Improved CP-Based Lagrangian Relaxation Approach with an Application to the TSP

Raphaël Boudreault Claude-Guy Quimper

Université Laval, Québec, Canada

UNIVERSITÉ LAVAL

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

$$Z = \min f(x_1, \ldots, x_n)$$
$$\text{s.t.} \quad \cdots$$

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 $L[x_i = \mu] > U$, where $L[x_i = \mu]$ is the optimal value of the relaxation with the **additional constraint** $x_i = \mu$, μ is **removed** from $\mathrm{dom}(x_i)$

Lagrangian Relaxation

$$Z = \min \mathbf{c}^T\boldsymbol{x} \quad \text{s.t.} \; A\boldsymbol{x} \le \boldsymbol{b},\; B\boldsymbol{x} \le \boldsymbol{d},\; \boldsymbol{x} \in X \quad \xrightarrow{\text{LR}} \quad Z_{LR}(\boldsymbol{\lambda}) = \min \mathbf{c}^T\boldsymbol{x} + \boldsymbol{\lambda}^T(A\boldsymbol{x} - \boldsymbol{b}) \quad \text{s.t.} \; B\boldsymbol{x} \le \boldsymbol{d} \; (\mathcal{B}),\; \boldsymbol{x} \in X$$

where $\boldsymbol{\lambda} \ge 0$ are **Lagrangian multipliers**

- **For any** $\boldsymbol{\lambda} \ge 0$, $Z_{LR}(\boldsymbol{\lambda})$ is a lower bound of Z
- To obtain the **best bound**: $\max_{\boldsymbol{\lambda} \ge 0} Z_{LR}(\boldsymbol{\lambda})$
- **Iterative methods** (subgradient descent)

CP-Based Lagrangian Relaxation (CP-LR)

[Sellmann, 2004] Given $\mathrm{Prop}(\mathcal{B})$, an efficient **cost-based filtering algorithm** for the substructure $\mathcal{B}$:

- Maximize $Z_{LR}(\boldsymbol{\lambda})$ while using $\mathrm{Prop}(\mathcal{B})$ for each **subproblem** encountered during the descent
- **Suboptimal multipliers** can lead to **more filtering**

Improved CP-LR Approach

Question:

Given a variable x and a value μ in its domain, could we **temporarily** change $\boldsymbol{\lambda}$ so that μ is **filtered** due to costs?

Goal: Find $\boldsymbol{\lambda}'$ such that $Z_{LR}(\boldsymbol{\lambda}')[x = \mu] > U$

Proposed framework:

1. Find **conditions** on $\boldsymbol{\lambda}$ such that the relaxed solution $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ remains optimal
2. For each x_i and $\mu \in \mathrm{dom}(x_i) \setminus \{x_i^*\}$, try to find $\boldsymbol{\lambda}'$ under these **conditions** that reach the **goal**

TSP and WEIGHTEDCIRCUIT

Given $G = (V, E)$, weight function $w \colon E \to \mathbb{Z}$

- **Binary** variables $\boldsymbol{x} = \left(x_{e_1}, \ldots, x_{e_{|E|}}\right)$
- Integer variable $z \in [0, K]$

$\text{WEIGHTEDCIRCUIT}(\boldsymbol{x}, z, G, w)$ is **satisfied** iff

- $T = \{e : x_e = 1\}$ is a **Hamiltonian cycle** of G
- $\sum_{e \in E} w(e) x_e \le z$

Traveling Salesman Problem (TSP)

$$\min \; z \quad \text{s.t.} \; \text{WEIGHTEDCIRCUIT}(\boldsymbol{x}, z, G, w)$$

Filtering algorithms [Benchimol *et al.*, 2012]:

- **CP-LR approach** using the **1-tree TSP relaxation** [Held & Karp, 1970]:

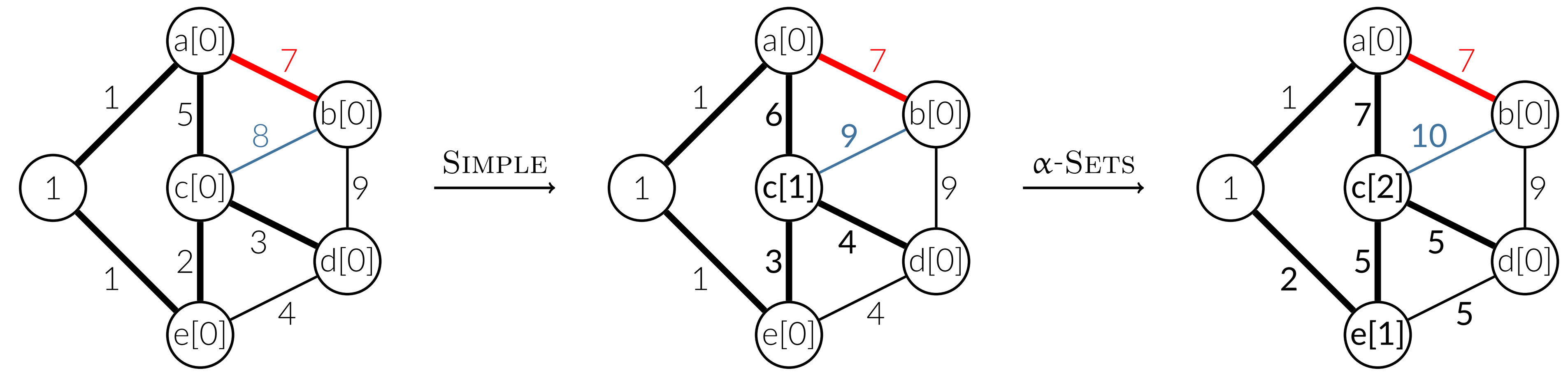
$$Z_{LR}(\boldsymbol{\lambda}) = \min \sum_{\{i,j\} \in E} \underbrace{(w(i,j) + \lambda_i + \lambda_j)}_{\tilde{w}(i,j)} x_{\{i,j\}} - 2 \sum_{i \in V} \lambda_i$$
$$\text{s.t.} \quad T = \{e : x_e = 1\} \text{ is a 1-tree}$$

- **1-tree**: Spanning tree on $V \setminus \{1\}$ + 2 distinct edges adjacent to 1
- $Z_{LR}(\boldsymbol{\lambda})$ obtained with a **minimum 1-tree** using $\tilde{w}(i, j)$

$U = 23$

Reduced cost: $\bar{c}(\{b,c\}) = \tilde{w}(b,c) - \tilde{w}(a,b)$

$Z_{LR}(\boldsymbol{\lambda})[x_{\{b,c\}} = 1] = Z_{LR}(\boldsymbol{\lambda}) + \bar{c}(\{b,c\}) > U \implies \{b,c\}$ is forbidden



$Z_{LR}(\boldsymbol{\lambda}) = 19 - 2(0) = 19$; $\bar{c}(\{b,c\}) = 8 - 7 = 1$ ✗

$Z_{LR}(\boldsymbol{\lambda}) = \mathbf{22} - 2(\mathbf{1}) = \mathbf{20}$; $\bar{c}(\{b,c\}) = \mathbf{9} - 7 = \mathbf{2}$ ✗

$Z_{LR}(\boldsymbol{\lambda}) = \mathbf{27} - 2(\mathbf{3}) = \mathbf{21}$; $\bar{c}(\{b,c\}) = \mathbf{10} - 7 = \mathbf{3}$ ✓

SIMPLE Algorithm

Given $\{i, j\} \in E$...

- Considers particular cases of a lemma to modify, if possible, **simultaneously** λ_i and λ_j with *safe* values that can only **increase** $Z_{LR}(\boldsymbol{\lambda}')[x_e = b]$
- Two versions: ***Relaxed*** and ***Complete***, where the former uses a faster pre-processing with potentially a smaller increase
- Worst-case time complexity is $O(|V|)$

α-SETS Algorithm

Given an edge, we could formulate the constraints on $\boldsymbol{\lambda}'$...

$$\lambda'_a + \lambda'_b - \lambda'_c - \lambda'_d \le w(c, d) - w(a, b)$$

However, we would have $O(|V|^4)$ constraints!

