The Bee-Benders Hybrid Algorithm with application to Transmission Expansion Planning

Cameron A.G. MacRae
Melih Ozlen
School of Science, RMIT University
Melbourne, Australia

Andreas T. Ernst
School of Mathematics, Monash University
Melbourne, Australia

ABSTRACT

This paper introduces a novel hybrid optimisation algorithm that combines elements of both metaheuristic search and integer programming. This new matheuristic combines elements of Benders decomposition and the Bees Algorithm, to create the Bee-Benders Hybrid Algorithm (BBHA) which retains many of the advantages of both methods. Specifically, it is designed to be easily parallelisable, to produce good solutions quickly while still retaining a guarantee of optimality when run for a sufficiently long time. The algorithm is tested using a transmission network expansion and energy storage planning model, a challenging and very large scale mixed integer linear programming problem. The BBHA is shown to be a highly effective hybrid matheuristic algorithm for this challenging combinatorial optimisation problem that performs at least as well as either Benders decomposition or the Bees Algorithm on their own, and significantly improves upon the individual approaches in many instances. While the paper demonstrates the effectiveness on an electricity network planning problem, the algorithm could be readily applied to any mixed integer linear program, and is expected to work particularly well whenever this has a structure that is amenable to Benders decomposition.

KEYWORDS

decomposition, matheuristic, optimisation, power transmission

ACM Reference Format:
Cameron A.G. MacRae, Melih Ozlen, and Andreas T. Ernst. 2021. The Bee-Benders Hybrid Algorithm with application to Transmission Expansion Planning. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3449726.3463158

1 INTRODUCTION

The need to solve large scale mixed integer programming problems arises in many applications. As the size increases, these problems become quickly too difficult to solve in a reasonable amount of time. In such cases it is common to use metaheuristics or to attempt decomposition approaches in order to obtain good quality solutions. In this paper we propose a new hybrid approach that draws both on the Bees Algorithm [23] and Benders Decomposition [4]. This paper uses the planning of electricity networks to cope with renewable generation as an example to both motivate the need for the proposed new algorithm and to evaluate its effectiveness.

Integrating renewable energy generation, especially variable generators such as wind and solar, into the electrical transmission network is a considerable design challenge currently facing network planners. For example, a recent blackout in South Australia saw 315MW of wind generation disconnect from the grid amid voltage dips and loss of load [5]. As a consequence a 100MW battery was installed by Tesla to prevent this from happening again [16]. Correspondingly, there has been a renewed interest in electricity network planning problems [7]. One such problem is the transmission expansion planning problem (TEP). Here the objective is to minimise the investment and operational costs of the network while meeting a set of operational constraints, for example, generation, demand, geographical, and environmental constraints [18]. This leads to very large scale combinatorial optimisation problems that require making expensive long-term capital investment decisions (building transmission lines or installing energy storage facilities) while considering the use of these to meet variable demand. For a recent review of this application area see [7].

The TEP and related problems are often modeled as Mixed Integer Programs (MIPs) with either non-linear constraints or more commonly a linear approximation. Advances in MIP solvers mean small to medium sized instances can generally be solved to optimality within a few minutes. However, a considerable body of research is dedicated to solving larger or more complicated instances. Novel approaches to these problems include branch and bound with a GRASP meta-heuristic [3], Projection-Adapted Cross Entropy [12], particle swarm optimisation [1], and a hybrid with a variant of the bat algorithm [21]. Of particular interest for this paper is that often TEP problems can be decomposed into investment and operational subproblems. Benders decomposition, with alternately continuous or discrete decision variables in the master (investment) problem and DC approximation or transportation operational subproblems, has been shown to be very effective for many variants of TEP problems over many years, for some examples see [8, 14, 20]. A survey of the literature on optimisation methods for TEP is given in [27].

Where the models become larger and more complex MIP solvers can struggle to find good solutions and metaheuristic methods are often employed. Some recent examples, include constructive heuristics [2], the social spider algorithm [10] and bat algorithm [21]. However, as noted by the recent survey paper of metaheuristics used in this area, at least for single-objective models pure metaheuristic approaches published in this area often fail to out-perform mathematical programming based methods [6]. Hence in this paper

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3463158

we explore a novel *matheuristic* that combines the strengths of both matehmatical (integer) programming and metaheuristics.

Of interest here are methods that combine Benders decomposition with metaheuristic algorithms. Benders decomposition breaks large MIPs into a master problem and subproblems. The subproblems are used to evaluate and test the feasibility of solutions proposed by an optimisation process solving the master problem. Information from the subproblems is also fed back to the master in terms of additional constraints (cuts) generated from the subproblem solutions. While the original idea of Benders decomposition algorithm goes back to 1962 [4], there have been a lot of advances in recent years that improved the effectiveness of the approach in practice. For a recent review see [25]. There have also been a limited number of attempts to combine Benders decomposition with metaheuristic approaches. Poojari and Beasley [24] created a method based on Benders decomposition where the master problem is always solved with a Genetic Algorithm (GA) while the subproblem is solved using Linear Programming (LP) as in the standard Benders approach. When tested on general MIPs, this proved to be more effective than using the Benders method on its own but often less effective than simply using an MIP solver without any decomposition. A little earlier Sirikum et al. [26] independently developed a nearly identical GA-Benders hybrid and applied it to a problem in power generation expansion planning, a similar problem to the one considered here but only looking at power generation investment without network transmission or energy storage.

In this paper, we present a hybrid exact/metaheuristic algorithm that melds Benders decomposition and a Bees Algorithm (BA) [23] inspired approach. Unlike most other matheuristics, this method retains the ability to generate provably optimal solutions. The ideas of multiple bees that have different functions, scouts and workers, is used to balance diversification and intensification in the solution of the master problem. Multiple parallel optimisation processes are used to speed up the search, that learn from each other not just through the exchange of good feasible solutions but also by sharing dual (cut) information obtained when solving Benders subproblems. Using the transmission network expansion and energy storage planning model (TESP) to test the model, we show the Bee-Benders hybrid algorithm (BBHA) to be an effective hybrid algorithm that exhibits equivalent performance to its component parts in the segments of the problem domain where those parts are strongest, and significantly improves upon the individual approaches where neither component part has a pronounced advantage.

