Statistical Analysis of Algorithms: A Case Study of Market-Clearing Mechanisms in the Power Industry*

Chris Barrett[†] Doug Cook[‡] Gregory Hicks[§] Vance Faber[¶]
Achla Marathe[†] Madhav Marathe[†] Aravind Srinivasan[||]
Yoram Sussmann[**] Heidi Thornquist[††]

November 12, 2002

Abstract

We carry out a detailed empirical analysis of simple heuristics and provable algorithms for bilateral contract-satisfaction problems. Such problems arise due to the proposed deregulation of the electric utility industry in the USA. Given a network and a (multi)set of pairs of vertices (contracts) with associated demands, the goal is to find the maximum number of simultaneously satisfiable contracts. Four different

---

*A preliminary version of this work appeared as the paper "Experimental Analysis of Algorithms for Bilateral-Contract Clearing Mechanisms Arising in Deregulated Power Industry" in *Proc. Workshop on Algorithm Engineering*, 2001. Work by Doug Cook, Vance Faber, Gregory Hicks, Yoram Sussmann and Heidi Thornquist was done when they were visiting Los Alamos National Laboratory.

[†]National Infrastructure Simulation and Analysis Center and CCS3, P.O. Box 1663, MS M997, Los Alamos National Laboratory, Los Alamos NM 87545. Email: `barrett,achla,marathe@lanl.gov`.

[‡]Department of Engineering, Colorado School of Mines, Golden CO 80401.

[§]Department of Mathematics, North Carolina State University, Raleigh. Email: `gphicks@unity.ncsu.edu`.

[¶]Mapping Sciences Inc. Email: `vance@mappingscience.com`.

[||]Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. Parts of this work were done while at Lucent Technologies, Murray Hill, NJ 07974-0636, and while at the National University of Singapore. Email: `srin@cs.umd.edu`.

[**]Department of Computer Science, State University of West Georgia. Email: `yoram@westga.edu`.

[††]Department of Computational and Applied Mathematics, Rice University, Houston, TX. Email: `heidi@caam.rice.edu`.

algorithms (three heuristics and a provable approximation algorithm) are considered and their performance is studied empirically in fairly realistic settings using rigorous statistical analysis. For this purpose, we use an approximate electrical transmission network in the state of Colorado. Our experiments are based on the statistical technique *Analysis of Variance* (ANOVA), and show that the three heuristics outperform a theoretically better algorithm. We also test the algorithms on four types of *scenarios* that are likely to occur in a deregulated marketplace. Our results show that the networks that are adequate in a regulated marketplace might be inadequate for satisfying all the bilateral contracts in a deregulated industry.

1 Introduction

The U.S. electric utility industry is undergoing major structural changes in an effort to make it more competitive [21, 17, 19, 11]. One major consequence of the deregulation will be to decouple the controllers of the network from the power producers, making it difficult to regulate the levels of power on the network; consumers as well as producers will eventually be able to negotiate prices to buy and sell electricity [18]. In practice, deregulation is complicated by the facts that all power companies will have to share the same power network in the short term, with the network's capacity being just about sufficient to meet the current demand. To overcome these problems, most U.S. states have set up an ISO (independent system operator): a non-profit governing body to arbitrate the use of the network. The basic questions facing ISOs are how to decide which contracts to deny (due to capacity constraints), and who is to bear the costs accrued when contracts are denied. Several criteria/policies have been proposed and/or are being legislated by the states as possible guidelines for the ISO to select a maximum-sized subset of contracts that can be cleared simultaneously [18]. These include: (a) Minimum Flow Denied: The ISO selects the subset of contracts that denies the *least* amount of proposed power flow. This proposal favors clearing bigger contracts first. (b) First-in First-out: The contract that comes first gets cleared first; this is the least discriminating to the contractors. (c) Maximum Consumers Served: This clears the smallest contracts first and favors the small buyers whose interests normally tend to go unheard.

There are three key issues in deciding policies that entail specific mechanisms for selecting a subset of contracts: *fairness* of a given policy to producers and consumers; the *computational complexity* of implementing a policy, and how *sound* a given policy is from an economic standpoint. (For instance, does the policy promote the optimal clearing price/network utilization etc.) Here we focus on evaluating the efficacy of a given policy with regard to its computational resource requirement and network resource utilization. It is intuitively clear that the underlying network, its capacity and topology, and the spatial locations of the bilateral contracts on the network, will play an important role in determining the efficacy of these policies. We do not discuss here the fairness and economics aspects of these policies: these are subjects of a companion paper. The work reported here is done as part of a simulation based analytical tool for deregulated electrical power industry being developed at the Los Alamos National Laboratory.

We experimentally analyze several algorithms for simultaneously clearing a maximal number of bilateral contracts. The qualitative insights obtained in this paper can be useful to policy makers who carry the ultimate responsibility of deploying the best clearing mechanism in the real world. The algorithms were chosen according to provable performance, ability to serve as a proxy for some of the above-stated policies, and computational requirement. The algorithms are as follows; see § 3 for their specification. The ILP-RANDOMIZED ROUNDING (RR) algorithm has a provable performance guarantee under certain conditions. The computational resource requirement is quite high, but the approach also provides us with an upper bound on any optimal solution and proves useful in comparing the performance of the algorithms. The LARGEST-FIRST HEURISTIC (LF) is a proxy for the *Minimum Flow Denied* policy. The SMALLEST-FIRST HEURISTIC (SF) serves as a proxy for the *Maximum Contracts Served* policy. The RANDOM-ORDER HEURISTIC (RO) clears the contracts in the random order. This algorithm was chosen as a proxy for the *First-in First-out* policy. Such a policy is probably the most natural clearing mechanism and is currently in place at many exchanges.

To compare the algorithms in a quantitative and (semi-)rigorous way, we employ statistical tools and experimental designs. Many of the basic tools are standard in statistics and their use is common in other fields. But to the best of our knowledge, the use of formal statistical methods in experimental algorithmics for analyzing/comparing the performance of algorithms has not been investigated. *Analysis of Variance* (ANOVA) is one such technique that can help identify which algorithms and scenarios are superior in performance. We believe that such statistical methods should be investigated further by the experimental algorithmics community for deriving more (semi)-quantitative conclusions when theoretical proofs are hard or not very insightful. For instance, consider a given approximation algorithm that has a worst-case performance guarantee of ρ. First, the algorithm may perform much better on realistic instances that are of interest. Quantifying the special structure of such instances is often hard; this often makes it difficult to develop further theoretical improvements on the performance of the algorithm. Second, many heuristics that have poor worst-case performance perform very well on such instances. Statistical methods such as ANOVA can facilitate the comparison of such heuristics and provable algorithms in settings that are of interest to the users of such algorithms.

We used a coarse representation of the Colorado electrical power network (see § 4) to qualitatively compare the four algorithms discussed above in fairly realistic settings. The realistic networks differ from random networks and structured networks in the following ways: (i) Realistic networks typically have a very low average degree. In fact, in our case the average degree of the network is no more than 3. (ii) Realistic networks are not very uniform. One typically sees one or two large clusters (downtown and neighboring areas) and small clusters spread out throughout. (iii) For most empirical studies with random networks, the edge weights are chosen independently and uniformly at random from a given interval. However, realistic networks typically have very specific kinds of capacities since they are constructed with particular design goal.

From our preliminary analysis, it appears that although the simple heuristic algorithms do not have worst-case performance guarantees, they outperform the theoretically better randomized rounding algorithm. We tested the algorithms on four carefully chosen scenarios. Each scenario was designed to test the algorithms and the resulting solutions in a deregulated setting. The empirical results show that networks that are capable of satisfying all demand in a regulated marketplace can often be inadequate for satisfying all (or even a acceptable fraction) of the bilateral contracts in a deregulated market. Our results also confirm intuitive observations: e.g., the number of contracts satisfied crucially depends on the scenario and the algorithm.

As far as we are aware, this is the first study to investigate the efficacy of various policies for contract satisfaction in a deregulated power industry. Since it was done in fairly realistic settings, the qualitative results obtained here have implications for policy makers. Our results can also be applied in other settings, such as bandwidth-trading on the Internet. See, e.g., [2]. Also, to our knowledge, previous researchers have not considered the effect of the underlying network on the problems; this is an important parameter especially in a free-market scenario.

The rest of this paper is organized as follows. The problem definitions and algorithms considered are described in Sections 2 and 3 respectively. Our experimental setup is discussed in Section 4. Section 5 presents our experimental results and analyzes them and Section 6 concludes the paper. In the appendix, we discuss interesting optimization issues that arise from deregulation, and also show problem instances on which our algorithms do not perform well.

