OPTIMAL POLITICAL DISTRICTING: THE ANCHOR METHOD

Wes Gurnee
MIT ORC
wesg@mit.edu

ABSTRACT

We study the problem of optimal political redistricting. That is, the problem of splitting up a state's voting precincts in to k population balanced and continuous regions that are optimal with respect to one or more objectives, typically involving political (un)fairness. We present a novel formulation, based on the modeling idea of an anchor, which enables us to solve real scale problems with some convergence guarantees.

1 Overview

The political districting problem (PDP) is the decision problem of splitting n precincts into k contiguous and population balanced districts. In the United States, the districts of the House of Representatives are redrawn every 10 years following the decennial census and reapportionment. Given the outsized influence that district composition has on electoral outcomes, how districts are drawn and who draws them, is an extremely contentious (read: litigious) matter. While most districts are drawn by partisan officials with political motives, optimization algorithms offer a more objective and less biased alternative to redistricting.

The most natural MIP formulations for political redistricting resemble a facility location problem: n binary variables for each precinct indicating whether to "open a district" at this location (with supply equal to the prescribed district population), and an additional n^2 binary decision variables indicating whether or not precinct j is apart of the district centered on precinct i (i.e., each precinct has a demand equal to its population, and can only be served by one facility). Unfortunately, these n^2 formulation do not scale to real world problems with $O(10^5)$ precincts, both due to the high number of variables, and the higher number of constraints required to enforce contiguity $O(kn^2)$. Furthermore, decomposition algorithms do not work well in this setting due to the massive degeneracy of the primal problem. These challenges necessitate a different approach.

One of the original solution techniques for the PDP proposed by [1] adopted a heuristic similar to Lloyd's algorithm for k-means: fix the centers y, optimize x, update y to be the new centroids, and iterate until convergence. However, this approach did not explicitly enforce contiguity and only optimized for compactness. We propose a modernized formulation capable of incorporating the full range of legal constraints. A key part of our approach is based on a generalization of the center block, which we call an anchor. The purpose of an anchor is to break symmetry and act as the source node for the flow variables which enforce contiguity. The critical difference, however, is that we can define an *anchor invariant* formulation such that for the optimal districting, *any* choice of anchors (that is, any choice of precincts in disjoint districts) will yield the same optimal plan. Conceptually, the anchors implicitly define the feasible space, but by construction, can be decoupled from the objective value. This feature is what makes it possible to create optimality guarantees.

2 Formulation

In this section, we will assume a set of anchors are given, and describe the inner loop formulation. Our integer programming model is given below where N is the set of blocks, $A \subset N$ is the set of anchors, and E is the edge set of the adjacency graph. The main decision variables are x_{ij} indicating if precinct j is assigned to district (anchor) i. We enforce that the assignment is a strict partition of precincts (2), are population balanced (3) and contiguous (4-6). We implement contiguity with flow variables f_{ijk} indicating the flow from anchor i that traverses the arc between adjacent precincts j and k, and require that all precincts demand one unit of flow of type i if assigned to the anchor i.

$$\begin{aligned}
\min \quad & f(x) && && (1)\\
\text{s.t.} \quad & \sum_{i \in A} x_{ij} = 1 && \forall j \in N && (2)\\
& 1 - \epsilon \leq \sum_{j \in N} x_{ij}\hat{p}_j \leq 1 + \epsilon && \forall i \in A && (3)\\
& \sum_{k \in \delta(j)} (f_{ikj} - f_{ijk}) = x_{ij} && \forall i \in A, j \in N && (4)\\
& \sum_{k \in \delta(j)} f_{ikj} \leq M x_{ij} && \forall i \in A, j \in N && (5)\\
& \sum_{j \in \delta(i)} f_{iji} = 0 && \forall i \in A && (6)\\
& u_{jk} \geq x_{ij} - x_{ik} && \forall i \in A, (j,k) \in E && (7)\\
& && && (8)
\end{aligned}$$

where $\hat{p}_j$ is the population of block j divided by the ideal district population, and $\delta(i)$ is the set of blocks adjacent to i.

We leave a generic objective $f(x)$ for now, as the hardest part of the PDP is in efficiently enforcing feasibility. An advantage of this formulation is that it is flexible in accommodating additional modeling logic (i.e., there is no special structure we must be careful to preserve). In the next section, we describe many such extensions relevant to redistricting, that will serve as terms in our objective or satisfy legal criteria specific to certain states.

2.1 Extensions

Compactness For many reasons—legal, aesthetic, normative—it is common to either constrain or optimize for compactness. Here we describe how to incorporate three different notions of compactness: dispersion, flow distance, and edge cuts/perimeter.

$$\text{(Dispersion)} \qquad \sum d_{ij}^{\alpha} x_{ij} \qquad (9)$$

Dispersion, sometimes also called moment or inertia when $\alpha = 2$, is a compactness metric parameterized by α and seeks to minimize the l_α distance of the assignment.

$$\text{(Flow distance)} \qquad \sum w_{ijk} f_{ijk} \qquad (10)$$

The flow distance metric is much less common, but we noticed that it was very efficient to optimize, because it broke the symmetry implicit in the flow variables. That is, under this objective, not all flow paths from i to j are of equal cost, and hence the resulting flow structure will favor compact spanning trees.

$$\text{(Cut Edges)} \qquad u_{jk} \geq x_{ij} - x_{ik} \qquad \forall i \in A, (j,k) \in E \quad (11)$$

Lastly, the edge cut formulation is an appealing graph theoretic measure of compactness, as it is scale agnostic, unlike the dispersion metric. To implement it, we introduce binary variables u_{jk} for each edge which gets set to 1 whenever x_{ij} and x_{ik} are different, that is, when blocks j and k are assigned to different districts. We can also add a weight w_{jk} when summing the edge cuts to get the interior perimeter of the districting plan, which penalizes cutting edges with long shared borders (this favors aesthetically pleasing districts with clean borders). Another advantage of this measure is that it is anchor invariant, which we will explain further in Section 3.

Despite all of these advantages, this linear relaxation of the edge cut formulation is terrible. The optimal solution will set $x_{ij} = \frac{1}{k}$ to make all edge differences 0. We found it was intractable to optimize for all but the smallest synthetic networks using this objective.

Political Objectives Of course the main set of objectives in political redistricting are *political*. For simplicity, we consider two-party election results where v_j gives the Republican vote share for block j. Then if $\hat{v}_j = \hat{p}_j v_j$, we introduce vote share variables r_i and seat share variables ψ_i

$$\text{(Vote Share Variables)} \qquad r_i = x_i^T \hat{v} \qquad \forall i \in A \quad (12)$$

$$\text{(Seat Share Variables)} \qquad \psi_i = \sigma_{PWL}(r_i) \qquad \forall i \in A \tag{13}$$

to model the expected political outcomes of a district plan. Here σ_{PWL} is a piecewise linear approximation of the cumulative density function corresponding to the probability that a seat is occupied by a Republican, given the vote share r_i. Since there are only $k \ll n$ of these variables, it is feasible to introduce this more elaborate modeling logic. We note one significant drawback with this approach is the implicit assumption of uniform turnout, to avoid introducing a fractional formulation.

