Abstract

We adapt the competitive location model based on maximal covering to include the knowledge that a competitor will enter the market later with a single new facility. The objective is to locate facilities under a budget constraint in order to maximise the remaining market share after the competitor’s later entry.

We develop mixed zero-one programming formulations for each of the following three strategies: the maxmin strategy where the worst possible competitor choice is considered, the minimum regret strategy, and the von Stackelberg strategy in which the competitor also optimises its market share. Some computational results show the feasibility and limits of these models.

1 Introduction

Consider an open market with demands \( d_i > 0 \) given at several known locations \( i \in I \). Decision maker \( L \) (the ‘leader’– or Lieselot, explaining why we use the female form for \( L \)) has a given budget \( B \) to spend on opening facilities in order to serve part of this market. Facilities may be opened anywhere at a finite number of potential sites \( s \in S \), at fixed costs \( f_s \).

The process of consumer capture is described by the patronising sets \( S_i \subset S \ (i \in I) \) as follows: as soon as \( L \) locates a facility within \( S_i \), consumer \( i \) will be captured and its full demand is earned. In other words we consider an ‘all or nothing rule’ or ‘the winner gets it all’ as expressed in [4]. This assumption will be slightly amended in section 5.

The patronising sets \( S_i \) may arise in different ways. They might for example be the set of potential sites lying within a threshold distance from \( i \)’s location. This threshold distance will depend on the type of service rendered, some types of service being useless above a certain distance. In the competitive setting we look at here, this threshold distance will possibly also be conditioned by the presence of existing (competing) facilities which already offer the same or similar service. The exact shape of the patronising sets reflects the consumer behaviour rules of the model, a full discussion of which falls beyond the scope of this paper, but may be found in several survey papers, e.g. [7, 3]. A few remarks about them will be given in section 1.3.

1.1 Maximal covering revisited

The leader’s aim to obtain a largest possible market share leads to a model which is mathematically equivalent to the now classical maximal covering model, introduced by Church and ReVelle [1]. It may be formulated using the following sets of location and capturing variables:

---

*This paper corrects and extends earlier work described in [9]
†Corresponding author
• $L_s \ (s \in S)$ answering the question ‘Does $L$ open a facility at $s$ ?’
• $x_i \ (i \in I)$ answering the question ‘is consumer $i$ captured ?’

\[
\begin{align*}
\text{MAX} & \quad \sum_{i \in I} d_i x_i \\
\text{such that} & \quad x_i \leq \sum_{s \in S_i} L_s \quad (i \in I) \\
& \quad \sum_{s \in S} f_s L_s \leq B \\
\text{variables} & \quad L_s \in \{0, 1\} \quad (s \in S) \\
& \quad 0 \leq x_i \leq 1 \quad (i \in I)
\end{align*}
\]

Note that we allow the capturing variables $x_i$ to be continuous and bounded between 0 and 1, despite their interpretation which seems to call for a binary nature. This is contrary to the classical formulation with binary $x_i$ [1] which has been used in all subsequent literature.

In fact the optimisation process will automatically push these auxiliary variables to one of their extreme values 0 or 1. Indeed, the only constraint in which a particular $x_i$ appears is the covering constraint $x_i \leq \sum_{s \in S_i} L_s$, the right-hand side of which can only have integer values. Either none of the $s \in S_i$ obtains a facility, the right-hand side is zero and $x_i = 0$, or a facility is opened at some site $s \in S_i$, the right-hand side is at least 1, so $x_i$ can still have any value between 0 and 1, and since $d_i > 0$ the maximisation process calls for setting $x_i = 1$.

This observation is important in practice, since the presence of too many binary variables may prohibit the use of commercial software due to built-in limitations. Also, when a branch and bound process is used to solve the covering problem to optimality, which is usually the case, it avoids possible branchings on $x_i$ variables, which would be totally ineffective. Therefore any reduction in the number of binary variables, as obtained here, may allow to increase the problem size without reaching the limits of tractability.

This remark will acquire particular significance in the sequel when developing more complicated models.

It might be useful, however, to enforce more directly the effect obtained here by the optimisation process: make sure that as soon as one site $s \in S_i$ has been opened, that then demand $i$ is indeed captured. This is obtained by adding the constraints

\[
L_s \leq x_i \quad \text{for all } s \in S_i
\]

which push $x_i$ to 1 as soon as $L_s = 1$.

As argumented in [8] adding these constraints to the covering model above ensures correct behaviour of the auxiliary variables $x_i$ for any type of objective, even for a minimising objective or one with negative objective coefficients.

1.2 Taking foresight into account

Although the maximal covering model may take into account existing competition in the definition of the patronising sets, it does not consider the possible effects of future competition. But what would happen if a competitor $F$ (the ‘follower’ – or ‘Frank’, explaining the male form used for $F$, which emphasizes the distinction with the female form used for leader $L$) would enter the market once leader $L$ has already established her facilities? $L$ will then probably lose some of her current consumers to $F$.

In the remainder of this paper we consider three different ways the leader might consider to take this foresight into account. Assuming she considers $F$’s future entry site as totally unpredictable, she is playing a ‘game against nature’, for which several strategies have been proposed. In the next two sections we discuss the two main classical strategies of this type: maximisation of worst case and minimisation of regret, and show how to formulate them as mixed binary programs. These give rise to variants of some discrete competitive location models with uncertainty already studied
in [14]. Then in the following section \( L \) assumes rational choice of the follower \( F \), and wants to maximise her market share remaining after \( F \)’s entry. The resulting two-level optimisation model was called the ‘preemptive’ model in [12], where a heuristic approach is described for its solution. In this paper we show how to obtain a full MIP formulation of this model, under the restrictive assumption that \( F \) will open a single facility only. The paper ends with a section describing setup and results of some computational experiments aimed at studying the feasibility of these models.

An in depth comparison of the behaviour of the three models will be left for future work. But one important observation may already be made. It is well known that in the case of a zero-sum game all three strategies are equivalent, so lead to the same optimal solutions. In our context this would mean that for any leader and follower facility sites, their market shares would be complementary and cover the whole market. This would happen for example in case no other competition pre-exists and the service offered is essential, in the sense that any demand will always either go to the leader or to the follower. We therefore assume to work here in a market which is large enough for the leader’s and follower’s market shares not to cover the whole market.

### 1.3 Concepts and notations about competition rules

In order to forecast what will happen, we need to make some precise assumptions on the consumer’s behaviour. In this paper we will adhere to the traditional covering philosophy, where each consumer is supposed to make a definite choice between \( L, F \) or none of them, and be either fully served from one of them or not served at all by \( L \) nor \( F \). The actual choice might go to the competitor with the nearest facility, or the cheapest facility in terms of price (inclusive travel), or more generally the most attractive facility, when close enough.

What happens in case of ties with respect to the criterion used in making this choice is an additional question. Several possibilities exist: each consumer might either opt for conservative behaviour, letting the consumer stay with its former supplier \( L \), or for novelty orientation, where the consumer prefers to move to the newcomer \( F \), see [7]. Other behaviour, actually splitting demand over the tied competitors, leads to extended covering models, going by the name of capture, see [10], and are not covered here. In this paper we will start by (partially) avoiding the question of ties by working with separate potential sites for leader and follower. Discussion of how to relax this hypothesis is postponed till section 5.

Let \( T \) be the set of sites available for the (single) facility \( F \) will possibly open. As just said, we assume for the moment that \( T \) is disjoint from \( S \).

Just like for the leader’s sites, we also need to specify which sites in \( T \) would capture consumer \( i \) in absence of \( L \)’s facilities. This set is denoted by \( T_i \subset T \).