2 Problem Definitions

We briefly define the optimization problems studied here. We are given an undirected network (the power network) $G = (V, E)$ with capacities c_e for each edge e and a set of source-sink node pairs (s_i, t_i), $1 \le i \le k$. Each pair (s_i, t_i) has: (i) an integral *demand* reflecting the amount of power that s_i agrees to supply to t_i and (ii) a negotiated *cost* of sending unit commodity from s_i to t_i. As is traditional in the power literature, we will refer to the source-sink pairs along with the associated demands as *a set of contracts*. In general, a source or sink may have multiple associated contracts. We find the following notation convenient to describe the problems. Denote the set of nodes by N. The contracts are defined by a relation $R \subseteq (N \times N \times \Re \times \Re)$ so that tuple $(v, w, \alpha, \beta) \in R$ denotes a contract between source v and sink w for α units of commodity at a cost of β per unit of the commodity. For $A = (v, w, \alpha, \beta) \in R$ we denote $source(A) = v$, $sink(A) = w$, $flow(A) = \alpha$ and $cost(A) = \beta$. Corresponding to the power network, we construct a digraph $H = (V \cup S \cup T \cup \{s, t\}, E')$ with source s, sink node t, capacities $u : E' \to \Re$ and costs $c' : E' \to \Re$ as follows. For all $A \in R$, define new vertices v_A and w_A. Let $S = \{v_A \mid A \in R\}$ and $T = \{w_A \mid A \in R\}$. Each edge $\{x, y\}$ from G is present in H as the two arcs (x, y) and (y, x) that have the same capacity as $\{x, y\}$ has in G, and with cost 0. In addition, for all $A = (v, w, \alpha, \beta) \in R$, we introduce: (i) arcs (v_A, v) and (w, w_A) with infinite capacity and zero cost; (ii) arc (s, v_A) with capacity $flow(A) = \alpha$ and cost 0; and (iii) arc (w_A, t) with capacity $flow(A) = \alpha$ and cost equaling $cost(A)$. By this construction, we can assume without loss of generality that each node can participate in exactly one contract. A *flow* is simply an assignment of values to the edges in a graph, where the value of an edge is the amount of flow traveling on that edge. The value of the flow is defined as the amount of flow coming out of s (or equivalently the amount of flow coming in to t). A generic *feasible flow* $f = (f_{x,y} \ge 0 : (x, y) \in E')$ in H is any non-negative flow that: (a) respects the arc capacities, (b) has s as the only source of flow and t as the only sink. Note that for a given $A \in R$, in general it is not necessary that $f_{s,v_A} = f_{w_A,t}$. For a given contract $A \in R$, A is said to be *satisfied* if the feasible flow f in H has the additional property that for $A = (v, w, \alpha, \beta)$, $f_{s,v_A} = f_{w_A,t} = \alpha$; i.e., the contractual obligation of α units of commodity is shipped out of v and the same amount is received at w. Given a power network $G(V, E)$, a contract set R is *feasible* (or *satisfied*) if there exists a feasible flow f in the digraph H that satisfies every contract $A \in R$. Let $B = supply(s) = demand(t) = \sum_{A \in R} flow(A)$.

In practice, it is typically the case that R does not form a feasible set. As a result we have two possible alternative methods of relaxing the constraints: (i) relax the notion of feasibility of a contract and (ii) try and find a subset of contracts that are feasible. Combining these two alternatives we define the following types of "relaxed feasible" subsets of R. We will concern ourselves with only one variant here. A contract set $R' \subseteq R$ is a *0/1-contract satisfaction* feasible set if, $\forall A = (v, w, \alpha, \beta) \in R'$, $f_{s,v_A} = f_{w_A,t} = \alpha$.

Definition 2.1 *Given a graph $G(V, E)$ and a contract set R, the* (0/1-VERSION, MAX-FEASIBLE FLOW) *problem is to find a feasible flow f in H such that $\sum_{A \in R'} f(A)$ is maximized where R' forms a 0/1-contract satisfaction feasible set of contracts. In the related* (0/1-VERSION, MAX-#CONTRACTS) *problem, we aim to find a feasible flow f in H such that $|R'|$ is maximized, where R' forms a 0/1-contract satisfaction feasible set of contracts.*

Though such electric flow problems have some similarities with those from other practical situations, there are many basic differences such as reliability, indistinguishability between the power produced by different generators, short life-time due to inadequate storage, line effects etc. [22]. The variants of flow problems related to power transmission studied here are intuitively harder than traditional multi-commodity flow problems, since we *cannot distinguish between* the flow "commodities" (power produced by different generators). As a result, current solution techniques used to solve single/multi-commodity flow problems are not directly applicable to the problems considered here.

3 Description and Discussion of Algorithms

We work on the (0/1-VERSION, MAX-#CONTRACTS) problem here. Let n and m respectively denote the number of vertices and edges in the network G. In [5], it was shown that (0/1-VERSION, MAX-#CONTRACTS) is *NP*-hard; also, unless *NP* $\subseteq ZPP$, it cannot be approximated to within a factor of $m^{1/2-\epsilon}$ for any fixed $\epsilon > 0$, in polynomial time. Thus, we need to consider good heuristics/approximation algorithms. First, there are three simple heuristics. The SMALLEST-FIRST HEURISTIC considers the contracts in non-decreasing order of their demands. When a contract is considered, we accept it if it can be feasibly added to the current set of chosen contracts, and reject it otherwise. The LARGEST-FIRST HEURISTIC is the same, except that the contracts are ordered in non-increasing order of demands. In the RANDOM-ORDER HEURISTIC, the contracts are considered in a random order.

We next briefly discuss an approximation algorithm of [5]. This has proven performance only when all source vertices s_i are the same; however, this algorithm extends naturally to the multi-source case which we work on. An integer linear programming (ILP) formulation for the problem is considered in [5]. We have indicator variables x_i for the contract between s_i and t_i, and variables $z_{i,e}$ for each (s_i, t_i) pair and each edge e. The intended meaning of x_i is that the total flow for (s_i, t_i) is $d_i x_i$; the meaning of $z_{i,e}$ is that the flow due to the contract between (s_i, t_i) on edge e is $z_{i,e}$. We write the obvious flow and capacity constraints. Crucially, we also add the valid constraint $z_{i,e} \leq c_e x_i$ for all i and e. In the integral version of the problem, we will have $x_i \in \{0, 1\}$, and the $z_{i,e}$ as non-negative reals. We relax the condition "$x_i \in \{0, 1\}$" to "$x_i \in [0, 1]$" and solve the resultant LP; let y^* be the LP's optimal objective function value. We perform the following rounding steps using a carefully chosen parameter $\lambda > 1$. (a) Independently for each i, set a random variable Y_i to 1 with probability x_i/λ, and $Y_i := 0$ with probability $1 - x_i/\lambda$. (b) If $Y_i = 1$, we will choose to satisfy $(1 - \epsilon)$ of (s_i, t_i)'s contract: for all $e \in E$, set $z_{i,e} := z_{i,e}(1 - \epsilon)/x_i$. (c) If $Y_i = 0$, we choose to have no flow for (s_i, t_i): i.e., we will reset all the $z_{i,e}$ to 0. A deterministic version of this result based on *pessimistic estimators*, is also provided in [5]; see [5] for further details.

Theorem 3.1 *([5]) Given a network G and a contract set R, we can find an approximation algorithm for ($0/1$-VERSION, MAX-#CONTRACTS) when all source vertices are the same. Let OPT be the optimum value of the problem, and m be the number of edges in G. Then, for any given $\epsilon > 0$, we can in polynomial time find a subset of contracts R' with total weight $\Omega(OPT \cdot \min\{(OPT/m)^{(1-\epsilon)/\epsilon}, 1\})$ such that for all $i \in R'$, the flow is at least $(1 - \epsilon)d_i$.*

4 Experimental Setup and Methodology

To test our algorithms experimentally, we used a network corresponding to a subset of a real power network along with contracts that we generated using different scenarios. The network we used is based on the power grid in Colorado and was derived from data obtained from PSCo's (Public Service Company of Colorado) Draft Integrated Resources Plan listing of power stations and major sub stations. The network is shown in Figure 1. We restricted our attention to major trunks only.

Sources: The location and capacities of the sources was roughly based upon data obtained from PSCo's Draft Integrated Resources Plan listing of power stations and major sub stations.

Sinks: The location and capacity of the sinks were roughly based upon the demographics of the state of Colorado. In order to determine the location and capacity of the sinks we used the number of households per county obtained from the US Census bureau. By assigning counties (load) to specific sub stations (sink nodes) the data for the sinks were derived.

The following three websites can be accessed to obtain the necessary information:

- http://www.census.gov/population/estimates/county/co-99-1/99C1_08.txt gives the population per county as of 1995.
- http://www.census.gov/datamap/www/08.html contains a map of Colorado counties.
- Finally, http://ccpg.basinelectric.com/ is the PSCo Colorado Website.

Edges: The edge capacities were derived from test data obtained by running the network through a max-flow program with the source and sink capacities at maximum and no capacity limits placed upon the connecting edges. The total sink capacity equaled the total source capacity. The sink capacity was distributed to the various sink nodes in correspondence with population percentages assigned to each sink node. The edge capacities were then roughly assigned and the model was rerun through the max-flow program until all edge limits were defined. The criteria used for defining all of edge limits was that the network must be feasible under the condition of maximum source/sink capacity. Once the feasibility criteria was satisfied, some edge limits were set at capacity, while others were set higher than capacity in order to provide flexibility in contract development for the later problems.

Software and Data Format. DIMACS (http://dimacs.rutgers.edu) has developed a standard format for storing network data for input into existing network solvers. For the network being examined the need exists to include two directed arcs for each edge since the network is undirected. Addition of a master source and a master sink node with edges to the individual source and sink nodes was needed in order to conform to the format requirement of a single source and a single sink node. The edge capacities of the edges from the master source and sink nodes were set to be the capacities of the respective individual source or sink node.

4.1 Creation and Description of Test Cases

All the test cases were generated from the basic model. The general approach we used was to fix the edge capacities and generate source-sink contract combinations, using the capacities and aggregate demands in the basic model as upper bounds. To ensure that the test cases we generated corresponded to (1) difficult problems, i.e. infeasible sets of contracts, and (2) problems that might reasonably arise in reality, we developed several scenarios that included an element of randomness (described in § 4.2).

4.2 Description of Scenarios

The current implementation is still based upon a network which should be feasible only if the total source capacity is greater than the total sink capacity and the only requirement is that the total sink capacity be satisfied regardless of which source provides the power. **Scenarios 1, 2, 3** and **4** are based around the network with total generating capacity **6249 MW**, and reduced sink capacities near **4400MW** combined.

1. **Scenario 1:** This scenario is based upon the network with a total sink capacity (i.e. customer demand) of 4400MW. The source capacity (supplier's maximum production capacity) was reduced by a constant proportion from the total generating capacity based upon population density of Colorado counties. The source capacity of the network was reduced until the running the MAXFLOW code indicated that the maximum flow in the network to be slightly less than the demand. This reduction in the sources total production capacity increased the chances of refusing customers (contracts).

