An analytical lower bound for a class of minimizing quadratic integer optimization problems

Christian Schmitt[a], Bismark Singh[b,*]

[a] *Department of Mathematics, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, 91058, Germany*
[b] *School of Mathematical Sciences, University of Southampton, Southampton SO17 1BJ, UK*

Abstract

Lower bounds on minimization problems are essential for convergence of both branching-based and iterative solution methods for optimization problems. They are also required for evaluating the quality of feasible solutions by providing conservative optimality gaps. We provide an analytical lower bound for a class of quadratic optimization problems with binary decision variables. In contrast to traditional lower bounds typically obtained from solving relaxed optimization models, our lower bound is analytical and does not require a numerical solution of any mathematical optimization model. This especially provides value for instances of our optimization model that are computationally difficult to even generate, before being handed to a solver for a numerical solution. Further, we propose a greedy heuristic to determine a feasible solution that, in turn, allows us to evaluate the quality of this lower bound. Numerical results for instances that previously could not be generated in over five hours demonstrate an optimality gap of under 11% in less than a minute using our bounds.

Keywords: quadratic integer optimization, non-convex, analytical bounds, lower bounds, facility location models

1. Introduction

We revisit the following non-convex quadratic optimization model with binary variables proposed in [9]:

$$z^* = \min_{x,y,u} \sum_{j \in J} C_j (1 - u_j)^2 \tag{1a}$$

*Corresponding author
Email address: b.singh@southampton.ac.uk (Bismark Singh)

$$\text{s.t.} \quad u_j = \frac{\sum_{i \in I} U_i P_{i,j} x_{i,j}}{C_j} \qquad \forall j \in J \tag{1b}$$

$$\sum_{i \in I} U_i P_{i,j} x_{i,j} \leq C_j \qquad \forall j \in J \tag{1c}$$

$$\sum_{j \in J} y_j \leq B \tag{1d}$$

$$\sum_{j \in J} x_{i,j} = 1 \qquad \forall i \in I \tag{1e}$$

$$y_j \geq x_{i,j} \qquad \forall i \in I, j \in J \tag{1f}$$

$$u_j \in [0,1] \qquad \forall j \in J. \tag{1g}$$

$$y_j \in \{0,1\} \qquad \forall j \in J \tag{1h}$$

$$x_{i,j} \in \{0,1\} \qquad \forall i \in I, j \in J. \tag{1i}$$

Model (1) belongs to the family of facility location problems which have a rich history in the discrete and combinatorial optimization literature, see, e.g., [3]. Here, a set of users $i \in I$ with demands $U_i > 0$ are to be assigned to a set of facilities $j \in J$ with capacities $C_j > 0$. The quantity $P_{i,j} \in (0,1)$ denotes a discounting factor for each edge from i to j. The goal of model (1) is to select $B \leq |J|$ facilities that minimize the quantity given in the objective function (1a) while ensuring the capacities of the selected facilities are not exceeded and each user is assigned to exactly one facility. The constraints (1b)-(1i) enforce these restrictions. Here, the binary decision variables in constraints (1h) and (1i) represent whether j is selected and whether i is assigned to j, respectively. Constraint (1f) ensures that j is selected if any i is assigned to it, while if j is not selected then no i is assigned to it. Each i is assigned to exactly one j (represented by constraint (1e)), at most B facilities are selected (represented by constraint (1d)) and the capacities of any facility are not exceeded (represented by constraint (1c)). The decision variable u is an auxiliary variable, defined by constraints (1b) and (1g), that is directly computed from the decision variables x and y. As such, it can be removed from the optimization model (1) by substituting its definition in the objective function (1a); however, we include it for reasons we explain later.

The aim of this work is to present an *analytical* lower bound for model (1). By an analytical bound, we mean a lower bound that is expressed explicitly in terms of algebraic functions alone, or a formula, that does not require a numerical computation. We solve no additional optimization models to compute this bound. Such bounds are hard to obtain, even for highly-structured opti-

mization models and even for linear programming models. However, when available, an analytical bound — whether lower or upper — for an optimization problem presents several advantages over a bound computed by an iterative procedure even though the former is expected to be weaker. The foremost of these is that it does not even require the generation of a mathematical model. As such, the bound is not constrained by limitations of computational memory. This is the key motivation for this work as model (1) is computationally difficult to generate for large datasets [9]. Another advantage is that the bound is, naturally, deterministic and thus predictable as opposed to a randomized bound. Next, an analytical bound is easily introducible into numerical optimization solvers to warm-start a branch-and-bound tree; this has shown computational value for quadratic optimization models that are non-convex, such as ours, even when the warm-start is poor [2]. Similarly, such a bound could serve as the starting point for procedures that iteratively tighten lower bounds such as those from a convex underestimation, see, e.g., [1]. An analytical bound may further help in proving optimality, or determining deterministic optimality gaps, when a corresponding bound is also available from another mechanism, such as a numerical algorithm or a heuristic; and, our work is in this spirit. Finally, an analytical bound presents several insights into the structural properties of an optimization model which are not visible from its numerical analysis. These structural properties could then assist in developing tailored solution methods particularly for computationally challenging problems.

Lower bounds on minimization problem, such as model (1), are obtainable by solving its linear relaxation or a Lagrangian relaxation of its constraints. For example, relaxing constraint (1d) introduces just a single dual variable which can then be solved by a standard subgradient method. For such bounds of a quadratic optimization model, see, e.g., [1, 6]. However, for models that are difficult to generate, the corresponding relaxation might be difficult to generate as well; indeed, our previous work we find this to be the case for model (1), see [9]. Similar memory limitations have been reported before in the context of facility location problems, see, e.g., [5, 8]. We further mention that employing the Polyak step-size rule within the subgradient method [10] — one of the most common ways to update the step-size parameter when solving the Lagrangian relaxation — rests on the availability of a lower bound as well [7]. An analytical bound, such as that we propose, is then employable here as well.