With these political estimates, we can optimize for different measures of partisan fairness. The most basic measure is the degree to which the overall expected seat share deviates from some ideal seat share $h^*(v)$.

$$\text{(Partisan Fairness)} \qquad \min |\sum_i \psi_i - h^*(v)| \tag{14}$$

For pure proportionality, we would let $h^*(v) = 1$. We note that it is more straightforward to gerrymander than to optimize for fairness, as partisan advantage can be optimized by simply minimizing or maximizing the sum of ψ_i. Another example objective which would be straightforward to implement is maximizing the number of competitive seats. We can measure competitiveness as the distance of r_i from being 50-50, discounted by α:

$$\text{(Competitiveness)} \qquad \max \sum_i t_i \ ; \quad t_i \leq \max(1 - \alpha|0.5 - r_i|, 0) \tag{15}$$

Voting Rights Act Another important set of legal constraints are those laid out by the Voting Rights Act (VRA) on providing sufficient opportunities for minority candidates to get elected. These rules are complicated, and are not mathematically precise, but we can approximate them by introducing binary variables y_i^g to indicate whether or not district i is an opportunity district for group g (though, we could also make these continuous variables corresponding to a probability estimate).

$$\text{(VRA Compliance)} \qquad y_i^g \leq \alpha \sum_j x_{ij} \hat{p_j}^g; \quad \sum_j y_i^g \geq g_{\min} \qquad \forall g \tag{16}$$

Then, per the laws and demographics of a state, we enforce that the districting plan meet a minimum level of opportunity for every group.

Boundary Preservation Finally, we consider preservation of existing political subdivisions such as counties, wards, or townships and/or preserving communities of interest (COI). We note that is trivial to add a hard constraint on two or more blocks being together: just merge them in the underlying adjacency graph. However, this can lead to feasibility issues if many of the blocks become merged because the district population tolerances are very strict.

Usually a more practical approach is to again add binary variables indicating whether or not a particular region is kept whole. If also using an edge cut formulation, one can model this by simply checking if any edges in the region R_t are cut

$$\text{(Boundary Preservation I)} \qquad s_t \geq u_{jk} \qquad \forall t, \forall (j,k) \in E \cap C_t \tag{17}$$

However, as discussed previously, the linear relaxation of the u_{jk} is too weak to be tractable at large scale. Another modeling alternative is to check if the region is fully contained in district i

$$\text{(Boundary Preservation II)} \qquad |C_t|(1 - s_{it}) \leq \sum_{j \in C_t} x_{ij} \qquad \forall t. \tag{18}$$

However, this still uses a big-M formulation (i.e., the linear relaxation still is not tight) and requires k times as many variables.

2.2 Alternate Formulations

We also tried a disaggregated flow formulation [2] with flow variables

$$f_{ij}^{k\ell} = \begin{cases} 1 & \text{if edge } (k,\ell) \in E \text{ is on the path from anchor } i \text{ to block } j \\ 0 & \text{otherwise} \end{cases}$$

for all $(k,\ell) \in \Delta_{ij} = \{(k,\ell) \in E : d_{ik} + d_{\ell j} \leq \alpha d_{ij}\}$. That is, we only consider edges such that the distance of the shortest path i, k, ℓ, j is no more than a factor α of the shortest path from i to j. The constraints are very similar to the aggregated formulation:

$$\sum_{k \in \delta(i)} f_{ij}^{ik} - f_{ij}^{ki} = x_{ij} \qquad \forall i \in A,\ \forall j \in N \setminus A$$

$$\sum_{\ell \in \delta(k)} f_{ij}^{k\ell} - f_{ij}^{\ell k} = 0 \qquad \forall i \in A,\ \forall j \in N \setminus A,\ \forall k \in \Delta_{ij} \setminus A$$

$$\sum_{k \in \delta(i)} f_{ij}^{k\ell} = 0 \qquad \forall i \in A,\ \forall j \in N \setminus A$$

$$\sum_{\ell \in \delta(k)} f_{ij}^{\ell k} \leq x_{ik} \qquad \forall i \in A,\ \forall j \in N \setminus A,\ \forall k \in \Delta_{ij} \setminus A$$

$$f_{ij}^{k\ell} \geq 0.$$

While this formulation is tighter given that we remove the big-M constraints, the memory requirements of this formulation are too large for full scale problems. For North Carolina, a medium state with 13 districts and 2183 blocks, the formulation contained 122800365 rows, 143450406 columns, 488095135 nonzero coefficients and took almost an hour just to construct the model (and then promptly crashed due to memory pressure). It is possible that more aggressive variable fixing and clever engineering effort could ameliorate these challenges but we did not pursue this line of inquiry further.

2.3 Variable Fixing

To reduce the size of the problem we experimented with variable fixing, that is, setting $x_{ij} = 0$ for pairs of anchors and blocks which are unlikely to be feasible let alone optimal. Our first thought was just to set $x_{ij} = 0$ if i was not within the Γ closest anchors. However, this does not work well for blocks where there is a steep gradient in population density. It may be the case that all of the close anchors are in a large urban area, but in an optimal (or even feasible) plan, these blocks would be assigned to a much more distant rural anchor.

Instead, we leverage the fact that at each iteration we have a feasible plan as a warm start. In particular, we set the bound as

$$x_{ij} \leq \begin{cases} 1 & \text{if block } j \text{ is assigned to a district adjacent to district } i \\ 0 & \text{otherwise.} \end{cases}$$

In other words, we further restrict the blocks to only be assigned to neighboring anchors in the warm start solution. This accommodates for population density gradients and other geographic features that affect the range of feasible assignments. Of course, fixing x_{ij} to 0 also allows fixing f_{ijk} to be 0 for all k.

3 Anchors

The obvious question we have deferred until now is: how do we choose anchors?

In theory, we would like a routine which uses the faster to solve $O(nk)$ formulation above as a subroutine to find a globally optimal solution. To study convergence, we introduce the concept of an *anchor invariant* formulation. The defining property of an anchor invariant formulation is that, for a globally optimal districting plan, any choice of anchors such that each district contains exactly one will yield the same optimal solution. In other words, anchors only affect the feasibility of an optimal solution, not the objective value. We note that the base formulation is anchor invariant, and that all of the extensions are anchor invariant with the notable exceptions of dispersion and flow based compactness (i.e., the tractable compactness metrics).