2. **Scenario 2:** For this scenario, we took the basic network and increased the sink capacity while the source capacity remained fixed.

3. **Scenario 3:** For generating instances for this scenario, the edge capacities were adjusted, reduced in most cases, to limit the network to a maximum flow of slightly more than 4400MW given its source and sink distribution. Here, if the load is allowed to be fulfilled from any source (as is normally done with centralized control), the network and the edge capacities are enough to handle a total of 4400MW. However, if we insist that a particular source needs to serve a particular sink (as is done in bilateral contract satisfaction), then the capacities may not be enough to handle the same load of 4400MW.

4. **Scenario 4:** For this scenario, we took the network of Scenario 3 and biased the selection of source nodes towards the lower valued source units.

4.3 Methodology

We worked with the four scenarios and ran all four algorithms for each. For the three greedy heuristics the implementations are fairly straightforward, and we used public-domain network flow codes. Implementing the randomized rounding procedure requires extra care. The pessimistic estimator approach of [5] works with very low probabilities, and requires significant, repeated re-scaling in practice. Thus we focus on the randomized version of the algorithm of [5]; five representative values of ϵ varying from .1 to .5

were chosen. We believe that satisfying a contract partially so that a contract is assigned less than .5 of the required demand is not very realistic. For each scenario, and for each of the 5 values of ϵ, the programs implementing the algorithms under inspection produced 30 files from which the following information could be extracted:

1. The running time of each algorithm.
2. Total number of satisfied contracts by each algorithm.
3. The LP upper bound on the IP and thus an upper bound on the approximations given by the algorithms.
4. The IP approximation and objective function value.

The number 30 was chosen to ensure that a statistically "large" sample of each measure would be provided in order to make valid statistical inference. We consider two parameters to measure the performance of our algorithms – (i) the running time and (ii) the quality of the solution obtained. More attention is given to the quality of solution measure since from a social standpoint contract satisfaction may leave little room for finding solutions that are far from optimal.

We now describe how these measures are used to make inferences about the qualitative performance of these algorithms with respect to one another and independently. Since the intent is to make inferences concerning solution quality, a measure of this sort must be derived from the data generated. To do this, the data provided by the LP relaxation is examined. The $\lfloor y^*_{\mathcal{AS}} \rfloor = \lfloor y^* \rfloor$ provides the best-case number or upper bound on the objective function value our algorithms can produce for a scenario. Hence, if an algorithm produces an objective function value of $\lfloor y^*_{\mathcal{AS}} \rfloor$, it has produced an optimal solution for a given scenario. For a given algorithm $\mathcal{A}$ and scenario $\mathcal{S}$, let $Value_{\mathcal{AS}}$ denote the number of contracts that are satisfied by $\mathcal{A}$ under $\mathcal{S}$. The fraction

$$p_{\mathcal{AS}} = \frac{Value_{\mathcal{AS}}}{\lfloor y^*_{\mathcal{AS}} \rfloor}$$

provides a measure of the quality of the algorithm's solution.

4.4 Experimental Objective

The objective of our experiments was to find out which, if any, of the algorithms discussed here performs better than the others, in terms of quality of solution and running time for different contract scenarios. The design of the experiment was developed keeping this objective in mind. Since the performance depends on the type of algorithm used and the contract scenario, these are our factors of interest. As mentioned in the section 4.3, for a given ϵ, 30 runs were performed for each algorithm-scenario pair. We perform two separate sets of experiments, one for the quality of solution as measured by $p_{\mathcal{A}}$ and the other for running time. This was done because the quality of solution and running time were independent of each other. The number of contracts satisfied do not depend upon the length of the time it takes to run the algorithm.

5 Results and Analysis

5.1 General Conclusions

We first present general conclusions obtained from our results and experimental analysis. These will be elaborated on subsequently.

1. Although there exist instances where the three heuristics produce solutions as large as $\Omega(n)$ times the optimal fractional solution, most of our tests show that we could find integral solutions fairly close to optimal.

2. Our experiments show that different scenarios make a *significant* difference in the type of solutions obtained. For example, the quality of solution obtained using the fourth scenario is significantly worse than the first three scenarios. The sensitivity to the scenarios poses interesting questions for infrastructure investment. The market will have to decide the cost that needs to be paid for expecting the necessary quality of service. It also brings forth the equity-benefit question: namely, who should pay for the infrastructure improvements?

3. It is possible that for certain scenarios, the underlying network is incapable of supporting even a minimum acceptable fraction of the bilateral contracts. This observation – although fairly intuitive – provides an extremely important message, namely, networks that were adequate to service customers in a completely regulated power market *might* not be adequate in deregulated markets. This makes the question of evicting the bilateral contracts more important.

4. One expects a trade-off between the number of contracts satisfied and the value of ϵ, for the randomized rounding algorithm: as ϵ increases, and the demand condition is more relaxed, a higher number of contracts should get satisfied. But our experiments show that the change in the number of contracts satisfied for different values of ϵ is insignificant. Also, $\lambda = 2$ gave the best solutions in our experiments.

5. In practical situations, the Random-Order heuristic would be the best to use since it performs very close to the optimal in terms of quality of solution and has very low running time. Furthermore, though the Smallest-First

heuristic does even better on many of our experiments, Random-Order is a natural proxy to model contracts arriving in an unforeseen way. Also, since the heuristics deliver solutions very close to the LP upper bound, we see that this LP bound is tight and useful. To further evaluate the randomized rounding algorithm, we need to implement its deterministic version [5], which is a non-trivial task.

5.2 Statistical Background

We use a statistical technique known as *analysis of variance* (ANOVA) to test whether differences in the sample means of algorithms and scenarios reflect differences in the means of the statistical populations that they came from or are just sampling fluctuations. This will help us identify which algorithm and scenarios perform the best.[1]

ANOVA has the following three advantages over individual t-tests[2] when the number of groups being compared is greater than two. See [9] for more details. In our case, we have four algorithms and four scenarios. Standard statistics terminology for a hypothesis that we wish to test, is *null hypothesis*.

- It gives accurate and known type-I error probability.[3]
- It is more powerful i.e. if null hypothesis is false, it is more likely to be rejected.
- It can assess the effects of two or more independent variables simultaneously.

5.3 Mathematical Model

Quality of Solution: We first describe the experiment for the quality of solution i.e. $p_{\mathcal{AS}}$. We use a two-factor ANOVA model since our experiment

[1]The populations in each of the groups are assumed to be normally distributed and have equal variances. The effect of violation of ANOVA assumptions of normality and homogeneity of variances have been tested in the literature ([10]) and the results show:

- Non-normality has negligible consequences on type-I and II error probabilities unless the populations are highly skewed or the sample is very small.
- When the design is balanced, i.e. the number of observations are the same for each group, violation of homogeneity of variance assumption has negligible consequences on the accuracy of type-I error probabilities.

[2]t-test checks for the significance of the difference in the means of *two* samples. It can assess whether the difference in sample means is just due to sampling error or they really are from two populations with different means.

[3]The probability of rejecting a null hypothesis when it is actually true.

involves two factors which are:

1. The algorithms: $\mathcal{A}_i, i = 1, 2, 3$ and 4.

2. The scenario: $\mathcal{S}_j, j = 1, 2, 3$ and 4.

Following classical statistics terminology, we will sometimes refer to algorithms as *treatments* and the scenarios as *blocks*. We will use $\mathcal{A}$ to denote the set of algorithms and $\mathcal{S}$ to denote the set of scenarios. For each algorithm-scenario pair we have *30 observations (or replicates)*. When testing the efficacy of the algorithms, we use 4 algorithms, each having 120 observations (30 for each scenario) from the corresponding population. The design of experiment used here is a *fixed-effect complete randomized block*. *Fixed-effect* because the factors are fixed as opposed to randomly drawn from a class of algorithms or scenarios; the conclusions drawn from this model will hold only for these particular algorithms and scenarios. *Complete* implies that the number of observations are the same for each block. *Randomized* refers to the 30 replicates being drawn randomly. We wish to test the hypothesis:

> Is the mean quality of solution provided by different algorithms the same, against the alternative hypothesis that some or all of these means are unequal?

The model for randomized block design includes constants for measuring the scenario effect (block effect), the algorithm effect (treatment effect) and a possible interaction between the scenarios and the algorithms. The appropriate mathematical model is as follows:

$$X_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk},$$

where X_{ijk} is the measurement ($p_{\mathcal{AS}}$) for the kth sample within the ith algorithm and the jth scenario. τ_i is the algorithm effect. β_j is the scenario effect. $(\tau\beta)_{ij}$ captures the interaction present between the algorithms and the scenarios. ε_{ijk} is the random error. See [8, 9] for further details on ANOVA.

We use S-Plus [15] software to run two-factor ANOVA to test the following three different null hypotheses.

1. Are the means given by the 4 different algorithms equal? The null hypothesis here is, $H_0 : \tau_1 = \tau_2 = \tau_3 = \tau_4$.

2. Are the means given by the 4 different scenarios equal? The null hypothesis here is, $H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4$.

3. Is there any interaction between the two factors? The null hypothesis here is, $H_0 : (\tau\beta)_{ij} = 0$.

The results of two-factor ANOVA are shown in Table 1 and Table 2. In the following discussion, we explain the meaning of each column in Table 1. DF refers to the degrees of freedom, SS refers to the sum of squared deviations from the mean. MS refers to the mean square error, which is the sum of squares divided by the degrees of freedom.[4]

The p-value gives the smallest level of significance at which the null hypothesis can be rejected.[5] The lower the p-value, the lesser the agreement between the data and the null hypothesis. Finally the F**-test** is as follows. To test the null hypothesis, i.e., whether the population means are equal, ANOVA compares two estimates of σ^2. The first estimate is based on the variability of each population mean around the grand mean. The second is based on the variability of the observations in each population around the mean of that population. If the null hypothesis is true, the two estimates of σ^2 should be essentially the same. Otherwise, if the populations have different means, the variability of the population mean around the grand mean will be much higher than the variability within the population. The

[4]The sum of squares for the algorithm factor can be calculated as:

$$SS_{\mathcal{A}} = nJ\Sigma_i(\overline{X}_{i\cdot\cdot} - \overline{X}_{\cdot\cdot\cdot})^2$$

where n is the number of replicates, J is the number of scenarios, $\overline{X}_{i\cdot\cdot}$ is the mean of algorithm i across all scenarios and $\overline{X}_{\cdot\cdot\cdot}$ is the grand mean across all algorithms and scenarios. Recall that in our case $n = 30$ and $J = 4$ yielding a total sample size of 120.

