Open access • Journal Article • DOI:10.1109/TPWRS.2013.2287880

Comparison of Mixed-Integer Programming and Genetic Algorithm Methods for Distributed Generation Planning — Source link

James D. Foster, Adam Berry, Natashia Boland, Hamish Waterer

Institutions: University of Newcastle, Commonwealth Scientific and Industrial Research Organisation

Published on: 01 Mar 2014 - IEEE Transactions on Power Systems (Unpublished internal technical report)

Topics: Integer programming, Knapsack problem, Genetic algorithm, AC power and Integer (computer science)

COMPARISON OF MIXED-INTEGER PROGRAMMING AND GENETIC ALGORITHM METHODS FOR DISTRIBUTED GENERATION PLANNING

J.D. FOSTER, A.M. BERRY, N. BOLAND, AND H. WATERER

Technical Report

Abstract. This paper applies recently developed mixed-integer programming (MIP) tools to the problem of optimal siting and sizing of distributed generators in a distribution network. We investigate the merits of three MIP approaches for finding good installation plans: a full AC power flow approach, a linear DC power flow approximation, and a nonlinear DC power flow approximation with quadratic loss terms, each augmented with integer generator placement variables. A genetic algorithm based approach serves as a baseline for the comparison. A simple knapsack problem method involving generator selection is presented for determining lower bounds on the optimal design objective. Solution methods are outlined, and computational results show that the MIP methods, while lacking the speed of the genetic algorithm, can find improved solutions within conservative time requirements and provide useful information on optimality.

1. Introduction

Distribution networks equipped with distributed generation are increasingly common [1]. New concepts in distributed generation are being tested at experimental facilities the world over [2]. Our problem, which we refer to as the *Distributed Generation Planning Problem* or DGPP, is the following: given an existing power system and a budget constraint, determine an placement of generators chosen from various types of distributed generation technologies in order to minimise the network's power deficit. Stated another way, our objective is to minimize the power deficit (that is, the power imported at the slack bus) by specifying a placement of new generators, thereby ensuring the network has the smallest possible dependency on external energy sources. A solution to the DGPP is both a placement *design* and a list of voltages feasible for the network with the new design installed.

We simplify the problem by using steady-state power systems and considering peak (worse-case) loads within a single time period, and generators are characterised by their peak generation capacity and cost, the latter covering both the installation and maintenance. The question is then: what is the greatest benefit obtainable from a single investment in new generators? We assume that the relatively small size of the generators allows any proposed design to be built over a short enough time period such that the expected peak power demand does not significantly change.

Date: July 20, 2012.

This work was supported in part by the University of Newcastle Vice-Chancellor's Scholarship and the CSIRO Studentship scheme.

A. M. Berry is with the Secure Grids project in the Energy Technology division of CSIRO, Mayfield West, NSW, Australia.

N. Boland, J. D. Foster (email: james.foster@uon.edu.au) and H. Waterer are with the Operations Research group of the School of Mathematical and Physical Sciences, University of Newcastle, Callaghan, NSW, Australia.

In this paper we compare three solutions to the DGPP found by mixed-integer linear programming (MILP) and mixed-integer nonlinear programming (MINLP) approaches. The MILP approach uses the well-known DC linear approximation model [3]. One MINLP approach uses the full nonlinear AC power flow model, while another MINLP model is similar to the DC model albeit with quadratic line loss terms [4–6]. A baseline of comparison for these approaches is provided by a genetic algorithm (GA) [7]. All approaches are tested against a diverse set of networks and assessed relative to a lower bound on the global optimum found by an efficiently solvable MILP knapsack problem obtained by deleting the power flow constraints.

Our contributions lie in the formulation of the DGPP in the language of mixed-integer nonlinear optimization and exploiting the resulting model structure to analyse the optimality of this difficult problem. There is a good fit since the design variables are naturally integer-valued. While distribution expansion planning by MIP techniques has been used extensively right up to the present [8–11], recent work has formulated similar problems as a continuous optimal placement and sizing problem and solved with a sequential quadratic programming approach [12] or particle swarm optimization [13], and recent related research in the literature has focused on heuristic methods from evolutionary computation [14, 15]. In fact, the DGPP first appeared as one aspect of a multi-criteria optimization problem, solved using a genetic algorithm [7, 16]. A major part of our investigation is demonstrating the maturity of general purpose open-source code (in our case, Bonmin [17]) to solve the mixed-integer nonlinear programs arising from our new DGPP formulations.

We note that in our context of MINLP problems with non-convex functions, exact solution methods are often impractical or not available for large-scale problems, with the deterministic 'branch-and-bound' decomposition algorithm of Bonmin proving optimal points are globally optimal only for problems with convex constraints. Even apart from the proper handling of discrete design variables, the problem is complicated by the presence of quadratic non-convex constraints arising from the power flow and voltage control considerations on the power system. With non-convexity problem-specific knowledge is often the best way to finding good solutions or tractable approximate formulations, and two such formulations are explored in this paper along with the full model. The work in this paper forms a basis for more structured approaches that determine good lower bounds to the globally optimal value of the DGPP.

TABLE 1. Nomenclature

Node (Bus) Parameters	
$\mathcal{N} = \{1, \ldots, N\}$	Nodes (buses)
$\mathcal{N}_{-1} = \{2, \ldots, N\}$	Nodes excluding the slack bus
$\bar{P}_{Dn} + j\bar{Q}_{Dn}$	Net real (P) and reactive (Q) power demand at all nodes
$\bar{P}_D$	Vector of real power demand
$\bar{P}_{Dtot} \stackrel{\text{def}}{=} \sum_{n\in\mathcal{N}} \bar{P}_{Dn}$	Total real power demand
$\check{Q}_n$, $\hat{Q}_n$	Lower/upper reactive power limits
$V_1 \stackrel{\text{def}}{=} V_0 + j0$	Voltage at the slack node
$\check{V}_n$, $\hat{V}_n$	Lower/upper voltage magnitude limits
$Y = G + jB$	The network admittance matrix
Arc (Branch) Parameters	
$(n,k) \in \mathcal{A} \subset \mathcal{N} \times \mathcal{N}$	Arc set
$y_{nk} = g_{nk} + jb_{nk}$	Admittance (conductance, susceptance)
Generator Parameters	
Gens	Generator types
$\mathcal{B}$	Budget (cost limit of new generators)
c_{Gm}	Unit cost of generator m
$\bar{P}_{Gm}$	Peak real power output of generators
$\check{Q}_{Gm}$, $\hat{Q}_{Gm}$	Lower/upper reactive power limits
$V_0 (= \lvert V_1 \rvert)$	Nominal operating voltage magnitude
Variables	
$P_n + jQ_n$, $n \in \mathcal{N}$	Transmitted real (P) and reactive (Q) power at nodes
$V_n = e_n + jf_n$, $n \in \mathcal{N}$	Voltage at nodes — rectangular form
$V_n = \lvert V_n \rvert(\cos\theta_n + j\sin\theta_n)$	Voltage at nodes — polar form
I_n, $n \in \mathcal{N}$	Transmitted current at nodes
x_n	Binary variable indicating presence of generator(s) at node
z_{solver}	The optimal value of *solver*
$P^{\text{slack}}(V, X)$	The imported or slack power for a network of voltage V with design X
j	Imaginary unit, $j^2 = -1$
$\stackrel{\text{def}}{=}$	Defining relation
*	Complex conjugate
$\text{Re}(w)$	Real part of complex number w
$\text{Im}(w)$	Imaginary part of complex number w

2. Mixed Integer Models for Distributed Generation Planning

In this section we outline three different continuous power flow models and their augmentation with discrete placement design variables, in order to formulate the Distributed Generation Planning Problem in various forms. Our general problem is to minimise the objective of network power deficit, subject to power flow and voltage constraints, along with a budget constraint on the total cost of installing generators. The real power deficit of the network is equal to the sum of the transmission losses and the real power demands at nodes less the contribution of any new generation installed at the nodes.

Our new contribution in this section is the modified power flow problems with integer variables for generator siting. After some basic definitions, we look at the most detailed model of power flow, the AC model, before turning to the more approximate 'DC' flow models; the interested reader is referred to [18–20] for greater detail on the underlying equations of the continous-variable power flow models of this section. We first define and discuss the optimization problems before outlining issues relating to their solution.