The structure of the rest of this article is as follows. In Section 2, we present our analytical lower

bound via Theorem 1. The proof of this theorem rests on two lemmas that we also state in this section. In Section 3, we provide a greedy heuristic to achieve a fast feasible solution for model (1). Although there is merit in its own right for the heuristic as demonstrated by our computational experiments, the primary purpose of this heuristic is to compute a conservative optimality gap for our bound from Section 2; we provide these numerical results in Section 4. We conclude in Section 5.

2. An Analytical Lower Bound

In this section, we present an analytic lower bound for model (1) that does not require the numerical solution of any optimization model; Theorem 1 states this result. To this end, consider the following optimization model whose structural properties we exploit in the proof of Theorem 1.

$$z^*_{PF} = \min_{x,y,u} \sum_{j\in J} C_j(1-u_j)^2 \tag{2a}$$

$$\text{s.t.} \quad u_j = \frac{\sum_{i\in I} U_i \bar{P}_i x_{i,j}}{C_j} \qquad \forall j \in J \tag{2b}$$

$$\sum_{i\in I} U_i \bar{P}_i x_{i,j} \le C_j \qquad \forall j \in J \tag{2c}$$

$$\sum_{j\in J} y_j \le B \tag{2d}$$

$$\sum_{j\in J} x_{i,j} = 1 \qquad \forall i \in I \tag{2e}$$

$$y_j \ge x_{i,j} \qquad \forall i \in I, j \in J \tag{2f}$$

$$u_j \in [0,1] \qquad \forall j \in J. \tag{2g}$$

$$y_j \in \{0,1\} \qquad \forall j \in J \tag{2h}$$

$$x_{i,j} \in [0,1] \qquad \forall i \in I, j \in J. \tag{2i}$$

Model (2) differs from model (1) on only two grounds: (i) the discounting factor $\bar{P}_i$ is independent of j in model (2), and (ii) the binary restriction on the decision variable x in model (1) is replaced with its continuous relaxation in model (2). In [9], model (2) is used to study the so-called notion of proportional fairness which, although not relevant to this work, provides insight

into the structural properties of model (2). The following lemma states the pertinent results for proportional fairness for our work.

Lemma 1. *Assume that the set* $\bar{I} = \{i \in I, i : \bar{P}_i > 0, C_i > 0\}$ *is non-empty and that model* (2) *is feasible. Consider an optimal solution,* (x, y, u)*, of model* (2) *with* $J_O = \{j \in J : y_j = 1\}$*. Then,* $u_j = u_{j'} > 0, \forall j, j' \in J_O$*.*

Proof. From the hypothesis, there exists at least one i with strictly positive values for U_i and P_i. Then, from equation (2b) and equation (2e), we have at least one $j^* \in J$ that is selected and has $u_{j^*} > 0$; i.e., $|J_O| > 0$. Then, from Theorem 2 of [9] the utilization of all selected j are equal to u_{j^*}. The result follows. □

Additional to Lemma 1, we need the following lemma for the proof of Theorem 1.

Lemma 2. *Let* $\bar{P}_i = \max_{j \in J} P_{i,j}, \forall i \in I$*. Let*

$$z_1^* = \min_{x,y,u} \sum_{j \in J} C_j (1 - u_j)^2, \; s.t. \; \Big((1b) - (1i) \cap \{ \sum_{i \in I} U_i \bar{P}_i \leq \sum_{j \in J} y_j C_j \} \Big) \tag{3}$$

and

$$z_{PF}^* = \min_{x,y,u} \sum_{j \in J} C_j (1 - u_j)^2, \; s.t. \; \Big((2b) - (2i) \Big). \tag{4}$$

Then, $z_1^* \geq z_{PF}^*$*.*

Proof. Let (x^*, y^*, u^*) be an optimal solution of model (3). We show that there exists a feasible solution (x, y, u) for model (2), where $\bar{P}_i = \max_{j \in J} P_{i,j}, \forall i \in I$, with $u_j \geq u_j^*, \forall j \in J$. We first note that a feasible solution for the decision variables u always exists given a feasible solution for the decision variables x and y variables, since u is an auxiliary variable (i.e., directly computed from the x and y variables).

Let $J_O = \{j \in J : y_j^* = 1\}$ denote the set of selected facilities, $I_j = \{i \in I : x_{i,j}^* = 1\}, \forall j \in J$ denote the set of users that are assigned to facility j, and j_i denote the facility that user i is assigned to. Then, $\bar{P}_i \geq P_{i,j_i}, \forall i \in I$ and from equation (1e) $\sum_{j \in J} P_{i,j} x_{i,j}^* = P_{i,j_i}$; i.e., j_i is the facility j corresponding to $x_{i,j}^* = 1$. We distinguish two cases below: (i) $P_{i,j_i} = \bar{P}_i, \forall i \in I$ and (ii) $\exists i \in I : \bar{P}_i > P_{i,j_i}$.

(i) First, we show that $(x, y) \leftarrow (x^*, y^*)$ is a feasible solution for model (2).

- (x, y) satisfy constraints (2c):

  for $j \in J$, we have $\sum_{i \in I} U_i \bar{P}_i x_{i,j} =$

  $$\sum_{i \in I} U_i P_{i,j_i} x^*_{i,j} = \sum_{i \in I_j} U_i P_{i,j_i} = \sum_{i \in I_j} U_i P_{i,j} = \sum_{i \in I} U_i P_{i,j} x^*_{i,j} \leq C_j. \tag{5}$$

  The first equality holds from the hypothesis and from the construction of (x, y), while the second and fourth equality follows from the definition of I_j. The third equality holds from the fact that $j_i = j, \forall i \in I_j$, while the last inequality follows from equation (1c).