When the number of districts is small, we can do random sampling to obtain a simple probabilistic bound on the number of samples/iterations until optimality. In particular, it is the probability of sampling k elements from a set of n elements (without replacement) that are implicitly partitioned into k sets, where each of the k elements are from different sets:

$$\prod_{i=0}^{k-1} 1 - \frac{(i-1)}{k} \qquad (19)$$

assuming each district has exactly n/k blocks. For larger numbers of districts this bound is useless, and we would expect the vast majority of randomly sampled anchor sets to be infeasible. One could likely make this bound substantially

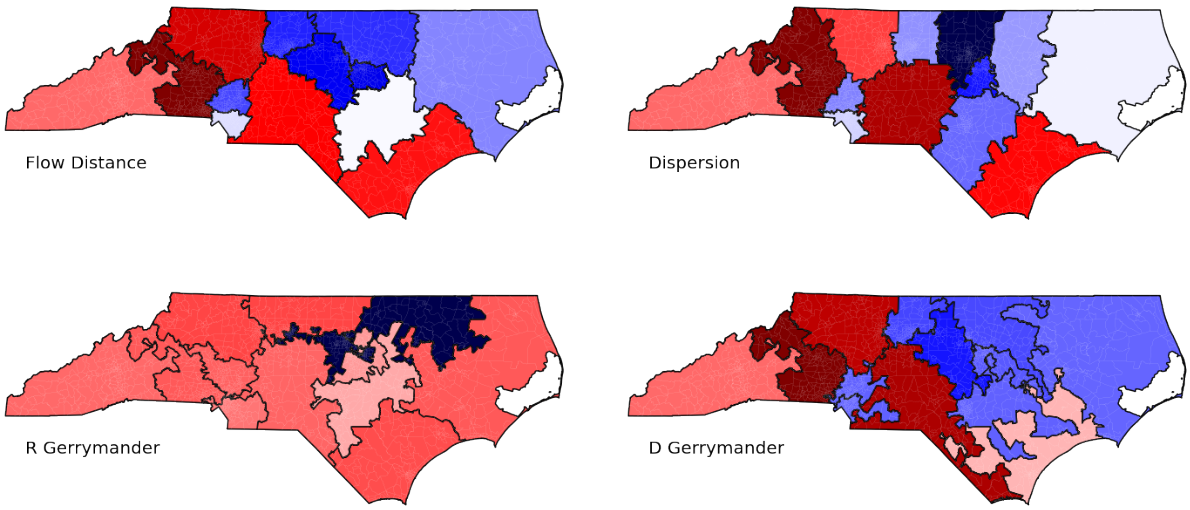

Figure 1: Districting plans optimized under different objective values.

tighter by exploiting the fact that these k sets are spanning trees of the underlying adjacency graph, but that is outside of the scope of this work.

In practice, there might be several natural choices of anchors, such as the home precincts of the incumbents to prevent "hijacking" gerrymanders, the historic center of a district, or the main town or geographic area associated with a district. Again, if we use an anchor invariant formulation, and we condition on no two incumbents being put into the same district (or similar disjoint criterion), then the solution is globally optimal. Of course, the number of districts may change between cycles, or we may not want to preserve incumbency advantages, so we cannot rely on such conditions.

A more general approach is to use a heuristic similar to Lloyd's algorithm for k-means. That is, starting from some set (like incumbent home locations, or the centroid of an existing or random plan), we iteratively solve the above MIP, and then use the blocks nearest to the optimized district centroids as the next set of anchors, and continue this iteration until there are no updates. While this only gives locally optimal solutions, we can utilize random restarts to get what are likely globally optimal solutions.

4 Results

Given the difficulty of the formulation, it was challenging to perform thorough full-scale experiments. Our main experimental result involve running our algorithm for a few different political and compactness objectives in North Carolina (see Figure 1. We comment on high level findings and attach some of our solver logs in the appendix.

We first create a random plan using a recursive spanning tree algorithm [3]. We then get the first set of anchors by taking the blocks nearest the population weighted centroid of the districts in the initial plan. We then iterate, using our formulation described in Section 2 with a one hour timeout, continuing to update the anchors to be the new centroids until there are no updates in the location of the anchors. We also always use the previous assignment as a warmstart.

We find that optimizing for compactness objectives, especially flow distance, is extremely fast, usually taking no more than a few seconds per iteration while converging in a very small number of iterations. The political objectives, in contrast, always take the full hour and usually timeout with a double digit gap (see Appendix). The solver spends a long time at the root node, generating hundreds or thousands of flow cover cuts in addition to hundreds of mixed-integer round cuts and a mix of all others. Nevertheless, the gerrymandering capabilities are quite impressive (see Figure 2 which also records the anchors and objective value of the iterations). Due to the piecewise-linear nature of the objective, the optimal solution tries to make as many districts as possible have 55% Republicans (at estimate 95% probability of victory), and manages to get a full 11/13 districts to about this threshold.

