

A Hybrid Cross-Entropy and Progressive Hedging Matheuristic with application to a RAPS System

A.T. Ernst^{a,c}, T. Moore^b, B. Owens^c and G. Singh^d

^a*School of Mathematical Sciences, Monash University, Clayton, Australia*

^b*CSIRO, Newcastle, Australia*

^c*CSIRO, Clayton, Melbourne, Australia*

^d*Technology, Geoscience and Engineering, BHP Billiton, 171 Collins St, Melbourne, Australia*

Email: Andreas.Ernst@monash.edu

Abstract: In many applications the design of a system needs to be optimised but the effectiveness of the design choices can only be evaluated by considering the behaviour of the system over a longer time period or multiple scenarios. Here we develop a general hybrid matheuristic method that can be used in such situations and apply it to a particular problem that arises in the design of a Remote Area Power Supply (RAPS) system in the presence of storage.

The major elements of our RAPS system include the load (demand for power), solar energy from photovoltaic (PV) panels, a diesel generator and a battery based storage facility. The aim is to find a fixed strategy for using the diesel generator and battery storage facility to efficiently meet demand. Similar types of problems arise in a range of other design applications such as expansion planning of electricity transmission networks; selection and sizing of power generation facilities in a RAPS or electricity grid; or supply chain design (sizing warehouse storage and transport infrastructure). All of these applications can be formulated as large MIP models with have a similar structure: high level design variables relating to infrastructure or policy, and additional variables for evaluating the effectiveness of these decisions in a number of scenarios.

In this paper we consider how to deal with such problems by combining ideas from progressive hedging, a Lagrangian decomposition based method, with the cross-entropy optimisation meta-heuristic. We describe the general structure of mixed integer programming (MIP) problems to which this applies, show how our application can be formulated in this structure and then describe the new hybrid matheuristic. Indicative computational results are provided comparing the new method against the CPLEX integer programming solver, progressive hedging and cross-entropy optimisation.

Keywords: *Matheuristics, progressive hedging, cross-entropy optimisation, remote area power supply*

The control strategy for making good use of the power supply system needs to determine when the generator should be turned on or off, the level of generation and how much power should be stored into or taken out of the battery at all times. In principle this is a continuous control problem. However in practice we are looking for a control strategy that is simple to implement so we restrict ourselves to a much smaller set of possible control functions. This is guided by the observation that if we fix the diesel generation, then the battery storage system will always be used to meet the residual demand and supply imbalance (within the physical constraints of the storage facility). This leads us to consider a control strategy where we are simply switching between the following 4 states:

- Off:** Turning the generator off completely and using only the battery system to meet any demand for power.
- Exc:** using the generator only to meet excess demand that cannot be met using energy from the battery, otherwise remaining idle
- Dem:** running the generator just fast enough to meet demand or keeping it idle if there is no demand.
- Max:** running the generator at its maximum (most efficient) output level subject to not producing more than we can either store in the battery or use to meet demand.

When we think about the system in terms of these modes of operation, the control problem becomes a much simpler one of deciding in what mode we should be operating at each point in time. In this paper we assume that the mode of operation depends broadly on two factors: the state of charge of the battery (eg no point running the generator beyond demand if the battery is already full), and the time of day. The latter serves as a proxy for PV generation and load patterns which as a first approximation can be assumed to follow a fairly regular daily pattern.

3 MATHEMATICAL FORMULATION

We first provide some notation and a MIP formulation that will help to define precisely what the optimisation problem is, that we are trying to solve. We use the following sets:

- T The set of time intervals into which a single day is split.
- H The set of periods (hours) during which we will use the same control strategy.
- B A coarse discretisation of the battery state of charge (groups of levels) over which the control strategy is to be kept constant.
- L A fine discretisation of the battery state of charge into a set of levels.
- G The possible generator states *on* and *off* - we could also consider more complicated states based on the level at which the generator runs but these are sufficient for most purposes.
- S The set of possible operating modes. $S = \{\text{Off,Exc,Dem,Max}\}$ as described above.
- D A set of demand scenarios or days for which the control strategy is to be optimised.

There are a number of ways the problem could be formulated. Here we use a network based formulation that is both computationally efficient and makes it easy to see how this kind of MIP structure can also arise in other applications where we are designing or using some kind of network.