Suppose now \( F \) opens a facility at \( t \in T \), what are the sites in \( S \) at which a leader’s facility would still capture consumer \( i \)’s demand ? We denote by \( S_{it} \) this set of sites of \( S \) which \( i \) prefers to \( t \).

The rules of the competition game are completely described by the sets \( S, T, S_i, T_i \) and \( S_{it} \). These sets are, however, not fully independent. The following properties always hold:

- \( S = \bigcup_{i \in I} S_i \), since any other site \( s \) would be unable to capture any demand, so would never be used by \( L \).
- \( T = \bigcup_{i \in I} T_i \), for the same reason w.r.t. \( F \).
- \( S_{it} \subset S_i \) by definition.
- \( S_{it} = S_i \) when \( t \not\in T_i \), since when \( F \) locates (only) at \( t \) \( i \) simply cannot be captured by \( F \), in other words this has no influence at all on \( i \)’s behaviour.

However, it might happen that \( S_{it} = S_i \) for some \( t \in T_i \) (see the example below).

On the other hand, for many \( i, t \)-combinations \( S_{it} \) will be empty. This means that if \( F \) locates at \( t \), consumer \( i \) is totally lost for \( L \), and it is then useless to consider the possibility of \( L \) capturing \( i \), which is why our formulations will often involve only \( i, t \)-pairs with nonvoid \( S_{it} \).
Consider for example the simplest consumer behaviour rule of allocation to the closest. Let \( d_i \) be the distance from consumer \( i \) to the current nearest competitor already present on the market before \( L \)'s or \( F \)'s entry, and \( d_i^s (d_i^t) \) the distance between consumer \( i \) and site \( s \) (\( t \)). \( S_i \) is then the set of \( L \)'s possible sites closer to \( i \) than to any other competitor already present on the market:

\[
S_i = \{ s \in S : d_i^s \leq d_i \}
\]

Similarly,

\[
T_i = \{ t \in T : d_i^t \leq d_i \}
\]

Also \( S_{it} \) is obtained in a similar way as \( S_i \), but taking also a competitor at \( t \) into account,

\[
S_{it} = \{ s \in S_i : d_i^s < d_i^t \}
\]

Now if \( t \not\in T_i \) this means that \( d_i^t > d_i \), and \( S_{it} = S_i \). However, it is very well possible that \( \max_{s \in S} d_i^s < d_i^t < d_i \), and then we also have \( S_{it} = S_i \) although \( t \in T_i \).

Observe that in the above we have (implicitly) assumed novelty orientation: when \( d_i^s = d_i \) consumer \( i \) will opt for a new facility at \( s \), and similarly for the follower at \( t \), while when \( d_i^s = d_i^t \leq d_i \) will prefer the more recent follower’s facility at \( t \) above the leader’s one at \( s \).

2 Worst Case Model : MAXMIN

A first strategy that \( L \) can use is to protect herself against a worst case scenario. In other words, \( L \) wants to ensure that the demand that she will still serve is the best possible wherever \( F \) will locate its facility.

For each fixed \( t \in T \) we have a maximal covering problem for \( L \) described by the patronising sets \( S_{it} \), and yielding a maximal covering value. The worst case corresponds to the lowest of all these values, and it is this minimum that should be maximised.

This can be formulated as an MIP by way of the following sets of location and (now conditional) capturing variables

- \( L_s \ (s \in S) \) answering the question ‘Does \( L \) open a facility at \( s \)?’
- \( x_{it} \ (i, t \sim S_{it} \not= \emptyset) \) answering the question ‘Will consumer \( i \) still be captured by \( L \) when \( F \) locates at \( t \)?’
- \( z \), an auxiliary variable for finding the minimum of all maximal covering values

\[
\text{MAX} \quad z
\]

such that

\[
z \leq \sum_{i \sim S_{it} \not= \emptyset} d_i x_{it} \quad (t \in T)
\]

\[
x_{it} \leq \sum_{s \in S_{it}} L_s \quad (i, t \sim S_{it} \not= \emptyset)
\]

\[
\sum_{s \in S} f_s L_s \leq B
\]

variables \( L_s \in \{0, 1\} \quad (s \in S) \)

\[0 \leq x_{it} \leq 1 \quad (i, t \sim S_{it} \not= \emptyset)\]

Note here the importance of being able to relax the auxiliary capturing variables \( x_{it} \) to being continuous: the number of binary variables in the model always remains equal to \( L \)'s number of feasible sites \(|S|\), irrespective of the number of feasible sites for the follower \( F \).

As before, one might consider adding the additional constraints

\[
L_s \leq x_{it} \quad \text{for all } s \in S_{it}
\]
in order to better enforce correct behaviour of all capturing variables. In the current model this is not really needed. However, when some additional constraints would have to be added for enforcement of particular managerial rules, absence of constraints (2) might lead to optimal solutions with fractional $x_{it}$ values, while their presence guarantees binary outcomes even with continuously relaxed variables.

The fully binary formulation may be seen as an application of the maximum capture problem with uncertainty introduced in [14], here simplified because no ties are present.

### 3 Minimum Regret Model: MINREGRET

Instead of a pure worst case profit oriented objective the leader $L$ might follow the principle of minimum regret. In order to measure her a-posteriori regret, $L$ must first know what would have been her best possible choice if she had known $F$’s choice a-priori.

In other words, suppose it were known that $F$ locates at $t \in T$, then $L$’s best choice is obtained by solving the maxcovering model given the patronising sets $S_{it}$ for all $i$ for which $S_{it} \neq \emptyset$, yielding the best conditional value $Z_t^*$, i.e.

$$Z_t^* = \max \sum_{i \sim S_{it} \neq \emptyset} d_i x_{it}$$

such that

- $x_{it} \leq \sum_{s \in S_{it}} L_s$ (for $i \sim S_{it} \neq \emptyset$)
- $\sum_{s \in S} f_s L_s \leq B$
- variables $L_s \in \{0,1\}$ (for $s \in S$)
- $0 \leq x_{it} \leq 1$ (for $i \sim S_{it} \neq \emptyset$)

These values can be obtained by solving the model above for each $t \in T$ separately.

When $L$ now chooses to locate in a way given by the $L_s$ ($s \in S$), and when $F$ locates at $t \in T$, $L$’s regret is given by

$$Z_t^* - \sum_{i \sim S_{it} \neq \emptyset} d_i x_{it}$$

where

- $x_{it} \leq \sum_{s \in S_{it}} L_s$ (for $i \sim S_{it} \neq \emptyset$)
- $L_s \leq x_{it}$ (for $i, t, s \sim s \in S_{it}$)

(note that we added the last constraints in order to fully enforce the correct value of the $x_{it}$ as a function of the $L_s$).

$L$ now wants to minimise the largest among all these regrets when varying $t \in T$. This is obtained by the following model where the auxiliary variable $z$ is now used to obtain the maximum regret:

$$\min \quad z$$

where

- $Z_t^* - \sum_{i \sim S_{it} \neq \emptyset} d_i x_{it} \leq z$ (for $t \in T$)
- $x_{it} \leq \sum_{s \in S_{it}} L_s$ (for $i \sim S_{it} \neq \emptyset$)
- $L_s \leq x_{it}$ (for $i, t, s \sim s \in S_{it}$)
- $\sum_{s \in S} f_s L_s \leq B$

variables $L_s \in \{0,1\}$ (for $s \in S$)
- $0 \leq x_{it} \leq 1$ (for $i \sim S_{it} \neq \emptyset$)
4 Optimising competitor: von Stackelberg model

Previous models consider the follower’s decision as totally uncertain, so do not consider rationality in his choice. However, \( F \) will face a similar problem than \( L \), but will be able to take \( L \)'s facilities into account, since these will already be known. His best choice for a facility site will be that one which captures the largest possible demand, thereby taking over some of \( L \)'s current consumers, probably next to consumers not served by \( L \). Unless \( L \) was already serving all demand, there is no guarantee that this would lead to the worst possible outcome for \( L \). In general, the resulting model for \( L \) is quite different from both previous ones and will probably lead to a very different solution strategy. Indeed it might be more interesting for \( L \) to secure a less dense market and not to needlessly invest in a region in which investment will be lost anyway to her follower.