2.1. **Basic Power Flow Modelling Terms and Relationships.** All electrical network models considered here are based on the abstraction of the network to a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with nodes $n \in \mathcal{N} = \{1, \ldots, N\}$ and arcs $\mathcal{A}$. Nodes in the network correspond to buses in the single-line diagram representation and arcs represent transmission lines. Table 1 contains descriptions of the various symbolic quantities and should be referred to for definitions. It is assumed that the network is connected and the bus admittance matrix Y is symmetric. At the slack (or swing) bus the voltage magnitude and phase angle are specified at reference values of V_0 and 0 respectively. In this work, the slack bus will always have label $n = 1$ and we denote $\mathcal{N}_{-1} \stackrel{\text{def}}{=} \{2, \ldots, N\}$ the set of nodes excluding the slack bus. Our continuous decision variables, the node voltage variables are collected in the vector V.

In modelling power flow, it is useful to define the *power flow functions* $P_n^{\text{flow}}(V)$ and $Q_n^{\text{flow}}(V)$ representing the real and reactive power injected from node n to the network as a function of the vector of node voltages V. Indeed, it is the choice of the form of the flow functions that is the key distinguishing feature of the different models, along with whether to include constraints on voltage and reactive power.

In the design problem, we choose from a given set of generator types. We denote this set by *Gens*. The essential properties distinguishing each generator type m in *Gens* are the peak real power output of the generator $\bar{P}_{Gm}$, the reactive power limits of the generator $(\check{Q}_{Gm}, \hat{Q}_{Gm})$ and the unit cost of the generator c_{Gm}.

The discrete *generator placement design* variables, one for each node and generator type are denoted

$$X_{nm}, n \in \mathcal{N}, m \in Gens,$$

with the value of the variable X_{nm} indicating the number of generators of type m installed at node n. The definition of a *design* is a set of values for the X_{nm} for a particular network and budget instance.

In the following section, we turn to outlining the different constraints resulting from each model of power flow incorporated within the mixed-integer generator placement design problem.

2.1.1. *Cost constraint.* With c_{Gm} as the unit cost of generator type m, and if the allocated budget for generator purchase is fixed at $\mathcal{B}$, the natural cost constraint is

$$\sum_{n \in \mathcal{N}} \sum_{m \in Gens} c_{Gm} X_{nm} \leq \mathcal{B}. \tag{1}$$

where the left hand side is the total cost of a design.

2.1.2. *Real power constraints.* The total contribution of real power from all installed generators will be

$$\sum_{m \in Gens} \bar{P}_{Gm} X_{nm}.$$

Consequently the conservation of power condition $P_n^{\text{flow}}(V) = \bar{P}_{Dn}$ at each node n is modified to incorporate the upgraded generation to

$$P_n^{\text{flow}}(V) = \bar{P}_{Dn} + \sum_{m \in Gens} \bar{P}_{Gm} X_{nm}, \quad n \in \mathcal{N}_{-1}. \tag{2}$$

Derivation of the models' objective functions is given in Appendix A.

2.2. **The ACrec Mixed-Integer Planning Problem.** The so-called 'full' model of power flow is described through the constraints of conservation of real power and bounds for reactive power and voltage magnitude. We denote as the *ACrec* model as the system of constraints with the choice of rectangular coordinates and power flow functions (3) and (4). Expressing the complex node voltage as $V_n = e_n + jf_n$ in rectangular form with $V = (e, f) \in \mathbb{R}^{2N}$, we have

$$P_n^{\text{flow}}(e,f) \stackrel{\text{def}}{=} \sum_{k\in\mathcal{N}} G_{nk}(e_n e_k + f_n f_k) + B_{nk}(f_n e_k - e_n f_k), \tag{3}$$

$$Q_n^{\text{flow}}(e,f) \stackrel{\text{def}}{=} \sum_{k\in\mathcal{N}} G_{nk}(f_n e_k - e_n f_k) - B_{nk}(e_n e_k + f_n f_k). \tag{4}$$

The ACrec problem has the following mathematical description in terms of rectangular variables: find

$$z_{ACrec} = \min_{e,f,X} P_{\text{AC}}^{\text{slack}}(e,f,X) \tag{5}$$

$$\text{where } P_{\text{ACrec}}^{\text{slack}}(e,f,X) \stackrel{\text{def}}{=} e^T G e + f^T G f - \sum_{n\in\mathcal{N}} \sum_{m\in Gens} \bar{P}_{Gm} X_{nm} + \bar{P}_{Dtot}$$

subject to budget constraint (1) and
Conservation of real power

$$P_n^{\text{flow}}(e,f) = \bar{P}_{Dn} + \sum_{m\in Gens} \bar{P}_{Gm} X_{nm}, \qquad n \in \mathcal{N}_{-1} \tag{6}$$

Bounds for reactive power

$$Q_n^{\text{flow}}(e,f) - \bar{Q}_{Dn} \le \hat{Q}_n + \sum_{m\in Gens} \hat{Q}_{Gm} X_{nm}, \quad n \in \mathcal{N}_{-1} \tag{7}$$

$$Q_n^{\text{flow}}(e,f) - \bar{Q}_{Dn} \ge \check{Q}_n + \sum_{m\in Gens} \check{Q}_{Gm} X_{nm}, \quad n \in \mathcal{N}_{-1} \tag{8}$$

Voltage bounds

$$e_n^2 + f_n^2 \le \hat{V}_n^2 + x_n(V_0^2 - \hat{V}_n), \qquad n \in \mathcal{N}_{-1} \tag{9}$$

$$e_n^2 + f_n^2 \ge \check{V}_n^2 + x_n(V_0^2 - \check{V}_n), \qquad n \in \mathcal{N}_{-1} \tag{10}$$

$$x_n \le \sum_{m\in Gens} X_{nm} \le M x_n, \qquad n \in \mathcal{N}_{-1} \tag{11}$$

$$x_n \text{ binary},$$

Slack bus reference values

$$e_1 = V_0, \quad f_1 = 0, \quad x_1 = 1 \tag{12}$$

$$(e,f) \in \mathbb{R}^{2N}, \qquad X_{nm} \ge 0, \text{ integer}, \quad x_n \text{ binary}, \tag{13}$$

where $V_0 \in \mathbb{R}$ is a given nominal voltage magnitude. We now outline the details of each constraint. We defer discussion of the objective function to section A.

Real power constraints: The equations describing real power flow (6) follow from the conservation of power, using the power flow function of (3).

Reactive power bounds: Relations (7) and (8) reflect the expanded reactive power compensation capacity that is available from generators with non-zero values of $\check{Q}_{Gm}$ or $\hat{Q}_{Gm}$. The constraints (7) and (8) on reactive power, with $\check{Q}_n \le \hat{Q}_n$, ensure conservation of power is achieved depending on the presence or absence of reactive power sources. When a source of reactive power, such as a certain type

of generator, is present at a node the value of reactive power can vary within the operational limits determined by the source, hence the relaxation of equality constraints to inequalities above. If no reactive power compensation is possible, the inequality constraints collapse to a single equality with $\check{Q}_n = \hat{Q}_n = 0$.

Voltage bounds: The voltage constraints (9) and (10) are magnitude bounds for some constants $\check{V}_n \leq \hat{V}_n$ (typically our bounds are taken within 6% of V_0, that is, $\check{V}_n = 0.94V_0$ and $\hat{V}_n = 1.06V_0$ after [21]). In a similar vein to the reactive power constraints in the presence of generators, we modify the voltage magnitude bounds. The introduction of the binary variables x_n is motivated by the requirement that a generator bus operates at a nominal system voltage magnitude of V_0 (see for instance [19, p266–7]). Therefore equations (9) to (11) represent the following logic: if there is a generator placed on node n then $0 < Mx_n$, implying $x_n = 1$ as x_n is a binary variable. Thus the voltage magnitude of the node is fixed to the network nominal operating voltage V_0; otherwise, $x_n \leq 0$, that is $x_n = 0$, and the voltage magnitude is bounded by $\check{V}_n$ and $\hat{V}_n$. The positive integer M is any upper bound on the sum in (11). The optimal choice of M in the absence of any other knowledge of the planning problem may be computed by a knapsack problem [22, Chap. 5]: fix n arbitrarily, and let M be the optimal objective value to

$$\max \sum_{m \in Gens} X_{nm} \text{ subject to } \sum_{m \in Gens} c_{Gm} X_{nm} \leq \mathcal{B}. \tag{14}$$

However a simple choice is to let M equal $\lfloor \mathcal{B}/c_G^{\min} \rfloor$, where $c_G^{\min} \stackrel{\text{def}}{=} \min_{m \in Gens} c_{Gm}$, since

$$c_G^{\min} \sum_{m \in Gens} X_{nm} \leq \sum_{m \in Gens} c_{Gm} X_{nm} \leq \mathcal{B}.$$

2.3. **The DC Power Flow Models.** In light of the harder nonlinear nature of the AC model, it is useful to look to models of a simpler structure, yet that still capture the essentials of power flow.