The sum of squares for scenario factor can be calculated as:

$$SS_{\mathcal{S}} = nI\Sigma_j(\overline{X}_{\cdot j\cdot} - \overline{X}_{\cdot\cdot\cdot})^2$$

where as before n is the number of replicates, I is the number of algorithms and $\overline{X}_{\cdot j\cdot}$ is the mean of scenario j across all algorithms. Again, in our case $n = 30$ and $I = 4$.

The sum of squares for algorithms and scenario interaction is:

$$SS_{\mathcal{AS}} = n\Sigma j\Sigma_i[\overline{X}_{ij\cdot} - (\overline{X}_{\cdot\cdot\cdot} + \hat{\tau}_i + \hat{\beta}_j)]^2$$

Here $\overline{X}_{ij\cdot}$ is the mean of observations for the algorithm i scenario j pair. $\hat{\tau}_i$ and $\hat{\beta}_j$ are respectively the estimated least square values of τ_i and β_j. The sum of squares "within" refers to the squared difference between each observation and the mean of the scenario and algorithm of which it is a member. It is also referred as the residual sum of squares. This can be calculated as:

$$SS_{\mathcal{W}} = n\Sigma j\Sigma_i\Sigma_k(X_{ijk} - \overline{X}_{ij\cdot})^2$$

The total sum of squares is

$$SS_{\mathcal{T}} = SS_{\mathcal{A}} + SS_{\mathcal{S}} + SS_{\mathcal{AS}} + SS_{\mathcal{W}}$$

[5]To obtain a p-value for say $F_{\mathcal{A}}$, the algorithm effect, we would look across the row associated with 3 degree of freedom in the numerator and 464 degrees of freedom in the denominator in the F-distribution table and find the largest value that is still less than the one obtained experimentally. From this value, we obtain a p-value of 0 for $F_{\mathcal{A}}$.

null hypothesis in the F-test will be accepted if the two estimates of σ^2 are almost equal.

In a two-factor fixed-effect ANOVA, three separate F**-tests** are performed: two tests for the factors, and the third for the interaction term. The null hypothesis for the first factor can be written as:

$$H_0^{\mathcal{A}} : \mu_{1\cdot\cdot} = \mu_{2\cdot\cdot} = \cdots = \mu_{j\cdot\cdot}$$

which is equivalent to writing: $H_0 : \tau_1 = \tau_2 = \tau_3 = \tau_4$. The F-test is:

$$F_{\mathcal{A}} = \frac{SS_{\mathcal{A}}/(I-1)}{SS_{\mathcal{W}}/IJ(n-1)}$$

and the null hypothesis for the second factor can be written as:

$$H_0^{\mathcal{S}} : \mu_{\cdot 1 \cdot} = \mu_{\cdot 2 \cdot} = \cdots = \mu_{\cdot j \cdot}$$

which is equivalent to writing: $H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4$. The F-test is:

$$F_{\mathcal{S}} = \frac{SS_{\mathcal{S}}/(J-1)}{SS_{\mathcal{W}}/IJ(n-1)}$$

and the null hypothesis for the interaction term can be written as:

$$H_0^{\mathcal{AS}} : (\tau\beta)_{ij} = 0.$$

The F-test is:

$$F_{\mathcal{AS}} = \frac{SS_{\mathcal{AS}}/(I-1)(J-1)}{SS_{\mathcal{W}}/IJ(n-1)}$$

If this F-ratio is close to 1, the null hypothesis is true. If it is considerably larger – implying that the variance between means is larger than the variance within a population – the null hypothesis is rejected. The F distribution table should be checked to see if the F-ratio is significantly large.

The results in Table 1 show that all the above three null hypothesis are rejected at any significance level. This implies that the performance (measured by $p_{\mathcal{AS}}$) of at least one of the algorithms is significantly different from the other algorithms. Also, different scenarios make a difference in the performance. Finally, the scenarios and the algorithms interact in a significant way. The interaction implies that the performance of the algorithms are different for different scenarios.

Source	DF	SS	MS	F**-test**	p**-value**
Scenario (Block)	3	0.14	0.05	43.38	0
Algorithm (Treatment)	3	22.78	7.59	6792.60	0
Scenario:Algorithm	9	0.12	0.01	15.90	0
Residuals	464	0.40	.0008		
Total	479	23.45			

Table 1: **Results of Two-Factor ANOVA:** This table shows results of two-factor ANOVA where the factors are algorithms and scenarios. The measurement is the quality of solution, given by $p_{\mathcal{AS}}$. The p-values show that the algorithm effect, scenario effect and the interaction between the algorithms and scenarios are all significant at any level of confidence.

5.3.1 Contrasts

The next question of interest is what really caused the rejection of the null hypothesis; just knowing that at least one of the algorithms is different does not help us identify which algorithm is significantly different. To answer this we use a procedure called *contrast*. A contrast $\mathcal{C}$ among I population means (μ_i) is a linear combination of the form

$$\mathcal{C} = \Sigma_i \alpha_i \mu_i = \alpha_1 \mu_1 + \alpha_2 \mu_2 + \cdots + \alpha_I \mu_I$$

such that the sum of contrast coefficients $\Sigma_i \alpha_i$ is zero. In the absence of true population means, we use the unbiased sample means which gives the estimated contrast as:

$$\hat{\mathcal{C}} = \Sigma_i \alpha_i \overline{X}_i = \alpha_1 \overline{X}_1 + \alpha_2 \overline{X}_2 + \cdots + \alpha_I \overline{X}_I.$$

The contrast coefficients α_1, α_2, $\cdots$, α_I are just positive and negative numbers that define the particular hypothesis to be tested. The null hypothesis states that the value of a parameter of interest for every contrast is zero, i.e., $H_0 : \mathcal{C} = 0$. The value of the contrast is tested by an F-test to see if the observed value of the contrast is significantly different from the hypothesized value of zero.

Table 2 shows the average value of the quality of solution for each algorithm-scenario pair. e.g. $\overline{X}_{11\cdot}$ means that we fix $i = 1$ and $j = 1$ and take the average of X_{ijk} over all k. From Table 2, it is clear that the randomized rounding algorithm (RR) is different from all the other algorithms for

Performance Measure: Quality of Solution (in %)					
	RR	SF	LF	RO	Scenario Means
Scenario 1	$\overline{X}_{11\cdot}$=48.68	$\overline{X}_{21\cdot}$=99.73	$\overline{X}_{31\cdot}$=97.97	$\overline{X}_{41\cdot}$=97.78	$\overline{X}_{\cdot 1\cdot}$=86.02
Scenario 2	$\overline{X}_{12\cdot}$=46.91	$\overline{X}_{22\cdot}$=99.56	$\overline{X}_{32\cdot}$=98.38	$\overline{X}_{42\cdot}$=98.93	$\overline{X}_{\cdot 2\cdot}$=85.94
Scenario 3	$\overline{X}_{13\cdot}$=45.69	$\overline{X}_{23\cdot}$=99.25	$\overline{X}_{33\cdot}$=97.10	$\overline{X}_{43\cdot}$=98.82	$\overline{X}_{\cdot 3\cdot}$=85.22
Scenario 4	$\overline{X}_{14\cdot}$=46.99	$\overline{X}_{24\cdot}$=98.03	$\overline{X}_{34\cdot}$=88.65	$\overline{X}_{44\cdot}$=93.41	$\overline{X}_{\cdot 4\cdot}$=81.77
Algo. Means	$\overline{X}_{1\cdot\cdot}$=47.07	$\overline{X}_{2\cdot\cdot}$=99.14	$\overline{X}_{3\cdot\cdot}$=95.51	$\overline{X}_{4\cdot\cdot}$=97.24	$\overline{X}_{\cdot\cdot\cdot}$ = 84.74

Table 2: **The Mean Values of the Quality of Solution:** This table shows the mean values of the quality of solution for each algorithm and each scenario.

all four scenarios. On an average, RR algorithm satisfies 49% less contracts than the Largest-First (LF) heuristic and 50% less than the Random-Order (RO) heuristic and 52% less contracts than the Smallest-First (SF) heuristic. The difference between SF, RO and LF heuristics appears only marginal. Based on this observation, we constructed the following contrast that tests if RR is statistically significantly different from the other three algorithms:

$$\mathcal{C}_{1Q} = \frac{1}{3}(\overline{X}_{2\cdot\cdot}) + \frac{1}{3}(\overline{X}_{3\cdot\cdot}) + \frac{1}{3}(\overline{X}_{4\cdot\cdot}) - \overline{X}_{1\cdot\cdot}$$

Using the value of algorithm means from Table 2 we can calculate the value of $\mathcal{C}_{1Q}$ (Q stands for the quality of solution) to be equal to 0.50.[6] The sum of squares of a contrast is expressed as:

$$SS(\mathcal{C}_{1Q}) = \frac{(\mathcal{C}_{1Q})^2}{\Sigma_i \alpha_i^2 / N_i}$$

Here α_i are the coefficients of the contrast and $N_i = 120$ is the number of observations (i.e. sample points for each algorithm across all scenarios). This results in $SS(\mathcal{C}_{1Q}) = 22.68$. Now we can use the following F-test to see the significance of the contrast:

$$SS(\mathcal{C}_{1Q})/MSE \sim F(1, 464)$$

[6]The table values are shown in percentages, but here we use actual values.