4.1 MIP formulation

In order to formulate the resulting two-stage von Stackelberg like model [15] as an MIP we need to introduce some different additional notions and variables. The Logical Implication Principle (LIP), developed in [8] to translate logical implications involving binary variables into inequality constraints, will be used without further explanation.

Let us start with variables answering the location and allocation questions for both players \( L \) and \( F \).

- \( L_s \ (s \in S) \) answering the question ‘Does \( L \) open a facility at \( s \) ?’
- \( x^L_i \ (i \in I) \) answering the question ‘Is consumer \( i \) captured by \( L \) ?’
- \( F_t \ (t \in T) \) answering the question ‘Will \( F \) open a facility at \( t \) ?’
- \( x^F_i \ (i \in I) \) answering the question ‘Will consumer \( i \) be captured by \( F \) ?’

We can then immediately write following constraints

\( L \) must remain within its budget \( B \)

\[
\sum_{s \in S} f_s L_s \leq B \quad (3)
\]

\( F \) will open one facility

\[
\sum_{t \in T} F_t = 1 \quad (4)
\]

Consumer \( i \) patronizes at most one facility

\[
x^L_i + x^F_i \leq 1 \quad (i \in I) \quad (5)
\]

Next we have the the capturing constraints, which relate the location variables to the allocation variables.

If \( i \) is captured by \( L \) then \( L \) must have opened somewhere in \( S_i \)

or ‘If \( L \) opens at none of the sites \( S_i \), then \( i \) cannot be captured by \( L \)

which is written as the LIP constraint:

\[
x^L_i \leq \sum_{s \in S_i} L_s \quad (i \in I) \quad (6)
\]

If \( i \) is captured by \( F \) then \( F \) must have opened somewhere in \( T_i \)

or ‘If \( F \) opens at none of the sites \( T_i \), then \( i \) cannot be captured by \( F \)

which is written as the LIP constraint:

\[
x^F_i \leq \sum_{t \in T_i} F_t \quad (i \in I) \quad (7)
\]
Patronising rules: for any demand point \( i \) there are three possibilities

- **In case** \( F \) **locates** at \( t \in T_i \) and \( L \) **has opened no site** in \( S_{it} \), then \( F \) **captures** \( i \).

  In other words: ‘If \( F_i = 1 \) and \( L_s = 0 \) for all \( s \in S_{it} \), then \( x^F_i = 1 \)’

  which is written as the LIP constraint:

  \[
  1 - x_i^F \leq 1 - F_i + \sum_{s \in S_{it}} L_s \quad (i \in I, t \in T_i) \tag{8}
  \]

  Note that this includes in particular all cases where \( t \in T_i \) and \( S_{it} = \emptyset \), and the constraint then reduces to

  \[
  F_i \leq x_i^F \quad (i \in I, t \in T_i, S_{it} = \emptyset) \]

- **But in case** \( F \) **locates** at \( t \in T_i \) and \( L \) **has opened some site** \( s \in S_{it} \), then it is \( L \) **who captures** \( i \). Note that for this constraint to exist we need \( S_{it} \neq \emptyset \). In other words:

  For any triplet \((i \in I, t \in T_i, s \in S_{it})\) we must have:

  If \( F_i = 1 \) and \( L_s = 1 \), then \( x_i^L = 1 \)

  which is written as the LIP constraint:

  \[
  1 - x_i^L \leq (1 - F_i) + (1 - L_s) \quad (i \in I, t \in T_i, s \in S_{it}) \tag{9}
  \]

  Note that we originally expected these constraints not to be needed (compare with [9]), but experimental results showed them to be mandatory to obtain a correct behaviour of the model. Also their presence will explicitly be used in the proof of the later lemma 1. This increases considerably the number of constraints in the model.

  However, these constraints may be rephrased in a much shorter way as follows, using the assumption that at most one facility will be located by the follower, expressed by constraint (4). This implies that the expression \( \sum_{t \in T'} F_i \) is an implicit binary variable for any \( T' \subset T \) (see [8]). Consider then any pair \((i, s)\) such that \( s \in S_i \), and for which there exist follower sites beaten by \( s \) for consumer \( i \), then we may state that \( L \) will capture \( i \) as soon as \( L \) locates at \( s \), but \( F \) locates at none of these sites beaten by \( s \) for consumer \( i \). In other words:

  As soon as \( \exists t \in T_i, s \in S_{it} \), if \( \sum_{t \in T_i, s \in S_{it}} F_i = 1 \) and \( L_s = 1 \), then \( x_i^L = 1 \)

  which is written as the LIP constraints:

  \[
  1 - x_i^L \leq (1 - \sum_{t \in T_i, s \in S_{it}} F_i) + (1 - L_s) \quad (i \in I, s \in S_i, \exists t \in T_i, s \in S_{it}) \tag{10}
  \]

  It is clear that there are not only less of these constraints than in (9), but at the same time they are also tighter, so this last formulation is to be preferred.

- **When** \( F \) **locates outside** \( T_i \), **consumer** \( i \) **cannot be captured by** \( F \). Capture by \( L \) is then conditioned by \( L \)'s presence or not within \( S_i \), which is already expressed by constraint (6) together with the maximisation of the objective discussed below. Therefore we do not need another set of constraints for these cases.

  But this property also allows us to further sharpen constraints (10): when \( F \) locates somewhere outside \( T_i \), but \( L \) locates at \( s \in S_i \), then we also know for sure that \( i \) will be captured by \( L \). Taking this together with previous information and the fact that \( F \) locates only a single facility, we may replace the constraint (10) by

  \[
  1 - x_i^L \leq (1 - \sum_{t \in T_i, s \in S_{it}} F_i - \sum_{t \notin T_i} F_i) + (1 - L_s) \quad (i \in I, s \in S_i, \exists t \in T_i, s \in S_{it}) \tag{11}
  \]
What remains is to describe the objective. The easy part of this task is the total demand captured by \( L \), which is the actual objective function to be maximised in our model.

\[
\text{MAX} \quad \sum_{i \in I} d_ix_i^L
\]  

(12)

Similarly we may write the demand captured by \( F \) as \( \sum_{i \in I} d_ix_i^F \), and we must ensure, according to the von Stackelberg principle, that this is the largest demand \( F \) is able to capture over all possible choices of \( t \in T \). Therefore we must be able to express separately the total demand that \( F \) would capture when locating at some site \( t \in T \).

To this end we use following auxiliary conditional capturing variables:

- \( y_{it} \) answering the question ‘Assuming that \( F \) locates at \( t \in T \), would \( i \) then be captured by \( F \)?’

We may consider several cases.

- When \( t \not\in T_i \) the answer to this question is certainly ‘No’, and the variable \( y_{it} \) is useless, since its value is fixed at 0.

