value of M^*_n, that is, the minimum number of elements necessary to select from $\mathbf{W}$ in order to assure the existence (a.s.) a feasible solution. In the case of spanning tree and k clique problem we immediately obtain from [ErR60] and [BOL85, LUK81]

$$M^*_n = n^{1-1/(n-1)} \qquad \text{and} \qquad M^*_n = n^{2-2/(k-1)+\epsilon}$$

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 [ErR60] implies that

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

hence $M_n^* = n^{2-1/(n-k)+\epsilon}$, as needed for the proof of Theorem 4. As a curiosity, on may ask whether a modified selection process in which elements are *returned* to $\mathbf{W}$, will harm significantly the optimal solution (but we can save some memory in this case!). More formally, we find the minimum number of elements selected randomly *with returns* from a $n \times n$ matrix that assures the existence of a feasible solution in the k center problem.

We derive here this result using the poissonization technique discussed above. We reformulate the problem in terms of urn-and-ball problem, and we argue in the language of Aldous (cf. [ALD89]). How many balls is needed to fill k urns with at least n balls? In the first step we replace the throwning process by a Poisson process with rate 1. Then, every urn receives a Poisson process with rate $1/n$, that is, $POIS(1/n)$. Call T_i the number of balls necessary to put at least n balls into the i-th urn. Naturally, due to the superexponentiality property of a Poisson process

$$\Pr\{T_i > t\} \approx e^{-t/n}(t/n)^n/n! .$$

Then, the required value M_n^* can be computed as $M_n^* = \max\{T_1, ..., T_k\}$. By Lemma 5 we know that $M_n^* \sim a_n$ where a_n solves the equation $k\Pr\{T_i > a_n\} = 1$. Using the Stirling's formula, one needs to solve asymptotically the following

$$k\sqrt{n/2\pi}(a_n n)^n e^{-a_n/n} = 1 .$$

After some algebra we find

$$M_n^* \sim n^{2-\sqrt{1/2n+\log k/(n\log n)}} .$$

We note that this differs from the previous solution by a factor of $O(n^{\sqrt{1/n}})$. In particular, our optimal solution would become $F^{-1}(1-\log n/\sqrt{2n})$.

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

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

Hungerian Method to check whether a perfect matching exists or not. In both algorithms BOTTLE and HEURISTIC we build 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 (note that this assure that m^* is properly selected).

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 the table one may immediately note very good accuracy of our theoretical results even for small size of the problem ($100 \leq n \leq 500$). This implies good convergence rates for our theoretical results. In additional, this suggests a good quality of our heuristic algorithms,

and it was confirmed by computer runs. 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. Furthermore, we point out that our algorithm BOTTLE does not differ significantly from an optimal algorithm for this problem, and our another computer experiments confirm this observation (hence our heuristic is even more valuable).

Finally, there are several directions 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 the order statistic approach can be used for other optimization problems such as linear assignment problem, traveling salesman problem, location problem, and so forth. Some preliminary results in this direction are reported in [SZP89].

ACKNOWLEDGMENT

It is a pleasure to acknowledge the excellent job done by Maryjane Scharenberg in preparing and running computer experiments.

References

[AnV79] D. Angluin and L. Valiant, Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings, *J. Computer and System Sciences,* 18, 155–193, 1979.

[BOL85] B. Bollobas, *Random Graphs,* Academic Press, London 1985.

[ChT78] Y. Chow and H. Teicher, *Probability Theory*, Springer-Verlag, New York 1978.

[ErR60] P. Erdös and A. Rényi, On the Evolution of Random Graphs, *Publ. Math. Inst. Hungar. Acad. Sci.*, 5, 17-61, 1960.

[ErR64] P. Erdös and A. Rényi, On Random Matrices, *Publ. Math. Inst. Hungar. Acad. Sci.*, 8, 455-461, 1964.

[FeL88] M. Fellows and M. Langston, Nonconstructive Tools for Proving Polynomial-Time Decidability, *J. of the ACM*, 35, 727-739, 1988.

[FRI88] A. Frieze, An Algorithm for Finding Hamiltonian Cycles in Random Directed Graphs, *J. Algorithms*, 9, 181-204, 1988.

[FRI89a] A. Frieze, On Matching and Hamilton Cycles in Random Graphs, in *Surveys in Combinatorics, 1989* (ed. J. Siemons), 84-114, Bambridge University Press, 1989.

[FRI89b] A. Frieze, Probabilistic Analysis of Graph Algorithms, Carnegie Mellon University, TR-88-42, 1989.

[GaT88] H. Gabow and R. Tarjan, Algorithms for Two Bottleneck Optimization Problems, *J. Algorithms*, 411-417, 1988.

[GAL87] J. Galambos, *The Asymptotic Theory of Extreme Order Statistics,* R.E. Krieger Publication Company, Malabar, 1987.

[GaG78] R. Garfinkel and K. Gilbert, The Bottleneck Traveling Salesman Problem: Algorithms and Probabilistic Analysis, *J. ACM,* 25, 435–448, 1978.

[HOL86] L. Holst, On Birthday, Collector's, Occupancy and other Classical Urn Problems, *International Statistical Review*, 54, 15-27, 1986.

[HoS86] D. Hochbaum and D. Shmoys, A Unified Approch to Approximate Algorithms for Bottleneck Problems, *J. of the ACM*, 33, 533-550, 1986.

[JaS89] Jacquet, P. and Szpankowski, W., Ultimate Characterizations of the Burst Response of an Interval Searching Algorithm: A study of a Functional Equation, *SIAM J. on Computing*, 18, 777-791, 1989.

[LaR78] Lai, T. and Robbins, H., A Class of Dependent Random Variables and Their Maxima, *Z. Wahrscheinhich.,* 42, pp. 89–111, 1978.

[LUK81] Lueker, G., Optimization Problems on Graphs With Independent Random Edge Weights, *SIAM J. Computing,* 10, pp. 338–351, 1981.

[MiV80] 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.

[SZP89] Szpankowski, Z., (Probability) Optimal Solution to Some Problems Not Only on Graphs, Purdue University CSD-TR-780, 1988; revised CSD-TR-872, 1989.

[WEI80] Weide, B., Random Graphs and Graph Optimization Problems, *SIAM J. Computing,* 9, pp. 552–557, 1980.

[ZUK89] M. Zuker, Computer Prediction of RNA Structure, *Methods in Enzymology*, 180, 262-288, 1989.