The rest of this paper is organised as follows. The Bee-Benders algorithm is introduced in Section 2. A MIP formulation of the TEP with storage model is given in Section 3. Numerical results, in which the algorithm is evaluated using the Brazilian 46-bus and Colombian 93-bus test systems are discussed in Section 4. We conclude in Section 5.

2 THE BEE-BENDERS HYBRID ALGORITHM

2.1 The Bees Algorithm

Here we present a hybrid exact/metaheuristic algorithm that combines Benders decomposition with an approach inspired by the Bees Algorithm. There are many variants of optimisation metaheuristics inspired by the behaviour of bees (see for example [15] for a review of one of the alternatives, the Artificial Bee Colony optimisation). Here we will follow the Bees Algorithm as proposed by [22, 23].

Algorithm 1: Bees Algorithm

Parameters: ns: no. of scout bees
$nre \geq nrb$: no. of recruited bees per elite/best site
$ne \leq nb$: no. of elite / best sites, $nb \leq ns$
$ngh, stlim$: initial size of nghbhd. & stagnation limit

Generate initial sites S = set of ns random solutions
while *not* stopping condition met **do**
  Evaluate the fitness (objective) of all sites in S
  Let $E, B \subset S$ be the ne fittest & $nb - ne$ next fittest sites
  foreach *solutions (sites)* $s \in E \cup B$ **do**
    Evaluate nre (or nrb) neighbours of s if $s \in E$ (resp. $s \in B$)
    if *better solution found* **then**
      Replace s with the best solution found
    else
      Reduce the neighbourhood size ngh
      Delete site s if no improvement for $stlim$ iterations
  Let $S := E \cup B$ and add random solutions until $|S| = ns$

In the most basic form, the algorithm comprises two phases: global search, and local search. A pseudo-code is given in Algorithm 1. Each solution is referred to as a flower or site in the terminology of the Bees Algorithm, and the local neighbourhood of a solution is called a flower patch. In the initialisation phase, "scout" bees leave the hive and fly to a random flower. The fitness of the flower is evaluated and the scout bees return to the hive. During the local search phase, the scouts who discover the ne elite and the nb best flowers (solutions) recruit "worker" bees to explore their respective flower patches, that is, the flowers in the neighbourhood of those the scouts discovered. Recruited worker bees fly to a random flower within the flower patch and evaluate its fitness. The fittest flower from the elite and best flower patches are combined with the fittest new flowers discovered by scouts during the subsequent global search phase to produce a new pool of elite and best solutions for further local exploration. Stopping conditions may include time, the number of iteration, or a test for convergence.

This can be thought as a multi-start local search algorithm which always works on a subset of best known solutions, with more effort expended in the neighbourhood of the elite solutions than the remaining solutions. This algorithm can be parallelised in a fairly straight forward manner by carrying out the search in each flower patch (neighbourhood of an elite or best solution) in parallel. Also the "scouting", that is, generation of new random solutions, can be carried out independently.

We have chosen to combine the BA with Benders decomposition because it has been shown to perform at least as well as standard evolutionary approaches, to be less sensitive to tuning parameters than other swarm approaches such as PSO, and yet retains a simplicity of implementation [23].

2.2 Benders Decomposition

Benders decomposition is a technique that allows a large, intractable problem, such as the TESP model described in Section 3, to be divided into more tractable component parts [4]. The first part is called the *master* problem and consists of a MIP that includes all of the integer variables and any applicable continuous variables. The second part consists of one or more *subproblems* that collectively contain the remaining continuous variables. The master problem yields a candidate solution of all of the integer variables. The subproblem(s) involving continuous variables only provide both an evaluation of the objective (as in black-box optimisation), and using the dual solution as a feasibility or optimality cut. In the problem considered here only optimality cuts exist, which effectively create a piece-wise linear surrogate function in the master problem. Particularly if $\boldsymbol{y}$ is the vector of variables of the master problem with z a variable representing the objective of the subproblem, then the subproblem generates optimality cuts of the form $z \geq r - \boldsymbol{b}^T\boldsymbol{y}$, i.e. the surrogate function is $z_C(\boldsymbol{y}) = \max_{c\in C}\{r_c - \boldsymbol{b}_c^T\boldsymbol{y}\}$ where C is the set of cuts. Typically this is done using a *branch-&-cut* approach where a single branch and bound tree of the master MIP is solved, with subproblems at each node of the tree solved to dynamically add cuts to the master problem (see [20] for details).

2.3 The hybrid method

The Bee-Benders hybrid algorithm (BBHA) is a hybrid of Benders decomposition and a local search phase that is largely based on the Bees Algorithm. The algorithm operates on a large MIP that has been decomposed in a manner suitable for Benders decomposition. The master problem contains binary variables representing certain investment decisions, and an LP subproblem containing continuous variables and largely operational constraints. The particulars of the mathematical model detailed in Section 3.

As with the BA, the algorithm comprises global search and local search phases. The global search phase commences as a conventional Benders decomposition using a "single tree" master approach. Lazy constraint callbacks are used to separate Benders cuts, as opposed to solving the master problem to optimality at each iteration. This single tree branch and bound search fulfills the role of the "scout" bee in the BA algorithm, ensuring that eventually the whole solution space is searched and the method can never remain stuck in a local optimum. It is assumed that the scout bee run in parallel to the other bees. Meanwhile, an initial set of random solutions is generated for exploration during the local search phase.

During the local search phase, "worker" bees (henceforth known as "workers") explore the local neighbourhood (subsequently referred to as a "site") of each solution, by estimating the fitness a subset of solutions using a surrogate function based on the set of known Benders cuts. The most heuristically promising solution discovered at the site is selected for full evaluation requiring solution of the subproblem LP. These worker bees can also run in parallel. As in the BA, the fittest solution from both the elite and best sites are combined with the incumbent solution of the scout bee to produce a new pool of solutions for further local search. The algorithm iterates in this way until stopping condition is met, or the Benders decomposition finds and proves the optimal solution. A pseudo-code description of the BBHA is given in Algorithm 2.