- (x, y) satisfy constraints (2b), (2d)-(2i):

  Constraint (2b) holds by definition, constraint (2g) follows from equation (2c) proven above, while the five constraints (2d)-(2f), (2h) and (2i) follow from the hypothesis and equations (1d)-(1h), respectively.

Next, we show that $u_j \geq u^*_j, \forall j \in J$. For $j \in J$, we have $u_j =$

$$\frac{\sum_{i \in I} U_i \bar{P}_i x_{i,j}}{C_j} = \frac{\sum_{i \in I} U_i P_{i,j_i} x^*_{i,j}}{C_j} = \frac{\sum_{i \in I} U_i P_{i,j} x^*_{i,j}}{C_j} = u^*_j.$$

The first and fourth equalities hold from constraints (2b) and (1b), respectively, the second equality follows from the hypothesis and the construction of x, and the third equality holds by the definition of j_i.

(ii) For this case, let $R_j = y^*_j \cdot (C_j - \sum_{i \in I_j} U_i P_{i,j}) \geq 0, \forall j \in J$ denote the surplus capacity for each facility; in particular, if j is not selected, then $R_j = 0$. Then, it follows from the hypothesis that

$$\begin{aligned}
0 &< \sum_{i \in I} U_i \bar{P}_i - \sum_{i \in I} U_i P_{i,j_i} && \text{(6a)} \\
&\leq \sum_{j \in J} y^*_j C_j - \sum_{i \in I} U_i P_{i,j_i} && \text{(6b)} \\
&= \sum_{j \in J} y^*_j C_j - \sum_{i \in I, j \in J} U_i P_{i,j} x^*_{i,j} && \text{(6c)} \\
&= \sum_{j \in J} y^*_j C_j - \sum_{j \in J, i \in I_j} U_i P_{i,j} && \text{(6d)} \\
&= \sum_{j \in J} y^*_j (C_j - \sum_{i \in I_j} U_i P_{i,j}) && \text{(6e)} \\
&= \sum_{j \in J} R_j. && \text{(6f)}
\end{aligned}$$

The first inequality holds from $\bar{P}_i \leq P_{i,j_i}, \forall i \in I$ and the hypothesis that this inequality is strict for at least one $i \in I$. The second inequality holds as y^* is feasible for model (3). The first, second, and fourth equalities hold from the definition of j_i, I_j, and R_j, respectively, while the third equality follows from equation (1f) and equation (1h).

Next, we construct a solution (x, y) of model (2) as follows:

$$x_{i,j} \leftarrow \frac{P_{i,j}}{\bar{P}_i} x^*_{i,j} + \frac{R_j}{\sum_{j \in J} R_j} \cdot \frac{U_i \bar{P}_i - U_i P_{i,j_i}}{U_i \bar{P}_i}, \qquad \forall i \in I, j \in J \tag{7a}$$

$$y_j \leftarrow y^*_j, \qquad \forall j \in J. \tag{7b}$$

Now, we show that (x, y) is feasible for model (2).

- (x, y) satisfy constraints (2b):

  This follows by definition.

- (x, y) satisfy constraints (2c):

  for $j \in J$, we have $\sum_{i \in I} U_i \bar{P}_i x_{i,j} =$

$$\sum_{i \in I} U_i \bar{P}_i \left(\frac{P_{i,j}}{\bar{P}_i} x^*_{i,j} + \frac{R_j}{\sum_{j \in J} R_j} \cdot \frac{U_i \bar{P}_i - U_i P_{i,j_i}}{U_i \bar{P}_i}\right) \tag{8a}$$

$$= \sum_{i \in I_j} U_i P_{i,j} + \frac{R_j}{\sum_{j \in J} R_j} \sum_{i \in I} \left(U_i \bar{P}_i - U_i P_{i,j_i}\right) \tag{8b}$$

$$\leq \sum_{i \in I_j} U_i P_{i,j} + \frac{\sum_{i \in I} U_i \bar{P}_i - \sum_{i \in I} U_i P_{i,j_i}}{\sum_{i \in I} U_i \bar{P}_i - \sum_{i \in I} U_i P_{i,j_i}} R_j \tag{8c}$$

$$= \sum_{i \in I_j} U_i P_{i,j} + R_j \tag{8d}$$

$$\leq \sum_{i \in I_j} U_i P_{i,j} + C_j - \sum_{i \in I_j} U_i P_{i,j} \tag{8e}$$

$$= C_j. \tag{8f}$$

  Equation (8a) holds from construction (7a), while equation (8b) follows from the definition of I_j. Equation (8c) holds from equation (6), while equation (8e) follows from the definition of R_j.

- (x, y) satisfy constraints (2d):

  This follows from construction (7b).

- (x, y) satisfy constraints (2e):

  for $i \in I$, we have $\sum_{j\in J} x_{i,j} =$

$$\sum_{j\in J} \left(\frac{P_{i,j}}{\bar{P}_i} x^*_{i,j} + \frac{R_j}{\sum_{j\in J} R_j} \cdot \frac{U_i\bar{P}_i - U_i P_{i,j_i}}{U_i\bar{P}_i}\right) \tag{9a}$$

$$= \frac{P_{i,j_i}}{\bar{P}_i} + \frac{U_i\bar{P}_i - U_i P_{i,j_i}}{U_i\bar{P}_i} \cdot \sum_{j\in J} \frac{R_j}{\sum_{j\in J} R_j} \tag{9b}$$

$$= \frac{P_{i,j_i}}{\bar{P}_i} + \left(1 - \frac{P_{i,j_i}}{\bar{P}_i}\right) \cdot \frac{\sum_{j\in J} R_j}{\sum_{j\in J} R_j} \tag{9c}$$

$$= 1. \tag{9d}$$

  Equation (9a) holds from construction (7a), equation (9b) follows from the definition of j_i, while equations (9c) and (9d) follow from algebra.

- (x, y) satisfy constraints (2f):

  We distinguish two cases: (a) $j \in J_O$ and (b) $j \in J \setminus J_O$.

