A New Formulation of a Parking Lot Design Problem

A. Michael Forbes, B. Mitchell Harris*

School of Mathematics and Physics, The University of Queensland, Brisbane, Australia
m.forbes@uq.edu.au, m.g.harris@uq.edu.au

Extended abstract submitted for presentation at the 11th Triennial Symposium on Transportation Analysis conference (TRISTAN XI)
June 19-25, 2022, Mauritius Island

April 4, 2022

Keywords: parking, composite variables, logic based Benders decomposition

1 INTRODUCTION

A recent paper in *Transportation Science*, Stephan *et al.* (2021), considered the "optimization of parking lots with the help of mathematical programming" where, for a fixed lot, the goal is to maximize the number of reachable perpendicular parking spaces. They propose several novel flow based models. *We* describe a new formulation based on composite variables and logic based Benders decomposition.

2 FORMULATION

Let S denote the set of 1×1 squares identified by their (i, j) coordinates. Parking spaces have size 1×2 or 2×1 and driving squares must be at least 2×2 squares wide to allow bidirectional traffic flow (*we could use another resolution*). We can generate the sets P of all legal parking tiles and D of all legal driving tiles *a priori*. Initialize $P = D = \emptyset$ and for all $(i, j) \in S$,

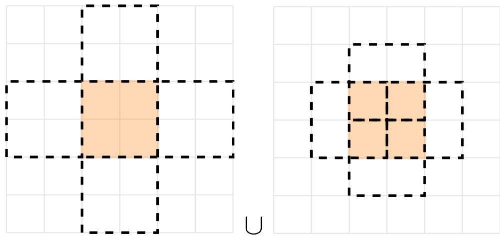
- Add $p = \{(i, j), (i + 1, j)\}$ to P if $p \subset S$,
- Add $p = \{(i, j), (i, j + 1)\}$ to P if $p \subset S$, and
- Add $d = \{(i, j), (i, j + 1), (i + 1, j), (i + 1, j + 1)\}$ to D if $d \subset S$.

We have a set $F \subset D$ which is the set of driving tiles which must be active; in particular $e \in D$ is the entrance tile. Driving lanes will be wide enough automatically since we only generate valid driving tiles. We need to find a valid packing of tiles into S which maximizes the number of parking tiles reachable from the entrance, where driving tiles may overlap. To that end we introduce the following binary variables:

- $x_p = 1$ if tile $p \in P$ is active, $x_p = 0$ otherwise;
- $y_d = 1$ if tile $d \in D$ is active, $y_d = 0$ otherwise; and
- $z_s = 1$ if square $s \in S$ is a driving square, $z_s = 0$ otherwise.

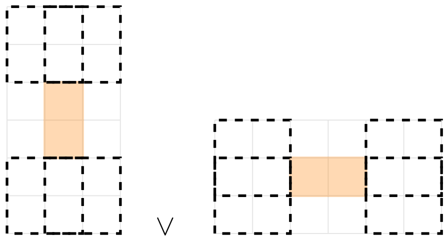
The number of variables is bounded above by $4|S|$. It will be useful to define the following neighbourhoods:

- $\mathcal{N}_D(d) \subset \mathcal{D}$ is the set of driving tiles adjacent to $d \in D$ including up to four properly adjacent tiles, and up to four overlapping tiles:



$\cup \qquad = \mathcal{N}_D(d).$

- $\mathcal{N}_P(p) \subset D$ is the set of up to 4 driving tiles adjacent to $p \in P$:



$\vee \qquad = \mathcal{N}_P(p).$

The following integer program gives a valid packing of tiles:

$$\max \quad \sum_{p \in P} x_p \tag{1}$$

$$\text{subject to} \quad x_p \leqslant \sum_{d \in \mathcal{N}_D(p)} y_d \qquad \forall p \in P, \tag{2}$$

$$z_s + \sum_{p \in P : s \in p} x_p \leqslant 1 \quad \forall s \in S, \tag{3}$$

$$y_d \leqslant z_s \qquad \forall d \in D,\ s \in d, \tag{4}$$

$$y_d = 1 \qquad \forall d \in F. \tag{5}$$

The objective function, (1), is the number of active parking tiles; (2) ensures that every parking tile is adjacent to a driving tile; (3) ensures that parking tiles do not overlap driving squares; (4) forces driving squares; and (5) fixes the set tiles. It is not hard to see that (1 - 5) does not guarantee each parking tile will be reachable from the entrance. We remedy this by adding logic based Benders cuts at incumbent nodes of the branch-and-bound tree. While there are several promising cutting schemes, here we summarize one.