MSE stands for the mean square error of the residuals. The contrast has one degree of freedom and residuals have 464 degrees of freedom (see table 1). F = 22.68/.0008 = 28350, since the observed value of F-test is greater than the critical F-value given in the F-distribution table, for any significance level, the null hypothesis is rejected. This confirms our earlier observation that the RR algorithm is significantly inferior in performance compared to the other three algorithms. The sum of squares of $SS(\mathcal{C}_{1Q}) = 22.68$ shows that 98% of the variation in factors sum of squares (total factors sum of squares being 23.05 i.e. total SS - residual SS, see table 2) is due to the difference in RR algorithm versus the other three algorithms.

Table 2 shows that the first three scenarios clear about 86% of the optimal number of contracts while under the fourth scenario, the number of contracts cleared is less than 82% of the optimal. Even though the difference in the number of cleared contracts is not very big, one would be curious to find out if the difference in performance under the first three scenarios versus the fourth scenario is significant or not. To answer this we created the following contrast which is orthogonal[7] to the first contrast ($\mathcal{C}_{1Q}$):

$$\mathcal{C}_{2Q} = \frac{1}{3}(\overline{X}_{\cdot 1 \cdot}) + \frac{1}{3}(\overline{X}_{\cdot 2 \cdot}) + \frac{1}{3}(\overline{X}_{\cdot 3 \cdot}) - \overline{X}_{\cdot 4 \cdot}$$

Just like $\mathcal{C}_{1Q}$, we can calculate the value of $\mathcal{C}_{2Q}$ using table 2:

$$SS(\mathcal{C}_{2Q})/MSE \sim F(1, 464) = 0.14/.0008 = 175$$

Again, the null hypothesis is rejected implying that the fourth scenario is indeed significantly different from the other three scenarios.

Now we look at two more contrasts to check if SF and LF are significantly different ($\mathcal{C}_{3Q}$) and LF and RO are significantly different ($\mathcal{C}_{4Q}$).

$$\mathcal{C}_{3Q} = \overline{X}_{2\cdot\cdot} - \overline{X}_{3\cdot\cdot}$$

$$\mathcal{C}_{4Q} = \overline{X}_{3\cdot\cdot} - \overline{X}_{4\cdot\cdot}$$

$$SS(\mathcal{C}_{3Q})/MSE \sim F(1, 464) = 2.178/.0008 = 2722.5$$

$$SS(\mathcal{C}_{4Q})/MSE \sim F(1, 464) = 1.038/.0008 = 1297.5$$

[7] Two contrasts $\mathcal{C}_1$ and $\mathcal{C}_2$ are said to be orthogonal if the sum of the products of their corresponding coefficients is zero. It is desirable to have independent or orthogonal contrasts because independent tests of hypotheses can be made by comparing the mean square of each such contrast with the mean square of the residuals in the experiment. Each contrast has only one degree of freedom.

Source	DF	SS	MS	F**-test**	p**-value**
Scenario (Block)	3	21152	7050.8	56.97	0
Algorithm (Treatment)	3	2161199	720399.8	5821.07	0
Scenario:treatment	9	28156	3128.5	47.78	0
Residuals	464	30381	65.5		
Total	479	2240888			

Table 3: **Results of Two-Factor ANOVA:** This table shows results of two-factor ANOVA where the factors are algorithms and scenarios. The measurement is the running time of the algorithm-scenario pair. The p-values show that the algorithm effect, scenario effect and the interaction between the algorithms and scenarios are all significant at any level of confidence.

For both $\mathcal{C}_{3Q}$ and $\mathcal{C}_{4Q}$, the observed value of the F-test is greater than the critical F-value given in the F-distribution table, the null hypothesis in both cases are rejected, implying that SF provides a better solution than LF and also that RO performs significantly better than LF.

In summary, all algorithms show significantly different performance when measured in terms of quality of solution. On an average, the best solution is given by the SF heuristic and the worst by the RR.

Running Time: Tables 3 and 4 show results of the same experiment when performance is measured by the *running time* of the algorithm. The factors, number of observations, kinds of tests, etc. remain the same as before, except the performance measure. Table 3's results clearly demonstrate that different algorithms take significantly different time to run and that different scenarios have significantly different running time. The interaction term is significant at any level of confidence implying that the running time of an algorithm is different for different scenarios.

Table 4 shows that the RR algorithm takes noticeably more time to run as compared to the other three heuristics. Among the three heuristics, LF and RO take about the same time but SF takes about 19 megaticks more than the LF and RO. Similarly, scenario 3 and 4 take about the same time but scenario 1 and 2 look different. To test all the above mentioned observations, we create the following different contrasts:

$$\mathcal{C}_{1t} = \frac{1}{3}(\overline{X}_{2\cdot\cdot}) + \frac{1}{3}(\overline{X}_{3\cdot\cdot}) + \frac{1}{3}(\overline{X}_{4\cdot\cdot}) - \overline{X}_{1\cdot\cdot}$$

Performance Measure: Running Time (in Megaticks)					
	RR	SF	LF	RO	Scenario Means
Scenario 1	$\overline{X}_{11.}$=163.33	$\overline{X}_{21.}$=41.23	$\overline{X}_{31.}$=24.57	$\overline{X}_{41.}$=25.50	$\overline{X}_{.1.}$=63.66
Scenario 2	$\overline{X}_{12.}$=218.23	$\overline{X}_{22.}$=49.63	$\overline{X}_{32.}$=29.73	$\overline{X}_{42.}$=30.23	$\overline{X}_{.2.}$=81.96
Scenario 3	$\overline{X}_{13.}$=181.70	$\overline{X}_{23.}$=45.70	$\overline{X}_{33.}$=23.30	$\overline{X}_{43.}$=26.43	$\overline{X}_{.3.}$=69.28
Scenario 4	$\overline{X}_{14.}$=184.33	$\overline{X}_{24.}$=44.53	$\overline{X}_{34.}$=27.00	$\overline{X}_{44.}$=27.27	$\overline{X}_{.4.}$=70.78
Algo. Means	$\overline{X}_{1..}$=186.90	$\overline{X}_{2..}$=45.27	$\overline{X}_{3..}$=26.15	$\overline{X}_{4..}$=27.36	$\overline{X}_{...}$ = 71.42

Table 4: **The Mean Values of the Running Time:** This table shows the mean values of the running time for each algorithm and each scenario.

$$\mathcal{C}_{2t} = \frac{1}{2}(\overline{X}_{3..}) + \frac{1}{2}(\overline{X}_{4..}) - \overline{X}_{2..}$$

$$\mathcal{C}_{3t} = \overline{X}_{3..} - \overline{X}_{4..}$$

$$\mathcal{C}_{4t} = \overline{X}_{.1.} - \overline{X}_{.2.}$$

All the above contrasts are orthogonal to each other. The first contrast, $\mathcal{C}_{1t}$ (here t stands for running time), checks if the RR algorithm takes more time to run than the other three heuristics. The second contrast, $\mathcal{C}_{2t}$, will find if the SF heuristic is significantly different from the LF and RO heuristic. The third contrast, $\mathcal{C}_{3t}$, checks if the LF and RO heuristics take about the same time to run. Finally, contrast $\mathcal{C}_{4t}$, check if the first scenario takes less time to run than the second scenario. The results of all the contrasts are shown below.

$$SS(\mathcal{C}_{1t})/MSE \sim F(1,464) = 2133700.9/65.5 = 32575.5$$

$$SS(\mathcal{C}_{2t})/MSE \sim F(1,464) = 27424.4/65.5 = 418.69$$

$$SS(\mathcal{C}_{3t})/MSE \sim F(1,464) = 72.6/65.5 = 1.11$$

$$SS(\mathcal{C}_{4t})/MSE \sim F(1,464) = 20093.4/65.5 = 306.76$$

As can be seen by looking at the F-distribution table, all the above contrasts except $\mathcal{C}_{3t}$ show that the observed value of the F-test is greater than the critical F-value. Hence the null hypothesis i.e. $H_0 : \mathcal{C}_{it} = 0$ where $i = 1, 2, 4$ can be rejected at any level of significance. This confirms our earlier hypothesis that RR indeed takes longer to run than the other three heuristics. SF takes more time to run than the LF and RO heuristics and the second scenario takes significantly more time to run than the first scenario.

The mean difference in running time across different algorithms shows that all algorithms are significantly different in terms of running time except for the *Largest-First* and the *Random-Order* heuristics. These two heuristics take about the same time to run and indeed a contrast done i.e. $\mathcal{C}_{3t}$ on LF and RO proves that and the null hypothesis, $H_0 : \mathcal{C}_{3t} = 0$, is accepted.

The randomized rounding algorithm takes significantly more time to run than any of the other heuristics. The gap in running time narrows when RR is compared against SF. RR takes 141 megaticks more time than the SF heuristic, 160 megaticks more than the LF and RO. SF takes more time to run than LF and RO but it clears more contracts than LF and RO.

All the above analysis was performed while keeping the value of ϵ constant at 0.1. The performance of the randomized rounding algorithm[8] does not change in any significant way, both in terms of $p_{\mathcal{AS}}$ and running time when ϵ varied from 0.1 to 0.5. So all the above results hold for $\epsilon = 0.1, 0.2, 0.3, 0.4$ and 0.5.[9]

Summary: It is clear that SF heuristic clears the most contracts, almost as good as the optimal but takes more running time as compared to LF and RO. However, it takes only a quarter of the time as compared to RR. As far as scenarios go, the first scenario clears most contracts in the least amount of time. From a practical standpoint, the RO heuristic seems to be the best since it performs very well both in terms of running time and quality of solution and is trivial to implement. It performs very close to optimal in terms of clearing contracts and yet takes minimal time to do it as compared to the other algorithms. The RR algorithm, although it gives good theoretical lower bounds, is not very appropriate for real-life situations where both time and a high level of contract satisfaction have a very high priority.

