Presented at the 19th Mini-Euro Conference on Operation Research Models and Methods in the Energy Sector, Coimbra, Portugal, 6-8 September, 2006

Evaluation of Open-Source LP Optimization Codes in Solving Electricity Spot Market Optimization Problems

Stuart R. Thorncraft
UNSW, Sydney NSW, s.thorncraft@student.unsw.edu.au

Hugh R. Outhred
UNSW, Sydney NSW, h.outhred@unsw.edu.au

David J. Clements
UNSW, Sydney NSW, d.clements@unsw.edu.au

ABSTRACT. The purpose of this paper is to provide an evaluation of the three open-source LP optimization codes, GLPK, COIN-LP (CLP) and LPSOLVE in solving electricity spot market optimization problems. The paper describes a technique for automatically generating over 100 locational marginal pricing (LMP) electricity spot market optimization problems, each being solved using the three optimization codes. Performance measures were then constructed using measurements of the CPU time spent in solving each optimization problem and the CPU time spent in each optimization code's interface. These measurements were used as the basis for performance characteristics for comparing the relative merits of each optimization code.

KEYWORDS. Energy markets; linear programming; benchmarking; open-source software.

1. INTRODUCTION

Given the availability of a number of open-source linear programming (LP) optimization codes including GLPK, COIN-LP (CLP) and LPSOLVE, it is useful to investigate their performance in solving electricity spot market optimization problems. The purpose of this paper is to develop an approach to evaluate these optimization codes for locational marginal pricing (LMP) electricity spot market optimization problems. Such problems contain features that are common to the electricity market models that have been implemented in many restructured electricity industries around the world. Thus, the results provide an initial step in the investigation of the suitability of the optimization codes for the purpose of electricity spot market dispatch and pricing.

2. ELECTRICITY SPOT MARKET OPTIMIZATIONS

2.1. Characteristics of electricity spot market models used in practice

Restructured electricity industries are generally governed by a set of electricity market rules, which in turn give rise to a specific electricity spot market optimization problem formulation that is used to simultaneously price and schedule resources. The optimization problems are solved on a periodicity between 5 to 60 minutes and form part of the overall function of power system operation. Thus, the time required to compute a solution is extremely important. The optimization input and output data is usually transferred to or from control systems, security assessment programs, online decision-support tools and to processes that communicate the outcomes to electricity market participants. This means that in addition to the time spent solving the optimization problem the software's ability to create an instance of the problem as well as retrieving its solution is also important.

Different electricity industries have different electricity spot market models. Some of the features that vary across different electricity industries include:

- The degree of granularity used for representing the electrical network. The network model may be approximated using a regional representation of the network such as the single region model used in the Ontario market (IESO, 2005) or the six-region model used in the Australian National Electricity Market (NEM) (NEMMCO 2005). Alternatively, the network model may be modeled in more detail and comprise a large number of buses such as the model used in the New Zealand Electricity Market (NZEM) (Alvrey et al. 1998), the National Electricity Market for Singapore (NEMS) (EMA 2006) or the Pennsylvania, New Jersey, Maryland Interconnection (PJM) in the mid-Atlantic region of the US east coast (PJM 2006);
- Complexity of the network model which ranges from simple radial systems, highly meshed systems or hybrids of these;
- Numbers of generators and dispatchable loads on the system; including ramping and other physical limitations as well as generation offer functions and/or load bid functions;
- Approach in modeling network losses;
- Inclusion of models for acquisition of ancillary service providers and for establishing prices for ancillary services; and
- Inclusion of security and other generic constraints to reflect the present state of the electricity system.

To assess the suitability of the open-source optimization codes for use in a number of electricity industries, this paper will restrict attention to only the first four of these items and will construct test cases for varying problem sizes.

2.2. Optimization formulation

A simplified formulation that includes only the fundamental aspects of an electricity spot market optimization is as follows. Complete formulations of electricity spot market optimizations are detailed in EMA 2006 or IESO 2005.