  (a) For $i \in I$, we have $x_{i,j} =$

$$\frac{P_{i,j}}{\bar{P}_i} x^*_{i,j} + \frac{R_j}{\sum_{j\in J} R_j} \cdot \frac{U_i\bar{P}_i - U_i P_{i,j_i}}{U_i\bar{P}_i} \le \frac{P_{i,j}}{\bar{P}_i} x^*_{i,j} + \left(1 - \frac{P_{i,j_i}}{\bar{P}_i}\right) \le \frac{P_{i,j_i}}{\bar{P}_i} + \left(1 - \frac{P_{i,j_i}}{\bar{P}_i}\right) = 1 = y^*_j = y_j.$$

  The first equality holds from construction (7a), the first inequality follows from algebra, the second inequality uses the definition of j_i, the second equality follows from algebra, while the last two equalities follow from $j \in J_O$ and construction (7b), respectively.

  (b) We have $y^*_j = 0, \forall j \in J \setminus J_O$; thus, $x^*_{i,j} = R_j = 0, \forall j \in J \setminus J_O, i \in I$. Hence, for $i \in I$, we have $x_{i,j} =$

$$\frac{P_{i,j}}{\bar{P}_i} x^*_{i,j} + \frac{R_j}{\sum_{j\in J} R_j} \cdot \frac{U_i\bar{P}_i - U_i P_{i,j_i}}{U_i\bar{P}_i} = 0 = y^*_j = y_j.$$

  The first equality holds from construction (7a), the second equality follows from algebra, the third equality holds from construction, while the last equality follows from construction (7b).

- (x, y) satisfy constraints (2g):

  This follows from constraint (2c) proven above.

- (x, y) satisfy constraints (2h):

  This follows from construction (7b).

- (x, y) satisfy constraints (2i):

  This follows from construction (7a), $\bar{P}_i \geq P_{i,j_i}, \forall i \in I$, and $R_j \geq 0, \forall j \in J$.

Next, we show that $u_j \geq u_j^*, \forall j \in J$. For $j \in J$, we have $u_j =$

$$\frac{\sum_{i \in I} U_i \bar{P}_i x_{i,j}}{C_j} = \frac{\sum_{i \in I} U_i \bar{P}_i \left(\frac{P_{i,j}}{\bar{P}_i} x_{i,j}^* + \frac{R_j}{\sum_{j \in J} R_j} \cdot \frac{U_i \bar{P}_i - U_i P_{i,j_i}}{U_i \bar{P}_i}\right)}{C_j} \geq \frac{\sum_{i \in I} U_i P_{i,j} x_{i,j}^*}{C_j} = u_j^*.$$

The first equality follows from equation (2b), the second equality holds from construction (7a), the last equality follows from equation (1b), while the inequality follows from $R_j \geq 0, \forall j \in J$ and $\bar{P}_i \geq P_{i,j_i}, \forall i \in I$.

Thus, $u_j \geq u_j^*, \forall j \in J$, and (x, y) is feasible to model (2). Since $C_j > 0, \forall j \in J$, it follows that:

$$z_1^* = \sum_{j \in J} C_j (1 - u_j^*)^2 \geq \sum_{j \in J} C_j (1 - u_j)^2 = z_{PF}^*.$$

□

We are now ready to present our key result.

Theorem 1. *Let $\bar{P}_i = \max_{j \in J} P_{i,j}, \forall i \in I$. Let J_L denote the set of B facilities with the largest values of C_j. Assume models (1) and model (2) are feasible. Let z^* denote the optimal objective function value for model (1), $\underline{z_1^*} = \sum_{j \in J} C_j + \frac{\left(\sum_{i \in I} U_i \bar{P}_i\right)^2}{\sum_{j \in J_L} C_j} - 2 \sum_{i \in I} U_i \bar{P}_i$, and $\underline{z_2^*} = \sum_{j \in J} C_j - \sum_{i \in I} U_i \bar{P}_i$. Then, $\underline{z}^* = \min\{\underline{z_1}, \underline{z_2}\} \leq z^*$.*

Proof. Consider the two optimization models

$$z_1^* = \min_{x,y,u} \sum_{j \in J} C_j (1 - u_j)^2, \text{s.t.} \left((1b) - (1i) \cap \{ \sum_{i \in I} U_i \bar{P}_i \leq \sum_{j \in J} y_j C_j \} \right), \tag{10}$$

and

$$z_2^* = \min_{x,y,u} \sum_{j \in J} C_j (1 - u_j)^2, \text{s.t.} \left((1b) - (1i) \cap \{ \sum_{i \in I} U_i \bar{P}_i > \sum_{j \in J} y_j C_j \} \right). \tag{11}$$

Since model (1) is feasible by the hypothesis, exactly one of model (10) or model (11) is also feasible; let $+\infty$ be the optimal objective function value of either model if it is infeasible. Then, we have $z^* = \min\{z_1^*, z_2^*\}$. Hence, it suffices to show (i) $z_1^* \geq \underline{z_1^*}$ and (ii) $z_2^* \geq \underline{z_2^*}$. To do so, we distinguish the two exclusive cases below.

(i) Consider the first case; i.e., model (10) is feasible. Then, it follows from Lemma 2 that $z_1^* \geq z_{PF}^*$. Thus, to prove $z_1^* \geq \underline{z_1^*}$ it suffices to show $z_{PF}^* \geq \underline{z_1^*}$.

