Facility Location Optimizer

Reference Manual

Version 1.2.2

February 12, 2016

Project Facility Location Optimizer

Project Facility Location Optimizer (Project FLO) is a research project created at the Institute for Mathematics of the Martin Luther University Halle-Wittenberg. The main purpose of the project is the development of a MATLAB-based software tool (aka FLO) for solving location problems.

Development of the software started on March 01, 2011 and the first version of FLO was released on April 22, 2015.

FLO Development leader

Christian Günther (Christian.Guenther@mathematik.uni-halle.de)

Coordinators of Project FLO

Christian Günther

Prof. Dr. Christiane Tammer (Christiane.Tammer@mathematik.uni-halle.de)

Members of Project FLO

Christian Günther

Marcus Hillmann (Marcus.Hillmann@mathematik.uni-halle.de)

Prof. Dr. Christiane Tammer

Brian Winkler (Brian.Winkler@mathematik.uni-halle.de)

Other contributors of Project FLO

Rico Khan
(Bachelor-Thesis “Algorithmen für restringierte Standortprobleme einschließlich Implementierung”, Martin Luther University Halle-Wittenberg, 08/2015 - 12/2015)

Manual FLO version 1.2.2

The Software FLO (version 1.2.2) manual was created and written by:

Christian Günther,

Marcus Hillmann,

Prof. Dr. Christiane Tammer,

Brian Winkler.

Contact

The members of Project FLO can be contacted at the following address:

Project FLO
Martin Luther University Halle-Wittenberg
Faculty of Natural Sciences II
Institute for Mathematics
Working Group Optimization and Stochastics
06099 Halle (Saale), Germany

E-Mail: contact@project-flo.de
Web: http://www.project-flo.de/contact-us/

Please feel free to contact us with questions, feedback or bug reports about the software or manual.

Contents

Project FLO Non-Commercial License Agreement

Project FLO is Christian Günther, Prof. Dr. Christiane Tammer, Marcus Hillmann and Brian Winkler of Martin-Luther-Universität Halle-Wittenberg in Halle (Saale), Germany ("we", "our","us"). We created and developed the software application Facility Location Optimizer including related supporting resources ("FLO", "the Software"). We operate and publish FLO through our website located at http://project-flo.de.

The terms of this License form a binding agreement between you, an individual user or non-commercial organization ("you" or "your"), and us regarding your non-commercial use of the Software. By downloading, accessing or otherwise using the Software you indicate your agreement to be bound by these License terms.

Non-commercial License Terms

1. The Software and any other related resources (including documentation) are licensed to you on a limited, non-exclusive, personal, non-transferable and royalty-free license under which you are free to use the Software and other resources PROVIDED THAT you only do so for non-commercial purposes (without charging a fee to any third party) and PROVIDED THAT you attribute the work to us by using an appropriate citation or (at least) mentioning our name, including an appropriate copyright notice and providing a link to our website located at http://project-flo.de.

2. The Software (and all related materials and resources) are licensed to you WITHOUT ANY WARRANTY and on an AS IS basis including without limitation the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. We accept no liability for your use of the Software (save to the extent such liability cannot be excluded as a matter of law).

3. The Software (and all related materials and resources) are licensed to you without any offer or promise of support or future development by the Project FLO or any third party.

4. You may install FLO on multiple devices in multiple locations PROVIDED THAT you always use the Software for non-commercial purposes and otherwise in accordance with these License terms.

5. The copyright and other intellectual property rights (including any trademarks) of whatever nature (arising anywhere in the world) in the FLO software (and all related resources) are and will remain our property (or in the case of third party materials (including software libraries) which we have the right of use, the property of the third party licensor), and we reserve the right to grant licenses to use the FLO software (and all related resources) to third parties.

6. An “appropriate copyright notice” for the purposes of this License shall take the following form:

   Copyright © Project FLO, 2015-2016

   An “appropriate citation” for use in scientific or other works is:

   C. Günther, M. Hillmann, Chr. Tammer and B. Winkler: Facility Location Optimizer (FLO) - A tool for solving location problems, www.project-flo.de

7. You may not use, copy, modify, or transfer the software or any copy, modification, or merged portion, in whole or in part, except as expressly provided for in this Agreement.

8. You may not decompile, disassemble, or reverse engineer any of the Software or attempt to do so.

9. This License is personal to you and you may not assign it to a third party or permit any third party to benefit from it without our prior written consent. You may not rent, lease, sublicense, or transfer the Software.

10. You will notify us immediately if you become aware of any unauthorised use of the whole or any part of the Software (and all related resources).

11. If any of the provisions of this License (including the additional terms incorporated by reference) are held to be invalid or unenforceable under any applicable statute or rule of law, it is to that extent to be deemed omitted from the License. Such an omission will not affect the validity of the remaining provisions of the License, which will remain in full force and effect.

12. This license is effective until terminated. You may terminate it at any time by destroying all provided copies of the Software covered by this

License and all related resources including support files generated by the Software. It will also terminate if you fail to comply with any term or condition of this License. You agree that upon such termination to destroy this Software, including all copies, functionally-equivalent derivatives, and all portions and modifications thereof in any form.

13. This License will be governed by and constructed in accordance with the laws of the Federal Republic of Germany.

Software FLO uses the following icon packages:

- Free 3d Glossy Interface Icon Set

  Author: Aha-Soft (http://www.aha-soft.com/)
  License Agreement: CC Attribution 3.0 United States

  You can find a full text of this license agreement here:
  http://creativecommons.org/licenses/by/3.0/us/

  Product and download page:
  http://www.softicons.com/toolbar-icons/free-3d-glossy-interface-icons-by-aha-soft

- 24x24 Free Button Icons

  Author: Aha-Soft (http://www.small-icons.com/)
  License Agreement: CC Attribution-ShareAlike 3.0 Unported

  You can find a full text of this license agreement here:
  http://creativecommons.org/licenses/by-sa/3.0/

  Product and download page:
  http://www.softicons.com/toolbar-icons/24x24-free-button-icons-by-aha-soft

- 16x16 Free Application Icons

  Author: Aha-Soft (http://www.small-icons.com/)
  License Agreement: CC Attribution-ShareAlike 3.0 Unported

  You can find a full text of this license agreement here:
  http://creativecommons.org/licenses/by-sa/3.0/

Product and download page:
http://www.softicons.com/toolbar-icons/16x16-free-application-icons-by-aha-soft

1 Introduction

Project Facility Location Optimizer (Project FLO) is a research project created at the Institute for Mathematics of the Martin Luther University Halle-Wittenberg. The main purpose of the project is the development of a MATLAB-based software tool (aka FLO) for solving location problems.

The development of the software started on March 01, 2011 and the first version of FLO was released on April 22, 2015.

In the following section we introduce some notions and basic concepts, which are essential for understanding the underlying location theory concerning the Software FLO, and moreover we will give a short introduction into the field of Multiobjective Optimization.

1.1 About Location Problems

Location problems appear in many variants and with different constraints depending on the practical application, for instance in the following areas:

- Urban and Regional Planning (e.g. locations for emergency facilities),
- Technology (e.g. placement of sensors on technical components),
- Economy (e.g. planning new production facilities),
- Geography (e.g. landscape design),
- Environment-Oriented Project Management (e.g. development of mining landscapes),
- Engineering.

Location problems and corresponding algorithms are well-studied in the literature, see for instance the books by Love, Morris and Wesolowsky [36], Hamacher [24], Drezner and Hamacher [9], Göpfert, Riahi, Tammer, Zălinescu [18], Göpfert, Riedrich and Tammer [19] and for an overview in the book sections by Nickel, Puerto and Rodriguez-Chia [41, 42].

Now we consider m points in the plane,

$$a^1 := (a_1^1, a_2^1), \cdots, a^m := (a_1^m, a_2^m) \in \mathbb{R}^2,$$

representing some a priori given facilities. The set

$$A := \{a^1, \dots, a^m\}$$

represents the set of all existing facilities. In many problems of locational analysis the decision maker is looking for new facilities such that the distances between the new facilities and existing facilities are minimal in a certain sense. One possibility is that these distances are described by an appropriate norm

$$||\cdot|| : \mathbb{R}^2 \to \mathbb{R}.$$

See Section 1.2 for more details about distance measures.

Now there are several possibilities to define a location problem. For instance we can consider a planar **median problem** that is defined by

$$\sum_{i=1}^{m} v_i \cdot ||x - a^i|| \to \min_{x \in \mathbb{R}^2}, \tag{1}$$

where v_i is a positive weight (e.g., significance of the facility) associated to the point a^i for all $i = 1, \dots, m$.
In its first and simplest form, such a problem (1) was posed by the jurist and mathematician Fermat in 1629. He asked for the point realizing the minimal sum of distances from three given points. In 1909 this problem appeared, in a slightly generalized form, in the pioneering work "Über den Standort der Industrien" of Weber [48]. Therefore, the problem given by (1) is called Fermat-Weber problem in the literature of location theory and involves, in his original formulation, the Euclidean norm as distance function. A comprehensive and recently published overview over methods for solving the Fermat-Weber problem is presented in the paper by Beck and Sabach [4] (2015).

Another class of location problems are planar **center problems**. The goal is to minimize the maximum of distances between the new facilities $x \in \mathbb{R}^2$ and existing facilities $a^1, \dots, a^m$, i.e., we consider the following problem

$$\max\{v_i \cdot ||x - a^i|| \mid i = 1, \dots, m\} \to \min_{x \in \mathbb{R}^2} \tag{2}$$

with weights $v_i > 0$ for all $i = 1, \dots, m$. For the special case with $v_i = 1$ for all $i = 1, \dots, m$ and the Euclidean norm as distance measure is the location problem (2) known as the smallest-circle problem or minimum covering circle problem in the literature of location theory. Applications of this model appear in, for instance, Emergency Management.

However, for the decision maker it is often difficult to choose the weights. If he has chosen the weights and computed a solution of this scalar location problem, it could be possible that the solution is not practicable. So it is more convenient for the decision maker to study a multiobjective location problem with the distances in the components of the vector-valued objective function. In this way, the decision maker gets an overview of the whole solution set, even on special solutions of the scalar problems, making it possible to better understand the problem.

The classical **multiobjective location problem** (also known in the literature as "point-objective location problem") consists in finding a new location such that the distances between given facilities and the new facility are minimized in the sense of multiobjective optimization (see Section 1.6):

$$\begin{pmatrix} ||x - a^1|| \\ \dots \\ ||x - a^m|| \end{pmatrix} \to \operatorname*{v-min}_{x \in \mathbb{R}^2} .$$

1.2 Choice of distance function

In this section we investigate the question how we can measure the distances between given points in $\mathbb{R}^2$. Therefore, in the following we introduce some well-known concepts for measuring distances.

1.2.1 Metrics and Norms

The distances between two points in the plane can be measured using an appropriate metric.

Definition 1. *Let Y be a non-empty set of $\mathbb{R}^2$. A function $d : Y \times Y \to \mathbb{R}$ is called metric on Y, if d fulfills the following conditions for all $x, y, z \in Y$:*

$(M1)$: $d(x, y) = 0 \iff x = y$ *(definiteness),*

$(M2)$: $d(x, y) = d(y, x)$ *(symmetry),*

$(M3)$: $d(x, z) \leq d(x, y) + d(y, z)$ *(triangle inequality).*

The real number $d(x, y)$ represents the distance between the points x and y.

Definition 2. *A function $||\cdot|| : \mathbb{R}^2 \to \mathbb{R}$ is called norm on $\mathbb{R}^2$, if $||\cdot||$ fulfills the following conditions for all $x, y \in \mathbb{R}^n$ and for all $\alpha \in \mathbb{R}$:*

$(N1)$: $||x|| = 0 \iff x = 0$ *(definiteness),*

$(N2)$: $||\alpha \cdot x|| = |\alpha| \cdot ||x||$ *(positive homogeneity),*

$(N3)$: $||x + y|| \leq ||x|| + ||y||$ *(triangle inequality).*

If $||\cdot|| : \mathbb{R}^2 \to \mathbb{R}$ is a norm, then it is possible to define a metric on $\mathbb{R}^2$ that is induced by the norm $||\cdot||$ in the following way

$$d(x, y) := ||x - y|| \quad \text{for all } x, y \in \mathbb{R}^2.$$

Now we present some well-known distance measures.