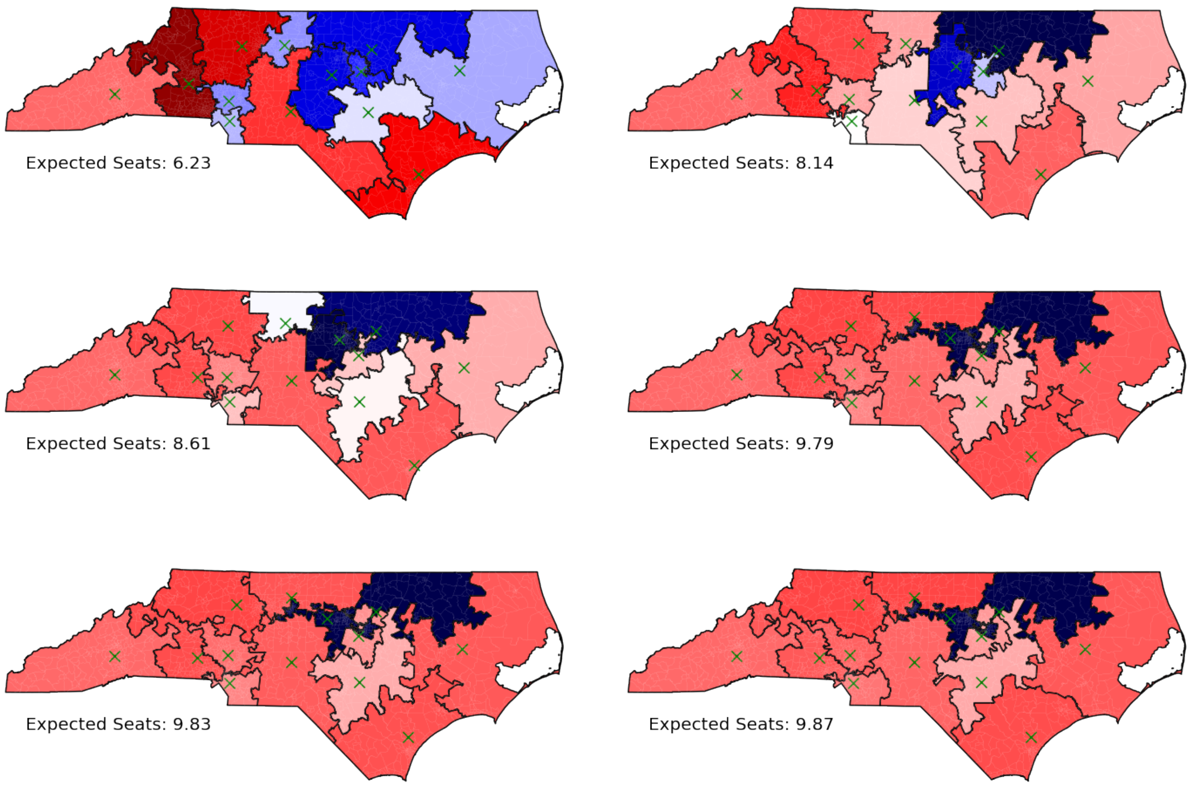

Figure 2: Anchors and objective value when optimizing for expected Republican seat share.

5 Conclusions

Our results show the scalability is mixed. We initially expected our algorithm would require $O(10)$ iterations taking each $O(1\text{min})$ but it turned out be closer to $O(1)$ iterations taking $O(1\text{hr})$. Potential ideas for future work include

- More aggressive variable fixing, potentially only add x_{ij} for j in an incident county to district i from the previous iteration.
- Use disaggregated flow formulation but with column generation of flow paths.
- Decay the value of M further away from the anchor.
- Derive valid inequalities to tighten the formulation.
- Gurobi tuning to find better solver parameters.
- Fix the bug which causes the western most district to never change.

References

[1] Sidney Wayne Hess, JB Weaver, HJ Siegfeldt, JN Whelan, and PA Zitlau. Nonpartisan political redistricting by computer. *Operations Research*, 13(6):998–1006, 1965.

[2] Hamidreza Validi, Austin Buchanan, and Eugene Lykhovyd. Imposing contiguity constraints in political districting models. *Operations Research*, 2021.

[3] Daryl DeFord, Moon Duchin, and Justin Solomon. Recombination: A family of markov chains for redistricting. *arXiv preprint arXiv:1911.05725*, 2019.