Algorithm 2: Bee-Benders hybrid algorithm

Parameters: $ne \leq nb$: no. of elite / best sites
  $nre \geq nrb$: no. of recruited bees per elite/best sites
  $ngh > 0$: max. Hamming distance defining a neighbourhood

Generate initial sites S = set of $nre + nrb$ random solutions
Let C be the set of Benders cuts
Start solving Benders decomposition with branch-&-cut, separating Benders cuts c at each node, $C := C \cup \{c\}$
foreach *solutions (sites)* $s \in S$ **do**
  Evaluate the fitness of s by solving the subproblem(s)
  Separate a Benders cut c and let $C := C \cup \{c\}$
while *not* stopping condition met **do**
  Let $E, B \subset S$ be the ne fittest & $nb - ne$ next fittest sites
  foreach *solutions (sites)* $s \in E \cup B$ **do**
    Heuristically evaluate nre solutions s' within distance ngh of s if $s \in E$ (or nrb solns if $s \in B$) using surrogate $z_C(s')$
    if *solution s' with (heuristic) fitness less than s* **then**
      Evaluate the fitness of s' by solving the subproblem(s)
      Separate a Benders cut c and let $C := C \cup \{c\}$
      Append s' to E (or B)
  Let $S := E \cup B \cup \{b\}$ where b is the incumbent solution of the branch-&-cut

2.3.1 *Initialisation.*

The algorithm is initialised with a population of $nre + nrb$ sites, which are distributed randomly with a uniform spread over the solution space. The fitness of each is evaluated exactly by solving the LP subproblem. Each LP subproblem also produces a Benders cut which is stored in a shared pool of cuts. The nb best sites are selected for neighbourhood search. Simultaneously, the algorithm commences solving the Benders decomposition using the "single tree" master problem approach: Lazy constraint callbacks are used to solve the LP subproblem and separate the cuts. This means that the master problem need only be solved to optimality once as opposed to once per iteration. Any generated cuts are added to the shared pool of cuts.

2.3.2 *The main loop.*

The main loop consists of two main phases: neighbourhood search and cut sharing. The neighbourhood search is carried out by each process independently, while the cut sharing represents a communication step between the processes. The cuts discovered by all processes are also shared at this point.

2.3.3 *Neighbourhood search.*

At each iteration, the workers that discovered the ne elite solutions, each recruit nre workers for neighbourhood search. Likewise, the workers who discovered the remaining $nb - ne$ best solutions each recruit nrb workers for neighbourhood search.

Neighbourhood search at a given site is performed by each worker producing a pool of candidate solutions using a Hamming distance function which randomly selects at most ngh binary variables to alter. For arbitrary MIPs with binary variables x_i for $i \in N$ and current solution s_i, this effectively imposes the constraint

$$\sum_{i \in N: s_i = 0} x_i + \sum_{i \in N: s_i = 1} (1 - x_i) \leq ngh.$$

In our application, a more specialised neighbourhood move can be defined based on the structure of the problem. We have binary variables that represent a unit increment in transmission capacity between two locations (a right of way). Each right of way has p binary variables denoting the installation of an equivalent line.

This means that individually installing the 1^{st} line is equivalent to installing the $2^{nd} \dots p^{th}$ line. Clearly there is no point in randomly replacing the installation of one line on a right of way with another. For this reason the neighbourhood move operates on groups of binary variables representing a single right of way and either adds or subtracts one or more lines.

The fitness of each candidate solution in the pool is estimated using the surrogate approach as shown in Algorithm 3. This is computationally very cheap, avoiding even the matrix multiplication for the case where the base cost in the master is already too high. Step 3 is simply computing the function $z_C(\hat{\boldsymbol{y}})$ from Section 2.2.

Each worker then solves the LP subproblem for the most promising heuristically determined solution in their solution pool, and the generated Benders cuts are added to the shared pool of known cuts. The fitness scores of the solutions found by the workers are combined with the incumbent solution of the Benders decomposition and are ranked from best to worst. The *nb* best solutions are selected for neighbourhood search during the next iteration.

2.3.4 Cut sharing. At the conclusion of the neighbourhood search phase any Benders cuts produced by worker or scout bees are added to the shared pool of cuts. These may be added to the MIP solver via the lazy constraint call-back in the branch-&-cut. The effect of the cut sharing is that each of the bees has a more accurate approximation of the objective function. With this approximation the scout bee avoids searching any solution that is not at least potentially better than the best found so far. The worker bees use the approximation to quickly evaluate solutions and to identify the most promising solution for which a full LP is solved.

2.3.5 Termination. The algorithm may terminate in several ways: After n^{max} iterations, t^{max} seconds, or if the Benders decomposition identifies and proves the optimal solution. Note that even if the time or iteration limit does not allow a provably optimal solution to be found, we are still able to extract a lower bound from the branch and bound tree that the scout bee has been searching. Thus even in the case where the method is only a heuristic, we have an estimate of the maximum gap to the globally optimal solution.

3 MATHEMATICAL MODEL OF TESP

We consider a electricity network consisting of nodes and arcs (referred to as "rights of way" in the literature or as a *link* in this paper). The ability to generate power and demand for power occurs at the nodes over a number of time periods. The objective of the complete TESP model is the is to minimise the investment cost of expanding the transmission network while simultaneously minimising a penalty for load curtailment at nodes with net demand. A discrete number of new or reinforcing circuits may be installed on each link, and the size of any Energy Storage System (ESS) at the nodes are determined.

Algorithm 3: Heuristic fitness evaluation

Data: $\hat{\boldsymbol{y}}$: candidate solution with costs $\boldsymbol{c}$ in the master

$B, \boldsymbol{r}$: coefficient matrix & constants vector of the Benders cuts

```
1 Let z := c^T ŷ
2 if z < fitness of the current incumbent then
3 |  Let v := r − Bŷ and let z := z + max_i{v_i}
4 return z
```

Cyclic discrete time is used to model the period of operation, and therefore the state of any installed ESS in the last time interval must be identical to the state in the initial time interval. This might model the typical power use over a day with the end of one day matching the start of the next. Generation is re-dispatchable and demand may vary between time intervals. Despite the introduction of time to the model, the planning is static, and only a single final expansion plan is produced. More complex models, for example with multiple scenarios for demand and renewable power generation capacity, are possible but not considered in this paper. The model determines the network expansion plan in the master problem and the operational characteristics (power flows, phase angles, energy storage) in the subproblem. We follow the common practice of modelling power flows using a DC approximation [17, p.36].