Example 1. *The l_p norm is defined for all $x \in \mathbb{R}^2$ by*

$$||x||_p := \begin{cases} \left(\sum_{i=1}^{2} |x_i|^p \right)^{\frac{1}{p}} & \textit{for } 1 \leq p < \infty, \\ \max_{i=1,2} |x_i| & \textit{for } p = \infty. \end{cases} \tag{3}$$

Moreover, we can define the l_p metric for all $x, y \in \mathbb{R}^2$ through

$$d_p(x, y) := \begin{cases} \left(\sum_{i=1}^{2} |x_i - y_i|^p \right)^{\frac{1}{p}} & \textit{for } 1 \leq p < \infty, \\ \max_{i=1,2} |x_i - y_i| & \textit{for } p = \infty. \end{cases} \tag{4}$$

Let $x := (x_1, x_2)$, $y := (y_1, y_2) \in \mathbb{R}^2$. Some important special cases of (3) *and* (4) *are*

$$||x||_1 := |x_1| + |x_2| \quad \textit{(Manhattan norm)},$$
$$d_1(x, y) := ||x - y||_1 = |x_1 - y_1| + |x_2 - y_2| \quad \textit{(Manhattan metric)},$$

$$||x||_\infty := \max\{|x_1|, |x_2|\} \quad \textit{(Maximum norm)},$$
$$d_\infty(x, y) := ||x - y||_\infty = \max\{|x_1 - y_1|, |x_2 - y_2|\} \quad \textit{(Maximum metric)},$$

$$||x||_2 := \sqrt{(x_1)^2 + (x_2)^2} \quad \textit{(Euclidean norm)},$$
$$d_2(x, y) := ||x - y||_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad \textit{(Euclidean metric)}.$$

In addition, we introduce the squared Euclidean norm

$$||x||_2^2 := (x_1)^2 + (x_2)^2 \quad \textit{(squared Euclidean norm)},$$
$$d_2^2(x,y) := ||x-y||_2^2 = (x_1-y_1)^2 + (x_2-y_2)^2 \quad \textit{(squared Euclidean metric)}.$$

Note that it can easily be proven that the squared Euclidean norm is not a norm in the sense of Definition 2 (in general the norm axioms (N2) and (N3) are not fulfilled).

Figure 1 shows the unit balls

$$B_i(0,1) := \{x \in \mathbb{R}^2 \mid ||x||_i \leq 1\},\ i \in \{1,2,\infty\}$$

of the Manhattan norm, the Euclidean norm and of the maximum norm.

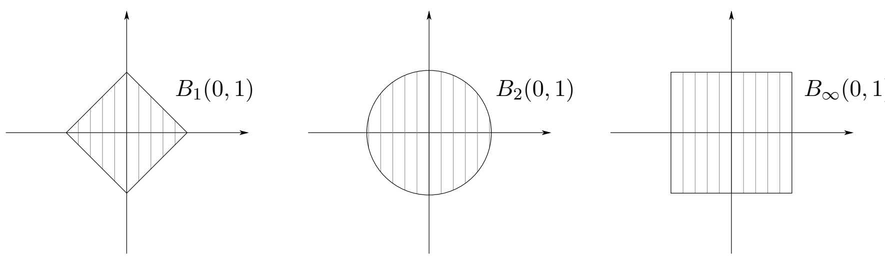


Figure 1: Unit balls of the norms $||\cdot||_1$, $||\cdot||_2$ and $||\cdot||_\infty$ on $\mathbb{R}^2$.

Example 2. *Additionally, we introduce a special distance measure by*

$$||x||_{1,\infty} := \frac{1}{4}\cdot||x||_1 + \frac{1}{4}\cdot||x||_\infty \quad \textit{(One-infinity-norm)},$$
$$d_{1,\infty}(x,y) := \frac{1}{4}\cdot||x-y||_1 + \frac{1}{4}\cdot||x-y||_\infty \quad \textit{(One-infinity-metric)}$$

for all $x,y \in \mathbb{R}^2$. Note that the (weighted) one-infinity-norm is a weigted sum of the Manhattan norm and the maximum norm.

1.2.2 Gauges

A more general concept for measuring distances (in comparison to metrics induced by norms) are distance functions induced by so called gauges.

In the following we introduce this concept and discuss some useful properties of gauges.

Definition 3. *Let B_μ be a compact and convex set in $\mathbb{R}^2$ with $0 \in \operatorname{int} B_\mu$. A gauge $\mu : \mathbb{R}^2 \to \mathbb{R}$ is defined by*

$$\mu(x) := \inf\{\lambda > 0 \,|\, x \in \lambda \cdot B_\mu\}$$

for all $x \in \mathbb{R}^2$.

Remark 1. *Note that a gauge function is also known as a special case of the Minkowski functional. Moreover, the function μ has the following useful properties:*

- $(G1)$: *Definiteness: It holds $\mu(x) = 0$ if and only if $x = 0$.*
- $(G2)$: *Non-negativity, i.e., for all $x \in \mathbb{R}^2$ it holds $\mu(x) \geq 0$.*
- $(G3)$: *Positive homogeneity, i.e., for all $x \in \mathbb{R}^2$ and for all $t \geq 0$ we have $\mu(t \cdot x) = t \cdot \mu(x)$.*
- $(G4)$: *Subadditivity (Triangle inequality), i.e., for all $x, y \in \mathbb{R}^n$ it holds $\mu(x + y) \leq \mu(x) + \mu(y)$.*
- $(G5)$: *μ is convex on $\mathbb{R}^2$.*
- $(G6)$: *μ is continuous on $\mathbb{R}^2$.*
- $(G7)$: *If B_μ is symmetric with respect to the origin (i.e., $B_\mu = -B_\mu$), then μ defines a norm on $\mathbb{R}^2$.*

We call a gauge with polyhedral unit ball B_μ a polyhedral gauge and a polyhedral gauge with symmetric unit ball B_μ is called a block norm.

For two points $x, y \in \mathbb{R}^2$ in the plane we can define a metric by

$$d(x, y) := \mu(x - y) = \inf\{\lambda > 0 \,|\, x - y \in \lambda \cdot B_\mu\}.$$

Example 3. *Let B_μ be a polytope in $\mathbb{R}^2$ with four extreme points and $0 \in \operatorname{int} B_\mu$. Assume that B_μ has the representation*

$$B_\mu = \operatorname{conv}\{e^1, e^2, e^3, e^4\}$$

for some extreme points $e^1, e^2, e^3, e^4 \in \mathbb{R}^2$ in clockwise order. The stationary vectors from the origin to the extreme points of the polytope B_μ are called fundamental directions in the literature of location theory. Moreover, the half lines

$$\{\lambda \cdot e^i \,|\, \lambda \geq 0\} \quad (i = 1, 2, 3, 4)$$

generated by the fundamental directions are called fundamental lines or construction lines.

We are now able to define so called fundamental cones (assume $e^5 := e^1$):

$$K_i := \{\lambda_1 \cdot e^i + \lambda_2 \cdot e^{i+1} \,|\, \lambda_1, \lambda_2 \geq 0\}$$

for all $i = 1, 2, 3, 4$. The Figure 2 visualizes the process of determining the function value of a polyhedral gauge. It is shown that

$$\mu(\tilde{x}) = \inf\{\lambda > 0 \,|\, \tilde{x} \in \lambda \cdot \mathrm{conv}\{e^1, e^2, e^3, e^4\}\} = 3$$

for the point $\tilde{x} \in \mathbb{R}^2$ (as defined in Fig. 2) holds.

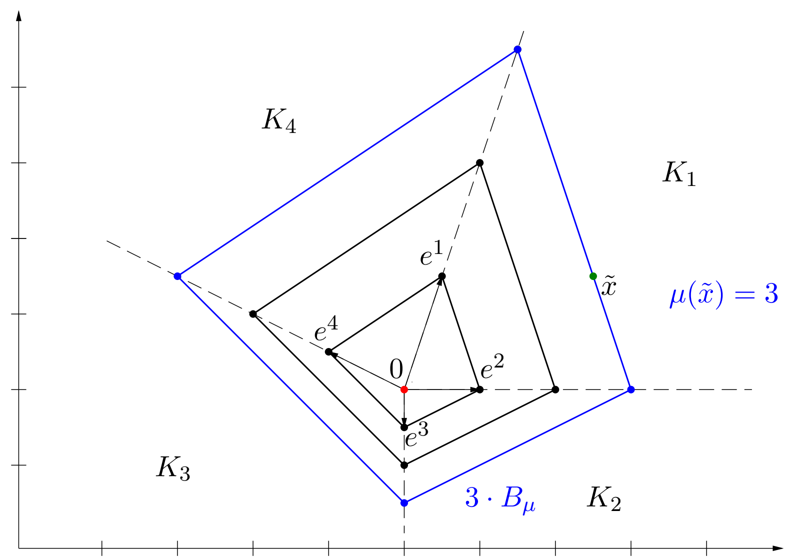


Figure 2: An example of determining the function value of a polyhedral gauge.

Example 4. *The unit ball of the one-infinity-norm $||\cdot||_{1,\infty}$ (see Example 2) can be represented by the convex hull of eight points*

$$\begin{array}{llll} e^1 := (0,2); & e^2 := (\frac{4}{3}, \frac{4}{3}); & e^3 := (2,0); & e^4 := (\frac{4}{3}, -\frac{4}{3}); \\ e^5 := (0,-2); & e^6 := (-\frac{4}{3}, -\frac{4}{3}); & e^7 := (-2,0); & e^8 := (-\frac{4}{3}, \frac{4}{3}), \end{array}$$

i.e., we have

$$B_\mu := \mathrm{conv}(\{e^1, \ldots, e^8\}) = \{x \in \mathbb{R}^2 \,|\, ||x||_{1,\infty} \leq 1\}.$$

Since B_μ is a compact and convex set in $\mathbb{R}^2$ with $0 \in \operatorname{int} B_\mu$, we know that $\mu = ||\cdot||_{1,\infty}$ defines a gauge. Moreover, we have

$$||x||_{1,\infty} = \inf\{\lambda > 0 \,|\, x \in \lambda \cdot B_\mu\} = \frac{1}{4} \cdot ||x||_1 + \frac{1}{4} \cdot ||x||_\infty$$

for all $x \in \mathbb{R}^2$. Figure 3 shows the unit ball of the one-infinity-norm.

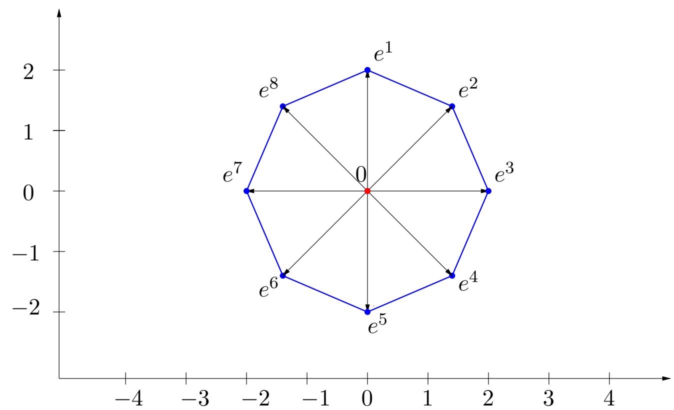


Figure 3: Unit ball of the one-infinity-norm on $\mathbb{R}^2$.

More general one can show that each norm $||\cdot|| : \mathbb{R}^2 \to \mathbb{R}$ defines a gauge μ using the corresponding unit ball $B_\mu := \{x \in \mathbb{R}^2 \,|\, ||x|| \leq 1\}$ of the norm.

Definition 4. *Let μ be a gauge with unit ball $B_\mu \subseteq \mathbb{R}^2$. The dual gauge $\mu^* : \mathbb{R}^2 \to \mathbb{R}$ of the gauge μ is defined by*

$$\mu^*(x) := \sup\{\langle y, x\rangle \,|\, y \in B_\mu\}$$

for all $x \in \mathbb{R}^2$, where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product.