Consider a network for each day $d \in D$ where nodes represent pairs $(t, l, g) \in T \times L \times G = \mathcal{N}$, ie they are associated with a time, a state of charge and whether the generator is on or off. We now create decision variables x_{vws}^d that connect pairs of such nodes $v, w \in \mathcal{N}$ if running with strategy $s \in S$ when starting at v reaches w for day d . Depending on how long the strategy s is applied for there may be different end points that could be reached. Given that the strategy stays constant for intervals defined by H and B , we only consider points $w = (t_w, l_w, g_w)$ that correspond to the first time that $t_w \in H$ or $l_w \in B$ with g_w as given by the strategy s . Let \lfloor_{vws}^d be the time period and \lfloor_{vws}^d be the battery state corresponding to this arc. We refer to the set of all such arcs for state s as \mathcal{A}_s^d . The total cost of this transition, which includes any generation fuel cost, battery charge/discharge costs, shortfall penalties and so on is denoted by c_{vws}^d .

Problem N

$$\min \frac{1}{|D|} \sum_{d \in D} \sum_{s \in S} \sum_{(v,w) \in \mathcal{A}_s^d} c_{vws}^d x_{vws}^d + \mathcal{D}(y_d^+ + y_d^-) \quad (2a)$$

$$\text{s.t.} \quad \sum_{s \in S} \sum_{\substack{(v,w) \in \mathcal{A}_s^d: \\ t_v=0 \ \& \ t_w=l}} x_{vws}^d = z_l \quad \forall d \in D \ l \in L \quad (2b)$$

$$\sum_{s \in S} \sum_{(w,v) \in \mathcal{A}_s^d} x_{vws}^d = \sum_{s \in S} \sum_{(v,w) \in \mathcal{A}_s} x_{vws}^d \quad \forall v \in \mathcal{N} : t_v \neq T, d \in D \quad (2c)$$

$$\sum_{l \in L} l z_l + y_d^+ - y_d^- = \sum_{l \in L} \sum_{s \in S} \sum_{\substack{(v,w) \in \mathcal{A}_s^d: \\ w=(T,l,g_w)}} l x_{vws}^d \quad \forall d \in D \quad (2d)$$

$$x_{vws}^d \leq z_{\lfloor_{vws}^d, \{d_{vws}, g_w, s\}} \quad \forall d \in D, s \in S, (v,w) \in \mathcal{A}_s^d \quad (2e)$$

$$\sum_{l \in L} z_l = 1 \quad \forall l \in L \quad (2f)$$

$$\sum_{s \in S} z_{bhs} = 1 \quad \forall b \in B, h \in H, g \in G \quad (2g)$$

$$z_{b,h,\text{off},\text{Off}} \geq z_{b,h,\text{on},\text{Off}} \quad \forall b \in B, h \in H \quad (2h)$$

$$z_{b,h,\text{on},s} \geq z_{b,h,\text{off},s} \quad \forall b \in B, h \in H, s \neq \text{Off} \quad (2i)$$

$$\sum_{\bar{s} > s} z_{b-1,h,g,\bar{s}} \geq z_{b,h,g,s} \quad \forall b \in B, h \in H, g \in G \quad (2j)$$

$$x_{vws}^d, y_d^-, y_d^+ \geq 0 \quad (2k)$$

$$z_{bhs} \in \{0, 1\} \quad (2l)$$

Let z_{bhs} be a binary variable that is one if control strategy s is to be used during battery interval b and time period h when the generator is in state g . We can now formulate our problem as a kind of network-design problem. We have one unit of flow going through the network using only arcs permitted by the z variables.

The only thing that remains is to incorporate a suitable way of dealing with the boundary conditions at the start and end of each day period. Ideally we would like the battery to start and finish in about the same state of charge each day, though what this level of charge should be is not yet know. Hence we define z_l variables that define the state of charge at the beginning of the day and penalise deviations from $l \in L$ at the end of the day. Here we penalty \mathcal{D} for deviations y_d^+/y_d^- from the desired level at the end of day $d \in D$.

The complete formulation is shown as Problem N. Here (2b) ensure we send one unit of flow from the chosen initial level, with equations (2c) ensuring conservation of flow through the day and (2d) capturing the final battery level. These flows are restricted by equation (2e) so that the only those transitions selected by the strategy are possible. These constraints can be strengthened slightly by aggregating over all of the arcs that are mutually exclusive for a given b, h, g, s combination.