In the DC model [3], so-named in analogy to the flow of current in a direct current circuit, the reactive power inequalities (7) and (8) are disregarded and the polar form of voltage variables and real power flow functions are used; writing $V_n = |V_n| \, \mathrm{e}^{j\theta_n} = |V_n|(\cos\theta_n + j \sin\theta_n)$ in *polar* form with $V = (\theta, |V|) \in \mathbb{R}^{2N}$, the power flow functions take the form

$$P_n^{\text{flow}}(\theta, |V|) \stackrel{\text{def}}{=} \sum_{k \in \mathcal{N}} |V_n||V_k|(G_{nk} \cos\theta_{nk} + B_{nk} \sin\theta_{nk}), \tag{15}$$

where $\theta_{nk} \stackrel{\text{def}}{=} (\theta_n - \theta_k)$ represents the change in node voltage phase angles across the arc from node n to node k. One looks to approximate (15). The specific derivations are now outlined.

2.3.1. *Linear DC Power Flow Model.* Assumptions commonly used in the literature (e.g. [23]) in deriving the DC model are:

(1) the line resistances are negligible, thus $G_{nk} = 0$ for all $k \in \mathcal{N}$ (observe that the assumption of negligible resistances implies no transmission losses over arcs);
(2) voltage magnitudes are close to unity (in normalised units of the nominal voltage V_0) and do not significantly affect real power flows, thus let $|V_n| = |V_k| = 1$ for all $n, k \in \mathcal{N}$;
(3) the voltage phase angles across lines, $\theta_{nk} \stackrel{\text{def}}{=} (\theta_n - \theta_k)$, are negligible in the second and higher orders, and so we approximate with the substitutions $\sin\theta_{nk} \cong \theta_{nk}$.

These three assumptions transform (15) into the linear function

$$P_n^{\text{flow}}(\theta) \stackrel{\text{def}}{=} \sum_{k\in\mathcal{N}} B_{nk}(\theta_n - \theta_k). \tag{16}$$

Therefore the linear DC power flow model is described by the voltage phase variables with a new linear version of the conservation equation (6) and a reference value for the phase angles. The DC Mixed-Integer Planning Problem has the following form: find

$$\begin{aligned} z_{DC} &= \min_{\theta,X} P_{\text{DC}}^{\text{slack}}(X) \\ &\stackrel{\text{def}}{=} \min_{\theta,X} -\sum_{n\in\mathcal{N}} \sum_{m\in Gens} \bar{P}_{Gm} X_{nm} + \bar{P}_{Dtot} \end{aligned} \tag{17}$$

subject to budget constraint (1) and a form of conservation of real power

$$\sum_{k\in\mathcal{N}} B_{nk}(\theta_n - \theta_k) - \sum_{m\in Gens} \bar{P}_{Gm} X_{nm} = \bar{P}_{Dn},\ n \in \mathcal{N}_{-1} \tag{18}$$

$$\begin{gathered} -\pi \le \theta_n \le \pi, \qquad n \in \mathcal{N}_{-1}, \\ \theta_1 = 0, \quad \theta = (\theta_i)_{i=1}^N \in \mathbb{R}^N, X_{nm} \ge 0, \text{ integer}. \end{gathered} \tag{19}$$

We refer to this as the DC Model. Arguably, the accuracy of this model when the assumptions are violated is poor [24]. Note that without new generation, power flow is reduced to finding a solution to a square linear system $B(\theta) = -\bar{P}_D$, where the matrix B that appears in the original bus admittance matrix Y is used throughout this paper.

2.3.2. *Nonlinear DC Power Flow Model.* We now turn to the derivation of a model that is similar to the DC Model yet with an attempt to incorporate transmission losses. The classical exposition of this model can be found in [4, Appendix D], and [5] is also useful. As in the DC Model, one disregards the reactive power equations and assumes that voltage magnitudes are close to unity. However we do not neglect the line resistances. Furthermore, the voltage phase angles across lines, θ_{nk}, are considered negligible only in the *third* and higher orders. As a result we approximate the trigonometric functions with the substitutions

$$\cos\theta_{nk} \cong 1 - \frac{1}{2}\theta_{nk}^2,\ \sin\theta_{nk} \cong \theta_{nk}.$$

These assumptions now transform (15) into the quadratic function

$$P_n^{\text{flow}}(\theta) = \sum_{k\in\mathcal{N}} -\frac{1}{2} G_{nk}(\theta_n - \theta_k)^2 + B_{nk}(\theta_n - \theta_k), \tag{20}$$

under the observation that by definition of G,

$$\sum_{k\in\mathcal{N}} G_{nk} = G_{nn} + \sum_{k\in\mathcal{N}:k\neq n} G_{nk} = G_{nn} - G_{nn} = 0.$$

This nonlinear model is essentially the DC model with line losses represented by quadratic functions of the variables, and so we refer to the following as the DCQL (DC Quadratic Loss) Model.

The DCQL Mixed-Integer Planning Problem has the following form: find

$$z_{DCQL} = \min_{\theta, X} P^{\text{slack}}_{\text{DCQL}}(X) \tag{21}$$

$$\text{where } P^{\text{slack}}_{\text{DCQL}}(X) \stackrel{\text{def}}{=} \sum_{n,k \in \mathcal{N}} \frac{1}{2} g_{nk}(\theta_n - \theta_k)^2 - \sum_{n \in \mathcal{N}} \sum_{m \in Gens} \bar{P}_{Gm} X_{nm} + \bar{P}_{Dtot}$$

subject to budget constraint (1) and
Conservation of real power

$$\sum_{k \in \mathcal{N}} -\frac{1}{2} G_{nk}(\theta_n - \theta_k)^2 + B_{nk}(\theta_n - \theta_k) - \sum_{m \in Gens} \bar{P}_{Gm} X_{nm} = \bar{P}_{Dn},\ n \in \mathcal{N}_{-1} \tag{22}$$

$$-\pi \leq \theta_n \leq \pi, \qquad n \in \mathcal{N}_{-1}, \tag{23}$$

$$\theta_1 = 0, \quad \theta = (\theta_i)_{i=1}^N \in \mathbb{R}^N, X_{nm} \geq 0, \text{ integer.}$$

Note that by definition of the matrix G, $-G_{nk} = g_{nk} \geq 0, n \neq k$ is the conductance of the arc from node n to node k.

2.4. **Solving the Planning Models.** With the description of our planning models completed, we now outline some issues surrounding their solution.

2.4.1. *The ACrec Planning Problem.* The ACrec DGPP is difficult even for fixed values of the integer variables. The feasible set of the AC model is defined through equalities involving nonlinear functions and inequalities involving the non-convex power flow functions [25] and the voltage magnitude lower bound (10). Consequently it should be expected that any continuous optimization problem (let alone one with discrete variables) involving the ACrec constraints will be difficult to solve to global optimality, and that any candidate global extrema will be difficult to verify [26]. Nice aspects of the ACrec constraints include the sparse nature of the Jacobian. Furthermore, the quadratic form of the ACrec model means the Jacobian depends linearly on the variables and the functions have no higher order terms beyond the second. In other words, the Hessian is constant, which can simplify computations that check local optimality conditions.

In principle, indefinite quadratically constrained quadratic programs are NP-hard [27] (though this is true even for MILP problems). On the positive side, the matrix G is positive-semidefinite and so the objective is convex in the voltage variables for fixed integer variables, and good initial points are often available from experience. However, the feasible set is non-convex even when the integer variables are fixed, and there may be many local minima in different components of the feasible set. For the specific case of power or load flow, multiple feasible points are known to exist even in simple cases [28] and for varying power injections at nodes (as occurs for varying generator types and placements) the number of solutions to the real power equations may change in a nonlinear manner. Quantitative analysis has focused on the set of vectors of power injections for which the network has a feasible voltage point [29] and this set is not convex in general [30]. This means that even a continuous relaxation (relaxing the integrality requirements on the design variables) of the ACrec Planning Problem may have solutions that are far from the true global optimal solution. Recognising this non-convex nature, global optimization approaches to power systems have employed schemes involving convex

outer approximation for functions specifically arising in the AC (polar) model and iterative refinement techniques [31].