A Example Solver Log

Solver log for Republican gerrymander with flow distance regularization of 10^{-5}.

```
Set parameter TimeLimit to value 3600
Set parameter MIPGap to value 0.001
Set parameter MIPFocus to value 1
Gurobi Optimizer version 9.5.1 build v9.5.1rc2 (mac64[rosetta2])
Thread count: 10 physical cores, 10 logical processors, using up to 10 threads
Optimize a model with 87216 rows, 185562 columns and 824772 nonzeros
Model fingerprint: 0x5a9f740c
Model has 8228 SOS constraints
Model has 13 piecewise-linear objective terms
Variable types: 157183 continuous, 28379 integer (28379 binary)
Coefficient statistics:
  Matrix range     [8e-06, 2e+02]
  Objective range  [1e-05, 1e-05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
  PWLObj x range   [5e-01, 6e-01]
  PWLObj obj range [5e-02, 9e-01]

User MIP start produced solution with objective -6.10069 (0.21s)
Loaded user MIP start with objective -6.10069

Presolve removed 13744 rows and 99150 columns
Presolve time: 0.89s
Presolved: 73532 rows, 86472 columns, 550471 nonzeros
Variable types: 55902 continuous, 30570 integer (30565 binary)

Deterministic concurrent LP optimizer: primal and dual simplex
Showing first log only...


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
   42087   -1.0810400e+01   0.000000e+00   6.542241e-01      5s
   65926   -1.0821663e+01   0.000000e+00   0.000000e+00      8s
   65926   -1.0821663e+01   0.000000e+00   0.000000e+00      8s
Concurrent spin time: 1.71s

Solved with primal simplex

Root relaxation: objective -1.082166e+01, 65926 iterations, 8.67 seconds (12.17 work units)
Total elapsed time = 10.24s

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  -10.82166    0 1072   -6.10069  -10.82166  77.4%     -   10s
     0     0  -10.82059    0 1274   -6.10069  -10.82059  77.4%     -   16s
     0     0  -10.82049    0 1260   -6.10069  -10.82049  77.4%     -   17s
     0     0  -10.81982    0 1369   -6.10069  -10.81982  77.4%     -   23s
H    0     0                        -6.1326211  -10.81982  76.4%     -   24s
     0     0  -10.81968    0 1385   -6.13262  -10.81968  76.4%     -   25s
     0     0  -10.81901    0 1530   -6.13262  -10.81901  76.4%     -   36s
     0     0  -10.81877    0 1508   -6.13262  -10.81877  76.4%     -   40s
     0     0  -10.81861    0 1807   -6.13262  -10.81861  76.4%     -   54s
     0     0  -10.81839    0 1863   -6.13262  -10.81839  76.4%     -   59s
     0     0  -10.81833    0 1655   -6.13262  -10.81833  76.4%     -   88s
H    0     0                        -6.6923075  -10.81833  61.7%     -   91s
H    0     0                        -6.9347445  -10.81833  56.0%     -  126s
```

```
H    0     0                     -7.2910480  -10.81833  48.4%     -  143s
H    0     0                     -7.2910580  -10.81833  48.4%     -  144s
H    0     2                     -7.2910680  -10.81833  48.4%     -  145s
     0     2  -10.81833    0 1628   -7.29107  -10.81833  48.4%     -  145s
H    1     4                     -7.3565526  -10.81833  47.1%  1196  539s
H    2     4                     -7.4125952  -10.81833  45.9% 154497  539s
H    3     8                     -7.5044017  -10.79802  43.9% 110591  607s
H    6     8                     -7.6713070  -10.78778  40.6% 75483  607s
H    7    16                     -7.8919720  -10.78778  36.7% 66782 2594s
H   15    22                     -8.1946217  -10.77172  31.4% 49687 4520s

Cutting planes:
  Gomory: 1
  Lift-and-project: 108
  Cover: 328
  Implied bound: 3
  MIR: 889
  StrongCG: 3
  Flow cover: 1275
  Network: 102
  RLT: 4
  Relax-and-lift: 461

Explored 21 nodes (881338 simplex iterations) in 4520.36 seconds (1977.78 work units)
Thread count was 10 (of 10 available processors)

Solution count 10: -8.19462 -7.89197 -7.67131 ... -6.93474

Time limit reached
Best objective -8.194621682188e+00, best bound -1.077172127117e+01, gap 31.4487%
Set parameter TimeLimit to value 3600
Set parameter MIPGap to value 0.001
Set parameter MIPFocus to value 1
Gurobi Optimizer version 9.5.1 build v9.5.1rc2 (mac64[rosetta2])
Thread count: 10 physical cores, 10 logical processors, using up to 10 threads
Optimize a model with 87216 rows, 185562 columns and 824722 nonzeros
Model fingerprint: 0x0feb6a3e
Model has 8228 SOS constraints
Model has 13 piecewise-linear objective terms
Variable types: 157183 continuous, 28379 integer (28379 binary)
Coefficient statistics:
  Matrix range     [8e-06, 2e+02]
  Objective range  [1e-05, 1e-05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
  PWLObj x range   [5e-01, 6e-01]
  PWLObj obj range [5e-02, 9e-01]

User MIP start produced solution with objective -8.19616 (0.20s)
Loaded user MIP start with objective -8.19616

Presolve removed 16549 rows and 102641 columns
Presolve time: 0.79s
Presolved: 70727 rows, 82981 columns, 517398 nonzeros
Variable types: 53618 continuous, 29363 integer (29342 binary)

Deterministic concurrent LP optimizer: primal and dual simplex
Showing first log only...
```

```
Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
   44634   -1.0763093e+01   0.000000e+00   2.877182e-01      5s
   56053   -1.0765299e+01   0.000000e+00   0.000000e+00      6s
   56053   -1.0765299e+01   0.000000e+00   0.000000e+00      6s
Concurrent spin time: 1.11s

Solved with primal simplex

Root relaxation: objective -1.076530e+01, 56053 iterations, 6.09 seconds (9.34 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  -10.76530    0  743   -8.19616  -10.76530  31.3%     -    7s
     0     0  -10.76418    0 1035   -8.19616  -10.76418  31.3%     -   12s
     0     0  -10.76413    0  987   -8.19616  -10.76413  31.3%     -   12s
     0     0  -10.76351    0 1090   -8.19616  -10.76351  31.3%     -   16s
     0     0  -10.76338    0 1073   -8.19616  -10.76338  31.3%     -   17s
     0     0  -10.76314    0 1252   -8.19616  -10.76314  31.3%     - 1064s
     0     0  -10.76286    0 1285   -8.19616  -10.76286  31.3%     - 1067s
     0     0  -10.76269    0 1333   -8.19616  -10.76269  31.3%     - 1077s
     0     0  -10.76262    0 1337   -8.19616  -10.76262  31.3%     - 1081s
     0     0  -10.76257    0 1494   -8.19616  -10.76257  31.3%     - 1090s
H    0     0                      -8.3069511  -10.76257  29.6%     - 1092s
H    0     0                      -8.5165105  -10.76257  26.4%     - 1127s
H    0     0                      -8.5456986  -10.76257  25.9%     - 1139s
H    0     2                      -8.5457086  -10.76257  25.9%     - 1140s
     0     2  -10.76257    0 1483   -8.54571  -10.76257  25.9%     - 1140s
H    1     4                      -8.5520719  -10.76257  25.8%  1551 2143s
     3     8  -10.31509    2 1400   -8.55207  -10.75982  25.8% 57549 4225s
H    4     8                      -8.5533614  -10.75982  25.8% 43162 4225s
H    5     8                      -8.6269920  -10.75982  24.7% 39289 4225s

Cutting planes:
  Gomory: 2
  Lift-and-project: 98
  Cover: 264
  MIR: 647
  Flow cover: 917
  Network: 61
  RLT: 6
  Relax-and-lift: 313

Explored 7 nodes (332835 simplex iterations) in 4225.80 seconds (614.28 work units)
Thread count was 10 (of 10 available processors)

Solution count 8: -8.62699 -8.55336 -8.55207 ... -8.19616

Time limit reached
Best objective -8.626992041630e+00, best bound -1.075494098954e+01, gap 24.6662%
Set parameter TimeLimit to value 3600
Set parameter MIPGap to value 0.001
Set parameter MIPFocus to value 1
Gurobi Optimizer version 9.5.1 build v9.5.1rc2 (mac64[rosetta2])
Thread count: 10 physical cores, 10 logical processors, using up to 10 threads
Optimize a model with 87216 rows, 185562 columns and 824747 nonzeros
```

```
Model fingerprint: 0x847ded7c
Model has 8228 SOS constraints
Model has 13 piecewise-linear objective terms
Variable types: 157183 continuous, 28379 integer (28379 binary)
Coefficient statistics:
  Matrix range     [8e-06, 2e+02]
  Objective range  [1e-05, 1e-05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
  PWLObj x range   [5e-01, 6e-01]
  PWLObj obj range [5e-02, 9e-01]

User MIP start produced solution with objective -8.62632 (0.18s)
Loaded user MIP start with objective -8.62632

Presolve removed 10593 rows and 95279 columns
Presolve time: 0.85s
Presolved: 76688 rows, 90348 columns, 567109 nonzeros
Variable types: 58081 continuous, 32267 integer (32240 binary)

Deterministic concurrent LP optimizer: primal and dual simplex
Showing first log only...


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
   47482   -1.0781517e+01   0.000000e+00   1.325208e-01      5s
   58645   -1.0782453e+01   0.000000e+00   0.000000e+00      6s
   58645   -1.0782453e+01   0.000000e+00   0.000000e+00      6s
Concurrent spin time: 1.76s

Solved with primal simplex

Root relaxation: objective -1.078245e+01, 58645 iterations, 6.87 seconds (9.40 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  -10.78245    0  832   -8.62632  -10.78245  25.0%     -    8s
     0     0  -10.76545    0 1047   -8.62632  -10.76545  24.8%     -   13s
     0     0  -10.76544    0 1007   -8.62632  -10.76544  24.8%     -   14s
     0     0  -10.76510    0 1110   -8.62632  -10.76510  24.8%     -   20s
     0     0  -10.76497    0 1127   -8.62632  -10.76497  24.8%     -   22s
     0     0  -10.76462    0 1292   -8.62632  -10.76462  24.8%     -   31s
     0     0  -10.76435    0 1193   -8.62632  -10.76435  24.8%     -   34s
     0     0  -10.76431    0 1214   -8.62632  -10.76431  24.8%     -  952s
     0     0  -10.76421    0 1314   -8.62632  -10.76421  24.8%     -  956s
     0     0  -10.76412    0 1311   -8.62632  -10.76412  24.8%     -  965s
H    0     0                       -8.6279178  -10.76412  24.8%     -  967s
H    0     0                       -8.6468276  -10.76412  24.5%     -  984s
H    0     0                       -8.6512993  -10.76412  24.4%     -  994s
H    0     2                       -8.6513193  -10.76412  24.4%     -  995s
     0     2  -10.76412    0 1303   -8.65132  -10.76412  24.4%     -  995s
H    1     4                       -8.6797796  -10.76412  24.0%  1774 1052s
H    2     4                       -8.7243899  -10.76412  23.4% 30724 1052s
H    3     8                       -8.7301665  -10.76050  23.3% 28347 1106s
H    5     8                       -8.7793736  -10.76050  22.6% 31115 1106s
H    7    16                       -8.7832757  -10.75271  22.4% 25501 1427s
```