Since model (2) is feasible by the hypothesis, an optimal solution to it exists. Let (x, y, u) denote an optimal solution to model (2), $J_O = \{j \in J : y_j = 1\}$ denote the set of selected facilities, $a_j = \sum_{i \in I} U_i \bar{P}_i x_{i,j}, \forall j \in J$ for this optimal solution, and $\bar{u} = \frac{a_j}{C_j}$. Then, we have

$$\sum_{j \in J_O} a_j = \sum_{j \in J_O} \sum_{i \in I} U_i \bar{P}_i x_{i,j} = \sum_{i \in I} U_i \bar{P}_i \sum_{j \in J} x_{i,j} = \sum_{i \in I} U_i \bar{P}_i, \tag{12}$$

where the first equality holds by definition, the second equality follows follow from equations (2f), and third equality follows from equation (2e). From Lemma 1, we have $\bar{u} = \frac{a_j}{C_j} > 0, \forall j \in J_O$. This implies $\bar{u} \sum_{j \in J_O} C_j = \sum_{j \in J_O} a_j = \sum_{i \in I} U_i \bar{P}_i$; i.e., $u = \frac{\sum_{i \in I} U_i \bar{P}_i}{\sum_{j \in J_O} C_j}$. The objective function value of model (2) corresponding to this feasible solution is as follows.

$$z_{PF}^* = \sum_{j \in J} C_j (1 - u_j)^2 \tag{13a}$$

$$= \sum_{j \in J_O} C_j \left(1 - \frac{\sum_{i \in I} U_i \bar{P}_i}{\sum_{j \in J_O} C_j}\right)^2 + \sum_{j \in J \setminus J_O} C_j (1 - 0)^2 \tag{13b}$$

$$= \sum_{j \in J_O} \left(C_j + C_j \cdot \frac{(\sum_{i \in I} U_i \bar{P}_i)^2}{(\sum_{j \in J_O} C_j)^2} - 2 C_j \cdot \frac{\sum_{i \in I} U_i \bar{P}_i}{\sum_{j \in J_O} C_j}\right) + \sum_{j \in J \setminus J_O} C_j (1 - 0)^2 \tag{13c}$$

$$= \sum_{j \in J_O} C_j + \frac{\left(\sum_{i \in I} U_i \bar{P}_i\right)^2}{\sum_{j \in J_O} C_j} - 2 \sum_{i \in I} U_i \bar{P}_i + \sum_{j \in J \setminus J_O} C_j \tag{13d}$$

$$= \sum_{j \in J} C_j + \frac{\left(\sum_{i \in I} U_i \bar{P}_i\right)^2}{\sum_{j \in J_O} C_j} - 2 \sum_{i \in I} U_i \bar{P}_i \tag{13e}$$

Equation (13e) contains only data except the set J_O. Thus, it is minimized when J_O is the set of the B largest facilities; i.e., $J_O = J_L$. Then, $z_{PF}^* \geq \underline{z_1^*}$ follows.

(ii) Consider the second case; i.e., model (11) is feasible. Again, let $J_O = \{j \in J : y_j = 1\}$ denote the set of selected facilities. Then, from the hypothesis that $\sum_{i \in I} U_i \bar{P}_i > \sum_{j \in J} y_j C_j = \sum_{j \in J_O} C_j$ it follows that:

$$\sum_{j \in J \setminus J_O} C_j = \sum_{j \in J} C_j - \sum_{j \in J_O} C_j > \sum_{j \in J} C_j - \sum_{i \in I} U_i \bar{P}_i. \tag{14}$$

Now, consider the objective function value of model (11). We obtain a lower bound for z_2^* as follows.

$$z_2^* \geq \sum_{j \in J_O} C_j(1-1)^2 + \sum_{j \in J \setminus J_O} C_j(1-0)^2 \tag{15a}$$

$$= \sum_{j \in J \setminus J_O} C_j \tag{15b}$$

$$> \sum_{j \in J} C_j - \sum_{i \in I} U_i \bar{P}_i \tag{15c}$$

$$= \underline{z_2^*}. \tag{15d}$$

Here, the inequality equation (15a) follows from the fact that the terms in the objective function of model (11) decrease by increasing the u values. From constraint (1g), the maximum value of u for any selected facility j is 1. Thus, we set $u_j = 1, \forall j \in J_O$ and $u_j = 0$ otherwise. Equation (15b) holds directly, equation (15c) follows from equation (14), while equation (15d) follows by definition. Thus, $z_2^* \geq \underline{z_2^*}$.

This completes the proof. □

Theorem 1 provides a very easy way to calculate a lower bound for model (2). The only assumptions we employ for the proof of Theorem 1 are feasibility of model (1) and (2). Feasibility of model (1) is natural to assume as we seek its own lower bound and it is a minimization problem. Feasibility of model (2) is a relatively mild assumption which is satisfied given large enough capacities of the facilities; note that model (2) does not include binary restrictions on the x variables. Finally, we mention that our proof does not employ the convexity of the objective function (1a) and neither the unimodular structure of constraints (1e)-(1f).

The bound in Theorem 1 is given by the minimum of the lower bounds of two related optimization models, namely model (10) and model (11). By construction, only one of these two models is feasible. We do not require a solution of either optimization model. Given sufficient capacity, model (10) is feasible and then the first term, $\underline{z_1}$, provides the lower bound. When capacity is insufficient, the second term, $\underline{z_2}$, provides the lower bound. Here, this second case improves upon relatively trivial bounds when model (10) is infeasible. An example of such a trivial, but analytical, lower bound is $\sum_{j \in J} C_j$ which is obtained by simply setting all the u variables to 0. This bound is improved by instead selecting the the B largest facilities and setting their corresponding u variables

to 1; this provides a better analytical lower bound, although again trivial, of $\sum_{j \in J \setminus J_L} C_j$. The term, $\underline{z_2}$, improves this trivial bound; while, the term $\underline{z_1}$ improves this further when sufficient capacity is assured. Both these terms coincide with the trivial bound when $\sum_{i \in I} U_i \bar{P}_i = \sum_{j \in J_L} C_j$; note that this condition can be verified before the optimization. We summarize this discussion in the corollary below.