Remark 2. *The following useful properties hold:*

($DG1$)*: The dual gauge of μ^* is the gauge μ itself.*

($DG2$)*: If μ is a polyhedral gauge then so is the dual gauge μ^*.*

($DG3$)*: Let μ be a polyhedral gauge. The unit balls of μ and μ^* (polytopes in $\mathbb{R}^2$) have the same number of extreme points.*

($DG4$)*: If μ is a norm then so is the dual gauge μ^*.*

1.2.3 Real world applications

The Euclidean metric is appropriate to model, for instance, the propagation of waves because one measures using the direct "line of sight" distance.

The Manhattan metric measures the distance between two points as the sum of the absolute differences of the single coordinates and is appropriate to model urban distances with mainly rectangular street profiles. For instance, the metric is appropriate to model distances in the borough of Manhattan in New York City and in fact, this is what grants it the name "Manhattan metric".

The maximum metric is useful as distance-function, if the motion takes place simultaneously in both directions and only the larger one of the distances determines how long the motion takes.

For more information about the estimation of travel distances, see [7] and the references therein.

1.3 Choice of the real world location coordinates

One way to solve real world location problems in a practical way (especially useful in small regions like cities) is through the loading of actual map images in the background of a coordinate system. Then the decision maker has only to specify the existing location points on the map to solve their preferred location problem with the Software FLO. In Figure 4, an example of a real world location problem is shown.
Another way is to use a rectangular coordinate system to give locations on the surface of the Earth. The most important 2-dimensional Cartesian coordinate systems in Europe are the Gauß-Krüger coordinate system and the Universal Transverse Mercator (UTM) coordinate system (available worldwide). With the help of one of the above coordinate systems, it is possible to overlay small regions of the Earth with a rectangular coordinate system. Instead of using angular measurements for the specification of the coordinates, the above mentioned rectangular coordinates are given in meters. In most cases, the coordinates are given as northing and easting values.

If a region is larger in comparison to a UTM grid zone, then the coordinate

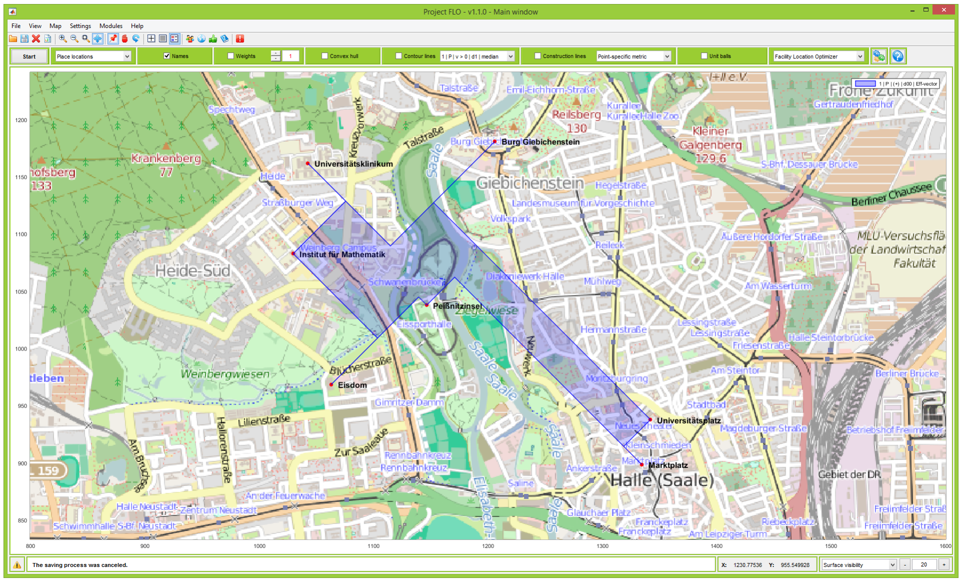


Figure 4: Screenshot of FLO with loaded a real map of the city Halle (Saale).

system of one UTM grid zone could be used across the boundary, provided that the increasing distortion allows for meaningful use.

Areas of application of the UTM grid:

- Geographic maps,
- Military, disaster control, firefighters, rescue services, police and other organizations,
- Surveying.

The decision maker can now use UTM coordinates for specifying the coordinates of the location points in FLO to solve their preferred location problem. There are other tools available:

http://www.thekompf.com/trekka/geoposition.php
http://www.deine-berge.de/Rechner/Koordinaten/Halle--Saale-,-Deutschland

to specify the UTM coordinates for real world locational data. In this case these tools can also be used for determining the real world location data of the solutions computed by FLO.

1.4 Classification of Location Problems

For identifying location problems, we use a classification scheme proposed in the literature of location theory by Hamacher and Nickel [26]. The classification contains five positions

$$\text{Pos 1.} \mid \text{Pos 2.} \mid \text{Pos 3.} \mid \text{Pos 4.} \mid \text{Pos 5.} \quad ,$$

where:

- Pos 1. : Number of new facilities (e.g., 1 for single-facility location problems),
- Pos 2. : Type of location problem (e.g., planar (P), discrete (D) or network location problem (N)),
- Pos 3. : Features of the location problem (e.g., positive weights, $v > 0$, i.e., $v_i > 0$ for all $i = 1, \ldots, m$; weights equal to one, $v = 1$, i.e., $v_i = 1$ for all $i = 1, \ldots, m$; negative weights, $w < 0$, i.e., $w_i < 0$ for all $i = 1, \ldots, m$; only attraction points, $(+)$; attraction and repulsion points, $(+, -)$; type of the feasible set, for instance $X = \mathbb{R}^2$ or X represents a polytope),
- Pos 4. : Definition of the distances (e.g., Manhattan metric, d_1; Maximum metric, d_∞; Euclidean metric, d_2; squared Euclidean metric, d_2^2; l_p metric, d_p; polyhedral gauge, μ; mixed polyhedral gauges, μ_i; attraction metric d and repulsion metric induced by gauge μ, (d, μ)),
- Pos 5. : Linkage of individual distances (e.g., Median problem (median), Center problem (center), Vector problem (sEff -vector, Eff -vector or wEff -vector).

1.5 Mathematical notions and concepts

In this section we introduce some mathematical notions and concepts which are important for understanding the use of the Software FLO.

1.5.1 Convex sets

At first, we introduce convex sets, which are especially important in optimization theory. We call a set $X \subseteq \mathbb{R}^2$ convex, if for all $x, y \in X$ the inclusion

$$[x, y] := \{\lambda x + (1 - \lambda)y \mid 0 \leq \lambda \leq 1\} \subseteq X$$

holds, i.e., the whole line segment $[x, y]$ between the points x and y is contained in the set X.

1.5.2 Convex optimization problems

Consider a objective function $h : \mathbb{R}^n \to \mathbb{R}$ and a feasible set $X \subseteq \mathbb{R}^n$. If h is convex on a convex set X, then the problem

$$h(x) \to \min_{x \in X}$$

is called a convex problem.

1.5.3 Convex hull of existing points

The convex hull of the set of existing facilities A is defined by

$$\mathrm{conv}(A) := \bigcap_{\substack{A \subseteq X \subseteq \mathbb{R}^2 \\ X \text{ convex}}} X,$$

i.e, conv(A) is the average over all convex upper sets of A. Note that conv(A) is the smallest convex set which contains the set A.

It is known that a solution of the location problem (1) involving the l_p norm $(1 < p < \infty)$

$$\sum_{i=1}^{m} v_i \cdot ||x - a^i||_p \to \min_{x \in \mathbb{R}^2}$$

is contained in conv(A) (see Juel and Love [30]).

1.5.4 Level lines

Let $X \subseteq \mathbb{R}^n$ be a nonemty set. A level line (also called level curve or contour line) of a real-valued function $h : X \to \mathbb{R}$ is a curve along which the function has a constant value. The contour line with regard to the level $z \in \mathbb{R}$ is defined by

$$L_=(X, h, z) := \{x \in X \mid h(x) = z\}.$$

Assume $x^* \in X$ is a solution of the problem $\min_{x \in X} h(x)$ and $z^* := h(x^*)$. Then we have

$$L_=(X, h, z^*) = \operatorname*{argmin}_{x \in X} h(x).$$

1.6 Multiobjective Optimization

In multiobjective optimization, one investigates optimization problems with a vector-valued objective function. Fundamental works concerning multiobjective optimization date back to F.Y. Edgeworth [11] (1881) and V. Pareto [44] (1896). Some recommended books concerning the topic of multiobjective optimization are books by Ehrgott [12], Eichfelder [13, 14], Göpfert *et al.* [18], Jahn [28], Khan *et al.* [32] and Löhne [34].

1.6.1 Problem formulation and solution concepts

Let us consider a multiobjective optimization problem defined by functions $f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ and a nonempty feasible set $X \subseteq \mathbb{R}^n$:

$$f(x) := \begin{pmatrix} f_1(x) \\ \ldots \\ f_m(x) \end{pmatrix} \to \operatorname*{v\text{-}min}_{x \in X}.$$

Throughout we define the set of all indices of the components of f by

$$I_m := \{1, \ldots, m\}.$$

We call a point $\tilde{x} \in X$ a Edgeworth-Pareto efficient solution (EP-efficient), if there is no $x \in X$ such that $f_i(x) \leq f_i(\tilde{x})$ for all $i \in I_m$ and $f_j(x) < f_j(\tilde{x})$ for some $j \in I_m$. Thus, the set of all EP-efficient solutions is given by

$$\operatorname{Eff}(X \mid f) = \{x^0 \in \mathbb{R}^n \mid \nexists\, x \in X : \ f(x) \in f(x^0) - \mathbb{R}^m_+ \setminus \{0\}\}.$$

In the following, we introduce a weaker concept. We call a point $\tilde{x} \in X$ a weakly Edgeworth-Pareto efficient solution (weakly EP-efficient) if there is no $x \in X$ such that $f_i(x) < f_i(\tilde{x})$ for all $i \in I_m$. Thus, the set of all weakly EP-efficient solutions is given by

$$\mathrm{wEff}(X \mid f) = \{x^0 \in \mathbb{R}^n \mid \nexists\, x \in X : \; f(x) \in f(x^0) - \operatorname{int} \mathbb{R}^m_+\}.$$

A stronger concept in comparison with the concept of Edgeworth-Pareto efficiency is the concept of strict Edgeworth-Pareto efficiency. We call a point $\tilde{x} \in X$ a strictly Edgeworth-Pareto efficient solution (strictly EP-efficient), if there is no $x \in X \setminus \{\tilde{x}\}$ such that $f_i(x) \leq f_i(\tilde{x})$ for all $i \in I_m$. Thus, the set of all strictly EP-efficient solutions is given by

$$\mathrm{sEff}(X \mid f) = \{x^0 \in \mathrm{Eff}(X \mid f) \mid |\{x \in X \mid f(x) = f(x^0)\}| = 1\}.$$

It can easily be seen that we have the inclusions

$$\mathrm{sEff}(X \mid f) \subseteq \mathrm{Eff}(X \mid f) \subseteq \mathrm{wEff}(X \mid f).$$

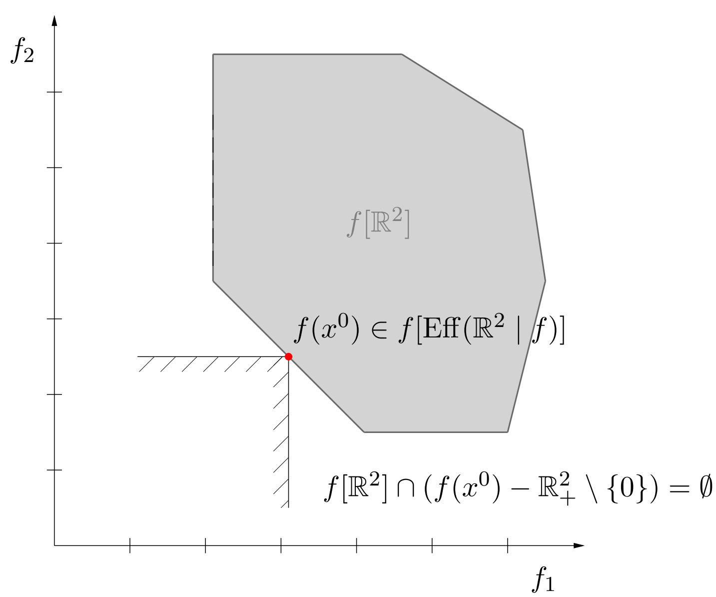


Figure 5: x^0 is a Edgeworth-Pareto efficient solution.