2.4.2. *The DC Planning Problem.* This problem involves affine functions alone. It is a MILP and is efficiently solved with state-of-the-art codes such as CPLEX. Since the problem has a convex (linear) objective and feasible set, any local optimum solution is a global one.

2.4.3. *The DCQL Planning Problem.* The considerations on solving the ACrec-based DGPP apply to the DCQL Planning Problem. In the case of the DCQL, there is a clearer separable structure in the phase angle differences $\theta_{nk} = (\theta_n - \theta_k)$. Exploiting this specific structure can usefully improve solution algorithms for separable mixed-integer problems [32]; while we do not use the solver of [32] in our computations, the separable structure is retained in passing the encoded problem to our chosen solver (Bonmin), which is able to recognise and potentially use the separability.

While the DCQL power flow functions are no longer linear (nor convex) they are positive-definite quadratic functions on the θ domain $\{0\} \times \mathbb{R}^{N-1}$, and relaxing equality in (22) to $\leq$, along with integrality on the X results in a convex feasible set (cf. [5]). The physical interpretation of replacing $P_n^{\text{flow}}(\theta) = \bar{P}_{Dn}$ with $P_n^{\text{flow}}(\theta) \leq \bar{P}_{Dn}$ is that while it still holds that power transmitted to the network from a node cannot exceed that available, not all power must be transmitted - put another way, "excess power can be shed at any node with no penalty" [5, p4]. This relaxation is justifiable in certain situations — for example, electricity market analysis [6, p316] or even battery storage applications — and again, we do not consider this specific relaxation here, but note this convex substructure is a promising avenue for further work on strong relaxations of the DCQL problem.

3. Objective Lower Bounds Obtained via Knapsack Problems

The nonlinear, nonconvex nature of the ACrec (§2.2) and DCQL (§2.3.2) Mixed-Integer Planning Problems means that it is not always immediate whether a found placement design is globally optimal, just locally optimal or even very good at all. Often a close lower bound to the objective value of the global optimum can indicate the quality of the best feasible solution known. In this section we give a simple means of obtaining lower bounds to the objective value through a simple relaxation. Such lower bounds serve as reference values for comparing the relative performance of our different modelling approaches in our computational comparisons. However, no designs are produced. Furthermore, we do not expect to obtain close lower bounds in every instance, considering the simple means of obtaining the relaxation. This relaxation of each Planning Problem is obtained by simply dropping all constraints from each problem other than the budget constraint and integrality. For reasons given below, we call the optimal value z_{KS} of the relaxation the *knapsack lower bound*:

$$z_{KS} \stackrel{\text{def}}{=} \min_{X} \quad -\sum_{n\in\mathcal{N}} \sum_{m\in Gens} \bar{P}_{Gm} X_{nm} + \bar{P}_{Dtot} \tag{24}$$

subject to

$$\sum_{n\in\mathcal{N}} \sum_{m\in Gens} c_{Gm} X_{nm} \leq \mathcal{B}, \tag{25}$$

$$X_{nm} \geq 0, \text{ integer.} \tag{26}$$

Equivalenty, we compute z_{KS} as the value of

$$\bar{P}_{Dtot} - \max_{X \geq 0,\ \text{integer}} \sum_{n \in \mathcal{N}} \sum_{m \in Gens} \bar{P}_{Gm} X_{nm}$$

subject to (25) and (26). Observing that the coefficients depend on $m \in Gens$ alone and consequently the variables X_{nm} may be aggregated at a node n, this suggests using a 'book-keeping' variable $\tilde{X}_m = \sum_{n \in \mathcal{N}} X_{nm}$, and our problem is finding the optimal value of the program

$$w_{KS} = \max_{\tilde{X} \geq 0,\ \text{integer}} \sum_{m \in Gens} \bar{P}_{Gm} \tilde{X}_m \tag{27}$$

subject to

$$\sum_{m \in Gens} c_{Gm} \tilde{X}_m \leq \mathcal{B}.$$

The lower bound is then $\bar{P}_{Dtot} - w_{KS}$. The above problem of finding w_{KS} is a classical knapsack problem [22, Chap. 5] in the variables $\tilde{X}_m$ and is efficiently solved by our MIP solver CPLEX [33].

The value z_{KS} is an underestimate of each original problem value as the losses, which are absent in this objective, are never less than zero, and the knapsack lower bounding problem is obtained from ignoring the power constraints; that is, $z_{KS} \leq z_{AC}$, and similarly $z_{KS} \leq z_{DC}$ and $z_{KS} \leq z_{DCQL}$. No direct relations exist however between the objective values of the other models. While no globally optimal values are guaranteed from ACrec and DCQL solutions, we expect that the difference between z_{KS} and the objective value of the best feasible solution (and z_{KS} and the actual global optimal value) will be governed by the size of the transmission losses.

4. Computational comparison of planning models

The aim of this section is to present the results of a computational comparison of the different modelling approaches to the DGPP previously outlined. To the best of our knowledge the direct comparison of three models for distributed generation planning, and with a genetic algorithm baseline of comparison, is novel (though comparisons of genetic algorithms in power system design [34] do exist). Here a *solver* is an implementation of a DGPP solution method, as distinct from the abstract problem statement.

4.1. **Computational Process.**

A set of small-scale (less than 5 MW capacity) generator types was taken from Table 3 of [7], excluding the two steam turbines but including the CHP varieties. While heat utility was not accounted for in the model, the CHP varieties of generator were kept in our available set as a check on the problem formulation: with all things being equal, the solvers should select a lower cost option out of generator types with identical electrical power properties, which they were indeed seen to do.

Theory, implementation and results for the genetic algorithm (GA) are described in [7,16]. The NSGA-II algorithm employed is based on fronts ranked by dominance and equipped to pursue a well-spread approximation set of the true Pareto optimal front by biasing the search toward regions of the objective space with sparse coverage. The NSGA-II algorithm forms the heart of a multi-objective solver for distributed generation planning encompassing objectives of imported power, cost, emissions and efficiency. The underlying power flow model is the a form of the polar AC equations, with voltage constraints enforced through penalties.

For a given network and a set of generator choices, the NSGA-II algorithm was run 20 times, starting each time from an initial population size of 50, to produce

TABLE 2. Test Networks. The table shows number of nodes (N) and arcs ($|\mathcal{A}|$), and total real power demand in MW ($\bar{P}_{Dtot}$).

Name (Source)	N, $\lvert\mathcal{A}\rvert$	$\bar{P}_{Dtot}$	Details
MP4gs [21]	4, 4	182.0	High demand with single fixed generator
MP6ww [21]	6, 11	100.0	Medium demand with two fixed generators
MP9a [21]	9, 9	67.0	Medium demand with two fixed generators
BE6a [35]	6, 8	5.450	Meshed topology, single fixed generator
CR12a [36]	12, 15	255.0	Meshed topology, 3 fixed generators
CR12rx [36]	12, 15	255.0	CR12a with resistance increased to equal reactance
GEA13a [37]	13, 12	3.900	Linear topology, equal loads at all nodes
GEA13ce [37]	13, 12	2.940	Linear topology, load distributed around centre node
GEA13in [37]	13, 12	3.510	Linear topology, increasing loads along nodes
P1sm [7]	14, 20	2.152	Meshed topology, same as IEEE 14 test case
P1md [7]	14, 20	2.152	P1sm with medium resistance and reactance
P1lg [7]	14, 20	2.152	P1sm with large resistance and reactance

20 independent imported-power/total-cost Pareto optimal fronts. Each run took 1000 evaluations. It should be noted that by increasing the number of evaluations one would expect to marginally improve the performance of NSGA-II. The 20 instances were then combined into an aggregate optimal front with the corresponding generator placement design recorded. The respective costs of the aggregate front were extracted in a list of budgets for the MIP problems. Each budget corresponds to an instance of the parameter $\mathcal{B}$ in the planning model statements. Each MIP problem was then solved sequentially by incrementing the budget parameter over the budget list derived fron the GA, thus obtaining a design (or infeasible result) for each budget value. The GA acts here as a reference for comparison of the MIP methods through its function of generating the budgets corresponding to candidate designs.