Minimize:

$$J = \sum_{b \in B_E} \sum_{i \in G} G_p(i,b) P_g(i) + B_p \sum_{b \in B} \delta(b) + C_p \sum_{b \in B} L_e(b) + C_p \sum_{b \in B} L_d(b) \qquad \ldots\ (1)$$

where B_E is the set of generator offer bands, G is the set of generators that participate in the electricity market, B is a set of network buses, J is the value of objective function which reflects the benefit of trade ($/h), $G_p(i,b)$ the generator i's price offer for band b ($/MWh), B_p a (very small) penalty cost associated with bus angles ($/rad-h), C_p a (large) penalty cost associated with not being able to satisfy the power balance constraint ($/MWh), $\delta(b)$ is a variable corresponding to the angle for bus b (rad), $L_e(b)$ and $L_d(b)$ correspond to excess and deficit supply (MW) - they are used to detect whether the optimization has failed to satisfy the energy balance at a given node thus indicating to the system operator that there is a need to take measures to manage system security.

Subject to the following constraints:

$$L_e(b) \geq 0, L_d(b) \geq 0, \delta(b) \geq 0, b \in B \qquad \ldots\ (2)$$

$$F_{\min}(k) \leq f(k) \leq F_{\max}(k), k \in L \qquad \ldots\ (3)$$

where L is the set of lines, $F_{min}(k)$ is the minimum flow on line k (MW), $F_{max}(k)$ the maximum flow on line k (MW) and $f(k)$ is the flow decision-variable on line k (MW). Equation (2) restricts the excess, deficit and bus angles to being positive and equation (3) ensures that line flows remain within flow limits.

$$B_{inj}(b) = \sum_{i \in G_b} P_g(i) + L_d(b) - L_e(b) - D_N(b), b \in B \qquad \ldots\ (4)$$

$$B_{inj}(b) = \sum_{(k,b) \in F} f(k) - \sum_{(k,b) \in T} f(k) + \frac{1}{2} \sum_{(k,b) \in F \cup T} P_{loss}(k), b \in B \qquad \ldots\ (5)$$

where G_b the set of generators at bus *b*, *F* is the set such that *(l,b)*$\in$*F* implies that line *l* leaves bus *b*, *T* is the set such that *(l,b)* $\in$*T* implies line *l* enters bus *b*, $B_{inj}(b)$ is a decision-variable that is introduced to simplify the power balance constraint by breaking it into two equations, $P_g(i)$ is generator *i*'s generation target (MW), $D_N(b)$ is the net demand (forecast) at bus *b* (MW), $P_{loss}(k)$ are the losses on line *k* (MW). Equations (4) and (5) reflect the power balance at each bus on the network.

$$f(k) = S_0 B_{sus}(k)\big(\delta(b_f) - \delta(b_t)\big), k \in L, (k, b_f) \in F, (k, b_t) \in T \qquad \ldots\ (6)$$

$$P_{loss}(k) = \frac{R(k)}{S_0} f(k)^2, k \in L \qquad \ldots\ (7)$$

where S_0 is the MVA base, $B_{sus}(k)$ is the susceptance of line *k* (pu) and *R(k)* is the resistance of line *k* (pu). Equation (6) is the DC power flow equation where line flows are linearly proportional to bus angle differences while equation (7) represents the line losses as a quadratic function of network line flows.

$$0 \le P_g(i) \le G_{\max}(i), i \in G \qquad \ldots\ (8)$$

$$G_I(i) - G_{rd}(i) T_s \le P_g(i) \le G_I(i) + G_{ru}(i) T_s, i \in G \qquad \ldots\ (9)$$

where $G_{max}(i)$, is the maximum output of generator *i* (MW), $G_i(i)$ is the initial condition for generator *i* (MW), $G_{rd}(i)$ is generator *i*'s ramp down rate (MW/min), $G_{ru}(i)$ is generator *i*'s ramp up rate (MW/min), T_s is the spot market period (min). Equation (8) reflects the upper limit to generation while equation (9) represents the limited ramping ability of each generator.

2.3 Lossless Model

In the lossless model, $P_{loss}(k)$ is set to zero and excluded from the optimization problem. The optimization therefore becomes linear and equation (7) is removed.

2.4. Lossy Model

In the lossy model, the quadratic losses are replaced by a set of linear constraints that approximate the quadratic losses. The linearization procedure and linear equations are described in EMA 2006.