1.6.2 Geometric interpretation

Using level lines and level sets, we obtain useful geometric characterizations of strictly EP-efficient, EP-efficient and weakly EP-efficient solutions.

Let $\tilde{x} \in X$. Then the following hold (see, e.g., [12, Theorem 2.30]):

$$\begin{aligned}
\tilde{x} \in \mathrm{sEff}(X \mid f) \quad &\Longleftrightarrow \quad \bigcap_{i \in I_m} L_{\leq}(X, f_i, f_i(\tilde{x})) = \{\tilde{x}\}, \\
\tilde{x} \in \mathrm{Eff}(X \mid f) \quad &\Longleftrightarrow \quad \bigcap_{i \in I_m} L_{\leq}(X, f_i, f_i(\tilde{x})) = \bigcap_{i \in I_m} L_{=}(X, f_i, f_i(\tilde{x})), \\
\tilde{x} \in \mathrm{wEff}(X \mid f) \quad &\Longleftrightarrow \quad \bigcap_{i \in I_m} L_{<}(X, f_i, f_i(\tilde{x})) = \emptyset.
\end{aligned}$$

Example 5. *We consider three component functions $f_1, f_2, f_3 : \mathbb{R}^2 \to \mathbb{R}$ defined by $f_i(x) := ||x - a^i||_1$ for all $x \in \mathbb{R}^2$ and all $i = 1, 2, 3$, where a^1, a^2, a^3 are three points like given in Figure 6. In the left part of Fig. 6 are the level lines of f_1 and f_2 at the point $\tilde{x} \in \mathbb{R}^2$ visualized. Due to the above geometric characterizations, we obtain*

$$\tilde{x} \in \mathrm{Eff}(\mathbb{R}^2 \mid [f_1, f_2]) \setminus \mathrm{sEff}(\mathbb{R}^2 \mid [f_1, f_2]).$$

In the right part of Fig. 6 it is shown that, by extension with one additional function f_3, we obtain that $\tilde{x}$ is no longer an EP-efficient solution for the problem concerning the objective function $[f_1, f_2, f_3]$. Note, however, that

$$\tilde{x} \in \mathrm{wEff}(\mathbb{R}^2 \mid [f_1, f_2, f_3]) \setminus \mathrm{Eff}(\mathbb{R}^2 \mid [f_1, f_2, f_3])$$

still holds.

1.6.3 Multiobjective Linear Programs

For multiobjective linear programs, a free solver (BENSOLVE) developed by Löhne and Weißing [35], is available. Further information can be found at

http://www.bensolve.org.

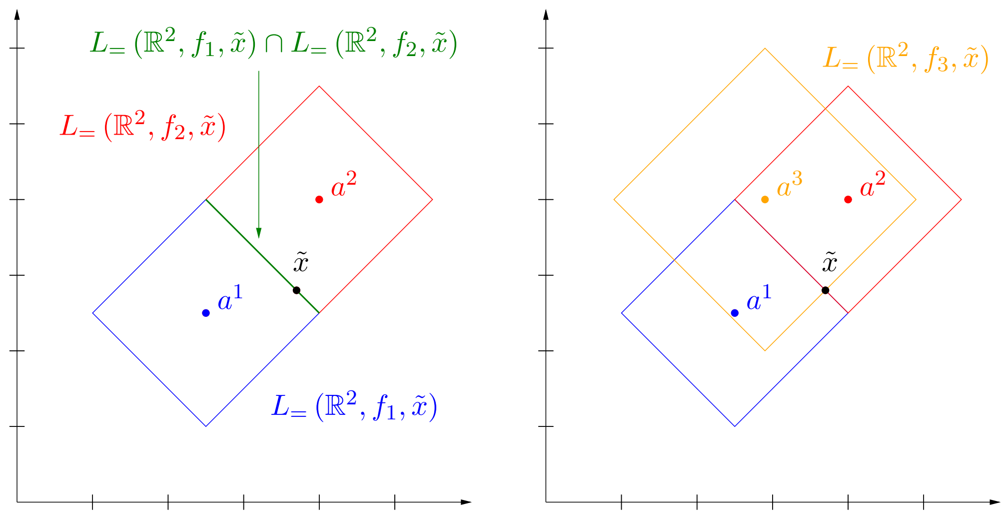


Figure 6: Geometric characterization of (weakly) EP-efficient solutions.

1.6.4 An application: A location-routing problem in touristy travelling management

A tourist is interested in visiting a certain region and he is looking for a hotel in this region from where he will visit points of interest $a^1, ..., a^m$ on several days. The time for the sightseeing tour on each day is restricted. After a certain time (given by time windows) the tourist goes back to the hotel on each day. On the next day the tourist starts again from the hotel. For each route one time window restricts the arrival time of the tourist in the hotel. Furthermore, other time windows are given concerning the opening hours of the points of interest. For this problem a multiobjective approach is very useful because in this concrete application in touristy routing planning the choice of the hotel as well as a route for the sightseeing tour is of interest.

For the formulation of the mathematical model of this problem the following points are important:

1. The tourist wants to find a suitable hotel from where he starts his sightseeing routes to the points of interest $a^1, ..., a^m$ on several days taking into account the distances to $a^1, ..., a^m$. On the one hand it is a user demand in touristy traveling management to restrict the set of available hotels and on the other hand it is important for the tourist

to get a decision support, not a fixed solution, i.e., to get a set of alternatives for the hotel.

So we formulate the problem as multiobjective location problem in order to get a set of alternatives for the hotel. This approach gives the decision maker a preselection of hotels and the tourist can chose his own preferences concerning the hotel (taking into account other criteria like quiet position, connections to further cultural centre, pricing, equipment or beautiful situated).

Moreover, the tourist wants to find a hotel as starting point for the sightseeing route on each day with a short length of the route under the given time windows.

2. Furthermore, we take the length of the route as well as the time for the route into consideration. These are two different criteria because the time for the route includes travelling time and waiting time, i.e., the time for the route is not proportional to the length of the route. A third criteria for the routing problem is a penalty function concerning the violation of time windows.

We consider as a subproblem a continuous multiobjective location problem in order to support the solution of the discrete multiobjective location problem for determining alternatives (hotels) as starting points for the multiobjective routing problem. We compute the whole set of EP-efficient solutions of the continuous multiobjective location problem. Then we select some alternatives (hotels) in this set as starting points for the routing problem. This will be done in the concrete application in touristy travelling management by computing the intersection of this set of solutions of the continuous multiobjective location problem with the set of available hotels. The reason for this approach is that we have an effective algorithm for solving the continuous multiobjective location problem and any point (hotel) belonging to the solution set of the multiobjective continuous problem is also a solution of the multiobjective discrete problem. This method is very useful from the numerical as well as from the practical point of view. With this approach it is possible to use the effective methods from continuous locational analysis in order to support the solution procedure of the discrete location-routing problem. Furthermore, the interests of the tourist in the concrete application are taken into consideration.

In order to formulate the multiobjective location-routing problem we are using the concept of Edgeworth-Pareto efficient solutions given in Section

1.6.1. We introduce the multiobjective location-routing problem to find a new location $x \in \mathbb{R}^2$ such that the distances $d(x, a^i)$ between m given facilities $a^i \in \mathbb{R}^2$ $(i \in I_m)$ and x as well as the length $L(x, y)$ of the routes with the starting point x to the given points a^i and the time $T(x, y)$ for the route and a penalty function $E(x, y)$ concerning the violation of time windows, where y is the decision variable defining the routes, are to be minimized in the sense of multiobjective optimization taking into account time windows:

$$(P_{LR}): \qquad F(x,y) := \begin{pmatrix} d(x,a^1) \\ \cdots \\ d(x,a^m) \\ L(x,y) \\ T(x,y) \\ E(x,y) \end{pmatrix} \longrightarrow \text{v-min},$$

where $x,\ a^i \in \mathbb{R}^2$, $(i \in I_m)$, $d(., a^i) : \mathbb{R}^2 \to \mathbb{R}_+$ is a distance function and $y = (y_{rsk})$, $y_{rsk} \in Y := \{0, 1\}$ $(r = 0, 1, ..., m;\ s = 1, ..., m + 1)$, $k = 1, ..., t$, are decision variables defining the routes.

The facilities for the routing problem are $x, a^1, ..., a^m$, where a^i $(i \in I_m)$ are the given facilities (points of interest). The starting point for each tour is the hotel x (facility 0). From x (facility 0) the tourist goes to a^{j_1} (facility j_1), from there to a^{j_2} (facility j_2) and so on. After a certain time given by the time window the tourist goes back to the hotel x. The decision variables define the routes for sightseeing, i.e., it holds $y_{rsk} = 1$ if in a route the facility s follows the facility r on day k, otherwise there is $y_{rsk} = 0$.

In the multiobjective approach we combine the continuous multiobjective location problem

$$(P_L): \qquad \begin{pmatrix} d(x,a^1) \\ \cdots \\ d(x,a^m) \end{pmatrix} \to \text{v-min}$$

with the routing problem (taking into account time windows)

$$(P_R): \qquad \begin{pmatrix} L(x,y) \\ T(x,y) \\ E(x,y) \end{pmatrix} \to \text{v-min}.$$

The decision maker has the possibility to describe the distance functions in the formulation of the location problem (P_L) by a norm $|| \cdot || : \mathbb{R}^2 \to \mathbb{R}$.

The distances (norms) between the new facility $x \in \mathbb{R}^2$ and the given facilities $a^i \in \mathbb{R}^2, i \in I_m$, can be chosen in different ways (see Section 1.2.3).

The location problem is formulated including the maximum norm or the Manhattan norm. This is motivated taking into account the following arguments:

- In many applications in locational analysis the road system is related to the Manhattan norm or to the maximum norm.
- It is possible to use the maximum norm as approximation for a l_p norm because of the well-known property

$$\lim_{p\to\infty}(\sum_{i=1}^{n}|x_i|^p)^{\frac{1}{p}} = \max\{|x_1|, ..., |x_n|\}$$

  for all $x := (x_1, \dots, x_n) \in \mathbb{R}^n$.

- There is an effective algorithm for computing the whole set of EP-efficient solutions of the multiobjective location problems involving the Manhattan or the maximum norm, respectively.

Using the maximum norm $\|\cdot\|_\infty$ we can formulate the location problem (P_L) as the problem

$$\begin{pmatrix} \|x - a^1\|_\infty \\ \cdots \\ \|x - a^n\|_\infty \end{pmatrix} \to \text{v-min}_{x\in\mathbb{R}^2}.$$

2 Facility Location Optimizer (FLO)

The first version of the Software FLO (version 1.0.0) was released on April 22, 2015. In this section we present some general information about the Software FLO.

2.1 About the logo of Project FLO

Figure 7 shows Project FLO's logo, which was created by Christian Günther.

Figure 7: Logo of Project FLO.

"FLO"is an acronym of the full name "Facility Location Optimizer", the software developed in our project. The three ellipses around the word FLO symbolize level curves (also called level lines, contour lines or isolines) of an objective function concerning a location problem in the plane. Notice that a level curve is a curve along which the objective function has a constant value. The red point in the middle of the letter "O" symbolizes the optimal solution of the underlying location problem. Moreover the letter "O" defines a shifted (to the red point) unit ball of a gauge (a special distance function for measuring distances between points). The horizontal and vertical red line segments are axes in a Cartesian coordinate system and symbolize the behaviour of specifying location points in FLO.

2.2 About the development of FLO

The development of the software started in 2011 with the initiation of Christian Günther's Bachelor's thesis (see [20]) under the supervision of Prof. Dr. Christiane Tammer. During his master program, which included the completion of a Master's thesis (see [21]), the program continued to evolve under his active development. The continued development of the Software FLO is now the main part of Christian Günther's PhD project. Additionally, since 2014, Marcus Hillmann and Brian Winkler are involved in the development of FLO as part of their own PhD projects.