Finally we have a set of equations that only involve the z variables which describe what an allowable daily control strategy is. This includes only having one setting for each (b, h, g) combination (2g); making sure that if we turn the generator off/on the strategy does not immediately force it on/off again (2h)/(2i); and that a lower battery state of charge level should always mean the generator state is higher (2j). The last of these is not strictly necessary. However in practice we often find that when considering only a single day (as we will for some of the subproblems below), there are parts of the $B \times H \times G$ space that are never traversed for which we still want to have a strategy that is consistent with the parts that are. Hence we impose this constraint to obtain strategies that match our intuition of what a good control strategy should look like.

This model, while in principle solvable for small data sets, becomes quite large as the number of days considered is increased or the discretisation becomes finer. Hence we want to look at matheuristic approaches that build on this formulation to find solutions in a reasonable amount of time. We first describe how progressive hedging or a cross-entropy optimisation method can be applied here. Then we provide the combined matheuristic that builds on both of these.

4 PROGRESSIVE HEDGING

Progressive hedging is a Lagrangian Decomposition based method originally proposed in Rockafellar and Wets [1991] in the context of continuous problems (where in fact this is an exact method), but has since been applied to combinatorial problems as well Watson and Woodruff [2011].

In the context of our problem this works by introducing z_{bhgs}^d variables that allow a separate strategy to be selected for each day and adding linking constraints that $z_{bhgs}^d = z_{bhgs}$. These constraints are then relaxed using Lagrange multipliers λ_{bhgs}^d . We use \bar{z}_{bhgs} to denote our current best estimate of the value of z_{bhgs} . This leads to the following subproblems for each $d \in D$ that need to be solved with an augmented Lagrangian objective:

Problem $N_d(\lambda^d, \bar{z})$

$$\begin{aligned} \min \frac{1}{|D|} \sum_{s \in S} \sum_{(v,w) \in \mathcal{A}_s^d} c_{vws}^d x_{vws}^d + \mathcal{D} y_d^+ + \mathcal{D} y_d^- + \lambda^d z^d + \rho/2 \sum_{b,h,g,s} (z_{bhgs}^d - \bar{z}_{bhgs})^2 \\ \text{s.t.} \quad (2b) - (2k), \quad z_{bhgs}^d \in \{0, 1\} \end{aligned}$$

Note that the quadratic term in the objective simplifies to a linear one of $(1 - 2\bar{z}_{bhgs})z_{bhgs}^d$ since for binary z variables $z^2 = z$ and the constant part of the objective can be ignored for the optimisation. These subproblems form the basis of the progressive hedging algorithm that is outlined in Algorithm 1.

Algorithm 1 Progressive Hedging

```

Initialise  $\bar{z}_{bhgs} := \frac{1}{|S|}$ ,  $\lambda := 0$ 
while not converged do
  for all  $d \in D$  do
    Solve  $N_d(\lambda^d, \bar{z})$ 
    (using a time limit of  $\mu$  seconds)
    Let  $\hat{z}^d$  be the best solution.
  end for
   $\bar{z} := \frac{1}{|D|} \sum_{d \in D} \hat{z}^d$ 
   $\lambda^d := \lambda^d + \rho(\hat{z}^d - \bar{z}^d) \forall d \in D$ 
end while

```

Algorithm 2 Cross-Entropy Heuristic

```

 $P_{bhgs} = \frac{1}{|S|}$  for all  $b, h, g$ 
while not converged do
  for  $i = 1, \dots, n$  do
    With probability  $P_{bhgs}$  generate  $s_{bhg} \forall b, h, g$ 
    Let  $\hat{z}^i = \arg \max_z \sum_{bhg} z_{bhgs_{bhg}}$  subject to (2g)–(2j).
    Evaluate  $\hat{z}^i$ : fix  $z = \hat{z}^i$  in Problem  $N$  and solve for  $x$ .
    Uses  $|D|$  path traversals through the network defined by  $\hat{z}^i$ 
  end for
  Select the best  $n/2$  solutions as  $\mathcal{Z}^*$ 
  Update the best solution found based on the best in  $\mathcal{Z}^*$ 
  Let  $P(\mathcal{Z}^*)_{bhgs} = |\{\hat{z} \in \mathcal{Z}^* : \hat{z}_{bhgs} = 1\}| / |\mathcal{Z}^*|$ 
  Update  $P_{bhgs} := (1 - \alpha)P_{bhgs} + \alpha * P(\mathcal{Z}^*)_{bhgs}$ 
end while

```

5 CROSS-ENTROPY OPTIMISATION

The cross-entropy method described in Boer et al. [2005] is a population based meta-heuristic. The basic structure of the algorithm is to iteratively generate a random set of solutions based on probability for variables to take on specific values. The set of solutions is then evaluated and a subset of "good" solutions selected (better than the others). Finally each iteration updates the probability distribution based on the frequency with which variables take on each value in the set of good solutions.