- When, contrarily, \( t \in T_i \) but \( S_{it} = \emptyset \), the answer to this question is certainly ‘Yes’, and the variable \( y_{it} \) is again useless, since its value is fixed at 1. We must, however, not forget \( i \)’s demand as being captured by \( F \) in this case.

- Finally for any pair \((i, t)\) with \( t \in T_i \) and \( S_{it} \neq \emptyset \), we must have \( y_{it} = 1 \) if and only if \( L_s = 0 \) for all \( s \in S_{it} \), which is written as the set of LIP constraints:

\[
1 - y_{it} \leq \sum_{s \in S_{it}} L_s
\]  

(13)

\[
L_s \leq 1 - y_{it} \quad s \in S_{it}
\]  

(14)

Using the variables \( y_{it} \) we may now obtain the demand captured by \( F \) in case it would locate at \( t \) as

\[
\sum_{i \sim t \in T_i, S_{it} \neq \emptyset} d_iy_{it} + \sum_{i \sim t \in T_i, S_{it} = \emptyset} d_i
\]

Therefore the Von Stackelberg principle, stating that \( F \)’s actually captured demand is the largest possible given \( L \)’s location, may be expressed by the set of constraints

\[
\sum_{i \sim t \in T_i, S_{it} \neq \emptyset} d_iy_{it} + \sum_{i \sim t \in T_i, S_{it} = \emptyset} d_i \leq \sum_{i \in I} d_ix_i^F \quad (t \in T)
\]  

(15)

It follows that \( y_{it} \)-variables are only needed for all \( i \) such that \( t \in T_i, S_{it} \neq \emptyset \), because in all other cases their value may be fixed in advance. Thus we obtain the following full formulation of the von Stackelberg model

\[
\text{MAX} \quad \sum_{i \in I} d_ix_i^L
\]  

(16)

such that

\[
\sum_{s \in S} f_sL_s \leq B
\]  

(17)

\[
\sum_{t \in T} F_t = 1
\]  

(18)

\[
x_i^L + x_i^F \leq 1 \quad (i \in I)
\]  

(19)
\[ x_t^L \leq \sum_{s \in S_t} L_s \quad (i \in I) \]  
\[ x_t^F \leq \sum_{i \in T_i} F_i \quad (i \in I) \]  
\[ 1 - x_t^F \leq 1 - F_i + \sum_{s \in S_{it}} L_s \quad (i, t \in T_i) \]  

\[ 1 - x_t^L \leq (1 - \sum_{i \in T_i, s \in S_{it}} F_i - \sum_{i \in T_i} F_i) + (1 - L_s) \quad \left( i, s \in S_t, \exists t \in T_i \sim s \in S_{it} \right) \]  
\[ 1 - y_{it} \leq \sum_{s \in S_{it}} L_s \quad (i, t \in T_i, s \in S_{it}) \]  
\[ \sum_{i \sim t \in T_i, S_{it} \neq \emptyset} d_i y_{it} + \sum_{i \sim t \in T_i, S_{it} = \emptyset} d_i \leq \sum_{i \in I} d_i x_t^F \quad (t \in T) \]

variables  
- \( L_s \in \{0, 1\} \quad (s \in S) \)
- \( F_i \in \{0, 1\} \quad (t \in T) \)
- \( x_t^L \in \{0, 1\} \quad (i \in I) \)
- \( x_t^F \in \{0, 1\} \quad (i \in I) \)
- \( y_{it} \in \{0, 1\} \quad (i, t \in T_i, S_{it} \neq \emptyset) \)

4.2 Equivalent relaxation

The following lemma shows that, just like for the maximal covering, the maximin and the minregret models, in the von Stackelberg model it is sufficient to consider only the leaders’ and followers’ location variables as binary.

**Lemma 1** If in the von Stackelberg competitive location model (16-31) the binary contraints (30) and (31) are relaxed to

\[ x_t^F \geq 0 \quad (i \in I) \]  
\[ y_{it} \geq 0 \quad (i, t \in T_i \sim S_{it} \neq \emptyset) \]

the set of feasible solutions remains unchanged.

If also the constraints (29) are relaxed to

\[ x_t^L \geq 0 \quad (i \in I) \]

the set of optimal solutions remains unchanged.

**Proof**

Using only the constraints of the relaxed von Stackelberg model (16-28,32-34) we will show, first for the \( y_{it} \) variables, and then also for the \( x_t^F \)-variables, that in any feasible solution with strictly positive value for the variable, the value must be actually 1. Then we will prove the same for the \( x_t^L \)-variables in any optimal solution, which will terminate the proof.

Consider thus any feasible solution in which for some \( i \in I, t \in T_i \sim S_{it} \neq \emptyset \) we have \( y_{it} > 0 \). Then we may choose an \( s \in S_{it} \) for which by (25) it then follows that \( L_s < 1 \), hence, by (27) that \( L_s = 0 \), and therefore by (24) that \( 1 - y_{it} \leq 0 \), so \( y_{it} \geq 1 \), while by (25) we have \( y_{it} \leq 1 \), so \( y_{it} = 1 \).

Next note that the nonnegativity constraints (32) and (34) together with (19) imply that all \( x_t^F \leq 1 \) and \( x_t^L \leq 1 \).

Consider now any feasible solution with \( x_t^F > 0 \) for some \( j \in I \).

Constraint (21) then implies that \( \sum_{i \in T_j} F_i > 0 \), which by (28) means that \( F_i = 1 \) for some \( t \in T_j \). In the sequel we consider this particular \( t \) only.
If now for some \( i \in I \) we have \( L_s = 1 \) for some \( s \in S_{it} \), then constraint (23) yields \( 1 - x_i^F \leq 0 \), which, together with \( x_i^F \leq 1 \), yields \( x_i^F = 1 \), and finally by (19) \( x_i^F = 0 \). Thus \( j \) can not be in this case. Denoting then by \( I' \) the set of all other indices not in this case, i.e. \( I' = \{ i \in I : L_s = 0 \text{ for all } s \in S_{it} \} \) (which includes \( j \) and also those \( i \) for which \( S_{it} = \emptyset \) we obtain that in constraint (26) the right-hand-side is reduced to \( \sum_{i \in I'} d_i x_i^F \).

Let us now look at the left-hand-side of this constraint. Constraints (24) and (25) were constructed to exactly mean that \( y_{it} = 1 \) if and only if \( L_s = 0 \) for all \( s \in S_{it} \) in case \( S_{it} \neq \emptyset \). Therefore the left-hand-side of constraint (26) reduces to \( \sum_{i \in T, t \in S_{it} \neq \emptyset} d_i + \sum_{i \in T, t \in S_{it} = \emptyset} d_i = \sum_{i \in I'} d_i \). This constraint therefore yields

\[
\sum_{i \in I'} d_i \leq \sum_{i \in I'} d_i x_i^F
\]

which, since all \( d_i > 0 \) and \( 0 \leq x_i^F \leq 1 \), implies that \( x_i^F = 1 \) for all \( i \in I' \), in particular \( x_j^F = 1 \).

Finally consider any optimal solution with \( x_j^F > 0 \) for some \( j \in I \). The only constraints in which \( x_j^F \) appears are (19), (20) and (23) in which all other variables either take value 0 or 1. Therefore setting the value of \( x_j^F \) to 1 will always leave the solution feasible, and in case its original value was less than 1, would strictly increase the objective value (\( d_i > 0! \)), which cannot be for an optimal solution. Therefore \( x_j^F = 1 \).