4.1.1. *Test data.* An extensive set of test networks of small node number (between 4 and 14) have been obtained from a cross-section of the power engineering literature. A summary of the test cases and their respective sources are shown in Table 2. The installed distributed generation was kept in the range of small to medium, 5 kW to 50 MW, following [1]. Consequently, the total net power demand over a network was important in determining the range of the budgets: for the networks with a net demand exceeding 50 MW we imposed a cap on the largest budget of $35 million, a figure approximately equal to the purchase of 50 MW worth of installed generation. For networks with a net demand much lower than 50 MW, a budget much lower than $35 million with a feasible design was found by the GA that met most, if not all, of the demand.

4.1.2. *Vaildation of Experiments.* The full AC Newton's method power flow module of MATLAB-based power flow package, MATPOWER [21], version 4.0, was used as the common evaluator of designs generated by each solution method. By using MATPOWER as our standard evaluator on each valid instance, the quality of designs can be compared independently of the solver. Each evaluation of a network with a candidate design was initialized from the same voltage point. The slack power was recorded after evaluation by MATPOWER if convergence was achieved. No voltage constraints are enforced in the algorithm, but statistics on the deviation of voltage magnitude outside bounds of ± 0.06 from 1.00 in normalized units are retained.

The knapsack lower bounds had no placement designs to evaluate and are taken directly from the solver output.

4.1.3. *Performance measurement methodology.* We use performance profiles to compare the different methods of solving the DGPP problems. Performance profiles as used here were introduced in [38] to summarize and compare the performance of different solvers running over the same instances. For timing performance, we use a ratio on the x-axis, and for evaluating the quality of designs we use a relative gap (or difference) from the knapsack lower bound on the x-axis.

Essentially, a performance profile is the cumulative distribution function of a performance metric for a solver, relative to all the solvers one looks to compare with (see [39, Chap. 22] for a useful exposition). The basic idea in the creation of a performance profile for each solver is to decide on a metric of comparison (be it iteration count, CPU run time or otherwise) where smaller values are better. Data is collated for each instance and solver. For each instance, the best value out of all solvers is identified, and the performance of a solver is defined as the ratio of the solver metric value to the best value. The desired profile of a solver is produced by sorting from best to worst the performance ratios over all instances, then plotting the proportion of instances against the performance ratio. Thus the point the profile leaves the vertical axis (at x-axis value 1) gives us the fraction of instances the solver had the best metric value. If the profile reaches a height of 1 then all instances were solved. Otherwise, the right side of the profile, where the profile has an asymptote, tells us the proportion of instances the solver failed to solve. Visually the better profiles are distributed closest to the upper left-hand corner of the plot, indicating that a larger number of instances had small ratios.

4.1.4. *Evaluating the quality of designs from lower bounds.* Our performance profile for power differs slightly from the CPU run time profile in that the power performance data are not ratios computed relative to the best solver but are values computed relative to the lower bounds found through the knapsack computation (the timing data, however, follows the original approach). The independent quantity (x-axis) is not a ratio, but a relative gap value, defined by

$$g_s(p) = \frac{|z_s(p) - z_{KS}(p)|}{|z_{KS}(p)|}, \tag{28}$$

if $|z_{KS}(p)|$ is greater than a set tolerance (we use 10^{-6}), otherwise

$$g_s(p) = |z_s(p) - z_{KS}(p)|, \tag{29}$$

where $z_s(p)$ is the solver value on budget instance p.

4.2. **Computational Results.**

4.2.1. *Platform specification.* Mixed-integer programming computations were run on a Dell PowerEdge 2950 with dual quad core 3.16GHz Intel Xeon X5460 processors and 64GB of RAM running Red Hat Enterprise Linux 5. The genetic algorithm was implemented in Java and run on a 2.67GHz Intel Core i5-560M processor and 3.24GB of RAM running Windows XP.

4.2.2. *Solving the MIP Problems.* The MIP tests were formulated and run using the AMPL modelling language [40]. The MINLP problems were solved with Bonmin [17], version 1.5. Default parameters were used for the `B-BB` branch-and-bound algorithm (based on solving a continuous nonlinear program at each node of the search tree and branching on variables [41]). The MILP problems were solved with CPLEX 12.3 [33]. CPLEX 12.3 is used as the NLP solver in Bonmin. The default NLP solver Ipopt 3.10.0 [42] is used within Bonmin, and being an interior-point method based solver there is limited use of warm-starting and previous feasible points. It appears the main use of good feasible solutions in the MINLP algorithm is to provide good upper bounds that help in the branch-and-bound scheme. The

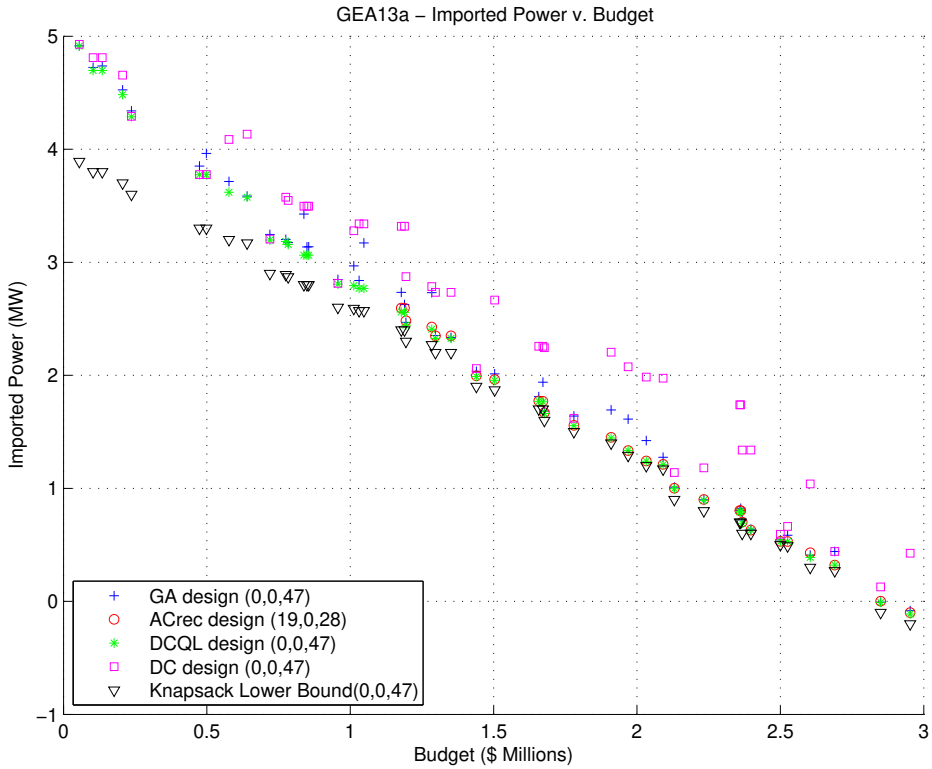

FIGURE 1. Multiple problem objectives against budget instances - case GEA13a

indications are that finding feasible points is not difficult in most cases; the difficulty occurs in establishing (local) optimality. On non-convex problems, Bonmin does not guarantee globally optimal solutions; indeed, the solutions of continuous subproblems found with Ipopt are not necessarily globally optimal. Bonmin attempts to find good feasible mixed-integer solutions with settings that allow the branch-and-bound algorithm to function as a robust heuristic in our non-convex setting.

A time limit of 1800 seconds (30 minutes) was imposed on each budget instance for both MIP solvers. It was observed that, if it was to do so, Bonmin returned an infeasible result virtually immediately, otherwise any time that was spent was done so trying to improve a known feasible solution.

4.3. **Results And Discussion.** Figure1 is shown as a representative instance; it provides insight into the relative values of objective values across all solvers for this test case. The three numbers following the key entry indicate successes of each solver: the first is the number of failures of the solver algorithm to find a feasible design (this is always 0 for the GA by construction), the second is the number of failures of MATPOWER to converge on a design, and the third is the total number of feasible designs which successfully converged with MATPOWER. Only the ACrec solver found no feasible designs (on budget instances below threshold of $1,177,964). This plot illustrates the gap to the knapsack lower bound: for the DC solver the gap varies widely over the instances, while the DCQL and GA methods steadily approach the lower bound as the budget (and the installed generation) increases.