6 Discussion and Concluding Remarks

We carried out an empirical study to evaluate the quality and running time performance of four different market clearing mechanisms. A novel aspect of our work is the use of statistical technique, *Analysis of Variance*, to compare the performance of different market clearing mechanisms. This tech-

[8] Other heuristics do not depend on the value of ϵ.

[9] The results are available from the authors upon request.

nique allows us to formally test which algorithm performs better in terms of each of the performance measures.

One heuristic was based on using a relaxation of integer linear program followed by randomized rounding of the fractional solution to yield an approximate integral solution. Although the algorithm had a provable performance guarantee, experiments suggest that the algorithm is not likely to be practically useful given the running time and the quality of solution produced. The result is not entirely unexpected; it has been observed that many approximation algorithms that are designed to work in the worst case typically do not have a very good average case behavior.

We also studied three different simple heuristics: experimental results suggest that each is likely to perform better than the theoretically provable approximation algorithm. This is in spite of the fact that it is very easy to construct instances where the heuristics have unboundedly poor performance guarantee.

One of the heuristics: the *random-order heuristic* was studied to emulate a simple "first-come first-serve" type clearing mechanism that is currently employed by many ISO. The heuristic performs surprisingly well even compared to a bound on an optimal solution obtained via linear programming. The results suggest that this simple clearing mechanism currently employed might result in near-optimal utilization of the network resources.

Our overall assessment is that for the purposes of developing large-scale microscopic-simulations of the deregulated power industry, the three heuristic methods give sufficiently good performance in terms of the quality of solution and the computational time requirement.

An interesting direction for future research is to study (both theoretically and experimentally) the performance of these algorithms when we have flow constraints modeling resistive networks. The additional constraints imposed on the system could conceivably make the problem easier to approximate. See [1] for further discussions on this topic.

Acknowledgments: All authors except Aravind Srinivasan were supported by the Department of Energy under Contract W-7405-ENG-36. Aravind Srinivasan was supported in part by NSF Award CCR-0208005. We thank Dr. Martin Wildberger (EPRI) for introducing us to the problems considered herein, and for the extensive discussions. We thank Mike Fugate, Dean Risenger (Los Alamos National Laboratory) and Sean Hallgren (Cal Tech) for helping us with the experimental evaluation. Finally, thanks to the referees of Workshop on Algorithmic Engineering (WAE) 2001 for their helpful comments on the preliminary version of this paper.

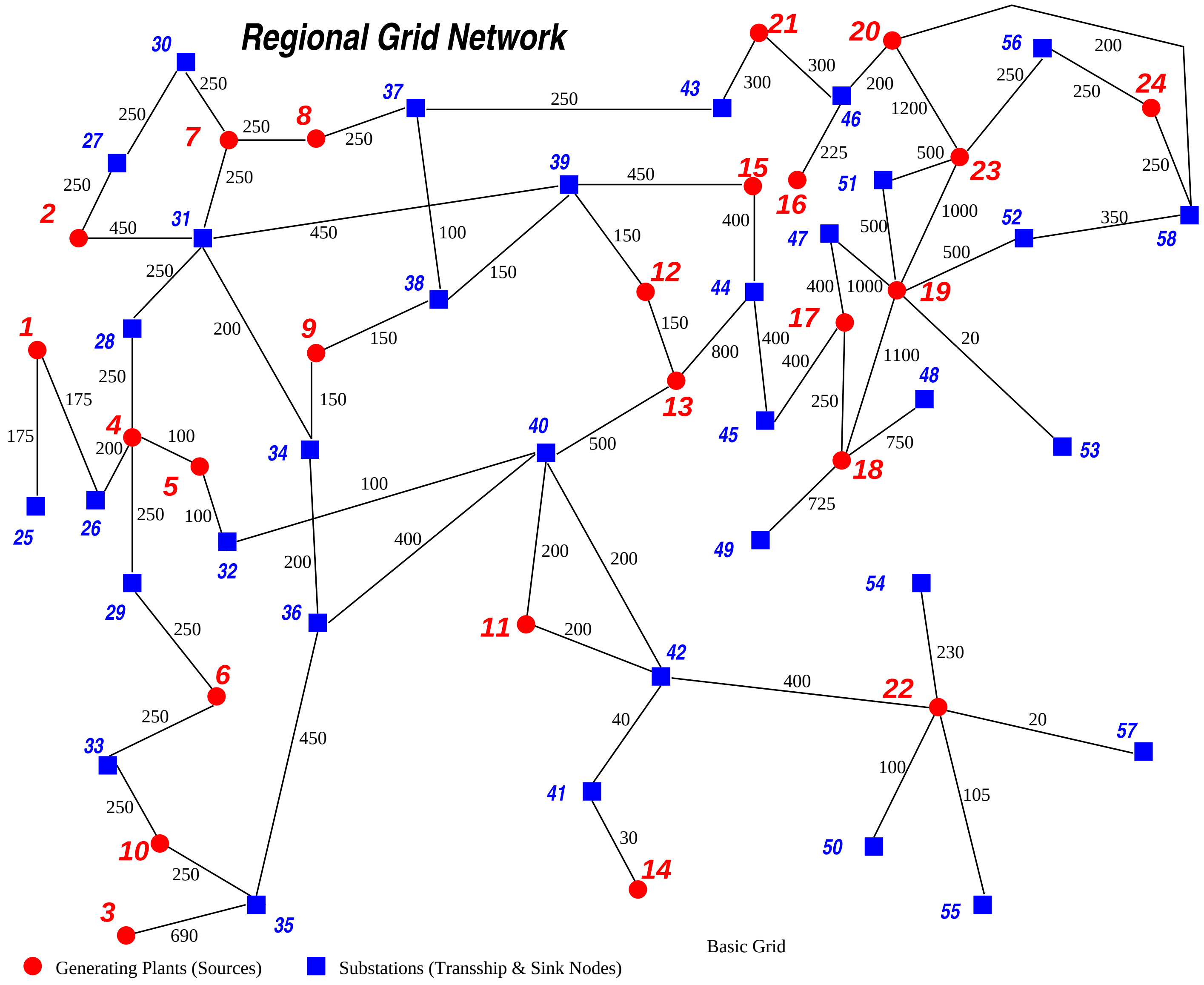


Figure 1: This shows the network with node numbered as they are referenced in all scenarios and edge capacities labeled at values used for **Scenarios 1 & 2**. The placement of the nodes and edges are what is probably the final form. The least number of edges cross and the nodes in the upper right are spread out a little bit maintaining the general feel of the distribution while allowing easier reading.

df

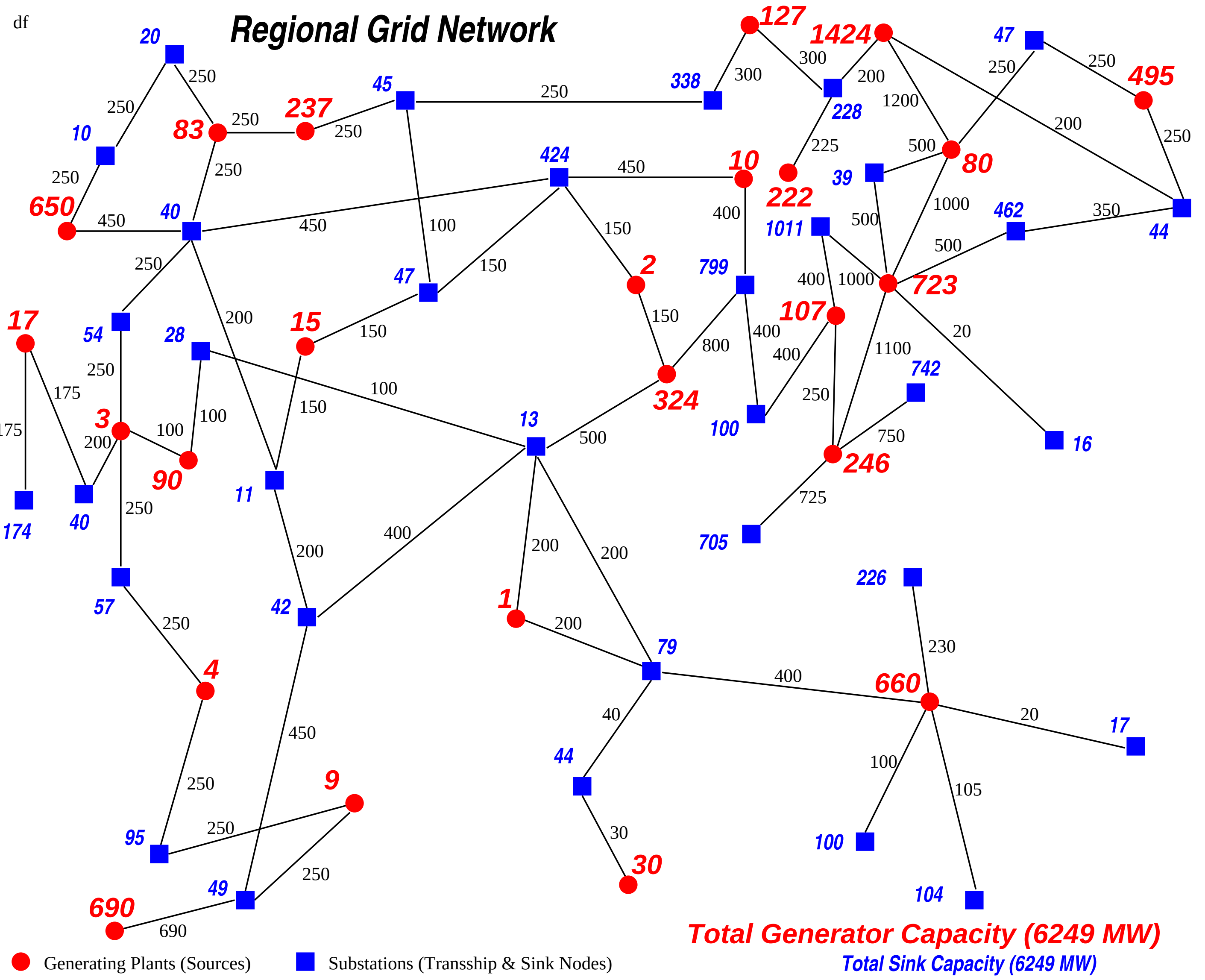


Figure 2: Shows the maximum capacities of the nodes and edges at the values used in **Scenario 2**. The positioning of the nodes and edges have not been changed to the same as the previous figure.