H	8	16				-8.7859122	-10.75271	22.4%	22421	1427s
H	9	16				-8.8667521	-10.75271	21.3%	27043	1427s
H	11	16				-8.8832163	-10.75271	21.0%	29925	1428s
	15	26	-10.31584	4	1288	-8.88322	-10.74713	21.0%	24613	2071s
H	25	36				-8.8832763	-10.74713	21.0%	89409	2209s
H	26	36				-8.8958841	-10.74713	20.8%	86059	2209s
H	27	36				-9.0544379	-10.74713	18.7%	82933	2209s
	35	46	-10.31573	6	1198	-9.05444	-10.74713	18.7%	67720	2216s
H	36	46				-9.0909370	-10.74713	18.2%	65839	2216s
H	37	46				-9.3134462	-10.74713	15.4%	64172	2216s
H	37	46				-9.3755701	-10.74713	14.6%	64172	2216s
H	45	56				-9.4106979	-10.74713	14.2%	53202	2257s
H	51	56				-9.4385129	-10.74713	13.9%	47125	2257s
	55	76	-10.31560	8	1116	-9.43851	-10.74713	13.9%	44026	2262s
H	75	86				-9.4785689	-10.74713	13.4%	32741	2305s
H	82	86				-9.4962538	-10.74713	13.2%	30178	2305s
	85	106	-10.31555	9	1120	-9.49625	-10.74713	13.2%	29184	2311s
	105	126	-10.31556	11	1118	-9.49625	-10.74713	13.2%	24098	2325s
H	125	136				-9.4962638	-10.74713	13.2%	20607	2377s
H	126	136				-9.4963538	-10.74713	13.2%	20464	2377s
H	127	136				-9.5274056	-10.74713	12.8%	20314	2377s
	135	166	-10.31542	13	1103	-9.52741	-10.74713	12.8%	19222	2382s
H	165	176				-9.5274156	-10.74713	12.8%	16161	2438s
H	165	176				-9.5377598	-10.74713	12.7%	16161	2438s
H	166	176				-9.5466375	-10.74713	12.6%	16070	2438s
H	167	176				-9.5559812	-10.74713	12.5%	15986	2438s
H	171	176				-9.5566982	-10.74713	12.5%	15706	2438s
H	172	176				-9.5704353	-10.74713	12.3%	15633	2438s
	175	212	-10.31527	17	1101	-9.57044	-10.74713	12.3%	15397	2443s
H	211	222				-9.5765816	-10.74713	12.2%	13125	2469s
H	213	222				-9.5789487	-10.74713	12.2%	13028	2469s
H	214	222				-9.5797060	-10.74713	12.2%	12972	2469s
H	215	222				-9.5849311	-10.74713	12.1%	12919	2469s
	221	268	-10.31452	21	1054	-9.58493	-10.74713	12.1%	12653	2476s
H	267	281				-9.5849611	-10.74713	12.1%	10785	2500s
H	268	281				-9.5920986	-10.74713	12.0%	10765	2500s
H	271	281				-9.5966355	-10.74713	12.0%	10666	2500s
	280	325	-10.31405	25	1090	-9.59664	-10.74713	12.0%	10366	2508s
H	324	335				-9.6105479	-10.74713	11.8%	9184	2639s
H	330	335				-9.6743019	-10.74713	11.1%	9049	2639s
H	332	335				-9.7334367	-10.74713	10.4%	9004	2639s
	334	392	-10.31402	28	1056	-9.73344	-10.74713	10.4%	8963	2645s
	391	402	-10.31397	33	1006	-9.73344	-10.74713	10.4%	7920	2843s
H	392	402				-9.7654927	-10.74713	10.1%	7899	2843s
H	394	402				-9.8383143	-10.74713	9.24%	7870	2843s
	401	465	-10.31395	34	1001	-9.83831	-10.74713	9.24%	7772	2852s
H	464	489				-9.8411747	-10.74713	9.21%	6952	2868s
H	466	489				-9.8412789	-10.74713	9.20%	6936	2868s
H	483	489				-9.8473196	-10.74713	9.14%	6752	2868s
	488	549	-10.31358	40	895	-9.84732	-10.74713	9.14%	6701	2886s
H	505	549				-9.8473399	-10.74713	9.14%	6543	2886s
H	548	577				-9.8475499	-10.74713	9.14%	6166	2902s
	576	642	-10.31283	45	852	-9.84755	-10.74713	9.14%	5941	3265s
	641	679	-10.31239	52	879	-9.84755	-10.74713	9.14%	5525	3309s
H	663	679				-9.8475599	-10.74713	9.13%	5397	3309s
H	664	679				-9.8546995	-10.74713	9.06%	5390	3309s
	678	750	-10.31233	55	854	-9.85470	-10.74713	9.06%	5307	3572s
H	749	798				-9.8550195	-10.74713	9.05%	4965	3600s
H	780	798				-9.8646703	-10.74713	8.95%	4819	3600s

```
Cutting planes:
  Gomory: 3
  Lift-and-project: 154
  Cover: 323
  Implied bound: 7
  Clique: 4
  MIR: 782
  StrongCG: 1
  Flow cover: 1040
  Network: 70
  RLT: 6
  Relax-and-lift: 390

Explored 797 nodes (3902114 simplex iterations) in 3600.59 seconds (3260.35 work units)
Thread count was 10 (of 10 available processors)

Solution count 10: -9.86467 -9.85502 -9.8547 ... -9.83831

Time limit reached
Best objective -9.864670267723e+00, best bound -1.074712536873e+01, gap 8.9456%
Set parameter TimeLimit to value 3600
Set parameter MIPGap to value 0.001
Set parameter MIPFocus to value 1
Gurobi Optimizer version 9.5.1 build v9.5.1rc2 (mac64[rosetta2])
Thread count: 10 physical cores, 10 logical processors, using up to 10 threads
Optimize a model with 87216 rows, 185562 columns and 824622 nonzeros
Model fingerprint: 0x15806a1d
Model has 8228 SOS constraints
Model has 13 piecewise-linear objective terms
Variable types: 157183 continuous, 28379 integer (28379 binary)
Coefficient statistics:
  Matrix range     [8e-06, 2e+02]
  Objective range  [1e-05, 1e-05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
  PWLObj x range   [5e-01, 6e-01]
  PWLObj obj range [5e-02, 9e-01]

User MIP start produced solution with objective -9.86967 (0.18s)
Loaded user MIP start with objective -9.86967

Presolve removed 9328 rows and 93318 columns
Presolve time: 0.86s
Presolved: 77953 rows, 92309 columns, 582658 nonzeros
Variable types: 59642 continuous, 32667 integer (32656 binary)

Deterministic concurrent LP optimizer: primal and dual simplex
Showing first log only...

Concurrent spin time: 1.26s

Solved with primal simplex

Root relaxation: objective -1.081600e+01, 52643 iterations, 4.86 seconds (6.61 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
```

```
     0     0  -10.81600    0  541   -9.86967  -10.81600  9.59%     -    6s
     0     0  -10.79945    0  700   -9.86967  -10.79945  9.42%     -   11s
     0     0  -10.79942    0  676   -9.86967  -10.79942  9.42%     -   11s
     0     0  -10.79893    0  661   -9.86967  -10.79893  9.42%     -   15s
     0     0  -10.79893    0  798   -9.86967  -10.79893  9.42%     -   16s