2.3 Program features of FLO

The current version 1.2.2 of FLO provides the following features:

- Solving of planar single objective location problems (median- and center problems) from the literature of location theory (e.g. the Fermat-Weber problem) taking into account different concepts for distances (norms and polyhedral gauges) and some types of restrictions.
- Solving planar multiobjective location problems with respect to different solution concepts of the theory of multiobjective optimization (e.g. concept of Edgeworth-Pareto efficiency).
- Solving special classes of non-convex single- and multiobjective location problems with attraction and repulsion points.
- Overview of the solution of different location problems on a map, where solutions can be identified through colors and a classification scheme for location problems proposed in the literature by Hamacher and Nickel.
- Detailed information about the output of all algorithms.
- Algorithm settings can easily be changed by the user along with detailed information about the algorithm, all directly from within FLO.
- Modern concepts of measuring distances between location points (for approximating real word distances) can be used through the definition of polyhedral gauges.
- Loading of actual map images in the background of the coordinate system.
- Location list can be exported to a spreadsheet, and a spreadsheet location list can be imported into FLO.
- Customization of the interface and the ability to save settings between sessions.

- Available in both German and English.

2.4 Computer Requirements

- FLO was developed to run primarily on Windows-based machines (Windows 7 or newer) and works best with large screen sizes and high resolutions (e.g. 1980 x 1024 pixels).
- FLO requires MATLAB 2011 or newer.
- The MATLAB app install file of FLO should work under MATLAB 2012b or newer.
- FLO should also run under MATLAB 2011 or newer on Linux and Mac systems but this is not officially supported.

2.5 Version notes

Previous releases and change logs:

Version 1.2.2 (packaged 12/02/2016)

- Revision of the main window map modes add and edit
- Revision of the import / export function concerning the list of location points.
- FLO solves the problem "1 | P | (+) | μ | wEff-vector", where μ is given by a special type of block norm.
- Revision of the Optimization Panel:
  - The optimization results can be exported to a spreadsheet or text file.
  - The running times of the algorithms can be visualized in a bar chart.
  - The solution sets of multiobjective location problems will be displayed in the list of optimization results.

- FLO can compute the dual gauge (see Definition 4) of a polyhedral gauge.
- Improvements for the use of FLO with Linux operating system (tested with Linux Mint 17.3).
- Some other improvements and bug fixes.

Version 1.2.1 (packaged 23/01/2016)

- Some improvements and bug fixes.

Version 1.2.0 (packaged 08/01/2016)

- New FLO panel “Restrictions”.
- FLO solves special classes of scalar location problems with some types of restrictions.
- FLO solves special classes of multiobjective location problems involving block norms.
- Some improvements and bug fixes.
- The documentation of FLO was updated with major changes.

Version 1.1.1 (packaged 11/09/2015)

- Added support for MATLAB release 2015b.
- Some improvements and bug fixes.
- New sidebar item: ”Snap to the grid” (see Section 4.1.4).
- New behaviour for adding, editing and moving points on the map (add mode and edit mode; see Section 4.1.3).
- The documentation of FLO was updated with some minor changes.

Version 1.1.0 (packaged 04/05/2015)

- Added support for MATLAB releases 2014b and 2015a.
- Some improvements and bug fixes.
- Added item “Changelog” in FLO’s help menu.

- The MATLAB app install file of FLO should work under MATLAB 2012b or newer.
- The documentation of FLO was updated with some minor changes.

Version 1.0.0 (packaged 22/04/2015)

Version 1.0.0 was developed and written by Christian Günther with the help of Marcus Hillmann and Brian Winkler. Moreover, the development of this version of the Software FLO benefited from the inspiration of Prof. Dr. Christiane Tammer, Marcus Hillmann and Brian Winkler.

- First published version of FLO.
- No support for MATLAB releases 2014b and 2015a.

2.6 Future development directions

Research topics:

- Location Theory,
- Vector Optimization,
- Uncertain Optimization (Robust and Stochastic Optimization).

Research directions of Project FLO:

- Single as well as multiobjective location problems involving constraints,
- Non-convex location problems,
- Multi-facility location problems,
- Extended multiobjective location problems,
- Location problems with uncertainties in the data,
- Approximation problems.

2.7 Related publications

Research articles of contributors of Project FLO, which are connected with the development of the Software FLO:

(*A*) C. Günther and Chr. Tammer. *Relationships between constrained and unconstrained multi-objective optimization and application in location theory.* Preprint Optimization-Online, http://www.optimization-online.org/DB_HTML/2015/11/5196.html, 2015 (submitted).

(*B*) S. Alzorba, C. Günther, N. Popovici and Chr. Tammer. *A new algorithm for solving planar multiobjective location problems involving the Manhattan norm.* Preprint Optimization-Online, http://www.optimization-online.org/DB_HTML/2016/01/5305.html, 2015 (submitted).

(*C*) A. Wagner, J. E. Martinez-Legaz and Chr. Tammer. *Locating a Semi-Obnoxious Facility - A Toland-Singer Duality Based Approach.* Journal of Convex Analysis, 23(4), 2016.

(*D*) S. Alzorba, C. Günther and N. Popovici. *A special class of extended multicriteria location problems.* Optimization, 64(5):1305-1320, 2015 (DOI: 10.1080/02331934.2013.869810).

(*E*) M. Hillmann. *Lagrange-Multiplikatoren-Regeln und Algorithmen für nichtkonvexe Standortprobleme.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

(*F*) C. Günther. *Dekomposition mehrkriterieller Optimierungsprobleme und Anwendung bei nichtkonvexen Standortproblemen.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

(*G*) S. Alzorba and C. Günther. *Algorithms for multicriteria location problems.* Numerical Analysis and Applied Mathematics ICNAAM, AIP Conference Proceedings, 1479:2286-2289, 2012 (DOI: 10.1063/1.4756650).

(*H*) C. Günther. *Standort-Medianprobleme mit variablen Anlagen.* Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2011.

Books of contributors of Project FLO:

(*A*) A. A. Khan, Chr. Tammer and C. Zălinescu. *Set-valued Optimization: An Introduction with Applications.* Springer, 2015.

(*B*) A. Göpfert, T. Riedrich and Chr. Tammer. *Angewandte Funktionalanalysis Motivationen und Methoden für Mathematiker und Wirtschaftswissenschaftler.* Vieweg+Teubner, Wiesbaden, 2009.

(*C*) H. W. Hamacher, K. Klamroth and Chr. Tammer. *Standortoptimierung.* In: B. Luderer (Ed.). *Die Kunst des Modellierens. Mathematisch-ökonomische Modelle.* Teubner-Verlag, 139-156, 2008.

(*D*) A. Göpfert, H. Riahi, Chr. Tammer and C. Zălinescu. *Variational Methods in Partially Ordered Spaces.* CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, 17, Springer-Verlag, New York, 2003.

3 Installation of the Software FLO

This section contains information concerning the license conditions and the installation procedure of the Software FLO.

3.1 License Conditions

The Software FLO is provided AS IS. We take no responsibility for damages, problems etc. resulting from use of this program and we also provide no warranty for bug-free operation, fitness for a particular purpose, or the appropriate behavior of the program.

Please see the complete license agreement at the beginning of this manual for more information.

You are free to make copies and run as many instances as required for your personal use, but no subsequent distribution of this software is allowed.

An "appropriate citation" for use in scientific or other works is:

C. Günther, M. Hillmann, Chr. Tammer and B. Winkler: Facility Location Optimizer (FLO) - A tool for solving location problems, www.project-flo.de

By downloading, accessing or otherwise using the Software you indicate your agreement to be bound by the License terms of Project FLO.

3.2 Installation

There are two possibilities for installing the Software FLO:

1. Installing the Software FLO as a MATLAB application:
   You can use the .mlappinstall file to install FLO as a MATLAB application. Detailed information about the installation process in MATLAB can be found in the MATLAB documentation:

   http://de.mathworks.com/help/matlab/creating_guis/install-and-run-app.html

2. Using the FLO folder system:
   Copy the folder FLO on your computer and open the path to the folder in MATLAB. You can start the Software FLO by typing

   flo_project

   in the command window of MATLAB.

4 Using the Software FLO

In this section we present detailed information about the structure and use of the Software FLO. Of course, please feel free to contact Project FLO with any questions or issues regarding the use of the software.

After installing and starting the software, you will be presented with Project FLO's License terms. Only after agreeing to the License terms (pressing the button "I agree" in the disclaimer window; see Figure 8), will you have the right to use the Software FLO.

Figure 8: Disclaimer window of FLO.

Using FLO for the first time, there are two windows: the main window and the module "Facility Location Optimizer" on the right side of the screen for interacting with the software available.

In Figure 9 you can see a screenshot FLO in action.

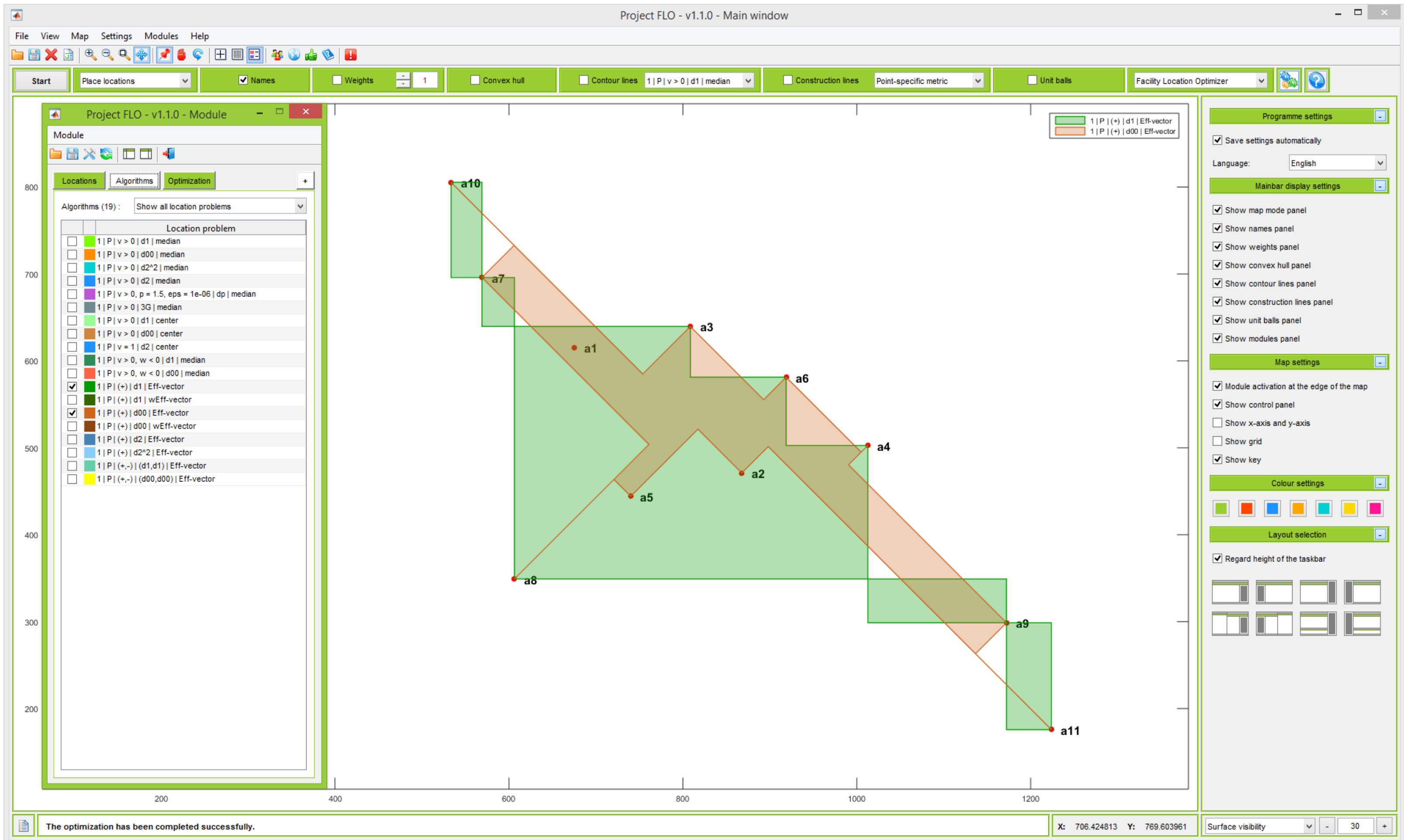


Figure 9: Screenshot of the Software FLO.

4.1 Main window

In FLO's main window, you can load and save workspaces, interact with the graphical plot and data, and change program-specific settings for visualization and optimization.

