**objective**

Solve large combinatorial problems

- Several problems in machine learning, computer vision and data analysis can be formulated using NP-hard combinatorial optimization problems.
- In many of these applications, approximate solutions for these NP-hard problems are 'good enough'.

We develop novel theory and algorithms to solve combinatorial problems approximately, in parallel. On three different problems, our solver can outperform Cplex (a commercial solver) by up to an order of magnitude in runtime, while achieving comparable solution quality.

**background**

- LP rounding is a 4 step scheme to approximate combinatorial problems with theoretical guarantees on solution quality.

```
Formulate Integer Program (IP)  
Relax IP to form a Linear Program (LP) 
Solve the LP  
Round LP solution
```

**main result**

- We can round approximate solutions to LPs (instead of exact solutions) without losing out on quality.

```
min c^T x  s.t.  
Ax = b,  x ≥ 0
```

**Definition** We define $x$ to be an $(\epsilon, \delta)$-approximate LP solution if

- Its objective is at most $\delta$ away from the optimal objective.
  $$|c^T x - c^T x^*| \leq \delta c^T x^*$$
- It is at most $\epsilon$ away from feasibility
  $$\|Ax - b\|_\infty \leq \epsilon$$

**example**

**The vertex cover problem**
Find a set of vertices that cover the graph

```
Integer Program (IP)  
min \sum_{v \in V} x_v  s.t.  
x_u + x_v \geq 1  \forall (u,v) \in E  
x_v \in \{0,1\}  \forall v \in V
```

```
Relaxed Linear Program (LP)  
min \sum_{v \in V} x_v  s.t.  
x_u + x_v \geq 1  \forall (u,v) \in E  
x_v \in [0,1]  \forall v \in V
```

Vertex cover has an approximation factor of 2

**combinatorial problems**

<table>
<thead>
<tr>
<th>Problem Family</th>
<th>Approximation Factor</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Covering</td>
<td>K or log(n)</td>
<td>Advertising, Classification, Tracking</td>
</tr>
<tr>
<td>Set Packing</td>
<td>eK + o(K)</td>
<td>MAP-Inference, Language processing</td>
</tr>
<tr>
<td>Multiway Cut</td>
<td>1.5 - 1/K</td>
<td>Entity resolution, Computer vision</td>
</tr>
<tr>
<td>Graphical Models</td>
<td>Heuristic</td>
<td>Clustering, Role labeling</td>
</tr>
</tbody>
</table>

**novel theory**

**Approximate LP solutions**

- We use a quadratic penalty (QP) formulation to solve the LP relaxation of the combinatorial problem.
- We solve the QP formulation with a parallel asynchronous co-ordinate decent method.
- We view the QP solutions as exact solutions of a perturbed LP, and use Renegar's analysis to quantify the difference with the exact LP solution.
- We derive convergence rates that translate to bounds on worst case running time for the entire LP rounding scheme.

```
min c^T x  s.t.  
Ax = b,  x ≥ 0
```

**novel implementation**

We compared our solver with Cplex v.12.5 (a state-of-the-art commercial LP solver).

- Our solver was, on average, around 10x faster than Cplex on vertex cover and independent set.
- Cplex was unable to solve any of the multiway cut problems in 3600 seconds.
- Our solver produced feasible integral solutions that were of comparable quality with Cplex.

**maximization problems**

```
Dataset  | Cplex | Vertex Cover | Time (s) | Solution  | Cplex | Time (s) | Solution  |
---------|-------|--------------|---------|-----------|-------|---------|-----------|
Amazon   | 2.04e+5 | 95.8e+1 | 297     | 2.04e+5   | 95.8e+1 | 297     |
DBLP     | 2.06e+5 | 153.8e+1 | 270     | 2.06e+5   | 153.8e+1 | 270     |
Google+  | 1.31e+5 | 99.9e+1 | 4.47    | 1.31e+5   | 99.9e+1 | 4.47    |
```

**minimization problems**

```
Dataset  | Cplex | Independent Set | Time (s) | Solution  | Cplex | Time (s) | Solution  |
---------|-------|-----------------|---------|-----------|-------|---------|-----------|
Amazon   | 1.56e+5 | 23.0      | 2.09    | 1.56e+5   | 23.0      | 2.09    |
DBLP     | 1.41e+5 | 32.2      | 2.72    | 1.41e+5   | 32.2      | 2.72    |
Google+  | 9.39e+4 | 44.5      | 4.37    | 9.39e+4   | 44.5      | 4.37    |