     0     0  -10.79864    0  815   -9.86967  -10.79864  9.41%     -   24s
     0     0  -10.79839    0  867   -9.86967  -10.79839  9.41%     -   26s
     0     0  -10.79837    0  902   -9.86967  -10.79837  9.41%     -   35s
     0     0  -10.79817    0  955   -9.86967  -10.79817  9.41%     -   38s
     0     0  -10.79817    0  821   -9.86967  -10.79817  9.41%     -   45s
     0     2  -10.79816    0  808   -9.86967  -10.79816  9.41%     -   70s
     1     4  -10.75797    1  811   -9.86967  -10.79816  9.41% 65315  122s
H    2     4                      -9.8697403  -10.79816  9.41% 32658  122s
H    3     8                      -9.8746309  -10.79427  9.31% 29072  152s
H    5     8                      -9.8757733  -10.79427  9.30% 27931  152s
H    6     8                      -9.8769124  -10.78951  9.24% 25344  152s
     7    16  -10.32476    3  833   -9.87691  -10.78951  9.24% 22377  240s
H    8    16                      -9.8770024  -10.78951  9.24% 19580  240s
H   11    16                      -9.8770324  -10.78951  9.24% 26004  240s
H   13    16                      -9.8796232  -10.78260  9.14% 23688  240s
    15    25  -10.32446    4  893   -9.87962  -10.78260  9.14% 21229 3929s

Cutting planes:
  Gomory: 1
  Lift-and-project: 111
  Cover: 204
  Implied bound: 2
  MIR: 567
  Flow cover: 737
  Network: 49
  RLT: 1
  Relax-and-lift: 283

Explored 24 nodes (1299753 simplex iterations) in 3929.70 seconds (1350.26 work units)
Thread count was 10 (of 10 available processors)

Solution count 8: -9.87962 -9.87703 -9.877 ... -9.86967

Time limit reached
Best objective -9.879623210978e+00, best bound -1.078259938854e+01, gap 9.1398%
Set parameter TimeLimit to value 3600
Set parameter MIPGap to value 0.001
Set parameter MIPFocus to value 1
Gurobi Optimizer version 9.5.1 build v9.5.1rc2 (mac64[rosetta2])
Thread count: 10 physical cores, 10 logical processors, using up to 10 threads
Optimize a model with 87216 rows, 185562 columns and 824722 nonzeros
Model fingerprint: 0x949ef9bb
Model has 8228 SOS constraints
Model has 13 piecewise-linear objective terms
Variable types: 157183 continuous, 28379 integer (28379 binary)
Coefficient statistics:
  Matrix range     [8e-06, 2e+02]
  Objective range  [1e-05, 1e-05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
  PWLObj x range   [5e-01, 6e-01]
  PWLObj obj range [5e-02, 9e-01]

User MIP start produced solution with objective -9.87859 (0.18s)
```

```
Loaded user MIP start with objective -9.87859

Presolve removed 9381 rows and 93353 columns
Presolve time: 0.86s
Presolved: 77900 rows, 92274 columns, 582338 nonzeros
Variable types: 59604 continuous, 32670 integer (32657 binary)

Deterministic concurrent LP optimizer: primal and dual simplex
Showing first log only...

Concurrent spin time: 1.61s

Solved with primal simplex

Root relaxation: objective -1.081487e+01, 52627 iterations, 4.65 seconds (6.03 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 -10.81487    0  524   -9.87859 -10.81487  9.48%     -    6s
     0     0 -10.79874    0  638   -9.87859 -10.79874  9.31%     -   10s
     0     0 -10.79874    0  559   -9.87859 -10.79874  9.31%     -   10s
     0     0 -10.79823    0  701   -9.87859 -10.79823  9.31%     -   13s
     0     0 -10.79818    0  834   -9.87859 -10.79818  9.31%     -   15s
     0     0 -10.79790    0  827   -9.87859 -10.79790  9.31%     -   22s
     0     0 -10.79778    0  830   -9.87859 -10.79778  9.30%     -   25s
     0     0 -10.79756    0  863   -9.87859 -10.79756  9.30%     -   32s
     0     0 -10.79753    0  862   -9.87859 -10.79753  9.30%     -   35s
     0     0 -10.79753    0  843   -9.87859 -10.79753  9.30%     -   42s
     0     2 -10.79753    0  834   -9.87859 -10.79753  9.30%     -   75s
     1     4 -10.75725    1  858   -9.87859 -10.79753  9.30% 17412   86s
H    3     8                      -9.8787244 -10.79379  9.26% 11386  110s
H    7    16                      -9.8788644 -10.78911  9.21% 16199  154s
H    8    16                      -9.8835314 -10.78911  9.16% 14476  154s
H    9    16                      -9.8874665 -10.78911  9.12% 16280  154s
    15    26 -10.32311    4  973   -9.88747 -10.78206  9.05% 15914  635s
    25    36 -10.32300    5 1007   -9.88747 -10.78206  9.05% 46054  645s
H   35    46                      -9.8874965 -10.78206  9.05% 36351  652s
H   37    46                      -9.8875165 -10.78206  9.05% 34547  652s
H   41    46                      -9.8875265 -10.78206  9.05% 32033  652s
H   45    56                      -9.8932852 -10.78206  8.98% 29319  718s
    55    76 -10.32266    8  961   -9.89329 -10.78206  8.98% 24452  722s
    75    86 -10.32195    9  930   -9.89329 -10.78206  8.98% 18590  763s
H   76    86                      -9.8936318 -10.78206  8.98% 18346  763s
H   76    86                      -9.8936970 -10.78206  8.98% 18346  763s
H   77    86                      -9.8945242 -10.78206  8.97% 18141  763s
    85   106 -10.32260   10  927   -9.89452 -10.78206  8.97% 16605  766s
   105   126 -10.32245   12  882   -9.89452 -10.78206  8.97% 13887  783s
   125   136 -10.32209   13  927   -9.89452 -10.78206  8.97% 11962  857s
H  127   136                      -9.8996045 -10.78206  8.91% 11780  857s
H  130   136                      -9.9232938 -10.78206  8.65% 11563  857s