- Considers instead an **incremental** set of constraints Ω
- Searches a set of nodes A and a value $\alpha \ge 0$ such that $\lambda'_u \leftarrow \lambda_u \pm \alpha, \forall\, u \in A$
- Each constraint $\omega \in \Omega$ can be written as $c_\omega \cdot \alpha \le m_\omega$
- If the maximal value $\alpha^* > 0$, $Z_{LR}(\boldsymbol{\lambda}')[x_e = b]$ is increased. Else, $\alpha^* = 0$ and a smart choice of node is made
- **Iterative**: Applied as long as A is found
- With $|A| \le C_m$, the worst-case time complexity is $O(C_m |V| |E| 4^{C_m})$
- HYBRID algorithm: Apply SIMPLE ***Complete*** first

Experiments

- Benchmark: **28 TSP instances** from TSPLIB
- Compared to the **state-of-the-art implementation** of WEIGHTEDCIRCUIT within *Choco* / *Choco Graph*

Instance	*Choco*		SIMPLE *Relaxed*		SIMPLE *Complete*		HYBRID	
	N	T	N	T	N	T	N	T
gr96	520	3.2	436	2.7	365	2.4	**323**	**2.4**
kroA100	1488	8.3	1066	5.9	1211	6.8	**873**	**5.5**
kroB100	2312	12.3	1838	8.9	2042	11.0	**1462**	**7.9**
kroC100	360	2.3	247	1.5	263	1.7	**210**	**1.5**
kroD100	128	1.1	132	1.0	127	1.0	**114**	**0.9**
kroE100	1915	9.8	1578	8.4	1532	8.8	**1113**	**6.9**
gr120	324	3.0	199	1.9	254	2.5	**145**	**1.6**
pr124	224	2.5	195	2.1	169	2.1	**157**	**1.9**
ch130	908	8.7	768	7.5	673	7.2	**518**	**5.4**
pr136	72574	561.4	78781	558.2	72224	**503.0**	**66481**	519.3
gr137	923	9.9	**716**	**8.1**	756	9.4	746	9.8
pr144	149	2.8	65	**1.3**	66	1.4	**62**	1.4
ch150	934	10.9	681	7.8	710	9.6	**498**	**7.3**
kroA150	5652	65.0	2461	**26.8**	2910	35.4	**2329**	27.4
kroB150	112078	1218.6	109715	1115.5	85765	925.6	**67134**	**671.2**
pr152	335	6.2	641	9.4	446	8.2	**277**	**5.7**
si175	38068	486.0	35755	436.1	38775	520.6	**29915**	**387.3**
rat195	18785	361.1	14905	290.3	13113	266.0	**10281**	**215.9**
d198	**5702**	**101.4**	7322	112.2	6695	109.0	6765	118.1
kroA200	1176978	15766.3	863416	12654.3	884545	12399.6	**687347**	**9618.6**
kroB200	46484	739.2	33405	555.6	32779	514.3	**27392**	**430.0**
gr202	1568	20.2	991	11.9	853	11.8	**644**	**10.0**
tsp225	125761	2426.2	72357	1416.6	74639	1546.7	**51766**	**1174.0**
gr229	458131	7367.2	303679	4568.1	268029	4559.1	**215354**	**3272.8**
pr264	123	8.8	130	8.9	114	10.0	**88**	**8.6**
a280	**1605**	**30.7**	2429	40.2	1832	33.7	2005	37.3
lin318	3794	115.1	2296	72.8	2409	81.7	**1128**	**39.1**
gr431	415036	20485.0	309152	17077.4	278439	15917.5	**241454**	**15074.7**
Mean	89031	1779.8	65906	1393.3	63276	1339.5	**50592**	**1130.8**

Table 1. Num. of search nodes (**N**) and solving time in seconds (**T**).

Contributions

- Introduction of an **improved CP-LR approach**
- Application to the WEIGHTEDCIRCUIT constraint filtering (SIMPLE and α-SETS algorithms)
- **Significant improvement** on the TSP solving time

raphael.boudreault.1@ulaval.ca, claude-guy.quimper@ift.ulaval.ca