In our problem we adapt this method to take into account the constraints on z as follows described in Algorithm 2. This makes use of a set of values P_{bhgs} with $\sum_s P_{bhgs} = 1$ for all b, h and g .

The main parameters controlling this algorithm are the population size n and the smoothing parameter α . Note that if we selected only the single best solution for inclusion in \mathcal{Z}^* then we arrive at an algorithm that is fairly similar to ant colony optimisation with α representing the evaporation rate.

6 A HYBRID METHOD

In order to guide the selection of a good population set \mathcal{Z}^* in the cross-entropy optimisation method and the "estimate" of the consensus solution \bar{z} in the progressive hedging method we can combine the two methods. We describe this hybrid in terms of the generic BMIP problem that we started with to show that it applies to a large class of MIP problems.

Algorithm 3 CEPHM

Initialise $\bar{z} = \frac{1}{2}, \lambda^i = 0 \forall i$
while not converged **do**
 for $i = 1, \dots, n$ **do**
 Let \hat{x}^i, \hat{z}^i be the optimal solution to

$$\min_{x^i, z^i} (c^0 + \lambda^i) z^i + c^i x^i + \frac{\rho}{2} \sum_j (z_j^i - \bar{z}_j)^2 \quad \text{s.t.} \quad A_0 z^i = b^0 \text{ and } A_i z^i + B_i x^i = b^i$$

 end for
 $\bar{z} := (1 - \alpha) \bar{z} + \frac{\alpha}{n} \sum_{i=1}^n \hat{z}^i$
 $\lambda^i := \lambda^i + \rho (\hat{z}^i - \bar{z}^i) \quad \forall i = 1, \dots, n$
 for $k = 1, \dots, nSample$ **do**
 $\forall j$ let $\zeta_j^k = 1$ with probability \bar{z}_j
 Let $\tilde{z}^k := \arg \max_{z^k} \{ \zeta^k z^k \mid A_0 z^k = b^0; z_j^k \in \{0, 1\} \forall j \}$
 Evaluate \tilde{z}^k by solving for each i : $\min_{x^i} c^i x^i \quad \text{s.t.} \quad B_i x^i = b^i - A_i \tilde{z}^k$
 end for
 Let S be the set of β best solutions
 Update \bar{z} using $\bar{z} := (1 - \alpha) \bar{z} + \frac{\alpha}{\beta} \sum_{k \in S} \tilde{z}^k$
end while

For BMIP we introduce variables z^i for each $i = 1, \dots, n$ and corresponding Lagrangian multipliers λ^i for relaxing the constraint $z^i = z$ to create a Lagrangian decomposition. We also use \bar{z} to denote a fractional vector that could be interpreted as the best estimate of z or as a vector of probabilities that each element of z is one. Using these we can define the generic Cross-Entropy Progressive Hedging Matheuristic (CEPHM) algorithm as described in Algorithm 3.

This algorithm requires the tuning of the following parameters:

- ρ A positive number. In the literature the value of half is often used. Alternatively this parameter may be progressively increased or decreased depending on the progress of the optimisation as suggested in Watson and Woodruff [2011].
- nSample The number of samples to be used for the Cross-Entropy part of the method. For the purposes of parallelising the loop over k in Algorithm 3 it is advisable to choose this to be a multiple of the number of cores available.
- β The fraction of the sample to be used for updating \bar{z} . This could be a fraction of *nSample* (eg half). Or as a more extreme case use an elitist update where $\beta = 1$.
- α The smoothing parameter controls how quickly \bar{z} is updated and hence controls the convergence. It is possible to use different values of α in the first update of \bar{z} than the second update of \bar{z} at the end of the while loop. This would modify the degree to which the progressive hedging as opposed to the cross-entropy sampling part of the algorithm are guiding the search.