Corollary 1. *In Theorem 1, let* $\sum_{i \in I} U_i \bar{P}_i = \sum_{j \in J_L} C_j$. *Then,* $\underline{z}^* = \underline{z_1} = \underline{z_2} = \sum_{j \in J \setminus J_L} C_j$.

Proof. The proof follows from the first case of Theorem 1. □

The simple nature of the bound we present in Theorem 1 is motivated by models that are difficult to generate when populated by large instances. As such, we do not consider bounds based on a linear programming relaxation or solving a Lagrangian dual since they both require the solution (and, hence, generation) of the underlying optimization model. For such bounds of a quadratic optimization model, see, e.g., [1, 6]. However, a natural question is the effectiveness of such a bound particularly in light of its highly simple nature. We answer this question by constructing a corresponding upper bound given by a feasible solution to model (1); the presence of an upper bound additionally allows us to compute an optimality gap. We develop this upper bound using a fast heuristic that we describe next.

3. A Fast Greedy Upper Bound

Generating model (1) requires large amounts of memory. As we demonstrate in Section 4, we are unable to generate instances beyond $|I| \approx 1,200$ and $|J| \approx 700$ naively. Hence, our previous work employs a number of techniques to reduce the size of model (1), see [9]. In this section, we present an algorithm that provides high-quality feasible solutions for model (1). These feasible solutions provide upper bounds for model (1). In the same spirit as Section 2, the foremost advantage of our algorithm is that it does not require the generation of any optimization model. Further, the algorithm includes a heuristic that is greedy, and, as we demonstrate later in this work, the computational effort is significantly lesser than that of a naive solution method. We introduce our algorithm below.

Solving model (1) involves two steps. In the first step, we solve a combinatorial problem to select a subset of facilities; this step determines the y variables. In the second step, we solve an

assignment problem that allocates, subject to capacity restrictions, users to the selected facilities; this determines the x variables. This observation motivates a straightforward greedy heuristic where we solve both these steps heuristically. For a survey of heuristics for pure FLPs, see, e.g., [4]. Algorithm 1 presents our greedy algorithm that we summarize below.

The algorithm takes as input a lower bound, $\underline{z}$, for model (1). Such a bound comes from any procedure, and not necessarily the one we present in Section 2. We begin by selecting a set of B facilities with the largest capacities. To compute the assignments, we first determine the "most preferred facility" for each user; this is the most accessible facility among those that are selected and have sufficient capacity to accommodate this user. Step 7 of Algorithm 1 computes this. Assigning all users to their most preferred facility, j', could result in exceedance of the capacity of the corresponding facility, hence we conduct the assignments iteratively up to the limit. To this end, we assign users in decreasing rank of their preferences to j' while iteratively reducing the capacities. We continue until the capacity of j' is exhausted, and repeat this assignment process until all the users are assigned, thereby completing both phases of our proposed algorithm. This provides a feasible solution and a corresponding upper bound for model (1), plus a gap from the input lower bound $\underline{z}$.

Next, we improve this upper bound. We store the best objective function value and the corresponding (x, y) variables in Step 19. We repeat the process by selecting a new set of facilities in the first phase. We do so by swapping the $K < B$ selected facilities that have the lowest utilization in the current solution with the K facilities that have the largest overall access among those that are not yet selected. We make this swap clear in Steps 24 - 27 of Algorithm 1. We terminate the algorithm when one of three conditions is reached: (i) the algorithm stalls and there is no improvement in the best objective function values in N iterations, (ii) the time limit T is reached, or (iii) the gap between the $\underline{z}$ and the best objective function values is lesser than an optimality tolerance, ε. In all three cases, Algorithm 1 reports the best feasible solution and its corresponding upper bound, the gap from the input lower bound, and the runtime.

Model (1) is infeasible if there is insufficient capacity to accommodate all the users. A sufficient condition to ensure feasibility is $\sum_{i\in I} U_i \max_{j\in J} P_{i,j} \leq \sum_{j\in J} C_j$; similarly, a sufficient condition to guarantee that model (1) is infeasible is $\sum_{i\in I} U_i \min_{j\in J} P_{i,j} > \sum_{j\in J} C_j$. However, the greedily computed assignments in Algorithm 1 can fail to determine a feasible solution even if the capacities

of the selected facilities are sufficiently large. This can happen if, e.g., larger facilities are exhausted in the earlier iterations. In this case, we ignore this assignment and proceed to the next iteration; we do this in Step 11 of Algorithm 1.

Algorithm 1 A heuristic for greedy assignments of users to facilities

Input: an instance of model (1); a lower bound, $\underline{z}$, for the instance of model (1); an integer $K < B$; T; $\varepsilon \geq 0$; an integer $N > 0$; a procedure $\tilde{J} \leftarrow$ `sort` $(m, j \in J, A_j)$ that outputs the indices of the decreasingly sorted m largest values of the set A_j.

Output: a feasible solution, (x, y, u), of the input instance; an upper bound for the instance of model (1), $\bar{z}$; optimality gap, δ, of $\bar{z}$ from $\underline{z}$.