3. OPEN-SOURCE OPTIMIZATION CODES

Open source software continues to provide an appropriate alternative to commercial software and proceeds to improve both in terms of the quality of individual applications and in terms of scope. This is especially true in the fields of numerical computing and operations research where many algorithms and codes are available to researchers and practitioners. In recent years, a number of open-source software optimization packages have become well established and proceed to be improved upon. This paper considers a subset of the available open-source optimization codes including GLPK, COIN-LP (CLP) and LPSOLVE. These codes are of interest because they have large user groups and have been used in a wide variety of research and commercial applications. However, results assessing their suitability in solving the problems that typically arise in the electricity industry are not available, thus it is of interest to explore the extent to which they can assist the electricity industry.

3.1. GNU Linear Programming Kit (GLPK) version 4.9

GLPK version 4.9 (GLPK 2006) is an ANSI C implementation of the two-phase simplex algorithm, the primal-dual interior point algorithm and the branch-and-bound method for solving mixed integer programs. It is intended to solve large-scale problems. GLPK provides an optional presolver, which transforms the problem into one that has better numerical properties for the simplex algorithm; this is particularly useful for large-scale problems.

3.2. COIN-LP (CLP) version 1.02.02

COIN-OR (Computational Infrastructure for Operations Research) is a collection of peer reviewed open source software (COIN-OR 2006). It was launched by IBM research in 2000, and from late 2004 has been run by the independent, non-profit COINOR foundation (Lougee-Heimer 2005). CLP is essentially a C++ implementation of the primal and dual simplex methods; however, there is also an implementation of the barrier method in addition to support for solving quadratic programs.

3.3. LPSOLVE version 5.5.0.6

LPSOLVE is a mixed integer LP solver, based on the revised simplex method and the branch-and-bound method for integers (LPSOLVE 2006). It solves pure linear, (mixed) integer/binary, semi-continuous and special ordered sets (SOS) models. As with the other optimization codes LPSOLVE provides a presolver.

4. EVALUATION METHODOLOGY

4.1. Performance measures

This paper focuses on evaluating the capability of the open-source optimization software in terms of its ability to solve linear electricity spot market optimization problems as a function of the problem size. Problem size is expressed in terms of the number of generators, network buses and transmission elements for the case where losses are not modeled and the case where a linearized loss model is included.

The following performance measures were examined:

- CPU time spent solving the optimization problem; and
- CPU time spent setting up the problem and retrieving its solution from the optimization code's interface (termed the 'interface time').

4.2. Evaluation procedure

The following procedure was used to identify suitable test cases and to evaluate the optimization codes:

1. Generate a database comprising a very large set of potential test network models based on the application of the algorithm outlined in section 4.3. From the database of potential test cases, a set of specific test cases was selected for the line-bus ratios of 1, 2, 5 and 7. The specific set of test cases that was used is summarized in Table 1, which lists the line bus ratio (N_l/N_b) corresponding numbers of network buses (denoted N_b) and number of generators (denote N_g) and loss model types. Note that N_l is the number of network lines and can be inferred (approximately) from the line-bus ratio.

2. For each test case and for each optimization code, the problem was set up and solved 3 times with the average being used in the performance profiles. The specific algorithm and options for each optimizer was set to be as similar as possible (to ensure a fair comparison). So for each solver the primal simplex algorithm was used, presolve capability was enabled and the solver time-limit was set to 15 minutes.

   The hardware used for the exercise was an Intel(R) Pentium(R) 4 CPU 2.66GHz with 512Mb RAM running on the Windows 2000 operating system. Clearly, the hardware is not comparable to what could be used in a practical situation; however, the results enable the relative performances to be assessed and also provide a lower bound on the performance that might be expected using more advanced hardware.

3. While each optimization test case was solved, the following were measured: the CPU time spent using the optimization code's interface to specify the problem, the CPU time spent computing the solution to the optimization problem and the CPU time spent in the optimization code's interface to retrieve the problem solution. The first and third of these were summed and termed the *interface time*. Failed or timed-out solve attempts were flagged appropriately.

   The CPU timings allow performance characteristics to be constructed for each optimization code. These are detailed in section 5.1.

Table 1. Summary of test cases.