7 EMPIRICAL PERFORMANCE

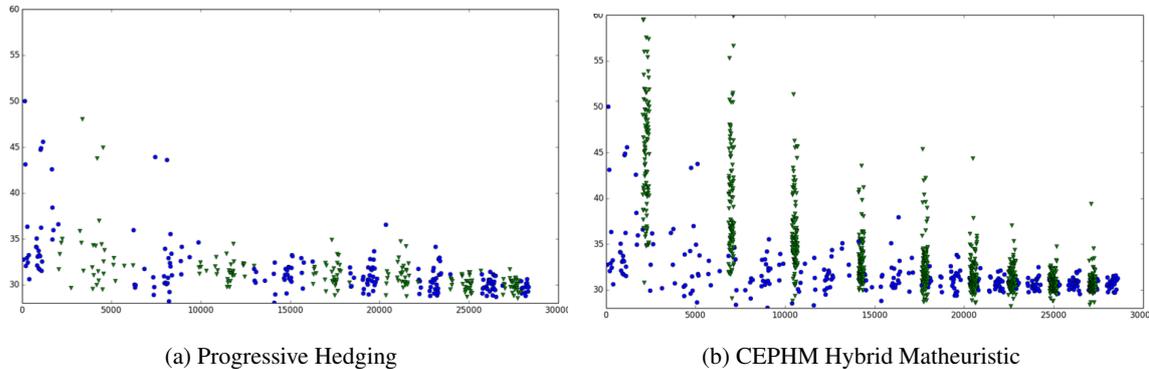
For the purpose of comparison the hybrid method as well as the basic progressive hedging and cross entropy methods were implemented in the python programming language using CPLEX 12.5. Experiments were carried out with 3 different demand and power generation data sets, $|D| = 7, 14$ day scenarios and two types of batteries for a total of 12 instances. In each case $|H| = 8$ and $|B| = 10$ with $|T| = 720$ and $|L| = 201$. These data sets were tested using 4 methods: CPX – using CPLEX for the full model; CEO – cross entropy optimisation (Alg. 2); PH – Progressive Hedging (Alg. 1); CEPHM – the new hybrid matheuristic (Alg. 3) optimisation; parameter configurations:

Quick: 4 hour CPU time limit, maximum 15 iterations; CEO: $n = 150, \alpha = 0.5$; PH: $\rho = 0.3, \mu = 300$;
 CEPHM: $nSample = 20, \alpha = 0.5, \rho = 0.3$

Slow: 8 hour CPU time limit, maximum 30 iterations; CEO: $n = 300, \alpha = 0.4$; PH: $\rho = 0.1, \mu = 600$;
 CEPHM: $nSample = 60, \alpha = 0.4, \rho = 0.1, \mu = 600$

Table 1. Mean and standard deviations of % gaps to the best solution found by any of the methods

Method	Quick		Slow	
	Mean	Std.Dev.	Mean	Std.Dev.
Cplex (CPX)	284.73	355.80	59.37	134.61
Cross-Entropy Opt. (CEO)	20.65	7.60	33.79	16.01
Progressive Hedging (PH)	2.95	3.20	1.66	1.79
Hybrid Matheuristic (CEPHM)	2.63	2.72	1.40	1.43

**Figure 1.** Solution values against run time for a typical instance.

Each of the instances with each method and setting was run 3 times. Table 1 gives an indication of how the performance of the various methods compares. It is very clear that the heuristics perform much better than CPLEX and progressive hedging is better than the cross entropy method. The improvement gained by the combined hybrid method over progressive hedging is somewhat more marginal. However both the average performance and the variability (robustness) of CEPHM as a solution method appears to be better than basic progressive hedging.

8 CONCLUSIONS

We have proposed a new hybrid matheuristic in the context of a specific application in designing a control strategy for a remote area power supply system. We have shown that the ideas from cross-entropy optimization and progressive hedging can be combined in a straight forward manner into a hybrid matheuristic. Initial numerical results indicate that the hybrid method performs significantly better than the cross-entropy optimization method on which it is based and that it provides good diversification to prevent premature convergence of the progressive hedging component. More extensive numerical evaluation and parameter tuning as well as testing on a range of different applications are required to determine the value of the approach as a generic matheuristic.

REFERENCES

- Boer, P.-T. d., D. P. Kroese, S. Mannor, and R. Y. Rubinstein (2005, February). A tutorial on the cross-entropy method. *Annals of Operations Research* 134(1), 19–67. 00512.
- Rockafellar, R. T. and R. J.-B. Wets (1991, February). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research* 16(1), 119–147. 00856.
- Watson, J.-P. and D. L. Woodruff (2011, November). Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science* 8(4), 355–370. 00032.
- Watters, L. J. (1967, November). Reduction of integer polynomial programming problems to zero-one linear programming problems. *Operations Research* 15(6), 1171–1174. 00206.