1: $J_B \leftarrow$ `sort` $(B, j \in J, C_j)$; $n \leftarrow 0$; $h \leftarrow$ `FALSE`.
2: **while** time $\leq T$
3: $R_j \leftarrow C_j, I_j \leftarrow \emptyset, \forall j \in J_B$; $I_A \leftarrow I$.
4: **while** $I_A \neq \emptyset$
5: $M_j \leftarrow \emptyset, \forall j \in J_B$.
6: **for** $i \in I_A$
7: $j' \leftarrow \arg\max_{\{j \in J_B : U_i P_{i,j} \leq R_j\}} \{P_{i,j}\}$.
8: $M_{j'} \leftarrow M_{j'} \cup \{i\}$.
9: **if** $M_j = \emptyset, \forall j \in J_B$
10: $n \leftarrow n + 1$.
11: go to Step 22.
12: **for** $j \in J_B : M_j \neq \emptyset$
13: $I_M \leftarrow$ `sort` $(|M_j|, i \in M_j, P_{i,j})$.
14: **for** $i \in I_M$
15: **if** $R_j - U_i P_{i,j} \geq 0$
16: $R_j \leftarrow R_j - U_i P_{i,j}$; $I_A \leftarrow I_A \setminus \{i\}$; $I_j \leftarrow I_j \cup \{i\}$.
17: $u_j \leftarrow \frac{\sum_{i \in I} U_i P_{i,j} x_{i,j}}{C_j}, \forall j \in J$; $h \leftarrow$ `TRUE`.
18: **if** $\sum_{j \in J} C_j (1 - u_j)^2 < \bar{z}$
19: $x_{i,j} \leftarrow 1, \forall i \in I_j, j \in J_B$, else $x_{i,j} \leftarrow 0$; $y_j \leftarrow 1, \forall j \in J_B$, else $y_j \leftarrow 0$; $\bar{z} \leftarrow \sum_{j \in J} C_j (1 - u_j)^2$; $\delta = \frac{\bar{z} - \underline{z}}{\bar{z}}$.
20: **else**
21: $n \leftarrow n + 1$.
22: **if** $n = N$ or $\delta < \varepsilon$
23: go to Step 29
24: $J_K \leftarrow$ `sort` $(K, j \in J_B, -u_j)$
25: $J_B \leftarrow J_B \setminus J_K$; $I_A \leftarrow I_A \cup I_j, \forall j \in J_K$.
26: $Q_j \leftarrow \sum_{i \in I_A} U_i P_{i,j}, \forall j \in J \setminus J_B$.
27: $J_B \leftarrow J_B \cup$ `sort` $(K, j \in J \setminus J_B, Q_j)$.
28: Update time to the cumulative wall-clock time.
29: **if** $h =$ `TRUE`
30: **Return:** $x, y, u, \bar{z}, \delta$, time.
31: **else**
32: **Return:** "No feasible solution constructed".

4. Numerical Results

In this section, we perform numerical evaluations for the lower and upper bounds of Section 2 and Section 3, respectively. We carry out all computational experiments on two high performance computing clusters with Intel Xeon E5-2643 v4 processors with 256 GB of RAM, Pyomo version 6.1.2 and Gurobi version 9.1.2; this setup is the same as in [9]. We create a pool of random instances for our experiments for different values of $|I|$ and $|J|$, and solve all instances with $B = 0.7|J|$. For reference, the original work has $|I| = 2{,}060$ and $|J| = 1,394$. The rest of our data is the same as in [9]. We use a maximum time limit of $\mathbf{T}$= 20,000 seconds for our experiments, and set the input parameters K and N of Algorithm 1 to $0.98|B|$ and 20, respectively.

Table 1 presents our results. The first row includes 10% of the original number of $|I|$ and $|J|$, and the remaining rows contain progressively 10% more $|I|$ and $|J|$ values. For comparison, we solve model (1) naively; i.e., without any of the computational enhancements suggested in [9]. Instances with at least 60% of the original $|I|$ and $|J|$ values could not be generated by a naive solution method; i.e., beyond 1,236 users and 836 facilities. Thus, five out of the ten instances could not even be generated by the naive solution method.

In Table 1, the "Gap" columns denote the respective optimality gaps; here, the "Naive" gap denotes the MIP gap reported by Gurobi, and the "Algorithm" gap denotes the gap between the best solution of Algorithm 1 and Theorem 1. Thus, the "Algorithm" gap provides a provable optimality guarantee for Algorithm 1. To further test the performance of Algorithm 1, we compare its best solution with that reported by the naive solution method. The "Guarantee" column provides the gap between these two quantities. A comparison of these two guarantees demonstrates that Algorithm 1 achieves results that are at most 6% away from the optimal. The analytical bounds by Theorem 1 demonstrate a provable optimality guarantee of at most 11%; note that this guarantee is conservative. On average, Algorithm 1 with Theorem 1 provides a provable optimality gap of 9.8%. These results demonstrate that Algorithm 1 is well-suited for cases where computing a solution directly is computationally expensive. The naive solution method could only solve a single instance to optimality within this time limit.

For the "Time", "Overall access" and "CV_w" columns in Table 1, Δ denotes the relative improvement of Algorithm 1 as compared to the naive solution method. As evidenced by the "Time" columns, the computational effort required by Algorithm 1 is minimal. Algorithm 1 solves the

largest instance to an under 10% provably-optimal gap in less than a minute, however the naive solution method is unable to even generate the instance in 20,000 seconds. This again demonstrates that Algorithm 1 is well-suited for cases where computing a solution directly is computationally expensive. The naive solution method could only solve a single instance to optimality within the time limit.

To measure the quality of a feasible solution, we use two metrics. For the users, $i \in I$, we define the overall access as $100\frac{\sum_{j\in J}\sum_{i\in I} U_i P_{i,j} x_{i,j}}{\sum_i U_i}$. For the facilities, $j \in J$, we define the weighted coefficient of variation, $CV_w = \frac{\sqrt{Var_w(u)}}{\bar{u}_w}$; here, $\bar{u}_w = \frac{\sum_{j\in J_O} C_j u_j}{\sum_{j\in J_O} C_j}$ and $Var_w(u) = \frac{\sum_{j\in J_O} C_j (u_j - \bar{u}_w)^2}{\sum_{j\in J_O} C_j}$ are the weighted average and weighted variance of the utilization of open facilities, respectively, and $J_O = \{j \in J : y_j = 1\}$ is the set of open facilities in a feasible solution. For a rationale of these metrics, see [9]. Then, as our results in Table 1 inform, the overall access achieved by Algorithm 1 slightly exceeds that obtained by the naive solution method for the five instances that can be generated. However, this increased access comes at a price of suboptimality measured by a reduction in proportional fairness. The CV_w, see the last three columns of Table 1, is about 40% more for the solutions of Algorithm 1 than those by the naive solution method. We mention again that our numbers are conservative underestimates since model (1) is not solved to optimality.