H  134   136                      -9.9296030 -10.78206  8.58% 11305  857s
H  164   176                      -9.9296282 -10.78206  8.58%  9597  870s
H  168   176                      -9.9302886 -10.78206  8.58%  9430  870s
   175   200 -10.32224   17  935   -9.93029 -10.78206  8.58%  9122  992s
H  180   200                      -9.9303086 -10.78206  8.58%  8901  992s
H  181   200                      -9.9387006 -10.78206  8.49%  8860  992s
H  182   200                      -9.9397085 -10.78206  8.47%  8833  992s
H  195   200                      -9.9459563 -10.78206  8.41%  8371  992s
```

```
   199   229  -10.32204   18  973   -9.94596  -10.78206  8.41%  8236 1009s
H  200   229                       -9.9459663  -10.78206  8.41%  8195 1009s
H  201   229                       -9.9460263  -10.78206  8.41%  8164 1009s
H  203   229                       -9.9462063  -10.78206  8.40%  8106 1009s
   228   278  -10.32200   19  991   -9.94621  -10.78206  8.40%  7431 1129s
   277   333  -10.32156   22  938   -9.94621  -10.78206  8.40%  6473 1266s
H  293   333                       -9.9463863  -10.78206  8.40%  6191 1266s
   332   396  -10.32078   25 1002   -9.94639  -10.78206  8.40%  5651 1412s
H  395   406                       -9.9463963  -10.78206  8.40%  4993 1453s
H  404   406                       -9.9464463  -10.78206  8.40%  4922 1453s
   405   473  -10.32038   30 1005   -9.94645  -10.78206  8.40%  4924 1626s
   472   536  -10.32015   35 1011   -9.94645  -10.78206  8.40%  4440 1752s
   535   600  -10.32012   37  988   -9.94645  -10.78206  8.40%  4076 2031s
   599   613  -10.31948   43  968   -9.94645  -10.78206  8.40%  3826 2090s
H  600   613                       -9.9464863  -10.78206  8.40%  3820 2090s
   612   681  -10.31949   44  957   -9.94649  -10.78206  8.40%  3777 2234s
   680   693  -10.31922   48  945   -9.94649  -10.78206  8.40%  3573 2253s
   692   765  -10.31845   49 1044   -9.94649  -10.78206  8.40%  3545 3600s

Cutting planes:
  Gomory: 1
  Lift-and-project: 117
  Cover: 178
  Implied bound: 3
  Clique: 1
  MIR: 486
  Flow cover: 651
  Network: 63
  RLT: 1
  Relax-and-lift: 278

Explored 764 nodes (2659211 simplex iterations) in 3600.09 seconds (2166.60 work units)
Thread count was 10 (of 10 available processors)

Solution count 10: -9.94649 -9.94645 -9.9464 ... -9.9387

Time limit reached
Best objective -9.946486312233e+00, best bound -1.078205502090e+01, gap 8.4006%
Set parameter TimeLimit to value 3600
Set parameter MIPGap to value 0.001
Set parameter MIPFocus to value 1
Gurobi Optimizer version 9.5.1 build v9.5.1rc2 (mac64[rosetta2])
Thread count: 10 physical cores, 10 logical processors, using up to 10 threads
Optimize a model with 87216 rows, 185562 columns and 824672 nonzeros
Model fingerprint: 0xabf8c9bf
Model has 8228 SOS constraints
Model has 13 piecewise-linear objective terms
Variable types: 157183 continuous, 28379 integer (28379 binary)
Coefficient statistics:
  Matrix range     [8e-06, 2e+02]
  Objective range  [1e-05, 1e-05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
  PWLObj x range   [5e-01, 6e-01]
  PWLObj obj range [5e-02, 9e-01]

User MIP start produced solution with objective -9.94733 (0.19s)
Loaded user MIP start with objective -9.94733
```

```
Presolve removed 9300 rows and 93169 columns
Presolve time: 0.87s
Presolved: 77981 rows, 92458 columns, 583817 nonzeros
Variable types: 59760 continuous, 32698 integer (32677 binary)

Deterministic concurrent LP optimizer: primal and dual simplex
Showing first log only...

Concurrent spin time: 1.71s

Solved with primal simplex

Root relaxation: objective -1.081498e+01, 50601 iterations, 5.02 seconds (6.62 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  -10.81498    0  570   -9.94733  -10.81498  8.72%     -    6s
     0     0  -10.79847    0  676   -9.94733  -10.79847  8.56%     -   11s
     0     0  -10.79847    0  649   -9.94733  -10.79847  8.56%     -   12s
     0     0  -10.79803    0  651   -9.94733  -10.79803  8.55%     -   15s
     0     0  -10.79801    0  701   -9.94733  -10.79801  8.55%     -   16s
     0     0  -10.79763    0  827   -9.94733  -10.79763  8.55%     -   23s
     0     0  -10.79743    0  809   -9.94733  -10.79743  8.55%     -   26s
     0     0  -10.79730    0  928   -9.94733  -10.79730  8.54%     -   34s
     0     0  -10.79723    0  913   -9.94733  -10.79723  8.54%     -   36s
     0     0  -10.79721    0  792   -9.94733  -10.79721  8.54%     -   43s
     0     2  -10.79721    0  787   -9.94733  -10.79721  8.54%     - 1074s
     1     4  -10.75632    1  824   -9.94733  -10.79721  8.54% 27220 1092s
     3     8  -10.32449    2  809   -9.94733  -10.79388  8.51% 27952 2028s
     7    16  -10.32395    3  827   -9.94733  -10.79054  8.48% 18198 2151s
    15    26  -10.32376    4  826   -9.94733  -10.78115  8.38% 20043 2829s
    25    35  -10.32362    5  862   -9.94733  -10.78115  8.38% 50342 3600s

Cutting planes:
  Gomory: 1
  Lift-and-project: 119
  Cover: 177
  Implied bound: 1
  MIR: 531
  Flow cover: 672
  Network: 47
  RLT: 3
  Relax-and-lift: 275

Explored 34 nodes (1453806 simplex iterations) in 3600.04 seconds (1066.66 work units)
Thread count was 10 (of 10 available processors)

Solution count 1: -9.94733

Time limit reached
Best objective -9.947326312232e+00, best bound -1.078114711613e+01, gap 8.3824%

Process finished with exit code 0
```