Test Case Number	**Approximate N_l / N_b**	**Values for N_b**	**Values for N_g**	**Type of loss model**
1-15	1	57, 207, 393, 777, 993	100, 300, 500	Lossless, using the model referred to in section 2.3
16-30	2	52, 200, 393, 747, 909	100, 300, 500	
31-45	5	51, 207, 396, 772, 927	100, 300, 500	
46-57	7	204, 402, 524, 653	100, 300, 500	
58-72	1	57, 207, 393, 777, 993	100, 300, 500	Lossy with 30 break points, using the model referred to in section 2.4
73-87	2	52, 200, 393, 747, 909	100, 300, 500	
88-102	5	51, 207, 396, 772, 927	100, 300, 500	
103-114	7	204, 402, 524, 653	100, 300, 500	

4. Finally, an overall performance characteristic for each solver across all of the problems was constructed. An outline of the methodology is provided in section 4.4 with the results being presented in section 5.2.

4.3. Algorithm for automatic test case generation

Because it is of interest to investigate how the optimization codes behave for different sized electricity networks, it is convenient to automatically construct networks of a prescribed degree of complexity. The following algorithm is a way of constructing different networks with different levels of complexity.

Let S_m be the number of algorithm steps used to construct a meshed portion of network, N_m a control parameter to govern how highly meshed the network is, S_r the number of algorithm steps used to construct a radial portion network and N_r a control parameter to govern how radial the network is. Finally, let $A(S_m, N_m, S_r, N_r)$ denote a network generated by the following algorithm:

- Step 1:

  a) If $S_m = 0$, then go to step 3;

b) Otherwise initialize mesh network to be a loop comprising 3 buses and set the assign E_b (a set of buses) equal to the 3 buses;

c) For each S_m >= 2 do step 2;

d) Go to step 3.

- Step 2:

1) Add $N_e +1$ (where $N_e = ||E_b||$, the number of edge buses) buses to the network and connect these new buses with lines to form a ring;

2) For each bus in the set of edges construct N_m connections to buses chosen successively from the set of the $N_e + 1$ newly added buses;

3) Replace the set of edges with the $N_e + 1$ newly added buses.

- Step 3:

1) If $S_r <= 0$ then finished;

2) If $S_m = 0$ then initialize by adding N_r buses and connecting them linearly with $N_r – 1$ lines (each bus is added to the set of edges);

3) For each bus in the set of edges connect N_r buses;

4) Replace the set of edges with the N_r newly added buses.

- Step 4:

1) N_g generators (only) are added to the problem by taking a total capacity and dividing it among the desired number of generators, the generators are assigned to alternating buses with the others being assigned as load buses; and

2) Each generator offers its capacity using 10 offer bands and has reasonable ramp rate rates and initial conditions assigned to it.

Three examples to illustrate the constructing a radial network, the construction of a meshed network and the construction of a mesh-radial hybrid are illustrated in Fig 1.

4.4. Determining overall performance characteristic

A way of addressing the issue of succinctly summarizing the overall performance of optimization codes is suggested by Dolan et al. 2002. The approach defines the following baseline for comparisons:

$$r_{p,s} = \frac{T_{p,s}}{\min\{T_{p,s} : 1 \le s \le N_s\}} \qquad \text{... (10)}$$

where $r_{p,s}$,is the performance ratio of solver s for problem p, $T_{p,s}$ is the CPU time spent on some activity in the optimization software and N_s is the total number of optimizers (3).

If solver s fails to solve problem p then we set $r_{p,s} = P_M$ where $P_M \ge r_{p,s}$ for all p, s. Using this definition, the overall assessment of solver performance is defined to be:

$$r_s(\tau) = \frac{1}{N_p} \left\| \{ p \in P : r_{p,s} \le \tau \} \right\| \qquad \text{… (11)}$$

where $r_s(\tau)$ is the cumulative distribution function for the performance ratio (reflecting the probability that performance ratio $r_{p,s}$ is within a factor τ of the best possible ratio), N_p is the number of problems and P is the set of all problems.