The mathematical model presented here and alternative modelling approaches in the literature, are discussed in detail in [19, 20]. As such, only an abridged discussion of the decomposed model follows. The key point to note here is that the main integer (binary) variables to be determined relate to the links to be installed, while a very large number of continuous variables and associated constraints have to be considered to determine the optimal generation, storage and power flows for any choice of network expansion.

The following notation will be used throughout this paper to define the TESP:

Sets:
- Γ the set of indices for buses;
- Ω_0 the set of rights of way for existing circuits;
- Ω_c the set of rights of way for candidate circuits;
- Ψ the set of uniform time intervals $\{1, 2, \dots, T\}$;

Parameters:
- α_{tk} cost of curtailment at time t at bus k;
- b_k cost of installing storage at bus k;
- c_{ij} cost of installing a circuit on link ij;
- d_{tk} demand at time t at bus k;
- $\bar{f}_{ij}$ maximum possible power flow on link ij;
- $\bar{g}_k$ maximum possible generation at bus k;
- γ_{ij} susceptance of circuits installed on link ij;
- M_{ij} the disjunctive parameter for link ij
- n^0_{ij} number of existing circuits on link ij;
- $\bar{n}_{ij}$ maximum installable circuits on link ij;
- $\bar{x}_k$ maximum installable storage capacity at bus k;
- $\hat{y}^p_{ij}$ master solution = 1 if the p^{th} circuit is installed on ij;

Decision variables:
- β_{tk} power flow to storage at bus k at time t;
- g_{tk} generation at time t at bus k;
- f^0_{tij} power flow for existing circuits at time t on link ij;
- f^p_{tij} power flow for the p^{th} circuit at time t on link ij;
- l_{tk} level of storage at bus k at time t;
- r_{tk} demand curtailment at time t at bus k ;
- θ_{tk} phase angle at time t at bus k;
- x_k storage capacity installed at bus k;
- y^p_{ij} binary variable = 1 if p^{th} circuit installed on link ij;
- v contribution of the subproblem to the master objective

3.1 The master problem

The objective of the master problem is to minimize the function

$$\sum_{(i,j)\in\Omega_c} c_{ij} y_{ij}^p + z \tag{1}$$

where c_{ij} is cost of installing a line on link ij and y_{ij}^p is a binary variable denoting the installation of the p^{th} circuit on ij. The estimated cost of the subproblem is given by z. The master initially only has symmetry breaking constraints (2) and basic variable bounds (3), with additional Benders cuts added incrementally:

$$y_{ij}^p \geq y_{ij}^{p+1} \qquad \forall\ (i,j)\in\Omega_c,\ p=1\ldots\bar{n}_{ij}-1 \tag{2}$$

$$z \geq 0, \quad y_{ij}^p \in \{0,1\} \qquad \forall\ (i,j)\in\Omega_c,\ p=1\ldots\bar{n}_{ij} \tag{3}$$

3.2 The subproblem

Given a set of new circuit installations determined by the master problem, the subproblem determines the cost of any installed ESS, and a penalty for load curtailment.

The objective of the subproblem is to minimise the function

$$z = \sum_{k\in\Gamma} b_k x_k + \sum_{t\in\Psi}\sum_{k\in\Gamma} \alpha_{tk} r_{tk} \tag{4}$$

where b_k is the fixed cost of installing x_k MW of storage at bus k, and αtk the cost of curtailing r_{tk} in each time interval t. It is assumed that the variable operating cost of ESS is low relative to fixed costs, and these are therefore omitted from the objective function.

The following technical constraints govern the operation of the network. *Nodal balance and power flow*:

$$d_{tk} = \zeta + g_{tk} + r_{tk} - \beta_{tk} \quad \forall\ t\in\Psi,\ \forall\ k\in\Gamma \tag{5}$$

$$\zeta = \sum_{(i,k)\in\Omega_0} f_{tik}^0 - \sum_{(k,j)\in\Omega_0} f_{tkj}^0 + \sum_{p=1}^{\bar{n}_{ij}}\sum_{(i,k)\in\Omega_c} f_{tik}^p - \sum_{p=1}^{\bar{n}_{ij}}\sum_{(k,j)\in\Omega_c} f_{tkj}^p$$

Nodal balance i.e. Kirchhoff's current law is ensured for each time interval by constraint (5). Power flows are modeled using a DC approximation requiring that the phase angle at each bus be determined for each time interval:

$$f_{tij}^0 - \gamma_{ij} n_{ij}^0 \left(\theta_{ti} - \theta_{tj}\right) = 0 \qquad \forall\ t\in\Psi,\ (i,j)\in\Omega_0 \tag{6}$$

$$\left| f_{tij}^p - \gamma_{ij}(\theta_{ti} - \theta_{tj}) \right| \leq M(1-\hat{y}_{ij}^p)\ \forall t\in\Psi, (i,j)\in\Omega_c, p\leq\bar{n}_{ij} \tag{7}$$

Kirchhoff's voltage law is implemented for existing and candidate circuits by (6), and (7). Here the absolute value function can be linearised and M is simply a sufficiently large number. Thermal limits on existing and candidate circuits are enforced by:

$$\left| f_{tij}^0 \right| \leq n_{ij}^0 \bar{f}_{ij} \quad \forall\ t\in\Psi,\ \forall\ (i,j)\in\Omega_0 \tag{8}$$

$$\left| f_{tij}^p \right| \leq \hat{y}_{ij}^p \bar{f}_{ij} \quad \forall\ t\in\Psi,\ \forall\ (i,j)\in\Omega_c,\ \forall\ p=1\ldots\bar{n}_{ij} \tag{9}$$

Storage level and charge/discharge limits with "wrap around" are implemented by the following constraints where $l_{0k} = l_{Tk}$:

$$l_{tk} = l_{t-1,k} + \beta_{tk} \quad \forall\ t\in\Psi,\ \forall\ k\in\Gamma \tag{10}$$

$$0 \leq l_{tk} \leq x_k \quad \forall\ t\in\Psi,\ \forall\ k\in\Gamma \tag{11}$$

$$0 \leq x_k \leq \bar{x}_k \quad \forall\ k\in\Gamma \tag{12}$$

Constraint (12) establishes bounds on the installable storage capacity at bus k, while constraint (11) ensures the stored energy does not exceed the installed capacity.