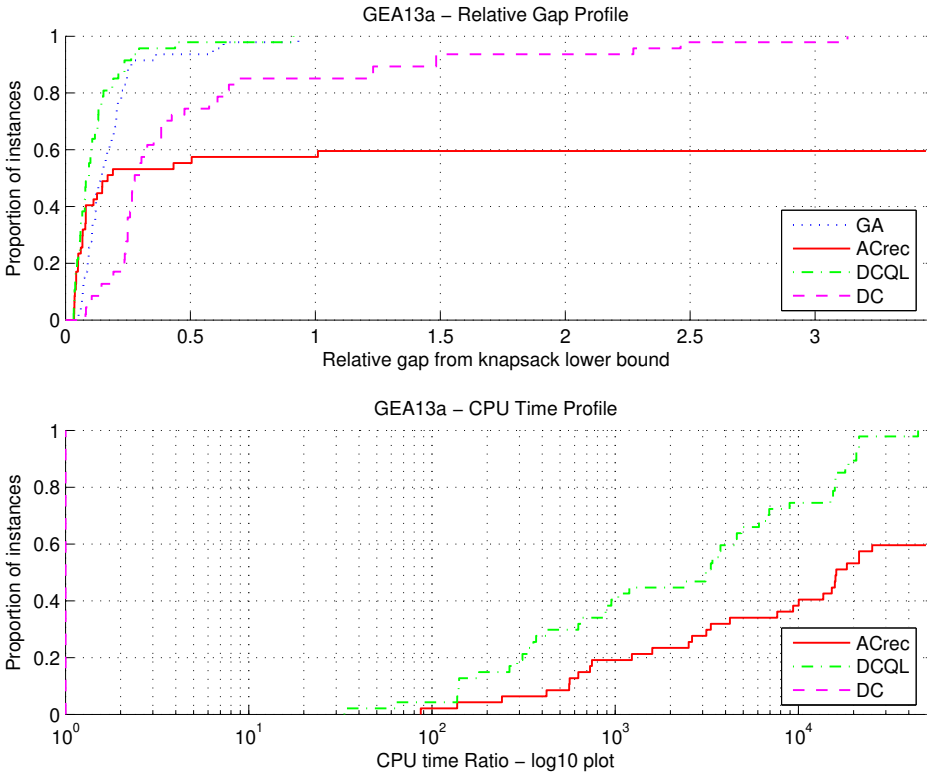

FIGURE 2. Case GEA13a Performance Profiles. Top: Relative Gap Profile. Bottom: CPU Time Profile (base 10 log scale).

This trend fits with the expectation that the gap reduces as the line losses reduce, which in turn occurs as more generation is installed nearby demand.

The indications of Figure 1 are borne out in the relative gap performance profile in the upper half of Figure 2. We see that all solvers return feasible designs for all instances except for ACrec, which found feasible designs in just under 0.6 (or 60%) of instances. The ACrec solvers found better designs than the GA for just over 40% of cases. The DCQL, however, out-performed all solvers although its largest relative gap value of 0.91 is comparable to the GA value of 0.93, with the largest gap of ACrec slightly larger at 1.01. The DC solver shows a poorer distribution with over 85% of designs within a relative gap of 0.70. In the lower half of Figure 2 is the timing profile with a base 10 log scale on the x-axis. The DC solver is uniformly the fastest on all instances — its series is shown by a vertical line at 1 — while the DCQL is likely to be faster than the ACrec solver.

If we now compare the MP9a test case in Figure 3 we see a reversal of roles with the ACrec solver dominating over DCQL in the timing data. The relative gap profile shows the MIP solvers are dominant over the GA in finding good solutions. No MIP solver distinguishes itself, though the DC is slightly worse. The relative gap profile has an offset of the origin of all series of around 0.07, with the most likely interpretation being that the knapsack lower bound is uniformly bad in this instance rather than the solvers failing to get close to the global minimum objective value.

In an unusual case, on the zero-budget (zero new generation) instance of network MP4gs, a feasible point was identified by MATPOWER in 3 iterations. In contrast,

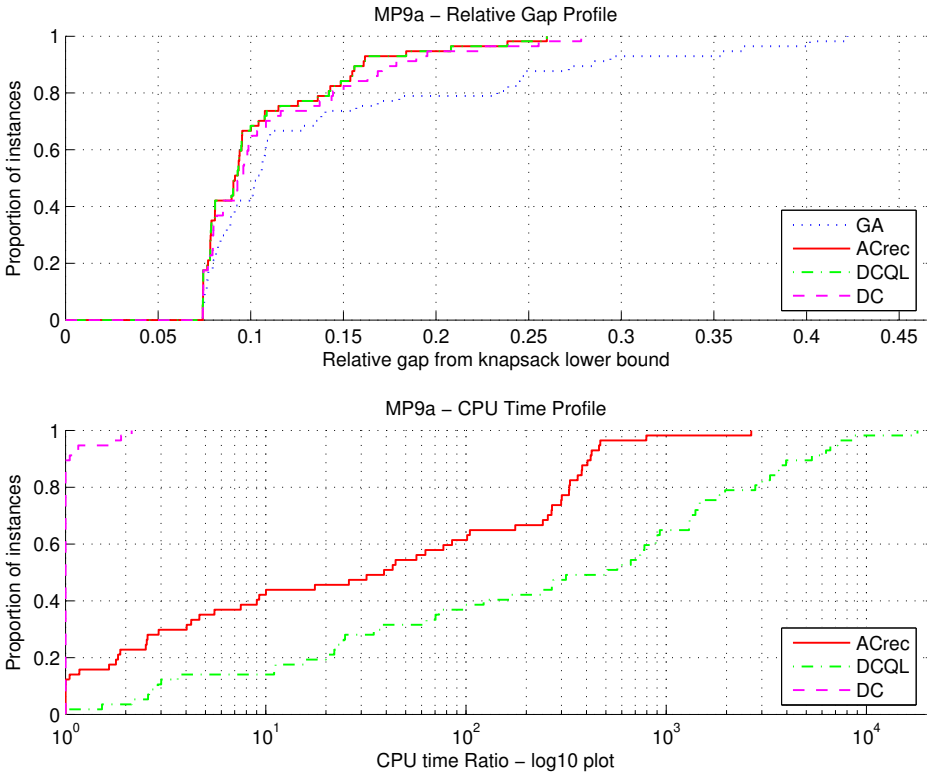

FIGURE 3. Case MP9a Performance Profiles. Top: Relative Gap Profile. Bottom: CPU Time Profile (base 10 log scale).

the ACrec solver converged to a locally infeasible point, with the same result occurring with 4 other nonlinear programming solvers tested via the NEOS server (`http://www.neos-server.org/neos/solvers/index.html`) and even after providing the initial feasible point. Similar infeasible results were returned for the non-zero budget instances. The DCQL and DC solvers, however, returned feasible designs across all budgets of the MP4gs network, indicating that approximations to the full nonlinear case can have a greater degree of flexibility in generating designs in certain difficult cases.

Finally, Figure 4 shows the aggregate performance of the solvers over all cases and budget instances; the relative gap profile is zoomed to show the lower end of the gap axis. The DCQL solver has better typical performance over the ACrec solver and the GA in both the proportion of smaller relative gap values and time ratios, helped by its ability to return feasible designs over all the test cases. We see on the 'all cases' relative gap profile at the 'cross-over' point of the GA and ACrec lines that the top 79.5% of relative gap values for the ACrec solver are better than the top 79.5% of the GA. Put another way, for any given instance there is a probability of approximately 79.5% of finding a improved relative gap in the ACrec solver compared to the GA, with gaps of less than 0.16.