The main window of FLO includes the following areas:

- Menu
- Toolbar
- Mainbar
- Plot
- Settings Sidebar (if activated)
- Footbar

Figure 10 shows the main window of FLO.

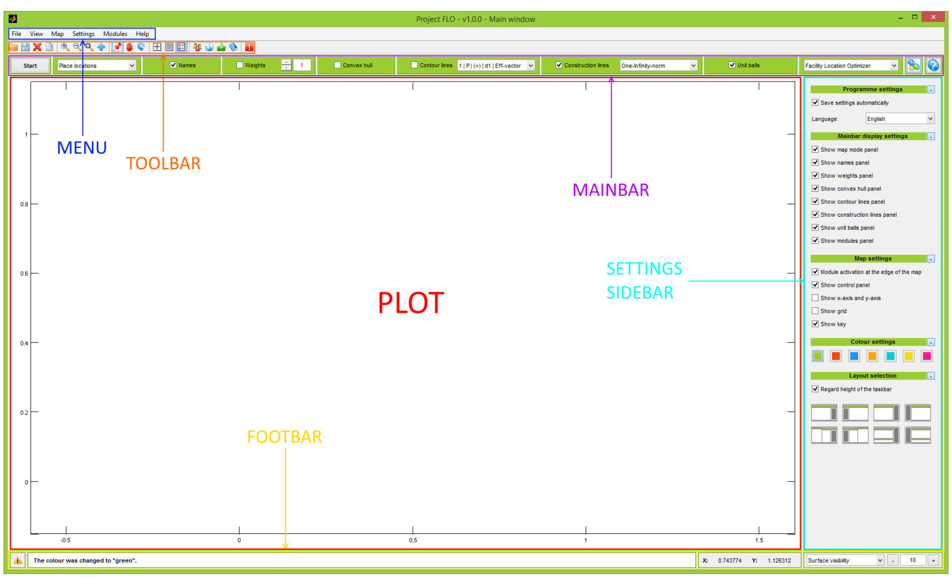


Figure 10: Main window of FLO.

4.1.1 Menu

The main window's menu allows you to change program-specific settings including special settings for visualization and optimization, to load workspaces or maps from external files, and access the help section.

Figure 11: Menu of FLO's main window.

The menu of FLO (see Figure 11) contains six parts:

1. **File**

   Load workspace
   (Load a workspace from a saved MAT-file)

   Save workspace as
   (Save a workspace to a MAT-file. A workspace contains the program-specific settings and settings concerning the module of FLO.)

   Save workspace
   (Save the currently loaded workspace to a MAT-file)

   Delete map data
   (Delete the current data on the map)

   Info
   (Get information about the currently loaded file)

   Exit
   (Close the program)

2. **View**

   Toolbars
   (Activation of specific panels of the main window of FLO)

      Mainbar
      (Activate specific panels of the Mainbar; see Section 4.1.3)

         Show map mode panel
         Show names panel
         Show weights panel
         Show convex hull panel
         Show contour lines panel
         Show construction lines panel
         Show unit balls panel
         Show restrictions panel
         Show modules panel

      Control panel
      (Activate the control panel including the control cross with zoom and drag options; see Section 4.1.4)

      Settings sidebar
      (Activate the settings sidebar; see Section 4.1.5)

   Layout default settings
   (Predefined positioning of the main window, module, and log window; see Section 4.1.5)

   Legend
   (Show the classification schemes in a legend on the top-right corner of the plot)

   Grid
   (Show the grid on the plot)

   Axes
   (Show X-axis and Y-axis on the plot)

Zoom in
(Zoom in on the map with zoom center in the middle of the coordinate system)

Zoom out
(Zoom out on the map with zoom center in the middle of the coordinate system)

Zoom on whole map
(Center all objects of the plot in the middle of the coordinate system)

Pan map section
(Activate the pan mode for moving map section)

Full screen
(Maximize the main window to full-screen size)

3. **Map**

   Load
   (Load real-world map from an image file)

   Export
   (Export the content of the current plot as a PDF-file; possible formats: $A0, \ldots, A6$)

   Delete
   (Remove the currently loaded map)

   Info
   (Get information about the currently loaded map)

4. **Settings**

   System options
   (Change general system-specific options)

   Language (German or English)
   Colour scheme (green, red, blue, orange, cyan, gold, pink)

Mathematical options
(Change plot-specific options; see Section 4.1.3)

Show names
Show weights
Show convex hull
Show construction lines
Show unit balls
Show contour lines
Show contour lines with values
Show restrictions

Reset program
(Reset FLO's program settings)

Note: The settings of the FLO module will only be reset if the module is open.

5. **Modules**

   Facility Location Optimizer
   (Open FLO's module for solving location problems)

6. **Help**

   Display manual
   (Show FLO's documentation)

   Open project website in browser
   (Open Project FLO's website)

   Send feedback
   (Contact Project FLO members)

   Download page of FLO
   (Get information about the current version of FLO)

   Changelog
   (Log of all the changes made to FLO)

Disclaimer
(Show the License agreement of Project FLO)

About the project
(Open a window with details about Project FLO)

4.1.2 Toolbar

You can also easily run many common program functions from the main window's toolbar (see Figure 12).

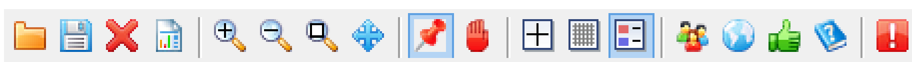


Figure 12: Toolbar of the FLO's main window.

The toolbar's nineteen functions are detailed below:

1. **Load workspace**
   (Load a workspace from a saved MAT-file)

2. **Save workspace**
   (Save a workspace in a MAT-file)

3. **Delete map data**
   (Delete the current data on the map)

4. **Open log window**
   (Open the log window for viewing the list of program messages)

5. **Zoom in**
   (Zoom in on the map with zoom center in the middle of the coordinate system)

6. **Zoom out**
   (Zoom out on the map with zoom center in the middle of the coordinate system)

7. **Zoom on whole map**
   (Center all plot objects in the middle of the coordinate system)

8. **Pan map section**
   (Activate the pan mode for moving the map section)
9. **Add mode**
   (Activate map mode for placing points on the map; see Section 4.1.3)
10. **Edit mode**
    (Activate map mode for editing / moving points on the map; see Section 4.1.3)
11. **Show axes**
    (Show X-axis and Y-axis on the plot)
12. **Show grid**
    (Show the grid on the plot)
13. **Show legend**
    (Show the classification schemes in a legend on the top-right corner of the plot)
14. **About the project**
    (Get details about Project FLO)
15. **Open project website in browser**
    (Open the Project FLO website)
16. **Send feedback**
    (Contact Project FLO members)
17. **Display manual**
    (Open FLO's documentation)
18. **Disclaimer**
    (View the Project FLO License agreement)

4.1.3 Mainbar

The mainbar provides several panels for changing visualization properties and mathematical options. Note that for changing properties of panels defined in the mainbar, it is necessary to load the "Facility Location Optimizer" module. This can be done via the menu option "Modules" or by using the module panel in the mainbar.

In the following, we present information about the panels contained in the mainbar of the main window (ordered from the left to the right):

1. **Start/Stop button**

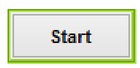


Figure 13: Start/Stop button of the mainbar.

By pressing the start button, the optimization process will begin execution. It is possible to stop the optimization process by clicking on the stop button, which will be displayed during the optimization.

2. **Map mode panel**

Figure 14: Map mode panel of the mainbar.

The list in the map mode panel is used to specify the map mode. The following modes are available:

- **Add mode**
  The user can place points on the map using the left mouse button.
- **Edit mode**
  The user can edit/move points on the map. If the mouse cursor is located in the near of location points, then one point with the smallest distance to the mouse cursor will be selected after single-clicking with the left mouse button on the map. By pressing of

the left mouse button and simultaneous moving of the cursor, the user can move existing points on the map.

For more information concerning the map modes, see Section 4.3.3 and Section 4.3.6.

3. **Names panel**

Figure 15: Names panel of the mainbar.

By activating this check box, the names of the location points will be displayed on the map next to each point.

4. **Weights panel**

Figure 16: Weights panel of the mainbar.

By activating this check box, the weight values will be displayed on the map next to the location points. Moreover, you can easily change the values of weights of the currently selected location points from within the weights panel of the mainbar.

5. **Convex hull panel**

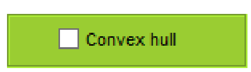


Figure 17: Weights panel of the mainbar.

By activating this check box, the convex hull of the existing location points will be displayed on the map.

6. **Contour lines panel**

Figure 18: Contour lines panel of the mainbar.

By activating this check box, the contour lines of the currently selected location problem's objective function will be displayed on the map.

7. **Construction lines panel**

Figure 19: Construction lines panel of the mainbar.

By activating this check box, the construction lines related to the given location points will be displayed on the map using the currently selected metric.

8. **Unit balls panel**

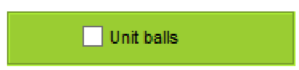


Figure 20: Unit balls panel of the mainbar.

By activating this check box, the unit balls related to the given location points will be displayed on the map using the currently selected metric in the construction lines panel.

9. **Restrictions panel**

Figure 21: Restrictions panel of the mainbar.

By activating this check box, the sets of restrictions will be displayed on the map.

10. **Modules panel**

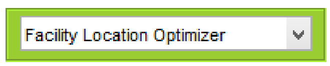


Figure 22: Modules panel of the mainbar.

This panel displays the currently active module name. You can also use this panel to deactivate the current module and activate another.

11. **Settings button**

Figure 23: Settings button of the mainbar.

Clicking on the settings button displays or hides the settings sidebar of the main window.

12. **Help button**

Clicking on the help button displays the FLO manual.

The visibility of the above mentioned panels 2 to 9 can be determined under the following menu path

View > Toolbars > Mainbar

or explicitly in the settings sidebar (see Section 4.1.5).

Figure 24: Help button of the mainbar.

4.1.4 Plot

The following describes several important components/features of the main window plot:

1. **Coordinate system**

   The main window of FLO provides a cartesian coordinate system, where the user can specify, edit, or move location points.

2. **Grid**

   A grid can be displayed on the plot with the grid's width defined in the footbar of FLO's main window (see Section 4.1.6).

3. **Axes**

   The X-axis and the Y-axis of the cartesian coordinate system can be explicitly displayed.

4. **Key**

   The classification schemes of location problems can be displayed (plot legend).

5. **Control panel**

   The control cross of the control panel contains zoom and drag options. In Figure 25, the control panel is highlighted using a large red rectangle.

6. **Activation of the module**

   FLO modules can be activated easily at the edge of the map. In Figure 25, the module activation areas are shown using slim blue rectangles at the edge of the map.