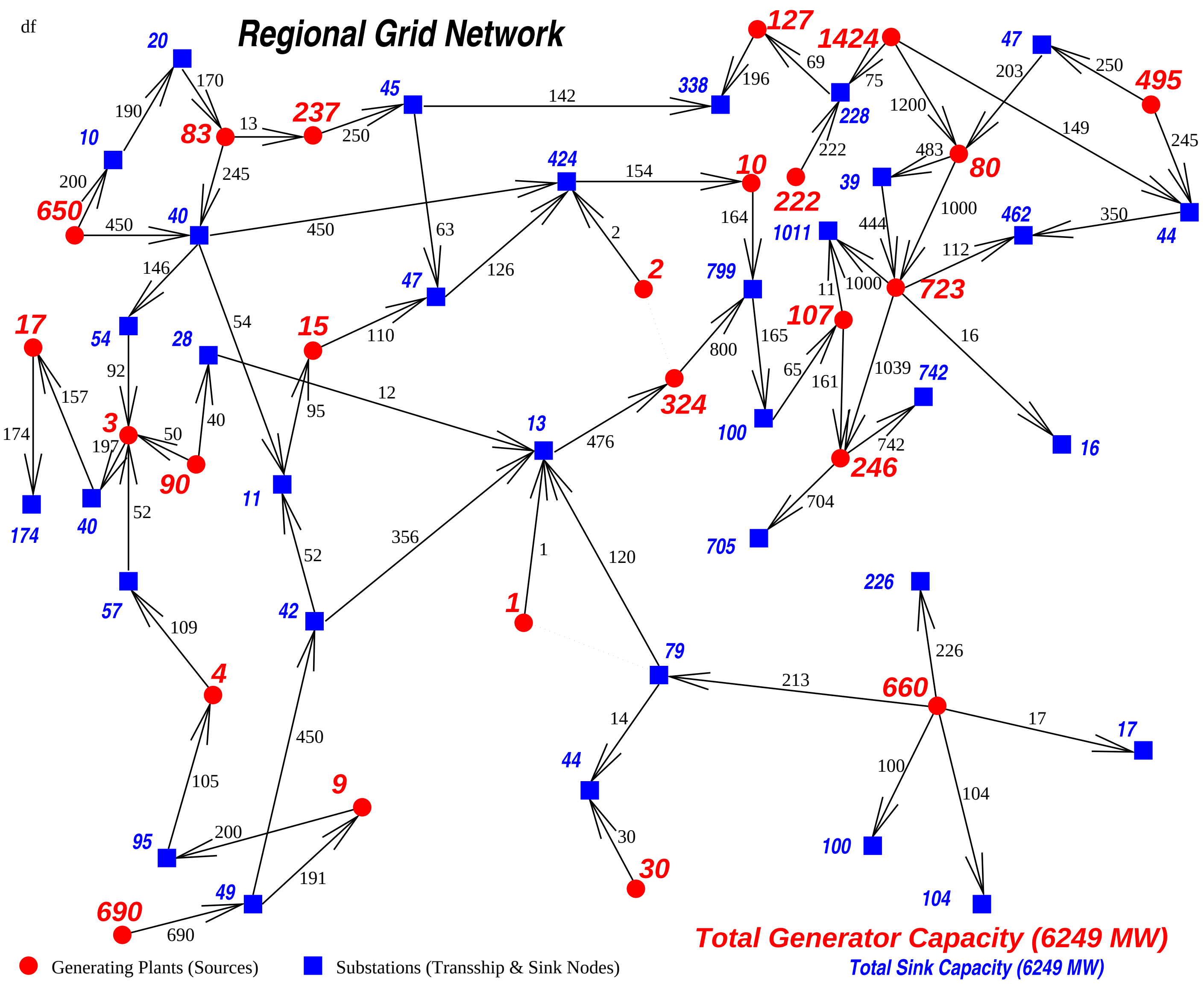


Figure 3: Shows the same network as the maximum capacities except the edges have been modified with arrows indicating direction of flow and the numbers associated with the edges are the flow values not the capacities. The edges with no flow have been changed to dotted lines although one or two of the dotted lines may look solid.

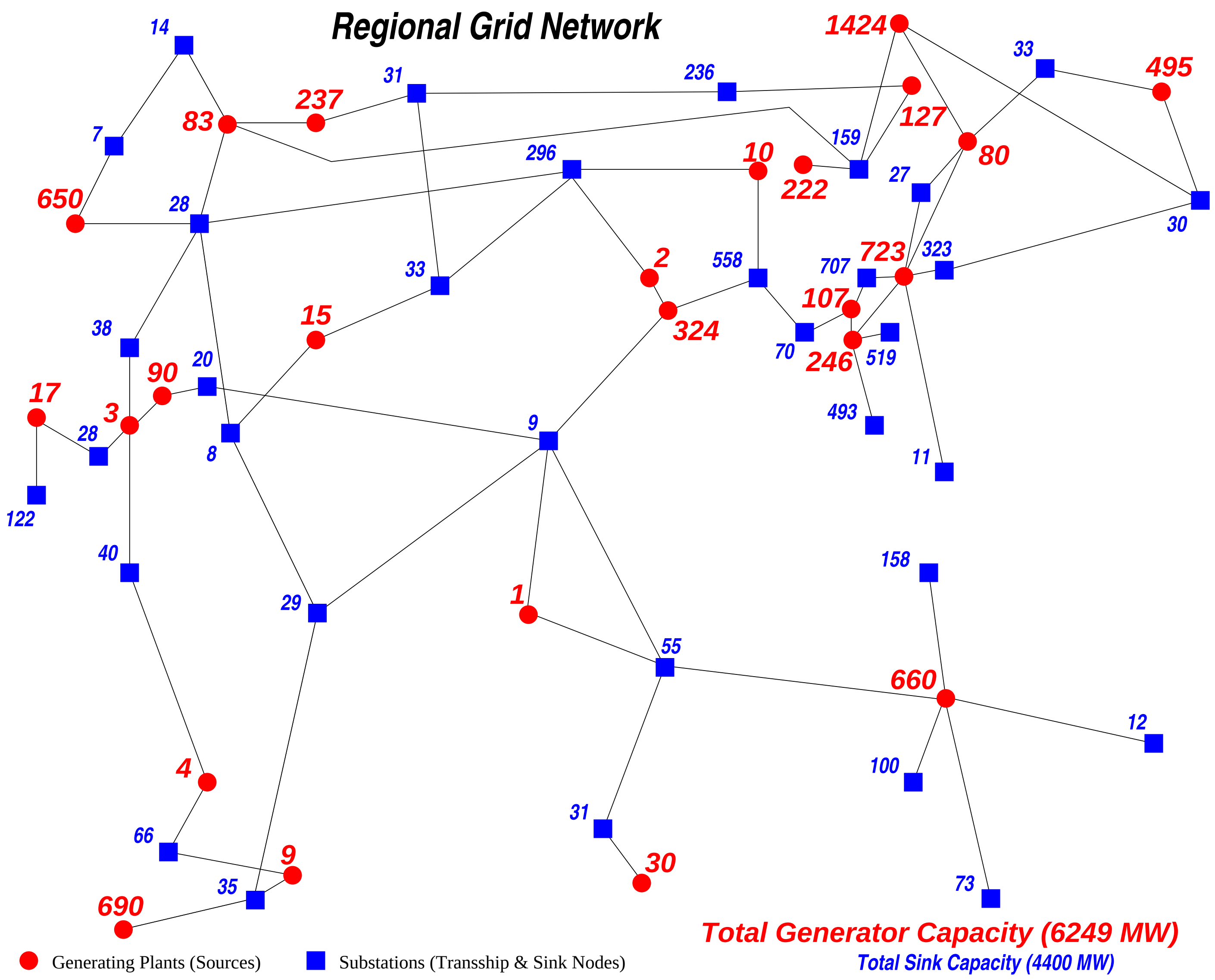


Figure 4: Shows the general network with the node capacities labeled with the sink capacities reduced to a total of 4400 MW. These are the basic capacities used in the creation of **Scenarios 1, 2, 3, & 4**.

References

[1] Ahuja, R.K., Magnanti, T.L., Orlin, J.B. Network flows: theory, algorithms, and applications. Prentice Hall, Englewood Cliffs, New Jersey. (1993)

[2] Example sites for information on bandwidth trading: http://www.lightrade.com/dowjones2.html and http://www.williamscommunications.com/prodserv/network/webcasts/bandwidth.html

[3] Cardell, J.B., Hitt, C.C., Hogan, W.W. Market Power and Strategic Interaction in Electricity Networks. Resource and Energy Economics Vol. 19 (1997) 109-137.

[4] California Public Utilities Commission. Order Instituting Rulemaking on the Commission's Proposed Policies Governing Restructuring California's Electric Services Industry and Reforming Regulation, Decision 95-12-063 (1995)

[5] Cook, D., Faber, V., Marathe, M., Srinivasan, A., Sussmann, Y.J. Low Bandwidth Routing and Electrical Power Networks. Proc. 25th International Colloquium on Automata, Languages and Programming (ICALP) Aalborg, Denmark, LNCS 1443, Springer Verlag, (1998) 604-615.