### 4.3 A slight further reduction

The largest set of variables in the Von Stackelberg model is the set \( y_{it} \) for all \( (i \in I, t \in T, t \sim S_{it} \neq \emptyset) \). This may be further reduced by the following observation:

When \( F \) locates at \( t \in T_i \), and \( |S_{it}| = 1 \), in other words \( S_{it} = \{ s \} \) for some \( s \in S \), then either \( L \) locates at \( s \) and \( i \) will be captured by \( L \), or not, and \( i \) will be captured by \( F \). In other words, in this particular case the variables \( y_{it} \) and \( L_s \) are exactly complementary:

\[
y_{it} = 1 - L_s
\]

which allows to dispense of the use of such \( y_{it} \), and simply replace them everywhere needed by \( 1 - L_s \).

### 5 Forbidding proximity

In practice it is often impossible for several facilities to share a same site or to be located too close to each other. This may be for simple physical reasons of space availability, or for ‘social’ reasons which (often legally) forbid such co-location. E.g. in several countries it is forbidden to locate a new pharmacy too close to an already existing one.

In the previous sections we have made the much stronger assumption that no sites are available to both leader and follower. But normally such common sites do occur, and it is only forbidden for both players to choose a same site. In that case the leader has a clear advantage in having access to all originally available sites, while the follower will have to avoid the leader’s chosen sites or other sites in their close vicinity.

As mentioned before, there is also a second, more technical reason for avoiding proximity in our models. As long as \( S \cap T = \emptyset \) one may usually assume that the sites for leader \( L \) and follower \( F \) are clearly distinguished for all consumers, so that it is possible to predict which way their preference will go in any case, leading to a clear definition of the patronising sets \( S_{it} \), central to all formulations. However, this assumption cannot reasonably be defended as soon as \( L \) and \( F \) share some potential sites: two facilities, one of each player, located at the same site, will necessarily become indistinguishable for the consumers (if not differentiated by some other characteristics as assumed here). The effect is particularly impressive when combined with novelty orientation: by locating right ‘on top’ of a leader’s facility the follower would completely take over all consumers of this latter!
More generally the same difficulty arises, as soon as both players can locate so close to each other that consumers would be indifferent between two facilities located at each of such sites. Therefore it is often necessary to take such indifference explicitly into account. In fact, the consumer indifference induced by facility proximity is most often the reason why proximity forbidding rules have been enforced.

In this section we indicate how such proximity forbidding rules may be included in the three location models discussed before.

5.1 Co-location exclusion

Co-location exclusion in the strict sense means that we accept common leader and follower sites \((S \cap T \neq \emptyset)\), but forbid the follower to locate at such a site in case the leader is already present there.

For logical coherence we must make sure that any facility at such a leader/follower common site can be patronised by the same consumers, irrespective of its owner. In other words, we must have always \(S_i \cap T = S \cap T_i = S_i \cap T_i\).

5.1.1 Maxmin and minregret models

In the worst case and minimum regret models of sections 2 and 3 this rule may simply be enforced by a proper choice of the patronising sets \(S_{it}\).

To this end it is sufficient to ensure that for any \(i\) and any common site \(u \in S_i \cap T_i\) we include \(u \in S_{iu}\).

Indeed, consider a common site \(u \in S_i \cap T_i\). By definition, as implemented in the model formulation, in case the follower locates at \(u\) and the leader locates anywhere in \(S_{iu}\), consumer \(i\) will patronise the leader’s facility. Therefore \(u \in S_{iu}\) for all \(i \in I, u \in S_i\) implies that in case both leader and follower would locate at \(u\), all these consumers \(i\) will be allocated to the leader. In other words this location of the follower would imply no loss for the leader.

Except in degenerate cases of no interest, any other location for the follower, distinct from any leader location, will lead to some losses to the leader, and hence also to some nonzero regret. This follower’s choice can therefore not be the worst case for the leader, neither in terms of losses, nor in terms of regret. This means that for both the maxmin model and the minregret model, the objective value cannot be generated by a solution involving co-location. In other words, the optimal solution will also not involve co-location.

5.1.2 Von Stackelberg model

It is not clear at all if the reasoning above fully applies to the von Stackelberg model. However this model uses location variables for the follower, which were not present in previous models, but very helpful in formulating our new requirements explicitly as additional constraints.

For every site \(s \in S \cap T\) we have to make sure that ‘If \(L\) locates at \(s\), \(F\) cannot locate there’, or, ‘If \(L_s = 1\) then \(F_s = 0\)’, which, by LIP, leads to

\[
F_s \leq 1 - L_s \quad (s \in S \cap T) \tag{35}
\]

In order to obtain a correct behaviour of the additional variables \(y_{it}\) we should take care to include the preference rule for common sites in the patronising sets as before, by enforcing \(u \in S_{iu}\) for any common site \(u \in S_i \cap T_i\).

5.2 Proximity forbidding rules

Suppose more generally that when the leader locates at \(s \in S\) there is a subset \(T_s \subset T\) of sites forbidden for the follower. In particular, forbidding co-location at \(s\) is ensured by \(s \in T_s\). Strict co-location exclusion, as discussed above then corresponds to \(T_s = \{ s \}\) for all \(s \in S \cap T\).
One must ensure that when $L$ locates at $s$, and $F$ would locate in some $t \in T_s$, $L$ will go on capturing all demand she had captured before. This is obtained by including $s$ in $S_i$ for all $i \in I$ for which $s \in S_i$ and all $t \in T_s \cap T_i$.

By the same reasoning as in section 5.1.1 this property will be sufficient for the minmax and minregret models.

Just like in section 5.1.2, in the von Stackelberg model one needs additional constraints to enforce the new rules. For each $s \in S$ and $t \in T_s$ we must have ‘If $L$ locates at $s$, $F$ cannot locate in $t$ ’, or, ‘If $L_s = 1$ then $F_t = 0$’, which is formulated by the LIP constraints

$$F_t \leq 1 - L_s \quad (s \in S, t \in T_s)$$ (36)

6 Computational experiments

The computational experiments presented in this section were set up in order to test the proposed models, and to study feasibility of their solution using standard software. To analyze the behavior of the different competitive models, we have performed two series of tests. First we have used artificially generated data on a grid, to be able to better control their size and judge the ensuing model complexity. In these test problems all leader and follower possible sites were distinct.

We formulated the models in AMPL (A Modeling Language for Mathematical Programming, Version 20021031) and solved them with ILOG CPLEX 9.0, using the default settings, on a Pentium 4, 1.8 MHz, with 256MB RAM, running Windows 2K. The slight reduction of the number of variables and constraints discussed in section 4.3 was not implemented, but we observed that AMPL’s presolve feature was able to automatically identify and carry out these substitutions.

Below we shortly discuss the setup of the 3 models, the data and the obtained solutions.

6.1 Basic assumptions in all models

The first step in concretization is the modelling of the patronage decision. We use the simple rule discussed in the introductory section 1: a consumer only takes into consideration facilities lying within a maximum travel distance from its location. The area consisting of all reachable facility locations is the ‘travel area’ of the consumer (analogy: service area of a company). Notice that for large maximum travel distances the travel areas will tend to the full market, ultimately leading to a zero-sum game situation in which all three models are equivalent. For this reason we used relatively small travel distances. Maximum travel distances are equal for all consumers, and we assume that the leader and follower have the same characteristics, thus we use an equal travel distance for each.

The patronised facility is taken as the nearest one, as proposed at the end of section 1.3. In case of a tie in distances, we adopt novelty orientation, described in section 1.3.

The distance measure used in the examples is the Euclidean distance throughout. Rectangular distance has also been tried, but although the shape of the consumer travel area changes, very similar results are obtained and are therefore not reported here. What may be said is that for equal threshold distance the travel areas are of course smaller, resulting in slightly smaller problem dimensions, and as expected somewhat improved solution times were observed.