Bounds: Constraint (13) imposes bounds on generator dispatch, while load curtailment is limited by demand. Note that the f_{tij}^0, f_{tij}^p, β_{tk} and θ_{tk} variables are unbounded.

$$0 \leq g_{tk} \leq \bar{g}_k \quad \forall\ t\in\Psi,\ \forall\ k\in\Gamma \tag{13}$$

$$0 \leq r_{tk} \leq d_{tk} \quad \forall\ t\in\Psi,\ \forall\ k\in\Gamma \tag{14}$$

3.3 Optimality cut

As noted above, load curtailment is permitted at any bus during any time interval so long as it does not exceed demand at that bus during the same time period. Therefore, the dual of the subproblem remains bounded for any feasible solution to the master problem. Accordingly, we only need to consider the following optimality cut:

$$\begin{aligned} z \geq & \sum_{t\in\Psi}\sum_{(i,j)\in\Omega_c} \left(\pi_{\gamma_{tij}^{+p}} + \pi_{\gamma_{tij}^{-p}}\right)\left(M_{ij}(1-y_{ij}^p)\right) + \sum_{t\in\Psi}\sum_{k\in\Gamma} d_{tk}\pi_{r_{tk}} \\ & + \sum_{t\in\Psi}\sum_{(i,j)\in\Omega_c} \left[\pi_{f_{tij}^{+p}} y_{ij}^p \bar{f}_{ij} + \pi_{f_{tij}^{-p}} y_{ij}^p \bar{f}_{ij}\right] + \sum_{t\in\Psi}\sum_{k\in\Gamma} d_{tk}\pi_{d_{tk}} \\ & + \sum_{t\in\Psi}\sum_{(i,j)\in\Omega_0} \left[\pi_{f_{tij}^{+0}} n_{ij}^0 \bar{f}_{ij} + \pi_{f_{tij}^{-0}} n_{ij}^0 \bar{f}_{ij}\right] + \sum_{t\in\Psi}\sum_{k\in\Gamma} \bar{g}_k \pi_{g_{tk}} + \sum_{k\in\Gamma} \bar{x_k}\pi_{x_k} \end{aligned} \tag{15}$$

Here the dual variables $\pi_{d_{tk}}$ correspond to constraint (5), $\pi_{\gamma_{tij}}$ to (6), $\pi_{\gamma_{tij}^{+p}}$ & $\pi_{\gamma_{tij}^{-p}}$ to (7), $\pi_{f_{tij}^{+0}}$ & $\pi_{f_{tij}^{-0}}$ to (8), and $\pi_{f_{tij}^{+p}}$ & $\pi_{f_{tij}^{-p}}$ to (9). The dual variables $\pi_{s_{tk}}$ correspond to constraints (10). Lastly, duals $\pi_{\bar{l}_k}$, $\pi_{g_{tk}}$, $\pi_{r_{tk}}$, and π_{xk} correspond to the bounds (11 - 14) respectively.

3.4 Limitations

The relative simplicity of this TESP formulation comes at the cost of addressing certain features of a real world electrical transmission network. The most obvious limitation is that power flows are modelled using a DC approximation to the AC power flow of most transmission networks.

It is also assumed that the variable operating cost of ESS is negligible, at least relative to the fixed cost of installing and operating the ESS over its lifetime, and that fixed costs increase linearly with capacity. Power flow to and from ESS is limited only by the total capacity and current level of the storage. The model allows that the storage completely charge or discharge within a single time interval. Furthermore, the model ignores losses during storage or transmission of power. The model also assumes generator dispatch does not incur costs, and that generators are not subject to technical constraints such as ramp rate limits.

While it is possible to address these and other limitations of the model with additional variables and constraints, these come at the cost of significant complexity in both notation and implementation. Here we have sought to balance to the realism of the modelling with the intent to use the model simply to demonstrate the use of the algorithmic approach.

4 NUMERICAL RESULTS

In each of the numerical experiments described in this section the model is implemented in Python and CPLEX 12.6.3 is used as the MIP solver. Parallelisation is achieved using multiple processes, not threading. The Benders decomposition is implemented with a "single tree" master using lazy constraint callbacks. While the LP solver may take advantage of multi-threading, the branch-&-cut is single threaded.

4.1 Parameter tuning

There are a number of parameters to the BBHA algorithm which may be tuned to find a set of default values that empirically demonstrate good performance. These are given in Table 1. The IEEE-25 bus test system is used to benchmark combinations of parameters presumed likely to perform well. A schematic and tabulated data are available in [9]. The system has 25 buses and 36 rights of way (links) with a total demand of 2750 MW. Without storage, and permitting a maximum of 4 new or reinforcing circuits on each link, the cost of the optimal expansion plan is US$107.7 million. An arbitrary cost coefficient of US$2000/MWh is used for storage in each network tested. Under the long peak scenario in Figure 1 the cost of the optimal expansion plan is US$43.8 million. This result is the benchmark objective for the parameter tuning.

In this tuning exercise, 34 sets of parameters are compared over the first 1800 seconds (30 minutes) of the optimisation. Rather than just comparing the final solution quality, we integrate the objective along the time axis with a composite trapezoidal rule and then rescale against the worst (largest) integral to obtain a *scaled trapz* score. The best result is that with lowest value. Only 5 sets of parameters find the optimal solution within the 30 minute window ([ne,nb,nre,nrb] = [1,2,30,10], [1,2,30,15], [2,3,20,10], [3,4,10,5], and [3,4,20,15]). Figure 2 plots the best objective against time to demonstrate the sensitivity of BBHA to the choice of parameters.

The Benders scout ensures that the BBHA is guaranteed to find the exact optimal solution to the problem given sufficient time to run to completion. Of course this may take a significant amount of time. The objective of the BBHA is to discover high quality solutions quickly, and as such we favour parameter sets which rapidly converge to such solutions in the case studies that follow. We explore the performance of three sets of [ne,nb,nre,nrb] parameters:

2,3,10, 5 : parameters with the smallest scaled trapz score.
1,2,30,10 : fastest convergence to the optimal solution.
1,2,10, 5 : parameters with the largest number of iterations.