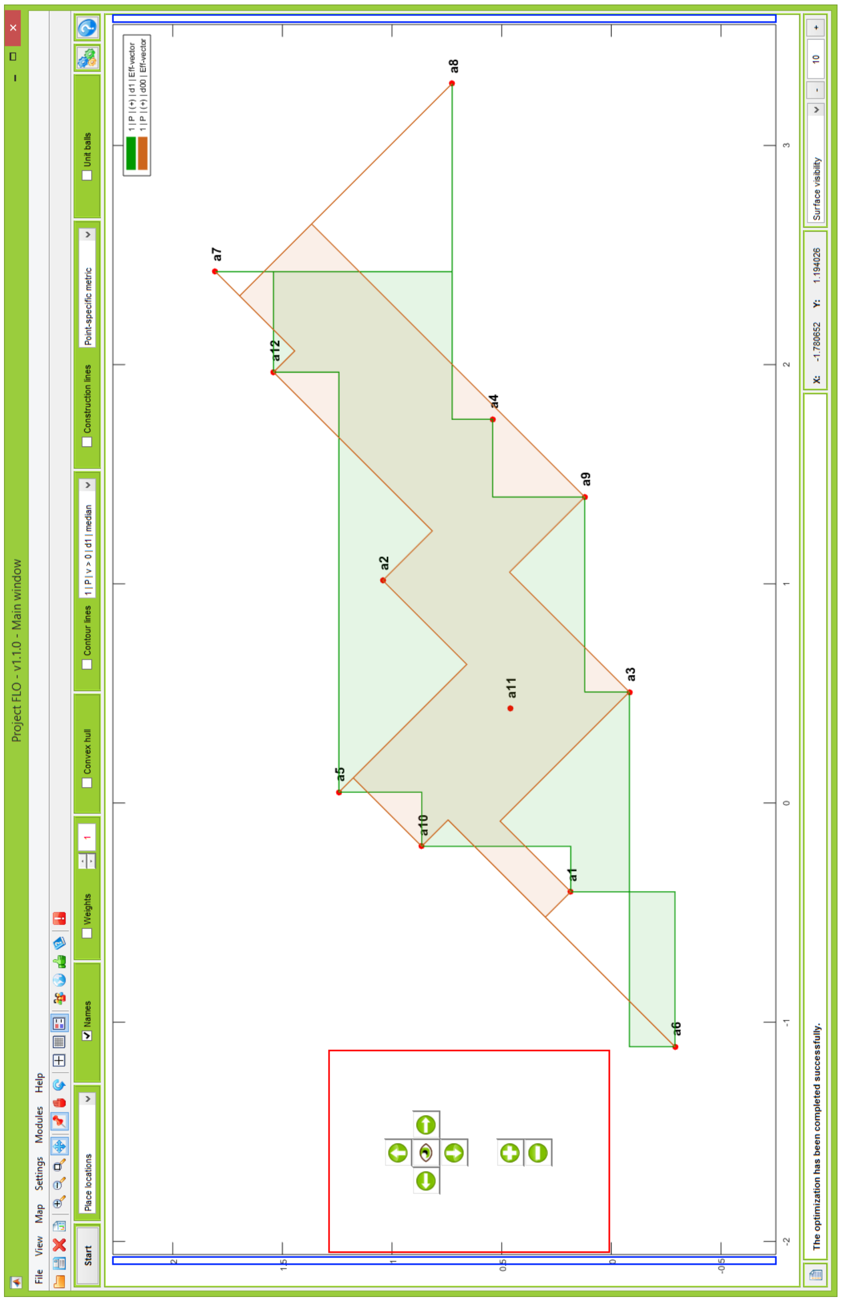


Figure 25: Plot with module activation panels at the edge of the map (blue rectangles) and activation area of the control cross (red rectangle).

7. **Popup menu**

   By right-clicking on the map, you can also access important program functions using a popup menu. The popup menu contains the following actions:

   Add mode
   (Activate map mode for placing points on the map; see Section 4.1.3)

   Edit mode
   (Activate map mode for editing / moving points on the map; see Section 4.1.3)

   Zoom in
   (Zoom in on the map with zoom center in the middle of the coordinate system)

   Zoom out
   (Zoom out on the map with zoom center in the middle of the coordinate system)

   Zoom on whole map
   (Center all objects of the plot in the middle of the coordinate system)

   Pan map section
   (Activate pan mode for moving the map section)

   Centre map here
   (Centre the current mouse-cursor point in the middle of the coordinate system)

   Load workspace...
   (Load a workspace from a saved MAT-file)

Save workspace...
(Save a workspace in a MAT-file)

Delete map data
(Delete the current data on the map)

Load map...
(Load a real-world map)

Export map...
(Export the content of the current plot as a PDF-file; possible formats: $A0, \ldots, A6$)

Delete map
(Remove the currently loaded map)

8. **Point-specific map zoom**

   By using the scroll wheel function of the computer mouse you can zoom in and out on specific cursor points on the map.

9. **Snap to the grid**

   The behaviour ”snap to the grid” for interaction with the plot can be activated in the settings sidebar (see Section 4.1.5). By clicking on the map the selection cross will be be fixed on a grid point with the smallest distance to the mouse cursor. For instance if you choose a value of one for the width of the grid in the footbar (see Section 4.1.6), then it is easy to create location points with integer coordinates on the map, provided the ”snap to the grid” checkbutton is activated in the settings sidebar.

4.1.5 Settings Sidebar

You can change program-specific settings in the settings sidebar which is activated using the settings button in the mainbar (see Section 4.1.3) or by using the menu (see Section 4.1.1).

Figure 26 shows the settings sidebar of FLO's main window.

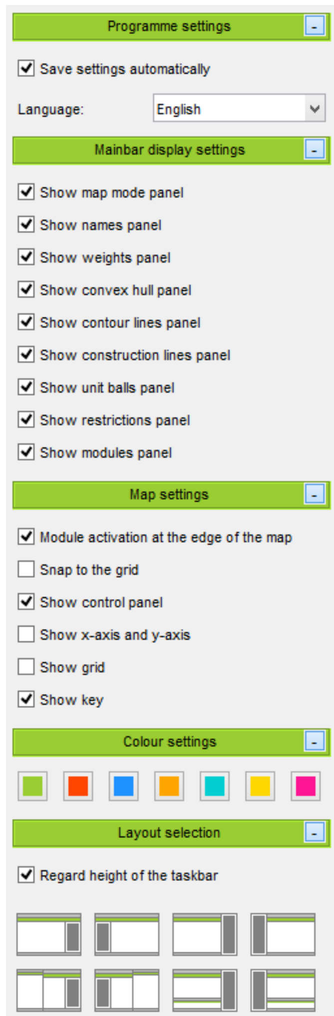


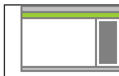
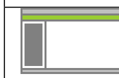
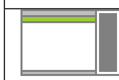
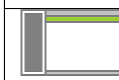
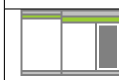
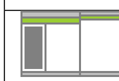
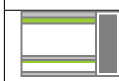
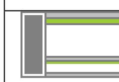
Figure 26: Settings sidebar of the main window of FLO.

The settings sidebar contains the following parts:

1. **Program settings**
   (Change general program-specific settings)
   - Saving settings automatically
   - Language (German or English)
2. **Mainbar display settings**
   (Change settings concerning the visibility of panels of the mainbar; see Section 4.1.3)
   - Show map mode panel
   - Show names panel
   - Show weights panel
   - Show convex hull panel
   - Show level lines panel
   - Show construction lines panel
   - Show unit balls panel
   - Show restrictions panel
   - Show module selection panel
3. **Map settings**
   (Change settings concerning the main window plot; see Section 4.1.4)
   - Activation of the module on the edge of the map
   - Snap to the grid
   - Show control panel
   - Show X-axis and Y-axis
   - Show grid
   - Show legend
4. **Colour settings**

   Select your preferred colour (green, red, blue, orange, cyan, gold, magenta) for using the Software FLO.
5. **Layout selection**

   Select one of six predefined layout settings for using the Software FLO.

	[Main window [Module]]
	[[Module] Main window]
	[Main window] [Module]
	[Module] [Main window]
	[Log window] [Main window [Module]]
	[[Module] Main window] [Log window]
	[[Main window] [Log window]] [Module]
	[Module] [[Main window] [Log window]]

4.1.6 Footbar

The footbar of the main window displays the most recent program execution message as well as the current coordinates of the mouse cursor. Additionally, the footbar can be used to change some settings concerning the display of objects on the plot.

Figure 27 shows the footbar of the main window of FLO.

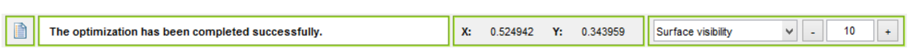


Figure 27: Footbar of the main window of FLO.

1. **Log status symbols**

   The log status symbol indicates warnings or errors which occurred during program execution.

The following symbols are used in the current version of FLO:

 normal: no warning or error occurred
 warning
 error

Clicking on the log status symbol in the footbar opens the log window of FLO (see Section 4.2).

Note: The log status symbol only changes when the log window is closed.

2. **Status message panel**

   The status message panel shows the most-recent program message. Previous messages can be found in the log window.

3. **Coordinates of the current point**

   The current coordinates (X and Y) of the cursor point are also displayed in the footbar of the main window.

4. **Settings panel**

   In addition, the footbar provides a settings panel for changing settings easily. The panel contains the following properties:

   - **Surface visibility**
     Change the degree of transparency for the displayed solution sets.
   - **Type size**
     Change the size of the text displayed in the plot.
   - **Number of contour lines**
     Specify the number of scalar objective function contour lines that FLO will display.
   - **Grid width**
     Change the width of the grid.

4.2 Log window

The log window contains all of FLO's previous program execution messages. Each row of the list in the log window (see Figure 28) gives the message's:

- date
- time
- type
- content

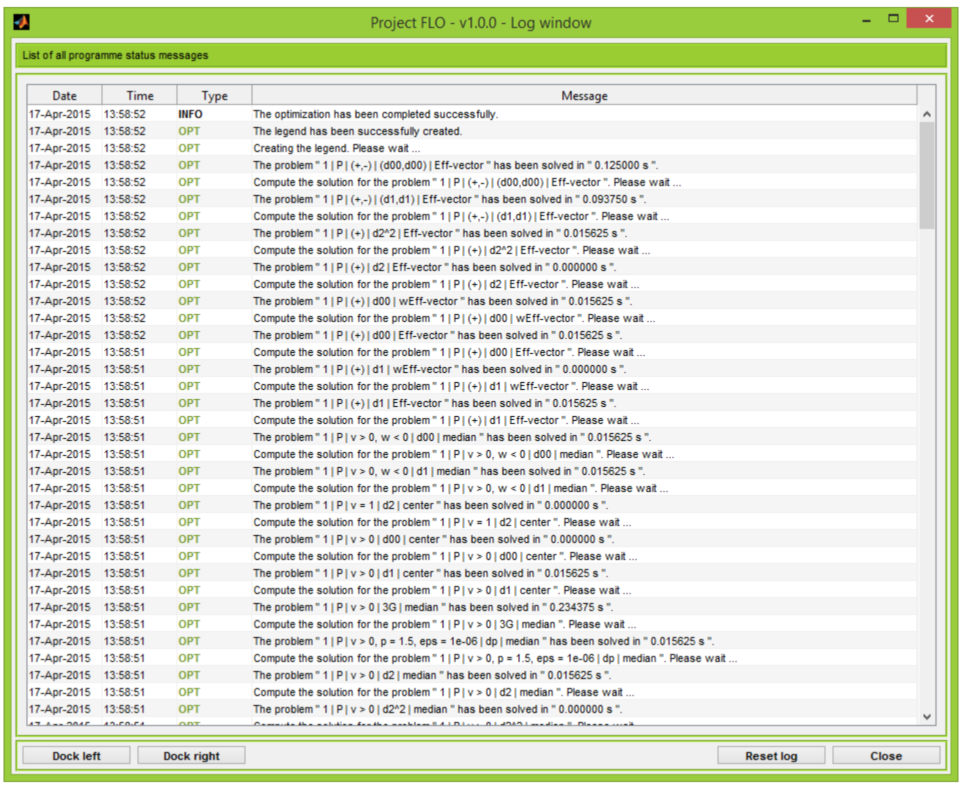


Figure 28: Screenshot of FLO's log window.

4.3 Module: Facility Location Optimizer

The FLO Module provides information about:

- the location points (Locations Panel),
- the location problem(s) and corresponding algorithms (Algorithms Panel),
- the output of the optimization (Optimization Panel),
- the restrictions (Restrictions Panel),
- the distance functions (Metrics Panel).

Figure 29 shows the FLO module in a normal-size view. The button on the top-right side of the module changes the module's view size. You can choose between two sizes:

- **Normal view**
  (normal width of the module; see Figure 29)
- **Extended view**
  (full width of the module; see Figure 32)

The button toggles between the extended view ("+") and the normal view ("–").

In addition to the four main panels (locations, algorithms, optimization and metrics), the module provides both a menu and a toolbar.