In the CPU time profiles the GA solver is omitted by experimental design as its timing data is not directly comparable, but it is by far the fastest solver on a per design basis: on the worst case run it took 362 seconds to find 15 designs with a range of budgets, or an average of 24.29 seconds per design, while in the best case it took 1.844 seconds to find 36 designs, or 0.051 seconds per design on average. In addition, the GA approach generates a diverse range of feasible designs

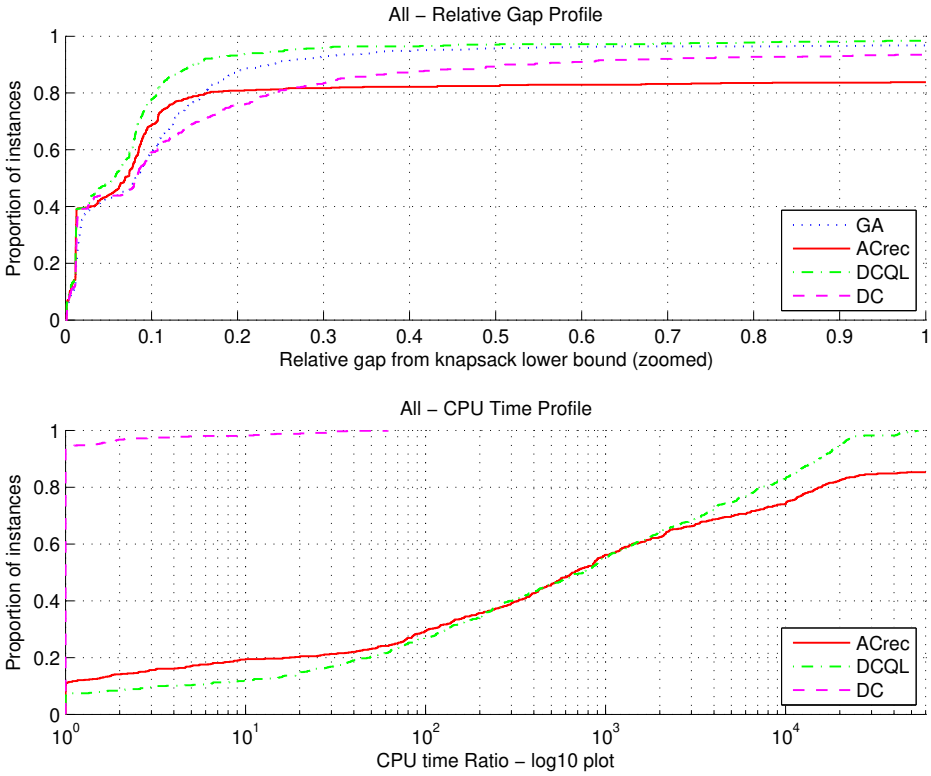

FIGURE 4. Performance Profiles, all cases. The relative gap profile is zoomed.

TABLE 3. MIP Timing Statistics. Median and minimum time are over feasible instances only.

Method:	ACrec	DCQL	DC
Feasible designs returned (/673)	574	673	673
Time limit of 1800s reached	163	252	45
Time limit reached (%)	28.4	37.4	6.69
Median time (seconds)	118.98	356.86	0.079
Minimum time (seconds)	0.012	0.009	0.013

in parallel over its run. The DC solver is by far and away the fastest of the MIP solvers, though ACrec or DCQL are faster on a small proportion (44/673 or 6.5%) of instances. Overall, the DC solver rivals even the simpler knapsack problem in time, yet is unlikely to produce the best designs as evidenced in the performance profiles. In the compared to the DCQL solver, the ACrec solver does have a lower median time and reaches its time limit in a smaller proportion of instances when it does not return an infeasible result. Further timing statistics are provided in Table 3. Indeed, the ACrec solver returned just under 15% (99/673) of instances as infeasible, which is explainable by the fact that it has the greatest number and type of nonlinear constraints of all methods. The DCQL solver finds equally good or better solutions to the ACrec solver, over a wider range of budget instances. The better performance of the DCQL solver over the ACrec solver, in terms of improved objective values, is unexpected as the ACrec model is the 'full' model from which the DCQL model is derived (including the fixing of voltage magnitude

variables). These results indicate that the DCQL model captures sufficient power flow structure from the AC (polar) model to enable the discovery of good designs across our set of networks and budgets. This is something that the linear DC solver fails to do to a sufficient extent; we postulate that the absence of line losses in the DC model allows the DC solver to ignore the complex interaction between network topology, demand and conservation of power.

5. Conclusion and further work

We have investigated the relative merits of different mixed-integer programming (MIP) problems and methods, placed in comparison to a genetic algorithm approach, for finding the optimal placement of small-scale generators in a distribution network under a budget constraint. The objective is to minimize the total power deficit in the system. The results of the optimization are a proposed placement of generators with a corresponding feasible steady-state voltage profile. We have show how three models of power flow - AC, DC and DCQL (DC with quadratic loss terms) - may be embedded in an mixed-integer programming framework for planning, and the relative quality of their respective solutions have been assessed over an extensive range of test networks and found to yield promising results. Indeed, each MIP model is a viable means of finding placement designs, with the nonlinear MIP approaches consistently showing objective value improvement over the stochastic genetic algorithm in numerous cases. While the speed of the GA is superior to the MIP approaches, the linear DC approach is able to find designs within the same order-of-magnitude time frame, of equal quality to the GA in certain cases. The AC power flow based MIP approach did surprisingly poorly, failing to find valid designs much less often than the more approximate DCQL approach. It has been seen that the DCQL model offers a good trade-off between accurately capturing the AC power flow and a tractable set of constraints leading to feasible designs over a wide set of budget instances.

One extension of this work is to consider a convex relaxation of the full AC power flow constraints, where an optimal solution to the relaxation provides a strong lower bound to the original non-convex program (e.g. [43]), retaining sufficient structure in order to identify good designs. Iterative refinement and decomposition of the domain may be applied in order to yield stronger bounds. Future work on the GA side includes increasing the GA evaluations to quantify their effect on the resulting population, and improvement of the internal AC power flow solver, including the integration of the DCQL power flow model into the GA to see if performance is improved as the MIP models indicate.

We have shown that current MIP methods are capable of tackling the DGPP over a variety of problems across numerous test networks, and in fact may improve, with respect to designs, upon the state-of-the-art in genetic algorithm approaches. A simple relaxation model has been introduced to provide lower bounds and shown to serve as a useful point of reference for discussing optimality.

Appendix A. Modelling the Objective Function

As previously mentioned, our objective is to minimise the network's *imported* or *slack* power, defined to be the real power deficit of the system and is equal to the sum of the transmission losses and the real power demands at nodes, less the contribution of newly installed generation, giving the objective function

$$P^{\text{slack}}(V,X) \stackrel{\text{def}}{=} P^{\text{loss}}(V) - \sum_{n\in\mathcal{N}} \sum_{m\in Gens} \bar{P}_{Gm} X_{nm} + \bar{P}_{Dtot}. \tag{30}$$

The last term $\bar{P}_{Dtot} \stackrel{\text{def}}{=} \sum_{n\in\mathcal{N}} \bar{P}_{Dn}$ is given from problem information, and though it is a constant, it is kept in the objective throughout to give the numerical results an immediate physical meaning. We still require the expression for line losses $P^{\text{loss}}(V)$, which is simply expressed as

$$P^{\text{loss}}(V) \stackrel{\text{def}}{=} \sum_{n\in\mathcal{N}} P_n^{\text{flow}}(V).$$

Because the loss function depends on the functions $P_n^{\text{flow}}(V)$, the form taken is specific to each model, as we now outline.

DCQL: The transmission losses in the DCQL model are computable as

$$\begin{aligned} P^{\text{loss}}(\theta) &\stackrel{\text{def}}{=} \sum_{n\in\mathcal{N}} P_n^{\text{flow}}(\theta) \\ &= \sum_{n,k\in\mathcal{N}} B_{nk}\theta_{nk} - \frac{1}{2}G_{nk}\theta_{nk}^2. \end{aligned}$$

Observing that $B_{nk} = B_{kn}$, so $B_{nk}\theta_{nk} = -B_{kn}\theta_{kn}$ and the sum over all n and k is zero,

$$P^{\text{loss}}(\theta) = \frac{1}{2}\sum_{n,k\in\mathcal{N}} (-G_{nk})\theta_{nk}^2 = \frac{1}{2}\sum_{n,k\in\mathcal{N}} g_{nk}\theta_{nk}^2 \tag{31}$$

where $-G_{nk} = g_{nk} \geq 0, n \neq k$ is the conductance of the arc from n to k.

DC: In this case, g_{nk} is set to 0 for all $n, k \in \mathcal{N}$, so $P^{\text{loss}}(\theta) = 0$.

ACrec: Since $P_n^{\text{flow}}(V) = \text{Re}(V_n(YV)_n^*)$,

$$\begin{aligned} P_{ACrec}^{\text{loss}}(e,f) &= \sum_{n\in\mathcal{N}} \text{Re}(V_n(YV)_n^*) = \text{Re}\sum_{n\in\mathcal{N}} V_n(YV)_n^* \\ &= \text{Re}(V^T(YV)^*) \\ &= e^TGe + f^TGf, \end{aligned}$$

where G is the real part of the network admittance matrix Y. Notably, the loss function is a convex quadratic form of the variables e, f since G is a positive-semidefinite matrix.