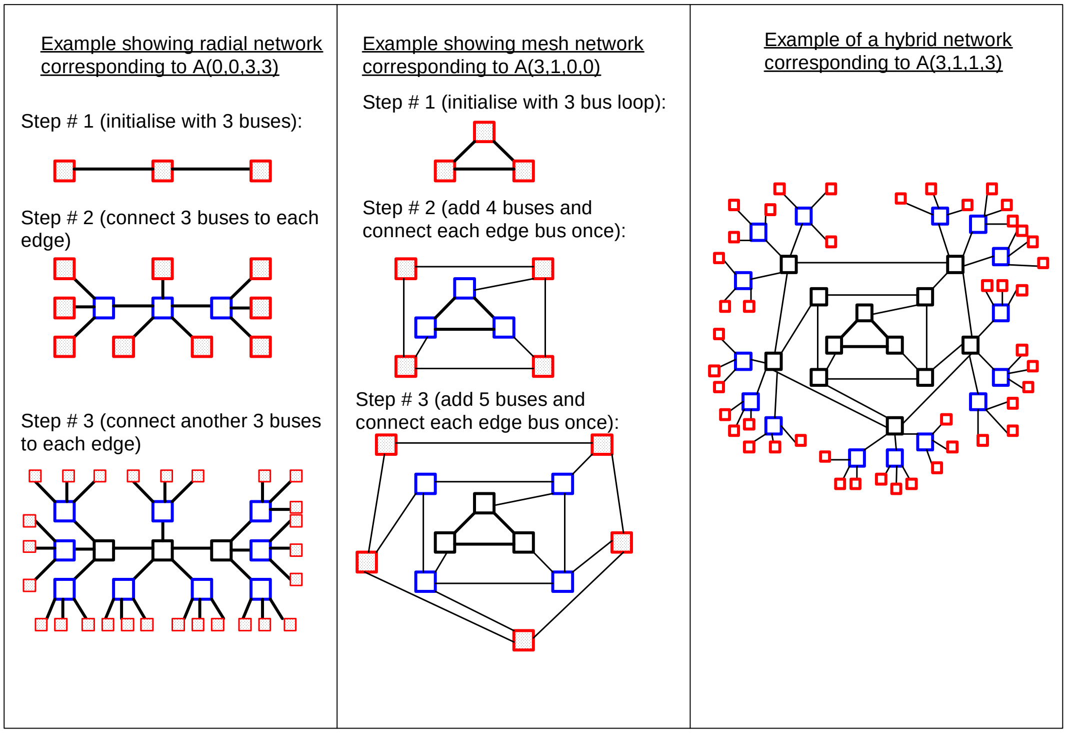
Example showing radial network corresponding to A(0,0,3,3)
Step # 1 (initialise with 3 buses):
Step # 2 (connect 3 buses to each edge)
Step # 3 (connect another 3 buses to each edge)
Example showing mesh network corresponding to A(3,1,0,0)
Step # 1 (initialise with 3 bus loop):
Step # 2 (add 4 buses and connect each edge bus once):
Step # 3 (add 5 buses and connect each edge bus once):
Example of a hybrid network corresponding to A(3,1,1,3)

Figure 1. Illustrations of the network generating algorithm for A(0,0,3,3), A(3,1,0,0) and A(3,1,1,3).

5. BENCHMARKING RESULTS

5.1. Sample of solve times as a function of network size for lossless models

Fig 2. shows the solve time and interface time performance characteristics for the three optimizers using data collected from solving test cases 31-45 from Table 1. Those plotted are for the case when the line-bus ratio is 5 and N_g =500 (the number of generators). While it is not shown in Fig 2., the observation was made that the effect of varying N_g over the values 100, 300, 500 did not make a significant difference to the basic shapes of the performance characteristics, hence we only consider results for N_g = 500.

Clearly, CLP out-performs GLPK and LPSOLVE for a large number of buses. It is also apparent that GLPK performs similarly to LPSOLVE for a small number of buses, however, as the number of buses in the optimization problem are increased, GLPK solve times begin to out-perform LPSOLVE. Fig. 2 suggests that both CLP and GLPK have similar interface time performance characteristics with the latter being fractionally faster while LPSOLVE spends more time in the interface.

5.2. Solve times as a function of network size for lossy models

For models that incorporate electrical losses, it was observed that the number of generators had very little effect on the optimization solve times or interface times. This is because each line on the network introduces significantly more constraints and variables (in total) compared to those used for modeling the generators.

