**Abstract**

CP-based Lagrangian relaxation (CP-LR) is an efficient optimization technique that combines cost-based filtering with Lagrangian relaxation in a constraint programming context. The state-of-the-art filtering algorithms for the WeightedCircuit constraint that encodes the traveling salesman problem (TSP) are based on this approach. In this paper, we propose an improved CP-LR approach that locally modifies the Lagrangian multipliers in order to increase the number of filtered values. We also introduce two new algorithms based on the latter to filter WeightedCircuit. The experimental results on TSP instances show that our algorithms allow significant gains on the resolution time and the size of the search space when compared to the state-of-the-art implementation.

**Cost-Based Filtering**

[Focacci et al., 1999] Considering

\[ Z = \min f(x_1, \ldots, x_n) \]

where the best solution gives an upper bound \( U \) and a relaxation gives a lower bound \( L \) of \( Z \):

- If \( L > U \), infeasibility
- Else, if \( Z(x_i = \mu, \mu > U \), where \( Z(x_i = \mu \) is the optimal value of the relaxation with the additional constraint \( x_i = \mu \) is removed from \( \{x_i\} \)

**Lagrangian Relaxation**

\[
Z = \min f(x_1, \ldots, x_n) \quad Z_{LR}(\lambda) = \min f(x_1, \ldots, x_n) + \lambda^T (A x - b)
\]

\[
B x \leq d \quad \text{s.t.}
\]

\[
A x \leq b \quad \text{s.t.}
\]

\[
B x \leq d \quad \text{B} \quad \text{s.t.}
\]

\[
x \in X \quad \text{s.t.} \]

where \( \lambda \geq 0 \) are Lagrangian multipliers
- For any \( \lambda > 0 \), \( Z_{LR}(\lambda) \) is a lower bound of \( Z \)
- To obtain the bound: max \( Z_{LR}(\lambda) \)

**Iterative methods (subgradient descent)**

**Improved CP-LR Approach**

**Simple Algorithm**

Given \((i, j) \in E\)...
- Consider particular cases of a lemma to modify, if possible, simultaneously \( \lambda \) and \( \lambda \) with safe values that can only increase \( Z_{LR}(\lambda) \)
- Two versions: Relaxed and Complete, where the former uses a faster pre-processing with potentially a smaller increase
- Worst-case time complexity is \( O(|V|^3) \)

**Simple Algorithm**

Given \((i, j) \in E\)...
- Consider particular cases of a lemma to modify, if possible, simultaneously \( \lambda \) and \( \lambda \) with safe values that can only increase \( Z_{LR}(\lambda) \)
- Two versions: Relaxed and Complete, where the former uses a faster pre-processing with potentially a smaller increase
- Worst-case time complexity is \( O(|V|^3) \)

**Improved CP-LR Approach**

**TSP and WeightedCircuit**

Given \( G = (V, E) \), weight function \( w: E \rightarrow \mathbb{R} \)
- Binary variables \( x = (x_1, \ldots, x_{|V|}) \)
- Integer variable \( w \in [0, \ldots, K] \)

**WeightedCircuit \( x, G, w \) is satisfied if**

\[
T = \{ (e : x_e = 1) \} \quad \text{is a Hamiltonian cycle of} \quad G
\]

\[
\sum_{e \in E} w(e)e \leq z
\]

Traveling Salesman Problem (TSP)

\[
\min \quad z \quad \text{s.t.} \quad \text{WeightedCircuit}(x, G, w)
\]

**Filtering algorithms** ([Benchimol et al., 2012]):
- CP-LR approach using the 1-tree TSP relaxation
- Held & Karp, 1970:

\[
Z_{LR}(\lambda) = \min \sum_{(i,j) \in E} \left( w(i,j) + \lambda \right) x_{i,j} + \sum_{i=1}^{n} \lambda
\]

\[
T = \{ (e : x_e = 1) \} \quad \text{is a 1-tree}
\]

- 1-tree: Spanning tree on \( \{1\} \) with 2 distinct edges adjacent to 1
- \( Z_{LR}(\lambda) \) obtained with a minimum 1-tree using \( \hat{c}(i, j) \)

**α-SETS Algorithm**

Given an edge, we could formulate the constraints on \( x \):

\[
\lambda_i + \lambda_j - \lambda_{i,j} \leq w(c, d) - w(a, b)
\]

However, we would have \( O(|V|^4) \) constraints!
- Consider instead an incremental set of constraints \( \Omega \)
- Searches a set of nodes \( A \) and a value \( \alpha \geq 0 \) such that \( \lambda_i = \lambda_j = \alpha, \forall \alpha \in A \)
- \( \Omega \) is defined as such a constraint
- If the maximal value \( \alpha \) is \( \Omega \), then it is written as such a constraint
- If \( \alpha = 0 \) and a smart choice of node is made
- Iterative: Applied as long \( A \) is found
- With \( |A| \leq C \), the worst-case time complexity is \( O(C^3 |E|/\beta^2) \)
- Hybrid algorithm: Apply SIMPLE Complete first

**Experiments**

- Benchmark: 28 TSP instances from TSPLIB
- Compared to the state-of-the-art implementation of WeightedCircuit with Choco / Choco Graph

| Instance | Choice | Source Relaxed | Source Complete | Human | T
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>tsp1</td>
<td>120.41</td>
<td>29.04</td>
<td>28.41</td>
<td>28.76</td>
<td>28.72</td>
</tr>
<tr>
<td>tsp2</td>
<td>120.41</td>
<td>29.04</td>
<td>28.41</td>
<td>28.76</td>
<td>28.72</td>
</tr>
</tbody>
</table>

\[
\text{Table 1: Num. of search nodes} (N) \text{ and solving time in seconds} (T)
\]

**Contributions**

- Introduction of an improved CP-LR approach
- Application to the WeightedCircuit constraint (Simple and α-sets algorithms)
- Significant improvement on the TSP solving time