6.2 Set up

The data configuration is artificially generated. We use a – relatively small– square grid to represent both the consumers’ locations and the possible leader and follower sites. All the grid cells stand for consumers (set $I$).

Each cell is also either a possible site for the leader (set $S$) or for the follower (set $T$), but never for both. Thus, by construction no common leader and follower locations are possible, which is an implicit way of forbidding co-location. The set $T$ consisted of those cells with sum of the cell coordinates (from $(1,1)$ to $(n,n)$) a multiple of 3, and $S$ consists of all other cells, so approximately $|S| \approx 2|T|$.
The consumer demands and the leader fixed costs were randomly generated, uniformly distributed integer in the respective ranges \([50,250]\) and \([5,10]\). The budget was chosen as 15.

### 6.3 Comparing the models

We started by generating data-instances on 6 different sized grids, from 5x5 up to 17x17, with maximum travel distances 2, 3 or 4. For each data-instance all three models were solved, and the total CPLEX-solve time was recorded (for the minregret model this cumulates solve-times for all subproblems). Table 1 allows to compare the performance of the solution procedure for the three models.

<table>
<thead>
<tr>
<th>Travel Distance</th>
<th>Grid Dimension</th>
<th>maxmin</th>
<th>minregret</th>
<th>Vonstackelberg</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5x5</td>
<td>0.310</td>
<td>0.630</td>
<td>0.260</td>
</tr>
<tr>
<td></td>
<td>7x7</td>
<td>0.200</td>
<td>1.782</td>
<td>0.721</td>
</tr>
<tr>
<td></td>
<td>10x10</td>
<td>3.665</td>
<td>10.845</td>
<td>0.470</td>
</tr>
<tr>
<td></td>
<td>12x12</td>
<td>10.525</td>
<td>50.002</td>
<td>0.981</td>
</tr>
<tr>
<td></td>
<td>15x15</td>
<td>22.635</td>
<td>1360.240</td>
<td>3.185</td>
</tr>
<tr>
<td></td>
<td>17x17</td>
<td>144.991</td>
<td>10258.510</td>
<td>3.395</td>
</tr>
<tr>
<td>3</td>
<td>maxmin</td>
<td>0.100</td>
<td>3.565</td>
<td>0.120</td>
</tr>
<tr>
<td></td>
<td>minregret</td>
<td>0.290</td>
<td>55.490</td>
<td>0.550</td>
</tr>
<tr>
<td></td>
<td>Vonstackelberg</td>
<td>12.318</td>
<td>33.468</td>
<td>6.579</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.282</td>
<td>117.611</td>
<td>14.290</td>
</tr>
<tr>
<td></td>
<td></td>
<td>28.191</td>
<td>115.919</td>
<td>149.277</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21.72</td>
<td>281.8</td>
<td>50.002</td>
</tr>
<tr>
<td>4</td>
<td>maxmin</td>
<td>0.111</td>
<td>0.580</td>
<td>0.540</td>
</tr>
<tr>
<td></td>
<td>minregret</td>
<td>0.290</td>
<td>103.328</td>
<td>6.529</td>
</tr>
<tr>
<td></td>
<td>Vonstackelberg</td>
<td>12.318</td>
<td>84.994</td>
<td>14.290</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.47</td>
<td>199.301</td>
<td>33.468</td>
</tr>
<tr>
<td></td>
<td></td>
<td>115.919</td>
<td>380.574</td>
<td>50.002</td>
</tr>
</tbody>
</table>

Table 1: Overview of the solution times for the three models

Clearly solution times increase generally with gridsize and travel distance. This corresponds to the increase in size of the formulations in terms of variables and constraints. Unexpectedly the von Stackelberg model performed very well, comparable with the maxmin model. The minregret model took much more time. A closer look at the result details indicated that this is not only due to the many maxcovering subproblems to be solved (each of these turned out to need almost negligible

<table>
<thead>
<tr>
<th>Travel Distance</th>
<th>Grid Dimension</th>
<th>b</th>
<th>c</th>
<th>C</th>
<th>√2</th>
<th>b</th>
<th>c</th>
<th>C</th>
<th>√3</th>
<th>b</th>
<th>c</th>
<th>C</th>
<th>√10</th>
<th>b</th>
<th>c</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5x5</td>
<td>25</td>
<td>78</td>
<td>195</td>
<td>25</td>
<td>100</td>
<td>357</td>
<td>25</td>
<td>120</td>
<td>527</td>
<td>152</td>
<td>857</td>
<td>25</td>
<td>164</td>
<td>1583</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7x7</td>
<td>49</td>
<td>153</td>
<td>379</td>
<td>49</td>
<td>202</td>
<td>742</td>
<td>49</td>
<td>248</td>
<td>1143</td>
<td>328</td>
<td>2037</td>
<td>49</td>
<td>361</td>
<td>4350</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10x10</td>
<td>100</td>
<td>319</td>
<td>792</td>
<td>428</td>
<td>1605</td>
<td>3774</td>
<td>100</td>
<td>534</td>
<td>2556</td>
<td>726</td>
<td>4772</td>
<td>100</td>
<td>811</td>
<td>527</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12x12</td>
<td>144</td>
<td>464</td>
<td>1154</td>
<td>625</td>
<td>2361</td>
<td>991</td>
<td>144</td>
<td>785</td>
<td>3811</td>
<td>1079</td>
<td>2264</td>
<td>144</td>
<td>1212</td>
<td>3803</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15x15</td>
<td>225</td>
<td>730</td>
<td>1817</td>
<td>924</td>
<td>2347</td>
<td>1284</td>
<td>225</td>
<td>1251</td>
<td>6154</td>
<td>1737</td>
<td>2364</td>
<td>225</td>
<td>1624</td>
<td>11333</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17x17</td>
<td>289</td>
<td>942</td>
<td>2347</td>
<td>1308</td>
<td>3260</td>
<td>1790</td>
<td>289</td>
<td>1624</td>
<td>8039</td>
<td>3182</td>
<td>3878</td>
<td>289</td>
<td>2758</td>
<td>13788</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20x20</td>
<td>400</td>
<td>3260</td>
<td>3260</td>
<td>15617</td>
<td>6872</td>
<td>2172</td>
<td>400</td>
<td>2758</td>
<td>11989</td>
<td>5878</td>
<td>5056</td>
<td>400</td>
<td>2758</td>
<td>13788</td>
<td></td>
</tr>
<tr>
<td></td>
<td>22x22</td>
<td>484</td>
<td>6872</td>
<td>6872</td>
<td>22172</td>
<td>8357</td>
<td>2818</td>
<td>484</td>
<td>2758</td>
<td>11989</td>
<td>5878</td>
<td>5056</td>
<td>484</td>
<td>2758</td>
<td>13788</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25x25</td>
<td>625</td>
<td>8357</td>
<td>8357</td>
<td>27172</td>
<td>10870</td>
<td>2818</td>
<td>625</td>
<td>2758</td>
<td>11989</td>
<td>5878</td>
<td>5056</td>
<td>625</td>
<td>2758</td>
<td>13788</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Dense Distribution for Leader Positions: variables and constraints
time) but rather to the difficulty of solving the final master problem (in two cases CPLEX was even unable to solve this final master problem by lack of available virtual memory). This may be explained by the fact that the regret values are much smaller than the demand values which appear in the maxmin model, yielding a more uniform objective function, and therefore much harder to solve.

### 6.4 von Stackelberg model

In view of the unexpected good behaviour of the seemingly much more complex von Stackelberg model, we decided to test it more thoroughly.