Let (x', y', z') be a feasible incumbent solution to (1 - 5). With special purpose code, we can find the set of all contiguous "regions" of active driving tiles; that is, a partition of the active driving tiles such that every tile in the region is reachable from every other tile in the region. If (x', y', z') is optimal and there is only one region, then we are done, since that region must contain the entrance.

If, on the other hand, there is more than one contiguous region, let $R \subset D \setminus \{e\}$ be a region not containing the entrance. Let

$$\mathcal{N}_D(R) = \left\{ d \in D : y'_d = 0,\ d \in N_d(d') \text{ for some } d' \in R,\ d \not\subseteq \cup_{d' \in R} d' \right\}$$

denote the set of strict inactive "neighbours" of R. Then $\mathcal{N}_D(R)$ is a cut set in D separating R from e. We can add the following Benders feasibility cuts as a lazy constraints:

$$y_d \leqslant \sum_{d' \in \mathcal{N}_D(R)} y_{d'} \quad \text{for all} \quad d \in R. \tag{6}$$

The principle difficulties with this formulation are the large number of legal packings, and the weak LP relaxation of (1 - 5); the main culprit being (3), since all five tiles may be partially active on each square. Advantages, however, include the flexibility of the composite variables, and the fact that no big M constraints are required.

3 IMPROVEMENTS

We propose a heuristic which constraints flow between adjacent driving tiles. Consider the network (D, A) with $A = \{(d, d') : d \in D, d' \in \mathcal{N}_D(d)\}$. For each $d \in D \setminus e$ we calculate the shortest distance from d to e in the graph with respect to two sets of arc lengths:

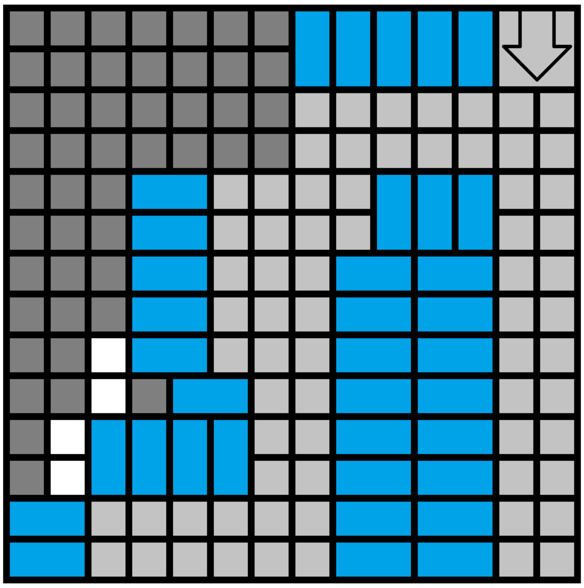
- First with respect to unit lengths, and
- Second, where the length of an arc (d, d') is penalized by the number of parking tiles adjacent to d. A promising penalty function is given by $1/|\mathcal{N}_D(d')|$.

Then we delete each arc (d, d') from A unless d' is *at least* as close to e as d is with respect to at least one of the two metrics. For each $d \in D \setminus \{e\}$ we add the following constraint to (1-5):

$$y_d \leqslant \sum_{\substack{d' \in \mathcal{N}_D(d): \\ (d,d') \in A}} y_{d'}. \tag{7}$$

The idea is to force the driving tiles to flow generally closer to the entrance. Unit lengths alone do not behave well with obstacles or corners, but the penalized metric allows us to deviate to make room for parking tiles. There is scope to explore more sophisticated heuristics of this nature.

This heuristic achieves the optimal objective value for the instance depicted in Figure 1 of Stephan *et al.* (2021) in under a minute. See the figure below. More detailed computational experiments are in progress, and will be described in the conference presentation.



References

Stephan, Konrad, Weidinger, Felix, & Boysen, Nils. 2021. Layout Design of Parking Lots with Mathematical Programming. *Transportation Science*, **55**(06).