Fig. 3 shows the solve times and interface times based on the results corresponding to test cases 73-87 from Table 1 for line-bus ratios of 2. The diagram shows that CLP continues to solve the problems in the shortest CPU times, followed by GLPK and then LPSOLVE. Fig. 4 shows the solve times for test cases 88-102 with line-bus ratio of 5. This confirms that the trend continues; that is, CLP solves the optimizations in the shortest periods of time, followed by GLPK and then LPSOLVE. Both GLPK and LPSOLVE exceeded the 15-minute time limit for higher numbers of network buses, explaining the truncation of results.

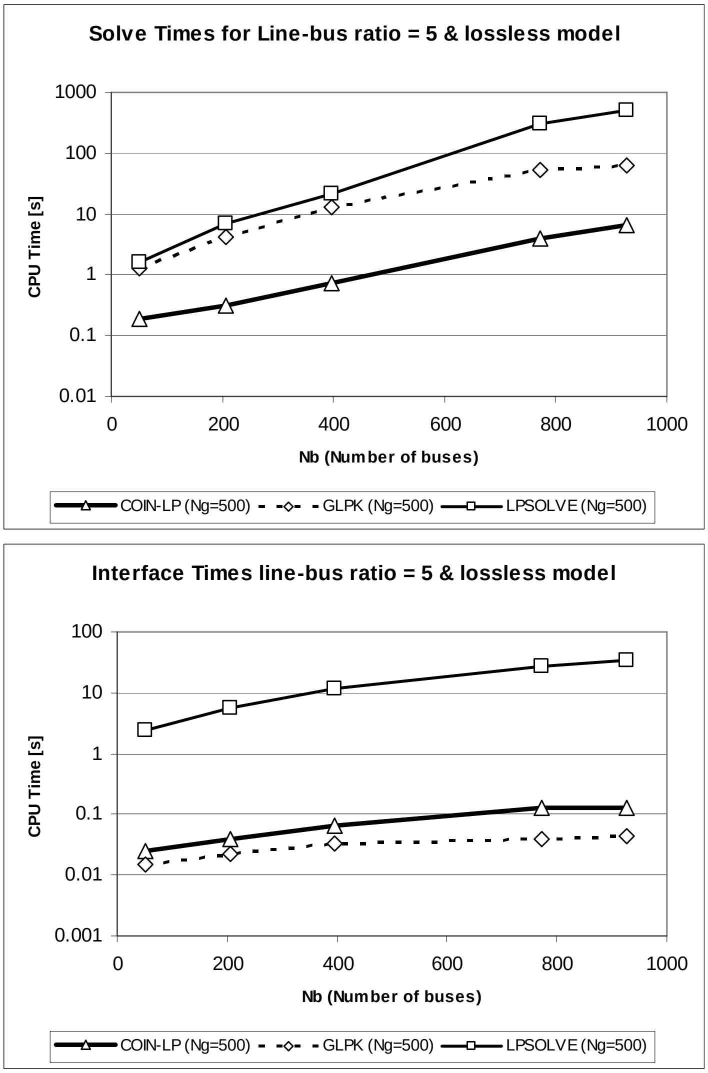

Figure 2. Solve & interface times for line-bus ratio = 5, Ng = 500 and using lossless network model

Fig. 3 also illustrates that the interface times for both GLPK and CLP flatten out and comprise an overhead of less than 1 second for a line-bus ratio of 2. It was observed that for a line-bus ratio of 5 the interface times were less than 10 seconds for CLP and 1 second for GLPK. LPSOLVE's interface time is similar to the amount of time it spends solving the optimization problem, which is a significant overhead. A more detailed examination of the parts of the LPSOLVE interface that consume the most time reveals that adding rows to the LP matrix is a relatively expensive operation; thus, a more efficient way of loading the problem into LPSOLVE would need to be investigated to reduce this overhead.

The results for the other test cases from Table 1 have similar characteristics to the cases that have been discussed in detail.