References

[1] Ackermann T, Andersson G, Söder L. Distributed generation: a definition. Electric Power Systems Research. 2001;57(3):195–204. Available from: `http://www.sciencedirect.com/science/article/pii/S0378779601001018`.

[2] Barnes M, Kondoh J, Asano H, Oyarzabal J, Ventakaramanan G, Lasseter R, et al. Real-World MicroGrids - An Overview. In: IEEE International Conference on System of Systems Engineering; 2007. p. 1–8.

[3] Stott B, Jardim J, Alsac O. DC Power Flow Revisited. IEEE Trans Power Syst. 2009 August;24(3):1290–1300.

[4] Scwheppe FC, Caraminis MC, Tabors RO, Bohn RE. Spot pricing of electricity. Kluwer Academic Publishers, Norwell, MA; 1988.

[5] Philpott AB. Experiments with Load Flow Pricing Models. Department of Engineering Science, The University of Auckland; 1999.

[6] Chen Y, Hobbs B, Leyffer S, Munson T. Leader-Follower Equilibria for Electric Power and NOx Allowances Markets. Computational Management Science. 2006;3:307–330. 10.1007/s10287-006-0020-1. Available from: `http://dx.doi.org/10.1007/s10287-006-0020-1`.

[7] Berry AM, Cornforth DJ, Platt G. An introduction to multiobjective optimisation methods for decentralised power planning. In: IEEE Power Engineering Society General Meeting; 2009. p. 1–9.

[8] Gönen T, Foote BL. Distribution-system planning using mixed-integer programming. Generation, Transmission and Distribution, IEE Proceedings C. 1981 March;128(2):70–79.

[9] Ponnavaikko N, Rao KSP, Venkata SS. Distribution System Planning through a Quadratic Mixed Integer Programming Approach. IEEE Trans Power Del. 1987 Oct;2(4):1157–1163.
[10] Haffner S, Pereira LFA, Pereira LA, Barreto LS. Multistage Model for Distribution Expansion Planning With Distributed Generation — Part I: Problem Formulation. IEEE Trans Power Del. 2008 April;23(2):915–923.
[11] Lavorato M, Rider MJ, Garcia AV, Romero R. Distribution network planning using a constructive heuristic algorithm. In: IEEE Power Engineering Society General Meeting; 2009. p. 1–6.
[12] AlHajri M. Sizing and placement of distributed generation in electrical distribution systems using conventional and heuristic optimization methods. Dalhousie University; 2009.
[13] Ziari I, Ledwich G, Ghosh A, Cornforth D, Wishart M. Optimal allocation and sizing of capacitors to minimize the transmission line loss and to improve the voltage profile. Computers & Mathematics with Applications. 2010;60(4):1003–1013.
[14] Vallem MR, Mitra J, Patra SB. Distributed Generation Placement for Optimal Microgrid Architecture. In: IEEE PES Transmission and Distribution Conference and Exhibition; 2006. p. 1191–1195.
[15] Haesen E, Driesen J, Belmans R. A long-term multi-objective planning tool for distributed energy resources. In: IEEE PES Power Systems Conference and Exposition. IEEE; 2006. p. 741–747.
[16] Berry AM, Cornforth DJ, Platt GM. The distributed generator placement and sizing test suite and analysis tool. In: IEEE PES Power Systems Conference and Exposition; 2009. p. 1–9.
[17] Bonami P, Biegler LT, Conn AR, Cornuéjols G, Grossmann IE, Laird CD, et al. An algorithmic framework for convex mixed integer nonlinear programs. Discrete Optimization. 2008;5(2):186 – 204. Available from: http://www.sciencedirect.com/science/article/pii/S1572528607000448.
[18] Bergen AR, Vittal V. Power Systems Analysis. 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.; 2000.
[19] Gross CA. Power System Analysis. 2nd ed. New York: John Wiley & Sons; 1986.
[20] Wood AJ, Wollenberg BF. Power generation, operation, and control. New York: John Wiley & Sons; 1996.
[21] Zimmerman RD, Murillo-Sánchez CE, Thomas RJ. MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education. IEEE Trans Power Syst. 2011 February;26(1):12–19.
[22] Wolsey LA. Integer programming. New York: Wiley; 1998.
[23] Kaye R, Wu F. Analysis of linearized decoupled power flow approximations for steady-state security assessment. IEEE Trans Circuits Syst. 1984 Jul;31(7):623 – 636.
[24] Purchala K, Meeus L, Van Dommelen D, Belmans R. Usefulness of DC power flow for active power flow analysis. In: IEEE Power Engineering Society General Meeting. vol. 1; 2005. p. 454–459.
[25] Rechtschaffen EEM. Algebraic properties of bus and line power equations with applications in network planning and analysis. In: Proceedings of the Sixth Power Systems Computation Conference, Darmstadt. IPC Science and Technology Press; 1978. p. 184–187.
[26] Horst R, Pardalos PM, Thoai NV. Introduction to global optimization. Kluwer Academic Publishers; 2000.
[27] Bao X, Sahinidis NV, Tawarmalani M. Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs. Optimization Methods and Software. 2009;24(4–5):485–504. Available from: http://www.tandfonline.com/doi/abs/10.1080/10556780902883184.
[28] Klos A, Wojcicka J. Physical aspects of the nonuniqueness of load flow solutions. International Journal of Electrical Power & Energy Systems. 1991;13(5):268–276. Available from: http://www.sciencedirect.com/science/article/pii/0142061591900506.
[29] Jarjis J, Galiana FD. Quantitative Analysis of Steady State Stability in Power Networks. IEEE Trans Power App Syst. 1981 January;PAS-100(1):318–326.
[30] Makarov YV, Dong ZY, Hill DJ. On Convexity of Power Flow Feasibility Boundary. IEEE Trans Power Syst. 2008 May;23(2):811–813.
[31] Leyffer S, Sartenaer A, Wanufelle E. Branch-and-Refine for Mixed Integer Nonconvex Global Optimization. Mathematics and Computer Science Division, Argonne National Laboratory; 2008.
[32] D'Ambrosio C, Lee J, Wächter A. A Global-Optimization Algorithm for Mixed-Integer Nonlinear Programs Having Separable Non-convexity. In: Fiat A, Sanders P, editors. Algorithms - ESA 2009. vol. 5757 of Lecture Notes in Computer Science. Berlin: Springer; 2009. p. 107–118.

[33] ILOG I. CPLEX Version 12 User's Manual. IBM; 2009.
[34] Mendoza F, Bernal-Agustin JL, Dominguez-Navarro JA. NSGA and SPEA Applied to Multiobjective Design of Power Distribution Systems. IEEE Trans Power Syst. 2006 November;21(4):1938–1945.
[35] Bandler JW, El-Kady MA. A New Method for Computerized Solution of Power Flow Equations. IEEE Trans Power App Syst. 1982 January;101(1):1–10.
[36] Chiradeja P, Ramakumar R. An approach to quantify the technical benefits of distributed generation. IEEE Trans Energy Convers. 2004 December;19(4):764 – 773.
[37] Gozel T, Hocaoglu MH, Eminoglu U, Balikci A. Optimal placement and sizing of distributed generation on radial feeder with different static load models. In: International Conference on Future Power Systems; 2005. p. 1–6.
[38] Dolan ED, Moré JJ. Benchmarking optimization software with performance profiles. Mathematical Programming. 2002;91:201–213. 10.1007/s101070100263. Available from: http://dx.doi.org/10.1007/s101070100263.
[39] Higham DJ, Higham NJ. MATLAB guide. Society for Industrial Mathematics; 2005.
[40] Fourer R, Gay DM, Kernighan BW. AMPL: A modeling language for mathematical programming. 2nd ed. Pacific Grove: Brooks/Cole; 2003.
[41] Bonami P, Lee J. BONMIN Users Manual. COIN-OR; 2011.
[42] Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming. 2006;106:25–57. 10.1007/s10107-004-0559-y. Available from: http://dx.doi.org/10.1007/s10107-004-0559-y.
[43] Taylor JA, Hover FS. Linear Relaxations for Transmission System Planning. IEEE Trans Power Syst. 2011 Nov;26(4):2533–2538.

E-mail address: james.foster@uon.edu.au