[6] Barrett, C.L., Marathe A. , Marathe, M. V. Parametrized Models for Large Scale Simulation and Analysis of Deregulated Power Markets. Internal Technical Report, Los Alamos National Laboratory (2002).

[7] The Changing Structure of the Electric Power Industry: Selected Issues. DOE 0562(98), Energy Information Administration, US Department of Energy, Washington, D.C. (1998)

[8] Fryer, H.C.: Concepts and Methods of Experimental Statistics. Allyn and Bacon Inc. (1968)

[9] G. Glass and K. D. Hopkins. Statistical Methods in Education and Psychology, 3rd ed., Allyn and Bacon, 1996.

[10] Glass G.V., Peckham P.D. and J.R. Sanders. "Consequences of failure to meet assumptions underlying the fixed-effects analysis of variance and covariance", Review of Educational Research, 42, (1972) 237-288.

[11] The U.S. Federal Energy Regulatory Commission. Notice of Proposed Rulemaking ("NOPRA"). Dockets No. RM95-8-000, RM94-7-001, RM95-9-000, RM96-11-000, April 24 (1996)

[12] Garey, M.R., Johnson, D.S. Computers and Intractability. A Guide to the Theory of NP-Completeness. Freeman, San Francisco CA (1979)

[13] Hogan, W.: Contract networks for electric power transmission. J. Regulatory Economics. (1992) 211-242.

[14] Kolliopoulos, S.G., Stein, C. Improved approximation algorithms for unsplittable flow problems. In Proc. IEEE Symposium on Foundations of Computer Science. (1997) 426-435.

[15] S-Plus5, "Guide to Statistics", MathSoft Inc. September (1988)

[16] Srinivasan, A. Approximation Algorithms via Randomized Rounding: A Survey. Lectures on Approximation and Randomized Algorithms (M. Karoński and H. J. Prömel, eds.), Series in Advanced Topics in Mathematics. Polish Scientific Publishers PWN, Warsaw.(1999) 9-71.

[17] Thomas, R. J. and T. R. Schneider. "Underlying Technical Issues in Electricity Deregulation", Power Systems Engineering Research Center, (1997). http://www.pserc.wisc.edu/ecow/get/publicatio/1997public/

[18] Wildberger, A. M. Issues associated with real time pricing. Unpublished Technical report, Electric Power Research Institute (EPRI) (1997)

[19] Websites: http://www.magicnet.net/∼metzler/page2d.html and http://www.eia.doe.gov/ (Energy Information Administration)

[20] Wolak, F. An Empirical Analysis of the Impact of Hedge Contracts on Bidding Behavior in a Competitive Electricity Market, http://www.stanford.edu/∼wolak. (2000)

[21] The Wall Street Journal, August 04 2000, A2, August 14 2000, A6, and August 21 2000, November 27 2000, December 13, 14 2000.

[22] Wood, A.J., Wollenberg, B.F.: Power Generation, Operation and Control. John Wiley and Sons. (1996)

Appendix

A Illustrative Examples

Example 1. This example illustrates the issues encountered as a result of deregulation. Figure 5(a) shows an example in which there are two power plants A and B, and two consumers. Let us assume that each consumer has a demand of 1 unit. Before deregulation, say both A and B are owned by the same company. If we assume that the plants have identical operating and production costs, then the demands can be satisfied by producing 1 unit of power at each plant. Now assume that due to deregulation, A and B are owned by separate companies. Further assume that A provides power at a much cheaper rate and thus both the consumers sign contracts with A. It is clear that both the consumers now cannot get power by A alone. Although the total production capacity available is more than total demand and it is possible to route that demand through the network under centralized control, it is not possible to route these demands in a deregulated scenario.

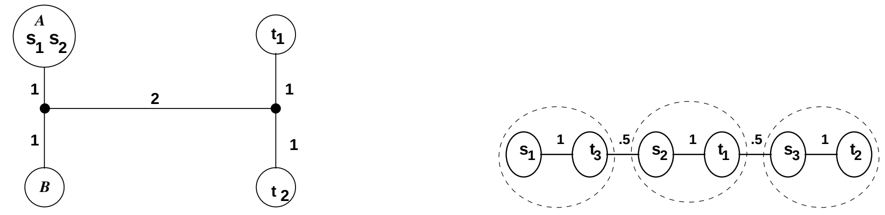
Example 2. Here, the graph consists of a simple line as shown in Figure 5(b). We have three contracts each with a demand of 1. The capacity of each edge is also 1. A feasible solution is $f(s_1, t_3) = f(s_2, t_1) = f(s_3, t_2) = 1$. The crucial point here is that *the flow originating at s_i may not go to t_i at all* — since power produced at the sources are indistinguishable, the flow from s_i joins a stream of other flows. If we look at the connected components induced by the edges with positive flow, we may have s_i and t_i in a different component. Thus we do not have a path or set of paths to round for the (s_i, t_i)-flow. This shows a basic difference between our problem and standard multi-commodity flow problems, and indicates that traditional rounding methods *may not be directly applicable*.

Example 3: In this example, we illustrate how different policies can yield different solutions. The graph is shown in Figure 6 with edge capacities as listed. Again, we have three contracts, whose details are given as follows:

1. Contract 1 – (s_1, t_1) demand $d_1 = 2$ and cost/unit $c_1 = .5$
2. Contract 2 – (s_2, t_2) demand $d_1 = 1$ and cost/unit $c_1 = 1$
3. Contract 3 – (s_3, t_3) demand $d_1 = 1$ and cost/unit $c_1 = 2$

The various solutions obtained under different policies are given below:

1. (0/1-Version, Max-Feasible Flow): Two possible solutions: (i) $f(s_1, t_1) = 2$, (ii) $f(s_2, t_2) = f(s_3, t_3) = 1$. Both solution route 2 units of flow in the network.



(a) (b)

Figure 5: Figures for Examples 1 and 2

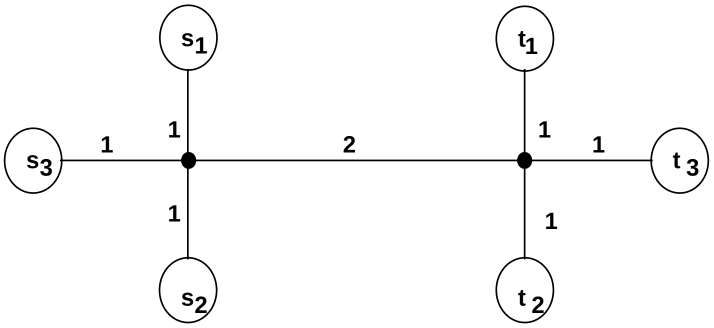


Figure 6: Example illustrating the various solutions under different contracts.

2. (0/1-VERSION, MAX-#CONTRACTS): In contrast to the previous case only one solution is possible: $f(s_2, t_2) = f(s_3, t_3) = 1$. This also routes 2 units of flow.

B Worst-Case Examples

The three heuristic methods of § 3 can be shown to have worst case performance guarantee that is $\Omega(n)$. (Recall that the performance guarantee of an approximation algorithm for a maximization problem Π is the supremum of the ratio of the optimal solution to the heuristic solution over all instances I of Π.) **Example 4** shows that all the heuristics can perform poorly w.r.t. an optimal solution. This is not too surprising given that the optimal solution gets to look at all of the input before clearing the contracts.

Example 4: Consider a network with two nodes A and B joined by an edge

(A, B). The capacity of the edge (A, B) is $C < 1$. There is an even number n of contracts $(s_1, t_1), \ldots, (s_n, t_n)$. Odd-numbered contracts have demand of 1 unit and the sources and sinks of these contracts are distributed as follows: source-nodes $s_1, s_3, \ldots s_{n-1}$ are located at node A and their corresponding consumers $t_1, t_3, \ldots t_{n-1}$ are located at B. Let us call this set *Odd-Set*. For the even numbered contracts (denoted *Even-set*) we have a demand of $1 + \frac{C}{2n}$ per contract and the source sink locations are reversed: the sources are located at B and the sinks at A. Note that

1. All *Odd-set* contracts have demand that is less than every contract in *Even-Set*.

2. In the absence of any other contracts, only one *Odd-set* contract can be cleared; similarly, exactly one *Even-set* contract can be cleared.

Now consider how many contracts can be satisfied by the each of three heuristic methods.

1. SMALLEST-FIRST HEURISTIC will clear only one contract (s_1, t_1).

2. LARGEST-FIRST HEURISTIC will also clear exactly one contract (s_2, t_2).

3. RANDOM-ORDER HEURISTIC will also perform poorly with high probability. This is because there is are a total of $n!$ ways to arrange the contracts and roughly only $O((\frac{n}{2})!(\frac{n}{2})!)$ good ways to do it.

4. An optimal solution can clear all the contracts simultaneously, since the flows from *Odd-set* appropriately cancel the flows from *Even-Set*. Thus the performance guarantee of the SMALLEST-FIRST HEURISTIC and LARGEST-FIRST HEURISTIC is $\Omega(n)$. The performance guarantee of RANDOM-ORDER HEURISTIC is also $\Omega(n)$ with high probability.

Example 5: Again, we have a single edge as the network. Denote the edge by (A, B) as before, with the endpoints being A and B respectively and the edge capacity being 1. We have n contracts. As before we divide them into *Even-Set* and *Odd-set* of contracts. The contracts' demands are strictly increasing: the i^{th} contract has demand $1 + (i-1)\epsilon$. The value ϵ is chosen so that $0 < \epsilon < 1$ and $(n-1)\epsilon > 1$. It is clear that SMALLEST-FIRST HEURISTIC can clear all the contracts, while LARGEST-FIRST HEURISTIC can clear exactly one contract. Again, a simple calculation shows that RANDOM-ORDER HEURISTIC will perform poorly with high probability.