Three additional larger grid sizes and several more travel distances were considered, as well as different leader/follower site densities. For these experiments two sets of demand data on a 25 x 25 grid were generated randomly. Nine lower-left subgrids of smaller sizes (between 5x5 and 25x25) of these basic grids were tested, in order to see the effect of an increase in the market area. In the first dataset we kept the 2:1 leader vs follower sites fraction (‘dense leader sites’) as before, but rather to the difficulty of solving the final master problem (in two cases CPLEX was even unable to solve this final master problem by lack of available virtual memory). This may be explained by the fact that the regret values are much smaller than the demand values which appear in the maxmin model, yielding a more uniform objective function, and therefore much harder to solve.

<table>
<thead>
<tr>
<th>Grid Dimension</th>
<th>5x5</th>
<th>7x7</th>
<th>10x10</th>
<th>12x12</th>
<th>15x15</th>
<th>17x17</th>
<th>20x20</th>
<th>22x22</th>
<th>25x25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>49</td>
<td>100</td>
<td>144</td>
<td>225</td>
<td>289</td>
<td>400</td>
<td>484</td>
<td>625</td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>153</td>
<td>319</td>
<td>464</td>
<td>730</td>
<td>942</td>
<td>1308</td>
<td>1583</td>
<td>2049</td>
</tr>
<tr>
<td></td>
<td>226</td>
<td>453</td>
<td>947</td>
<td>1378</td>
<td>2172</td>
<td>2802</td>
<td>3896</td>
<td>4723</td>
<td>6117</td>
</tr>
<tr>
<td>\sqrt{2}</td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>49</td>
<td>100</td>
<td>144</td>
<td>225</td>
<td>189</td>
<td>400</td>
<td>484</td>
<td>625</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>344</td>
<td>750</td>
<td>1107</td>
<td>1773</td>
<td>2304</td>
<td>3230</td>
<td>3934</td>
<td>5120</td>
</tr>
<tr>
<td></td>
<td>604</td>
<td>1335</td>
<td>2975</td>
<td>4423</td>
<td>7143</td>
<td>9324</td>
<td>13124</td>
<td>16015</td>
<td>20895</td>
</tr>
<tr>
<td>\sqrt{5}</td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>49</td>
<td>100</td>
<td>144</td>
<td>225</td>
<td>189</td>
<td>400</td>
<td>484</td>
<td>625</td>
</tr>
<tr>
<td></td>
<td>224</td>
<td>504</td>
<td>1134</td>
<td>1693</td>
<td>2743</td>
<td>3584</td>
<td>5054</td>
<td>6174</td>
<td>8064</td>
</tr>
<tr>
<td></td>
<td>978</td>
<td>2315</td>
<td>5411</td>
<td>8192</td>
<td>13448</td>
<td>17682</td>
<td>25098</td>
<td>30755</td>
<td>40331</td>
</tr>
<tr>
<td>\sqrt{10}</td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>49</td>
<td>100</td>
<td>144</td>
<td>225</td>
<td>189</td>
<td>400</td>
<td>484</td>
<td>625</td>
</tr>
<tr>
<td></td>
<td>312</td>
<td>775</td>
<td>1829</td>
<td>2776</td>
<td>4570</td>
<td>6008</td>
<td>8542</td>
<td>10485</td>
<td>13759</td>
</tr>
<tr>
<td></td>
<td>1684</td>
<td>4765</td>
<td>12313</td>
<td>19286</td>
<td>32674</td>
<td>43550</td>
<td>62778</td>
<td>77545</td>
<td>102613</td>
</tr>
<tr>
<td>\sqrt{15}</td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
<td>b</td>
<td>c</td>
<td>C</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>49</td>
<td>100</td>
<td>144</td>
<td>225</td>
<td>189</td>
<td>400</td>
<td>484</td>
<td>625</td>
</tr>
<tr>
<td></td>
<td>344</td>
<td>883</td>
<td>2129</td>
<td>3256</td>
<td>5402</td>
<td>7128</td>
<td>10174</td>
<td>12513</td>
<td>16459</td>
</tr>
<tr>
<td></td>
<td>1968</td>
<td>5979</td>
<td>16131</td>
<td>25628</td>
<td>44020</td>
<td>59034</td>
<td>85666</td>
<td>106179</td>
<td>141051</td>
</tr>
<tr>
<td>6</td>
<td>25</td>
<td>49</td>
<td>100</td>
<td>144</td>
<td>225</td>
<td>189</td>
<td>400</td>
<td>484</td>
<td>625</td>
</tr>
<tr>
<td></td>
<td>356</td>
<td>939</td>
<td>2289</td>
<td>3512</td>
<td>5842</td>
<td>7716</td>
<td>11026</td>
<td>13569</td>
<td>17859</td>
</tr>
<tr>
<td></td>
<td>2068</td>
<td>6597</td>
<td>18293</td>
<td>29286</td>
<td>50624</td>
<td>68062</td>
<td>99038</td>
<td>122917</td>
<td>163533</td>
</tr>
</tbody>
</table>

Table 3: Dense Distribution for Follower Positions: variables and constraints
the opposite case. This phenomenon is due to the combination of two effects. The variable $y_{\text{ht}}$ is indexed over the set representing all the pairs of consumers and follower's sites on the condition that the considered follower site lies within the travel area of the consumer and there must be at least one leader site that is preferred to it. Because there are more follower sites in the second case than in the first one, the number of $y_{\text{ht}}$'s is larger. But this effect alone cannot explain the less than proportional increment in $y$-variables. Due to the greater density of the follower sites in the second case, there will be less situations where a consumer is located closer to a leader site than to a follower site. Thus the set over which $y_{\text{ht}}$ is indexed, is smaller. This counteracts the augmenting effect of the first increase.

This twofold effect, among others, also explains the evolution in the number (C) of constraints.

Some statistics on solution results are shown in tables 4 and 5. Here (s) stands for solution time in CPU-seconds, (n) gives the number of branch and bound nodes generated and (F) indicates the optimal objective value.