The final parameter to consider is the size of the neighbourhood for local search *ngh* (Section 2.3.3). Figure 3 shows the distribution of the hamming distance over the range 1-10 required to produce the best improved solution of using 14000 workers. A value of 2 accounts for the largest number of improved solutions. This is unsurprising as it reflects a typical swap move of replacing one circuit with another. Given that the long tail of larger hamming distances mainly occurs at the beginning of the optimisation, the value of *ngh* was kept at 8 for all case studies.

Table 1: Default Parameters for the BBHA.

Name	Description	Val.
ne	number of elite sites	1
nb	number of best sites	2
nre	recruited bees for elite sites	10
nrb	recruited bees for remaining best sites	5
ngh	maximum size of neighbourhood for local search	8

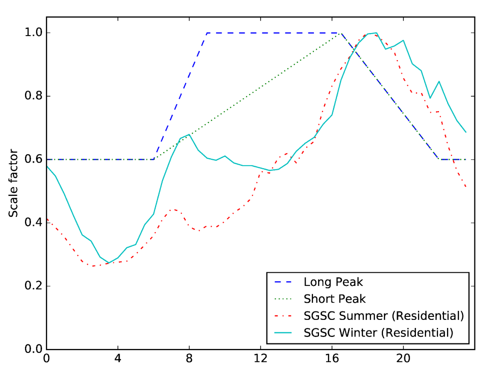

Figure 1: Load profiles used for each case study. (48× 30 min).

4.2 Results

The algorithms were tested on two larger networks, the Brazillian network with 46 buses, 79 links, total demand 6880MW, $\bar{n}_{ij} = 5$ (see [13]) and the Colombian network with 93 buses, 155 links total demand 14559MW, $\bar{n}_{ij} = 4$ (see [11]). Network expansion plans are optimised for four scenarios over a 24 hour period (see Figure 1).

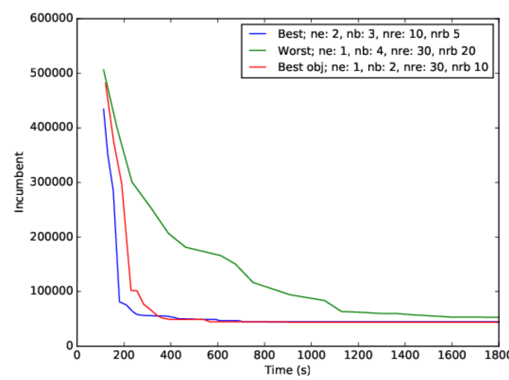

Figure 2: Best and worst parameter sets for IEEE 25-bus network and long peak scenario. The "Best" produces the best solution quality on average over the run, while "Best obj" produces the optimal solution the fastest for this instance.

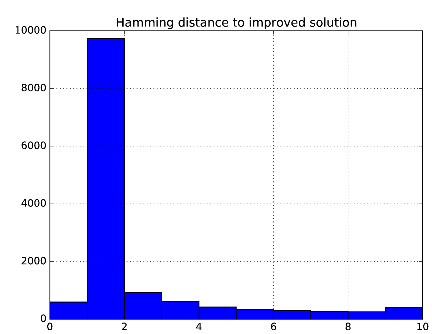

Figure 3: Histogram showing the hamming distance to best improved solution.

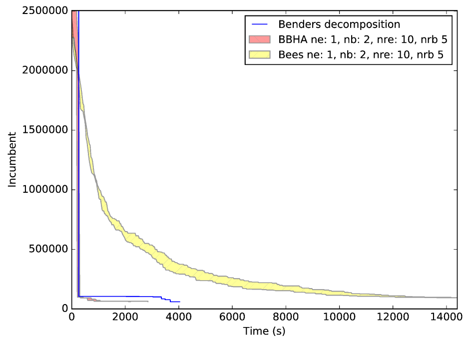

Figure 4: 46-bus SGSC Winter scenario, parameters:1,2,10,5

Each scenario is optimised $N = 5$ times for both the BBHA and Bees algorithm, and once using Benders decomposition which as a deterministic method exhibits little variance. Each run is limited to 4 hours. Results are shown in Table 2.

For the 46-bus instances, the parameter set [1,2,10,5] typically matches or exceeds the mean performance of the other parameter sets under investigation for the BBHA, whereas the parameter set [1,2,30,10] exhibits better performance for the BA. The BBHA finds the optimal solution for the short peak and SGSC summer and SGSC winter demand scenarios, and the Benders scout is able to prove optimality (as does the branch-&-cut run). For the long peak scenario none of the methods are able to prove optimality. The range of incumbent solution values over time is shown in Figure 4 for the SGSC winter scenario, demonstrating the ability of BBHA to converge more quickly than the pure Benders approach.

For the 93-bus test system the parameter set [1,2,10,5] exhibits consistently good performance for both the BBHA and BA. The BA achieves a lower mean for the long peak scenario. The Benders decomposition tends to lag behind both approaches for all scenarios except the SGSC winter demand profile. However, the [1,2,30,10] parameter configuration also does well. Insight into typical convergence behaviour is give in Figures 5 and 6.

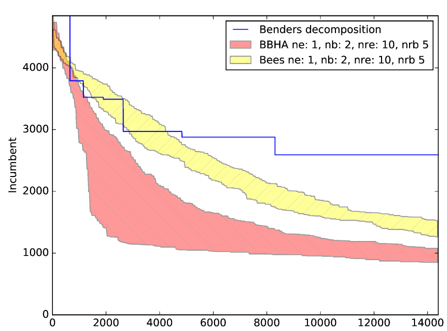

Figure 5: 93-bus SGSC Summer scenario, parameters:1,2,10,5

In instances, which Benders decomposition terminates with an optimal solution, the BBHA algorithm typically not only discovers the optimal solution heuristically well in advance of the Benders decomposition, but is also able to prove optimality prior. This is the case for the 46-bus network with the summer and winter scenario. However for the short-peak scenario BBHA, while still proving optimality, takes slightly longer than the pure Benders approach

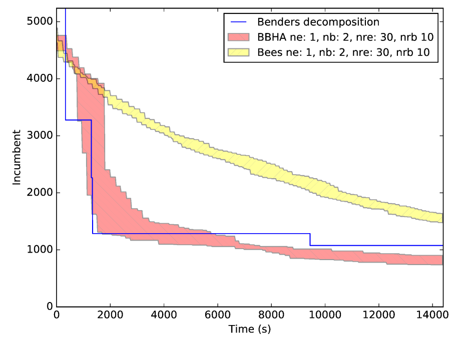

Figure 6: 93-bus SGSC Winter scenario, parameters:1,2,30,10

In other instances, where Benders struggles to find any good solutions, BBHA sometimes performs comparable to BA. This is the case for example with the 93-bus long-peak scenario in which BA has a slightly better mean performance, though the differences are relatively minor.