Instance		**Gap [%]**			**Time [s]**			**Overall access [%]**			**CV_w [%]**		
$\|I\|$	$\|J\|$	Naive	Algorithm	Guarantee	Naive	Algorithm	Δ	Naive	Algorithm	Δ	Naive	Algorithm	Δ
206	139		7.5	1.7	25.2	0.7	97.1%	64.3	64.6	0.5%	22.7	31.9	-40.3%
412	279	1.9	8.3	4.1	**T**	1.9	100.0%	63.2	63.3	0.1%	22.2	31.3	-41.4%
618	418	3.1	10.8	5.7	**T**	3.9	100.0%	60.6	60.8	0.3%	22.6	32.7	-44.7%
824	558	3.5	10.7	5.6	**T**	6.3	100.0%	61.6	61.7	0.1%	23.5	31.8	-35.5%
1,030	697	3.6	11.0	5.9	**T**	9.9	100.0%	59.0	59.1	0.1%	24.4	33.3	-36.6%
1,236	836	-	9.6	-	∞	14.2	-	-	61.5	-	-	32.1	-
1,442	976	-	10.1	-	∞	19.2	-	-	59.6	-	-	32.7	-
1,648	1,115	-	10.7	-	∞	25.8	-	-	59.5	-	-	34.1	-
1,854	1,255	-	9.7	-	∞	31.2	-	-	59.8	-	-	32.9	-
2,060	1,394	-	9.6	-	∞	39.2	-	-	60.3	-	-	33.9	-

Table 1: Comparison of the computational performance of Algorithm 1 ("Algorithm") and the naive solution method ("Naive") for model (1). The "Δ" columns denote the relative difference between Naive and Algorithm. The Δ values in the "Overall access" and "CV_w" columns are calculated corresponding to the best feasible solution from the naive solution method; here, a positive Δ suggests Algorithm 1 provides an improvement over the naive solution method. The "Gap" columns provide the MIP gap reported by Gurobi for the naive solution method and the the gap reported by Algorithm 1, respectively, while the "Guarantee" column denotes the conservative gap between the upper bound reported by Algorithm 1 and the lower bound reported by the naive method. A blank denotes the instance is solved to optimality, while "-" denotes the instance could not be constructed naively due to a lack of memory. Entries marked with "**T**" in the "Time" columns denote the maximum time limit is reached. For details, see Section 4.

5. Conclusions

Motivated by a recently proposed facility location problem with a quadratic objective function that is computationally challenging to solve, we propose a heuristic that provides an upper bound as well as derive a theoretical lower bound. Specifically, we are interested in bounds since the model is computationally difficult to generate even without proceeding to a solution method. Both of our bounds are obtained without requiring the solution of any optimization model. Computing lower bounds for a minimization problem are important for the convergence of a branching-based algorithm. However, while upper bounds are available relatively easily from feasible solutions, good quality lower bounds are harder to obtain. Analytical bounds are rarely available for any mathematical optimization models. However, the analytical nature of the lower bound we present

makes it simple to evaluate; computational results demonstrate its merit despite this simplicity. The upper bound is a simple greedy heuristic that employs a lower bound as input to determine an optimality gap; again, our numerical results demonstrate significant savings in computational effort. Future work could determine improved lower and upper bounds by studying whether the proposed model is submodular. Another direction of work is the development of such analytical lower bounds for the generalized quadratic facility location problem.

Declarations of interest: none.

Acknowledgements

The authors gratefully acknowledge the compute resources and support provided by the Erlangen Regional Computing Center (RRZE).

References

[1] Bomze, I.M., Locatelli, M., Tardella, F., 2007. New and old bounds for standard quadratic optimization: Dominance, equivalence and incomparability. Mathematical Programming 115, 31–64. doi:10.1007/s10107-007-0138-0.

[2] Burer, S., Vandenbussche, D., 2006. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. Mathematical Programming 113, 259–282. doi:10.1007/s10107-006-0080-6.

[3] Farahani, R.Z., Hekmatfar, M. (Eds.), 2009. Facility location: Concepts, models, algorithms and case studies. Contributions to Management Science, Physica-Verlag Heidelberg, Heidelberg. doi:10.1007/978-3-7908-2151-2.

[4] Jain, K., Mahdian, M., Saberi, A., 2002. A new greedy approach for facility location problems, in: Proceedings of the 34th Annual ACM Symposium on the Theory of Computing, pp. 731–740. doi:10.1145/509907.510012.

[5] Jena, S.D., Cordeau, J.F., Gendron, B., 2017. Lagrangian heuristics for large-scale dynamic facility location with generalized modular capacities. INFORMS Journal on Computing 29, 388–404. doi:10.1287/ijoc.2016.0738.

[6] Nowak, I., 1999. A new semidefinite programming bound for indefinite quadratic forms over a simplex. Journal of Global Optimization 14, 357–364. doi:10.1023/A:1008315627883.

[7] Polyak, B.T., 1987. Introduction to optimization. Translations Series in Mathematics and Engineering, Optimization Software, New York.

[8] Risanger, S., Singh, B., Morton, D., Meyers, L.A., 2021. Selecting pharmacies for COVID-19 testing to ensure access. Health Care Management Science 24, 330–338. doi:10.1007/s10729-020-09538-w.

[9] Schmitt, C., Singh, B., 2024. Quadratic optimization models for balancing preferential access and fairness: Formulations and optimality conditions. INFORMS Journal on Computing doi:10.1287/ijoc.2022.0308.

[10] Shor, N.Z., 1968. The rate of convergence of the generalized gradient descent method. Cybernetics and Systems Analysis 4, 79–80. doi:10.1007/BF01073933.