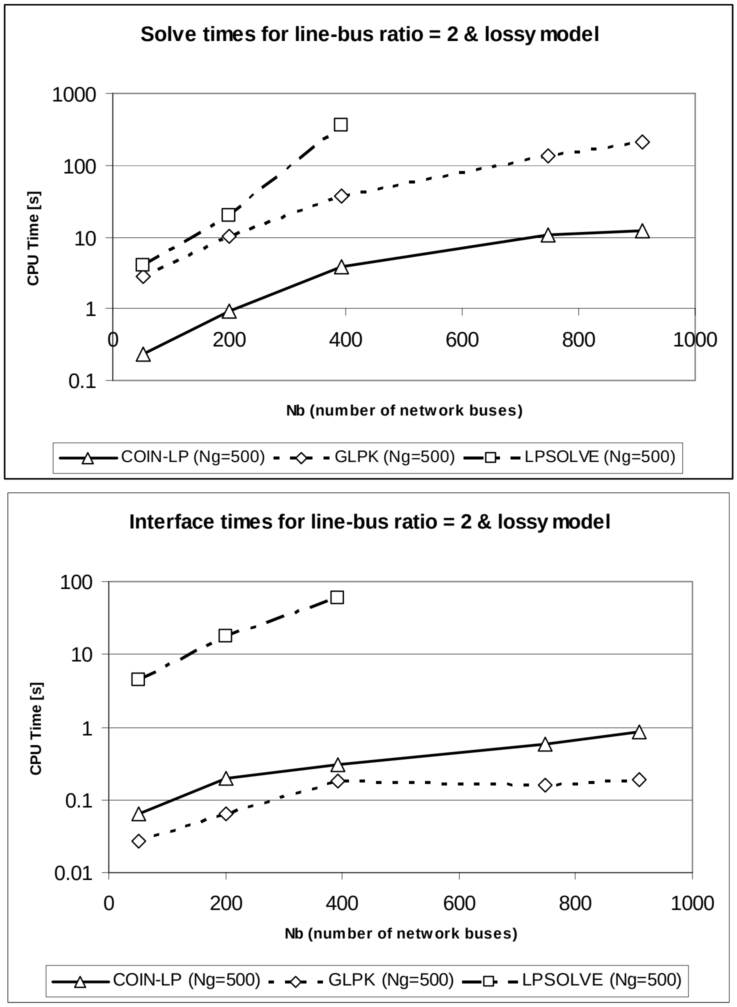

Figure 3. Solve & interface times for line-bus ratio = 2, Ng = 500 and using the lossy network model

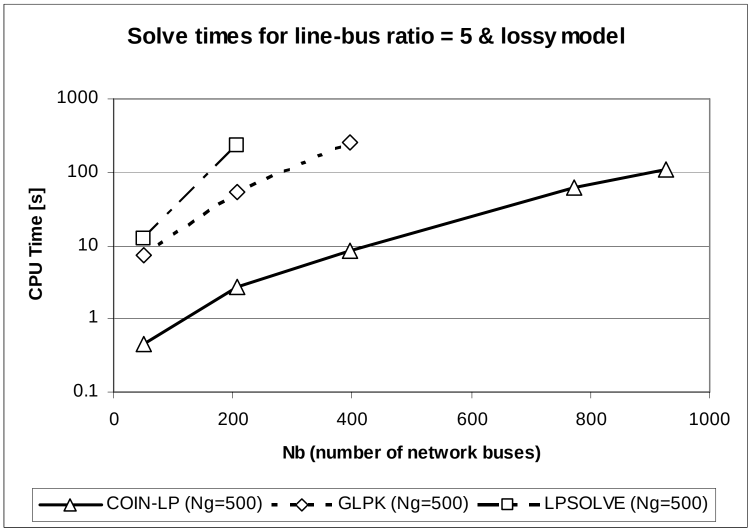

Figure 3. Solve times for line-bus ratio = 5, Ng = 500 and using the lossy network model

5.3. Overall performance profiles

The overall performance profiles for the solve times are displayed in Fig. 5. These were constructed using CPU time measurements from solving the full set of 114 test cases described in Table 1. Recall that the performance metric, $r_s(\tau)$ represents the fraction of problems that the optimization code was able to solve within a fraction τ of the best performing solver. On this basis, CLP out-performs GLPK and LPSOLVE across the full set of test problems. The graph also shows that GLPK consistently out performed LPSOLVE. Fig. 4 shows that GLPK and LPSOLVE were unable to solve all of the problems within the allotted time limit.