Like any other hybrid approach the BBHA is a compromise. A straight Benders decomposition implementation running on the same computing infrastructure will evaluate more of the search tree than the BBHA scout. Likewise, without the continuously running scout or the trade off between producing Benders cuts and local search the BA approach can dedicate more cores to evaluating candidate solutions. As a result, the straight Benders approach tends to outperform the BBHA in terms of the lower bounds it produces. However since these are generally of much less interest than finding good feasible solutions, and because the Bees algorithm is unable to produce any lower bounds, we have restricted ourselves to only compare upper bound solution quality in this analysis.

However, empirically we have found the benefits of cut sharing largely negate any compromise. In the first instance, the cuts generated by the Benders scout improve the heuristic estimate of the fitness of the candidate solutions in the worker solution pool. Likewise, the cuts generated in parallel by the elite workers are typically in the neighbourhood of the incumbent solution they prove useful to the Benders decomposition. Perhaps the clearest example of this is shown in Figure 4. Here, by sharing cuts between workers and the simultaneous Benders decomposition, the BBHA is able to prove the optimal solution faster than Benders decomposition alone, even though the Benders decomposition has a resource advantage on the compute infrastructure.

5 CONCLUSION

In this paper, we introduced a hybrid exact/metaheuristic algorithm that combines Benders decomposition and a Bees algorithm inspired approach. To the best of our knowledge this is the first such matheuristic based on Benders decomposition and the Bees algorithm. The approach has been tested using the transmission network expansion and energy storage planning model. This is a very challenging problem that has been shown previously to be intractable for standard MILP solvers [20]. The standard Benders decomposition into investment and operational subproblems, is able to solve some instances of our problem using branch-&-cut while finding only very low quality solutions for others. On the other hand the metaheuristic Bees Algorithm provides no proof of

Table 2: Results for the BBHA compared with the basic Bee Algorithm and Benders Decomposition on their own. The BBHA & Bee algorithm were run three times with different parameter settings for each instance.

Network	Scenario	Params	BBHA worst (US$$10^3$)	BBHA mean (US$$10^3$)	BBHA best (US$$10^3$)	Bee worst (US$$10^3$)	Bee mean (US$$10^3$)	Bee best (US$$10^3$)	Benders (US$$10^3$)
46-bus	Long peak	1 2 10 5	121,394.82	107,205.36	100,110.63	216,458.67	170,644.48	144,749.27	111,840.23
46-bus	Long peak	1 2 30 10	127,617.19	111,251.96	100,110.63	142,615.21	128,806.83	115,754.15	111,840.23
46-bus	Long peak	2 3 10 5	119,453.01	113,597.68	110,321.02	241,328.47	201,617.23	180,162.04	111,840.23
46-bus	Short peak	1 2 10 5	72,355.41	72,355.41	72,355.41	138,245.85	113,529.47	98,384.70	72,355.41
46-bus	Short peak	1 2 30 10	72,355.41	72,355.41	72,355.41	116,960.58	108,905.22	102,433.18	72,355.41
46-bus	Short peak	2 3 10 5	72,355.41	72,355.41	72,355.41	165,592.33	145,689.30	125,070.37	72,355.41
46-bus	SGSC summer	1 2 10 5	46,434.71	46,434.71	46,434.71	72,356.25	65,746.02	52,702.91	46,434.71
46-bus	SGSC summer	1 2 30 10	46,434.71	46,434.71	46,434.71	59,616.67	55,531.41	48,323.52	46,434.71
46-bus	SGSC summer	2 3 10 5	46,434.71	46,434.71	46,434.71	108,743.60	93,496.44	78,362.36	46,434.71
46-bus	SGSC winter	1 2 10 5	59,952.72	59,952.72	59,952.72	94,794.21	94,447.67	93,841.26	59,952.72
46-bus	SGSC winter	1 2 30 10	59,952.72	59,952.72	59,952.72	85,922.72	77,248.63	68,314.38	59,952.72
46-bus	SGSC winter	2 3 10 5	59,952.72	59,952.72	59,952.72	118,503.15	101,681.93	83,985.08	59,952.72
93-bus	Long peak	1 2 10 5	1,834.13	1,743.23	1,581.22	1,977.77	1,705.92	1,429.31	9,537.89
93-bus	Long peak	1 2 30 10	2,434.92	2,120.86	1,891.62	1,927.81	1,723.29	1,550.24	9,537.89
93-bus	Long peak	2 3 10 5	2,391.34	2,258.85	2,137.23	1,882.27	1,832.58	1,740.81	9,537.89
93-bus	Short peak	1 2 10 5	1,110.53	758.89	579.23	1,561.26	1,526.77	1,490.24	2,181.05
93-bus	Short peak	1 2 30 10	960.64	835.98	704.98	1,707.68	1,539.22	1,286.71	2,181.05
93-bus	Short peak	2 3 10 5	1,425.75	1,066.31	829.03	1,792.89	1,674.09	1,582.94	2,181.05
93-bus	SGSC summer	1 2 10 5	1,076.99	928.03	848.93	1,519.32	1,403.26	1,266.19	2,592.20
93-bus	SGSC summer	1 2 30 10	1,189.71	1,039.27	828.64	1,524.21	1,421.82	1,305.48	2,592.20
93-bus	SGSC summer	2 3 10 5	1,093.63	1,078.31	1,059.19	1,733.61	1,639.30	1,530.99	2,592.20
93-bus	SGSC winter	1 2 10 5	1,444.57	1,097.66	897.01	1,507.05	1,456.26	1,360.55	1,077.04
93-bus	SGSC winter	1 2 30 10	904.47	812.52	738.34	1,636.75	1,558.55	1,474.41	1,077.04
93-bus	SGSC winter	2 3 10 5	1,501.19	1,302.51	1,143.90	1,853.96	1,764.57	1,704.73	1,077.04

optimality and again is not able to reliably produce good solutions for this problem.

