



Research article

On lot sizing with capacity acquisition

Jia Guo¹ and Minjiao Zhang^{2,*}

¹ Department of Management, College of Business and Economics, California State University East Bay, Hayward, CA 94542, USA

² Department of Economics, Finance, and Quantitative Analysis, Kennesaw State University, Kennesaw, GA, 30144, USA

* **Correspondence:** Email: mzhang16@kennesaw.edu.

A. Supplementary computational results

A.1. Implementation-sensitivity analysis

We tested three root-loop cut-delivery strategies to evaluate how implementation choices affect the computational behavior of the proposed inequalities. Implementation 1 uses a fixed single-cut root loop, in which at most one violated cut from each selected group is added per round. Implementation 2 keeps the single-cut-per-group structure but adds adaptive stopping based on root-bound improvement. Implementation 3, used in Section 4.2, generates a pool of violated cuts, ranks them by violation, and adds a limited number of sufficiently strong cuts from each selected group.

Table 1. Aggregate comparison of root-loop implementations on the core instances.

Implementation	Method	Root gap after (%)	Root-gap red. (%)	Median time	Median nodes	Mean cuts
Fixed single-cut	Baseline	14.54	0.00	90.98	237,365	0.0
Fixed single-cut	First	13.68	5.60	70.84	215,008	15.0
Fixed single-cut	Second	11.22	28.00	66.18	175,795	14.87
Fixed single-cut	Both	10.75	30.70	60.66	172,568	30.0
Adaptive single-cut	Baseline	14.53	0.00	84.84	237,365	0.0
Adaptive single-cut	First	13.73	5.10	66.86	205,614	3.0
Adaptive single-cut	Second	11.24	27.80	96.81	238,766	13.2
Adaptive single-cut	Both	10.75	30.70	64.36	166,729	29.6
Pooled adaptive	Baseline	14.54	0.00	77.89	237,365	0.0
Pooled adaptive	First	13.69	5.65	174.49	305,271	9.0
Pooled adaptive	Second	11.27	22.43	53.63	142,490	37.6
Pooled adaptive	Both	10.66	26.67	95.56	243,199	79.2

Across the three implementations, the proposed inequalities consistently reduce the root gap, especially the second-group and combined cuts. However, the final MIP runtime and node count depend on the cut-management strategy. This confirms that the value of a valid inequality in a modern MIP solver depends not only on its strength, but also on how the cut is selected, added, and coordinated with the solver's internal search process.

A.2. Perturbed-demand robustness check

As a supplementary robustness check, we generated perturbed-demand instances by multiplying each period's demand by an independent random factor within $\pm 10\%$. These perturbed instances are not intended to replace the structured core benchmark. Rather, they provide a stress test for whether the proposed cuts continue to strengthen the root relaxation under noisy demand. Table ?? reports the aggregate results.

Table 2. Perturbed-demand robustness check.

Method	Solved	Time limit	Mean final gap (%)	Median final gap (%)
Baseline	8	7	1.044	0.010
First	8	7	1.083	0.010
Second	8	7	1.077	0.010
Both	8	7	1.134	0.010

Method	Root gap before (%)	Root gap after (%)	Median time	Median nodes	Mean cuts
Baseline	14.402	14.402	97.47	199,604	0.0
First	14.395	13.585	122.38	248,371	9.0
Second	14.374	11.271	114.28	176,578	32.4
Both	14.374	10.670	101.73	185,114	70.8

The proposed cuts continue to strengthen the root relaxation under demand perturbation. The average root gap decreases from 14.40% for the baseline to 13.58% with the first group, 11.27% with the second group, and 10.67% with both families. However, the root-bound improvement does not translate into a higher solved count; all four methods solve 8 of the 15 perturbed instances within the time limit. We therefore interpret the robustness check as evidence that the cuts remain formulation-strengthening under noisy demand, but not as evidence of uniform computational speedup.



AIMS Press

©2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)