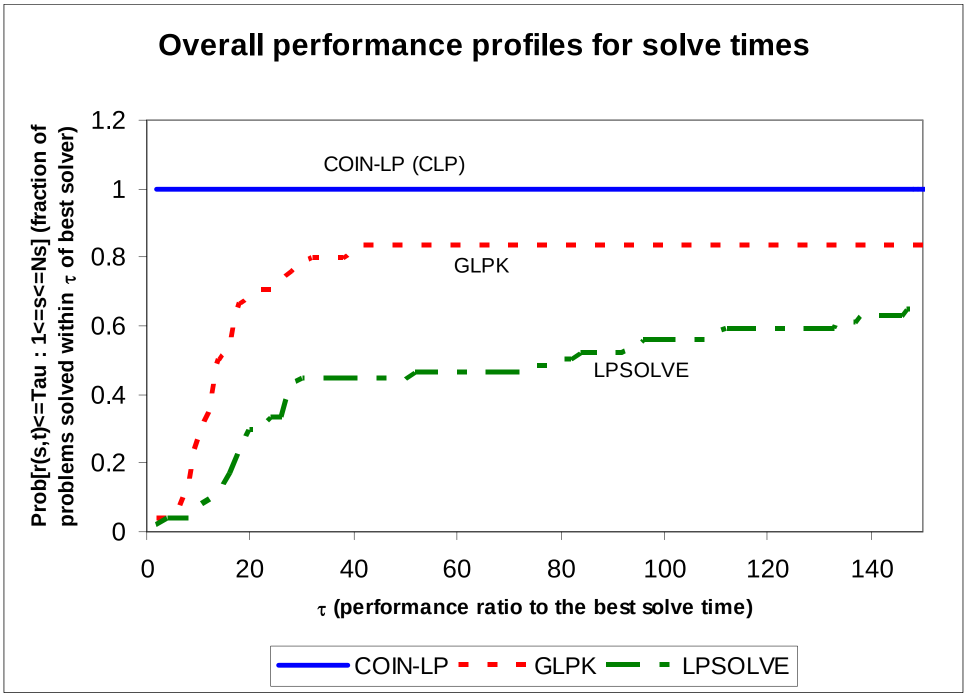

Figure 5. Overall performance profile across all test cases for each solver

6.CONCLUSIONS

This paper presented a technique to evaluate the open-source optimization codes: GLPK, COIN-LP (CLP) and LPSOLVE in terms of their ability to solve electricity spot market optimization problems. The benchmarking process showed that in terms of the CPU time spent solving the optimization problems; CLP performed the best for both small-scale and large-scale problems, followed by GLPK and then LPSOLVE. In terms of interface time, the benchmarking process indicated that GLPK and CLP spent similar CPU times that comprised only a minor overhead compared with the solve time. Consequently the study concludes that the CLP solver shows the highest potential in being able to satisfy the requirements for solving electricity spot market optimization problems. While this paper has only benchmarked a specific class of electricity optimization problem, it provides an initial starting point in terms of investigating the feasibility for a system or market operator to utilize open-source optimization codes for the purpose of electricity spot market dispatch and pricing.

REFERENCES

Alvrey, T, Goodwin D, Ma X, Streiffert D, Sun D, ‘A Security-Constrained Bid-Clearing System for the New Zealand Wholesale Electricity Market’, IEEE Transactions on Power Systems, vol. 13., No. 2, May 1998.

COIN-OR, www.coin-or.org, accessed: January 2006.

Dolan ED, More JJ, 'Benchmarking Optimization Software with Performance Profiles', Mathematical Programming, 2002, 91:201-213, available: www-unix.mcs.anl.gov/scidac-tops.

EMA, 'Singapore Electricity Market Rules', January 2006, available: www.emcsg.com.

GLPK, GNU Linear Programming Kit, version 4.9, www.gnu.org/software/glpk/glpk.html, accessed: January 2006.

IESO, 'Market Rules for the Ontario Electricity Market', March 2005, available: www.ieso.ca.

Lougee-Heimer R, 'COIN-OR Pays Off', Informs, OR/MS Today, Vol. 32, Number 5, October 2005.

LPSOLVE, 'Reference Guide', 2006, available: www.geocities.com/lpsolve.

NEMMCO, 'An Introduction to Australia's National Electricity Market', June 2005, available: www.nemmco.com.au.

PJM, 2006, www.pjm.com.