The BBHA exhibits the essential characteristics of a hybrid optimisation method. Where the problem is readily solved by one of the component optimisation methods the BBHA performs comparably at minimum. Where each component optimisation method performs similarly on a given problem, the hybrid approach exceeds this individual performance. In short, the whole is greater than the sum of its parts. The BBHA is general in nature and does not require any special problem structure beyond the decomposition required by the Benders decomposition method. Thus the approach may be applied to any decomposable mixed integer program.

REFERENCES

[1] J. Aghaei, K. M. Muttaqi, A. Azizivahed, and M. Gitizadeh. 2014. Distribution expansion planning considering reliability and security of energy using modified PSO (Particle Swarm Optimization) algorithm. *Energy* 65 (Feb. 2014), 398–411.

[2] F. A Assis, A. M. Leite da Silva, L. AF Manso, and J. FC Castro. 2021. Transmission expansion planning of large power networks via constructive metaheuristics with security constraints and load uncertainty analysis. *International Transactions on Electrical Energy Systems* 31, 1 (2021), e12595.

[3] L. Bahiense, G.C. Oliveira, M. Pereira, and S. Granville. 2001. A mixed integer disjunctive model for transmission network expansion. *IEEE Transactions on Power Systems* 16, 3 (2001), 560–565. https://doi.org/10.1109/59.932295

[4] J. F. Benders. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik* 4, 1 (1962), 238–252.

[5] C. Chang. 2016. South Australia blackout highlights energy security. *News Ltd* (Oct. 2016).

[6] G. Chicco and A. Mazza. 2021. *Metaheuristics for Transmission Network Expansion Planning*. Springer, Cham, 13–38.

[7] A.S. Dagoumas and N.E. Koltsaklis. 2019. Review of models for integrating renewable energy in generation expansion planning. *Applied Energy* 242 (2019), 1573–1587.

[8] K. Dilwali, H. Gunnaasankaraan, A. Viswanath, and K. Mahata. 2016. Transmission expansion planning using benders decomposition and local branching. In *2016 IEEE Power and Energy Conference at Illinois (PECI)*. 1–8.

[9] AO. Ekwue and B.J. Cory. 1984. Transmission system expansion planning by interactive methods. *IEEE Trans. on Power Apparatus & Sys.* 103 (1984), 1583–1591.

[10] MS El-Bages and WT Elsayed. 2017. Social spider algorithm for solving the transmission expansion planning problem. *Electric Power Sys. Res.* 143 (2017), 235–243.

[11] A. H. Escobar, R. A. Gallego, and R. Romero. 2004. Multistage and coordinated planning of the expansion of transmission systems. *IEEE Transactions on Power Systems* 19, 2 (May 2004), 735–744. https://doi.org/10.1109/TPWRS.2004.825920

[12] A. Eshragh, J. Filar, and A. Nazari. 2011. A Projection-Adapted Cross Entropy method for transmission network planning. *Energy Systems* 2, 2 (2011), 189–208.

[13] S. Haffner, A. Monticelli, A. Garcia, J. Mantovani, and R. Romero. 2000. Branch and bound algorithm for transmission system expansion planning using a transportation model. *Generation, Transmission and Distribution, IEE Proceedings-* 147 (2000), 149–156.

[14] S. Huang and V. Dinavahi. 2017. A branch-&-cut benders decomposition algorithm for transmission expansion planning. *IEEE Sys. Journal* 13 (2017), 659–669.

[15] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga. 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *The Artificial Intelligence Review* 42, 1 (2014), 21–57.

[16] F. Keck, M. Lenzen, A. Vassallo, and M. Li. 2019. The impact of battery energy storage for renewable energy power grids in Australia. *Energy* 173 (2019), 647–657.

[17] U.G. Knight. 1972. Some frequently used analytical techniques. In *Power Systems Engineering and Mathematics*, U.G. Knight (Ed.). Pergamon, 28 – 51.

[18] G. Latorre, R.D. Cruz, J.M. Areiza, and A. Villegas. 2003. Classification of publications and models on transmission expansion planning. *IEEE Transactions on Power Systems* 18, 2 (2003), 938–946.

[19] C. MacRae, M. Ozlen, and A. Ernst. 2014. Transmission expansion planning considering energy storage. In *International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. 1–5.

[20] C. A. G. MacRae, A. T. Ernst, and M. Ozlen. 2016. A Benders decomposition approach to transmission expansion planning considering energy storage. *Energy* 112 (2016), 795 – 803.

[21] C. A. Moraes, E. J. De Oliveira, M. Khosravy, L. W. Oliveira, L. M. Honório, and M. F. Pinto. 2020. A Hybrid Bat-Inspired Algorithm for Power Transmission Expansion Planning on a Practical Brazilian Network. In *Applied Nature-Inspired Computing: Algorithms and Case Studies*. Springer, Singapore, 71–95.

[22] DT Pham, E Koc, JY Lee, and J Phrueksanant. 2007. Using the bees algorithm to schedule jobs for a machine. In *Proc eighth int. conf. on laser metrology, CMM and machine tool performance*. Cardiff, UK, 430–439.

[23] D. T. Pham and M. Castellani. 2009. The Bees Algorithm: Modelling foraging behaviour to solve continuous optimization problems. *Proc. of the Inst. of Mechanical Engineers* 223, 12 (2009), 2919–2938.

[24] C. A. Poojari and J. E. Beasley. 2009. Improving benders decomposition using a genetic algorithm. *European Journal of Operational Research* 199, 1 (2009), 89–97.

[25] R. Rahmaniani, T.G. Crainic, M. Gendreau, and W. Rei. 2017. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research* 259, 3 (2017), 801–817.

[26] J. Sirikum, A. Techanitisawad, and V. Kachitvichyanukul. 2007. A New Efficient GA-Benders' Decomposition Method For Power Generation Expansion Planning With Emission Controls. *IEEE Trans. on Power Systems* 22, 3 (2007), 1092–1100.

[27] A. Sorokin, J. Portela, and P.M. Pardalos. 2012. Algorithms and Models for Transmission Expansion Planning. In *Handbook of Networks in Power Systems I*. Springer Berlin Heidelberg, 395–433.