<table>
<thead>
<tr>
<th>Grid Dimension</th>
<th>5x5</th>
<th>7x7</th>
<th>10x10</th>
<th>12x12</th>
<th>15x15</th>
<th>17x17</th>
<th>20x20</th>
<th>22x22</th>
<th>25x25</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 s</td>
<td>0.17</td>
<td>0.16</td>
<td>0.07</td>
<td>0.27</td>
<td>0.18</td>
<td>0.27</td>
<td>0.60</td>
<td>0.97</td>
<td>1.60</td>
</tr>
<tr>
<td>n</td>
<td>25</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>190</td>
<td>2196</td>
<td>2582</td>
<td>2813</td>
<td>2928</td>
<td>2995</td>
<td>3022</td>
<td>3052</td>
<td>3092</td>
</tr>
<tr>
<td>$\sqrt{3}$ s</td>
<td>0.17</td>
<td>0.5</td>
<td>0.771</td>
<td>0.55</td>
<td>0.48</td>
<td>0.961</td>
<td>1.931</td>
<td>1.161</td>
<td>3.395</td>
</tr>
<tr>
<td>n</td>
<td>16</td>
<td>31</td>
<td>15</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>2405</td>
<td>3156</td>
<td>4281</td>
<td>4640</td>
<td>5010</td>
<td>5016</td>
<td>5291</td>
<td>5291</td>
<td>5291</td>
</tr>
<tr>
<td>$\sqrt{5}$ s</td>
<td>0.37</td>
<td>1.091</td>
<td>1.922</td>
<td>0.871</td>
<td>1.993</td>
<td>2.043</td>
<td>2.704</td>
<td>1.712</td>
<td>5.539</td>
</tr>
<tr>
<td>n</td>
<td>50</td>
<td>72</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>2584</td>
<td>3749</td>
<td>5792</td>
<td>6663</td>
<td>6870</td>
<td>6965</td>
<td>7300</td>
<td>7300</td>
<td>7300</td>
</tr>
<tr>
<td>$\sqrt{7}$ s</td>
<td>0.53</td>
<td>2.904</td>
<td>27.329</td>
<td>5.327</td>
<td>11.356</td>
<td>2.453</td>
<td>14.351</td>
<td>44.535</td>
<td>28.682</td>
</tr>
<tr>
<td>n</td>
<td>48</td>
<td>40</td>
<td>53</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>2816</td>
<td>4605</td>
<td>8031</td>
<td>9852</td>
<td>9972</td>
<td>10759</td>
<td>11051</td>
<td>11051</td>
<td>11051</td>
</tr>
<tr>
<td>$\sqrt{10}$ s</td>
<td>0.53</td>
<td>5.315</td>
<td>57.479</td>
<td>18.574</td>
<td>2.754</td>
<td>9.774</td>
<td>25.567</td>
<td>44.589</td>
<td>34.841</td>
</tr>
<tr>
<td>n</td>
<td>50</td>
<td>37</td>
<td>272</td>
<td>2</td>
<td>0</td>
<td>12</td>
<td>2</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>F</td>
<td>2816</td>
<td>4879</td>
<td>9225</td>
<td>12205</td>
<td>13560</td>
<td>14213</td>
<td>14533</td>
<td>14632</td>
<td>14632</td>
</tr>
<tr>
<td>$\sqrt{15}$</td>
<td>s</td>
<td>0.821</td>
<td>10.204</td>
<td>56.883</td>
<td>201.253</td>
<td>14.861</td>
<td>6.69</td>
<td>38.887</td>
<td>89.921</td>
</tr>
<tr>
<td>n</td>
<td>40</td>
<td>138</td>
<td>108</td>
<td>170</td>
<td>2</td>
<td>0</td>
<td>12</td>
<td>13</td>
<td>66</td>
</tr>
<tr>
<td>F</td>
<td>2816</td>
<td>4839</td>
<td>10360</td>
<td>13286</td>
<td>16214</td>
<td>17513</td>
<td>18232</td>
<td>18246</td>
<td>18720</td>
</tr>
<tr>
<td>$\sqrt{20}$ s</td>
<td>0.881</td>
<td>12.248</td>
<td>84.296</td>
<td>349.245</td>
<td>77.462</td>
<td>11.381</td>
<td>19.048</td>
<td>31.587</td>
<td>69.198</td>
</tr>
<tr>
<td>n</td>
<td>46</td>
<td>111</td>
<td>51</td>
<td>431</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>2816</td>
<td>5013</td>
<td>11138</td>
<td>14644</td>
<td>18873</td>
<td>20660</td>
<td>23025</td>
<td>23025</td>
<td>23025</td>
</tr>
<tr>
<td>$\sqrt{25}$ s</td>
<td>1.331</td>
<td>22.573</td>
<td>183.748</td>
<td>383.739</td>
<td>270.394</td>
<td>68.499</td>
<td>42.361</td>
<td>88.479</td>
<td>88.139</td>
</tr>
<tr>
<td>n</td>
<td>64</td>
<td>292</td>
<td>289</td>
<td>339</td>
<td>71</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>2816</td>
<td>5013</td>
<td>11138</td>
<td>14861</td>
<td>19444</td>
<td>21714</td>
<td>23293</td>
<td>23376</td>
<td>23405</td>
</tr>
</tbody>
</table>

Table 4: Dense Distribution for Leader Positions: solution procedure and value

Generally speaking solution times are quite reasonable. They more or less increase with grid dimension and the travel distance, as is to be expected from the larger model sizes. However the main factor in solution time is clearly the number of nodes in the search tree. Because the von Stackelberg model is quite highly constrained, it is not amazing that the first LP often directly yields an integer optimal solution, as indicated by a 0 entry for n. This happens quite frequently for the larger size problems, but is not a general rule: branching is needed in several cases, but the search tree remains moderate.

In almost all cases, an increase of the maximum travel distances leads to an increase in optimal objective value. In both tables there is one exception showing this property does not hold in general. Observe also that for the smallest grid sizes the objective remains constant from a certain travel distance on; this happens as soon as a travel distance is reached which leads to a full market coverage by both leader and follower. Increasing the travel distance has then no effect anymore on the respective market shares.

Similarly one observes that for a fixed travel distance the optimal solution does not change
beyond a certain grid size. This was verified to be due to a higher concentration of demand in a central region of the basic grid, pulling the optimal solution within this range. The additional demand considered in the larger grid sizes did not influence the optimal solution at the lower travel distances, but started to modify the picture for higher travel distances.

Comparing the two tables we also observe that it takes longer to solve the first set with dense demand considered in the larger grid sizes did not influence the optimal solution at the lower travel distances, but started to modify the picture for higher travel distances.

The apparent simplicity of problems with large sized grids is explained by the fact that leader and follower were able to choose not-interfering locations.

### 7 Concluding remarks

We have shown how to incorporate the knowledge of a future entry of a competitor into maxcovering type models for competitive location. We developed full MIP formulations for three classical strategies for handling such foresight, using worst case analysis, minimising regret, and assuming rational optimising behaviour by the follower. From the computational point of view our tests have shown that exact solution of these models is feasible for relatively small scale situations. These tests also indicate that for larger sized instances only the first approach seems to remain feasible, while the two other approaches appear to become much harder due to respectively prohibitive solution times for the regret approach and prohibitive model size for the last model. Other strategies for exact solution, like constraint generation strategies should be looked into.

Further development of heuristic methods also remains useful, as discussed in [13]. As done in that work, one might also consider possible entry of several follower facilities. Extension of the MIP models developed here is theoretically possible, but would lead to a very important increase in model size, and therefore probably not exactly solvable by current means. Again constraint generation methods might be an outcome.
The study of the behaviour of the solutions generated by these models is underway. In particular it is of interest how one model, e.g. the apparently easiest maxmin model might function as proxy for the other two much harder models.

Other methods of resolving ties should be envisaged. On the one hand, one might replace our assumption of novelty orientation by conservative behaviour, meaning that when a consumer faces a tied choice between two facilities it is always the existing one that has the advantage. However, this (admittedly simple) rule would be stronger than merely forbidding proximity, since the tie in the consumer’s choice might also occur between facilities at quite different places. One might therefore assume on the other hand that in case of ties the consumer divides (e.g. equally) its demand between the tied facilities. Without foresight this leads to the maximal capture model initiated by ReVelle [10]. How to incorporate this assumption in our models with foresight is currently under investigation.

More research is also needed into the impact of proximity forbidding rules.

Finally it should be observed that the von Stackelberg assumption is not very precisely stated. What would happen if, given the leader facility locations, several optimal follower solutions exist? Would the follower choose any of these, the worst one for the leader, or rather the best? A little thought shows that the model formulated here assumes implicitly an optimising, but ‘friendly’ competitor, choosing for the last ‘win-win’ type solution: the way the $y_i$-variables were used only ensures that follower’s location is a best one for the follower, and the maximisation of the leader’s market share objective then leads to that optimal follower’s choice which maximises what’s left over to the leader. A model under the opposite assumption of a rational and ‘aggressive’ follower always opting for its optimal decision which harms the leader the most, is under study.

References