4.3.1 Menu

The module's menu is structured as follows:

- **Module**

  Activate module
  (Activate the FLO module)

  Normal / extended view of the module

(Change the view size of the module)

Close module
(Close the module)

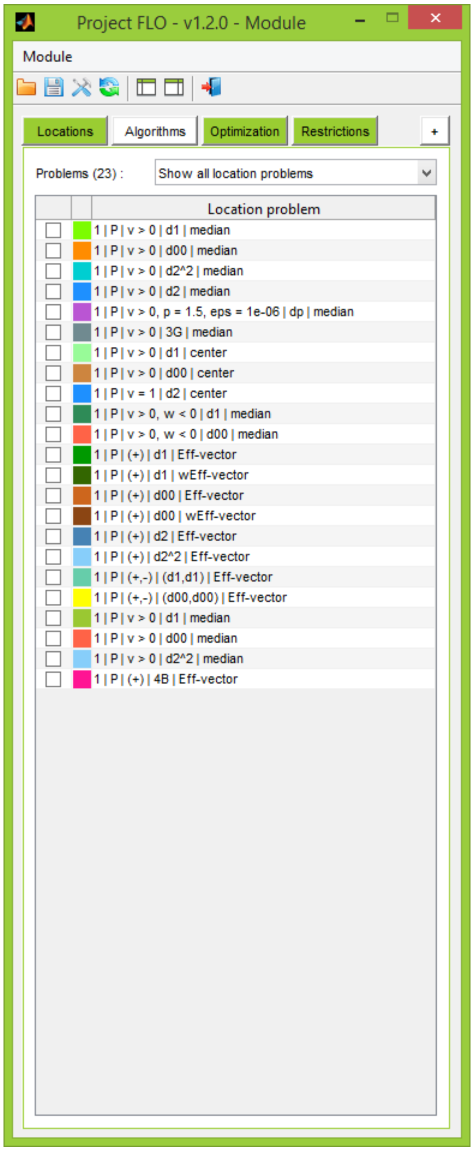


Figure 29: Module of FLO.

4.3.2 Toolbar

Using the toolbar (see Figure 30), you can call certain module functions in a simple way. The toolbar contains seven symbols (ordered from the left to

Figure 30: Toolbar of the module of FLO.

the right):

- **Import location points**
  (Import a spreadsheet location list into FLO; see Section 4.3.3)
- **Export location points**
  (Export a location list to a spreadsheet)
- **Synchronize data**
  (Synchronize data with map)
- **Algorithm settings**
  (Show the algorithm settings panel)
- **Dock module left**
  (Dock module on the left of the main window)
- **Dock module right**
  (Dock module on the right of the main window)
- **Close module**
  (Close the module)

4.3.3 Locations Panel

This panel (see Figure 31) is used for setting location point data.

The panel contains the following components:

- **List of location points**
  Each row of the location list contains the following column fields:

Figure 31: Screenshot of the Locations Panel of the module of FLO.

Selection
(The location point can be selected (marked) or deselected (unmarked))

Colour
(The display colour of the location point on the plot)

Name
(The name of the location point)

Weight
(The weight of the location point, i.e. the importance or demand of the location point)

X-coordinate
(The first coordinate of the location point)

Y-coordinate
(The second coordinate of the location point)

Status
(The status of the location point: active or inactive)

Note: Only location points with an active status will be included in the computations during the optimization process.

Distance function
(The distance measure for computing the distance from the new facility to the location point)

Note: You can create new distance measures using the Metrics panel.

Input date
(The input date of the location point)

Remarks
(Other remarks concerning the location point)

- **Buttons**
  Using the four buttons, it is possible to add, edit, delete or activate/deactivate location points.

Note: The desired operations will only be applied to selected location points (points with marked check boxes in the first column).

- **Grouping of the location points**
  The list of location points can be grouped in ascending or descending order by several categories:
  - **Show all** location points,
  - Location points with **positive weights**,
  - Location points with **negative weights**,
  - Location points with **active status**,
  - Location points with **inactive status**,
  - Location points with marked check boxes in the first column of the list, i.e., **selected** location points.
- **Sorting of the location points**
  The list of the location points can be sorted by several columns: entry date, name, weight, status, selection or metric.
- **Popup menu**
  You can right-click on the location list to access a popup menu with certain location point functions. The popup menu has the following options:

  Import list of location points
  (Import a spreadsheet or text file location list into FLO)

  Note: You have to use a special spreadsheet structure like the following example using location points (coordinates given in UTM format) in Halle (Saale) and Merseburg:

	A	**B**	**C**	**D**	E
1	Name	Weight	X	Y	Remarks
2	Halle (Saale)	100	706066	3570927	http://www.halle.de
3	Merseburg	50	708297	5693700	http://www.merseburg.de

Also note that the header row (row number 1 in the above table) is necessary for loading a spreadsheet into FLO.

If you want to use a text file (.txt format) for importing location points, then your text file needs the following structure:

Name # Weight # X # Y # Remarks
Halle (Saale) # 100 # 706066 # 3570927 # http://www.halle.de
Merseburg # 50 # 708297 # 5693700 # http://www.merseburg.de

Export list of location points
(Export a location list to a spreadsheet or a text file)

Generate location points randomly
(Generate location points randomly on a rectangle in the plane where you can choose the bounds on the components of the coordinates as well as the bounds for the values of the weights)

Clear the list of location points
(All location points in the list will be deleted)

Activate all location points
(All location points in the list will be activated)

Deactivate all location points
(All location points in the list will be deactivated)

Select all location points
(All location points in the list will be selected)

Deselect all location points
(All location points in the list will be deselected)

Invert the selection
(All selected rows will be deselected and all deselected rows will be selected)

Sort the list of the location points
(Sort the list by input date, names, weights or activity)

Reset colours of the location points
(Reset the location point's colour to the original used by FLO, (red for attraction points and blue for repulsion points))

Suppose that the panel "Restrictions" is not selected in FLO's module window. Then there are several options available to interact with FLO's main window:

- **Add mode**
  The user can place location points on the map using the left mouse button. After single-clicking on a position, a window will open where the user can specify the input parameter of the current location point.

- **Edit mode**
  The user can edit/move location points on the map. If the mouse cursor is located in the near of location points, then one location point with the smallest distance to the mouse cursor will be selected after single-clicking with the left mouse button on the map. After double-clicking on the map, a window will open where the user can edit the input parameter of the current selected location point. By pressing of the left mouse button and simultaneous moving of the cursor, the user can move existing location points on the map.

4.3.4 Algorithms Panel

In this panel (see Figure 32), you can change settings concerning the implemented optimization algorithms and change general options for how related objects are displayed on the plot in the main window.

The panel contains the following components:

- **List of location problems**

  This list contains all location problems that can be solved by FLO. The location problems can be identified through a classification scheme (contained in the third column of the list) proposed in the literature of location theory by Hamacher and Nickel in [26] (see Section 1.4).

Figure 32: Screenshot of the Algorithms Panel of the Software FLO.

Additionally, the models displayed in the list of location problems can be categorized according to predefined classes of location problems:

- Show all location problems,
- Median location problems,
- Median location problems with positive weights,
- Median location problems with positive and negative weights,
- Median location problems involving constraints,
- Center location problems with positive weights,
- Multiobjective location problems with attraction,
- Multiobjective location problems with attraction and repulsion.

See Section 5 for more details about the above location problems and their classifications.

You can choose your preferred models by marking checkboxes in the first column of the list of location problems. All algorithms for solving the marked models will run during the optimization process.

For each model, you can also select a colour to identify the solution set on the map of the main window. Left-click on the model-specific coloured rectangle in the second column of the list of location problems to open a window for selecting the colour.

- **Popup menu**
  You can also access certain operations through a popup menu by right-clicking on the list of location problems. The popup menu contains the following options:

  Show algorithm settings
  (The module will be displayed in extended-view size and the panels for changing algorithm settings will be shown)

  Note: This option is only visible in normal view size of the module.

  Select all location problems
  (All rows of the list of location problems will be selected)

Deselect all location problems
(All rows of the list of location problems will be deselected)

Duplicate all selected objects of location problems
(All rows of the list of location problems will be duplicated where the duplicated objects will be added to the list of location problems)

Delete all duplicated objects of location problems
(All rows of the list of location problems which are duplicate objects will be deleted)

Open the manual
(Open FLO's documentation)

About the classification scheme
(Get information about the used classification scheme)

- **Specific settings panel**

  By clicking on a column field of a row in the list of location problems, the corresponding settings for the model can be changed in the specific settings panel. In addition, it is possible to change algorithm-specific settings for models by using a selection field on the top of the panel.

  Now we present more details about the components of the specific settings panel:

  Name of the location problem
  The name of the current location problem will be displayed in a selection field on the top of the panel.

  Information Button
  Open detailed information (contained in a PDF file) about the model and the corresponding algorithm, all directly from within FLO.

  "+" Button
  Duplicate objects from the list of location problems. Clicking on this button adds a new object of the specific underlying model to the list of location problems. Using this, it is possible to compare solutions

and run times of different variants of algorithms (i.e. different starting solutions or values for tolerances).

"–" Button
Deletes the selected object from the list of location problems.

Note: This button is only visible for duplicated objects.

Algorithm details
The algorithm details panel displays information about the implemented algorithm: name of the algorithm, proposer, implementer and some details about the implementation date.

Algorithm options
In the algorithm options panel, you can change certain settings and parameters of the underlying algorithm for solving the selected location problem.

In the following we present a list of all the algorithm options available in the current version of FLO:

Stopping criteria

Let $f : \mathbb{R}^2 \to \mathbb{R}$ be the objective function of a scalar location problem in the plane. We look at the case where we compute approximate solutions of the problem $f(x) \to \min_{x \in X}$, where $X \subseteq \mathbb{R}^2$ represents the feasible set. Subsequently, let x^i, $i \geq 1$, be the approximate solution at iteration i and x^{i-1} of iteration $i-1$, respectively. x^0 is assumed to be the starting solution for the iteration procedure.

Function value tolerance:
The user can select a value $\varepsilon_1 > 0$ such that, if the following condition holds

$$|f(x^i) - f(x^{i-1})| < \varepsilon_1,$$

then the iteration procedure stops.

X-tolerance:
Another stopping criteria can be formulated by

$$\|x^i - x^{i-1}\|_2 < \varepsilon_2,$$

where the parameter $\varepsilon_2 > 0$ is again specified by the user.

Maximal number of iterations is reached:
The user can specify an upper bound $i_{max} \in \mathbb{N}$ for the number of iterations. That means if $i > i_{max}$ holds, then the iteration procedure stops.

Starting solution
The user can select a preferred starting solution x^0 for applying the iteration procedure.

Plotting options
Additionally, it is possible to display the following iteration-specific data:

- Iteration steps on the plot of the main window,
- X-tolerances on a sub plot,
- Function value tolerance on a sub plot.

Other algorithm-specific options
Furthermore, several other algorithm-specific options can be activated in the specific settings panel:

- Choice of metrics for measuring distances to attraction points and repulsion points, respectively.
- Display of the geometrical construction.
- Plotting component functions of the objective function.

- **General algorithm settings panel**

  Other plot-specific options can be changed in the general settings panel:

  - Show names,
  - Show weights,
  - Show convex hull,
  - Show restrictions,
  - Show contour lines,
  - Show contour lines with values,

  - Show construction lines,
  - Show unit balls.

See Section 4.1.3 for more details about the plot-specific options.

4.3.5 Optimization Panel

The optimization panel contains a list of the output of all performed algorithms for those problems where the optimization process has finished.

The panel has the following components:

- **List of optimization results**
  Each row of the list (see Figure 33), contains the following column fields:

  Location problem
  (Classification scheme of the location problem)

  Runtime
  (Execution time of the algorithm in seconds)

  Share
  (Share of the total execution time of the optimization)

  Solutions
  (Exact or approximate solutions for scalar location problems)

  Function value
  (Function value of exact or approximate solutions of a scalar location problem)

  Remarks
  (Further notes on the type of the solutions, error or warning messages, and other output information)

Figure 33: Screenshot of the Algorithms Panel of the Software FLO.

- **Popup menu**
  You can right-click on the list of optimization results to access a popup menu with certain list options. The popup menu has the following options:

  Export list of optimization results
  (Export the list of optimization results to a spreadsheet or a text file)

  Visualize running times
  (Visualize the running times of the algorithms in a barchart)

4.3.6 Restrictions Panel

In this panel (see Figure 34), you can add, edit and remove all settings concerning the restriction sets and restriction objects.

The panel contains the following components:

- **List of restriction sets**
  The list of restriction sets includes all sets of restriction objects that can be used for defining feasible sets of location problems.

  Each row of the list of restriction sets contains the following column fields:

  Selection
  (The restriction set can be selected (marked) or deselected (unmarked))

  Colour
  (The display colour of the boundary of the restriction objects contained in the corresponding restriction set)

  Name
  (The name of the restriction set)

  Input date
  (The input date of the restriction set)

  Remarks
  (Further notes on the type of restriction and other information)

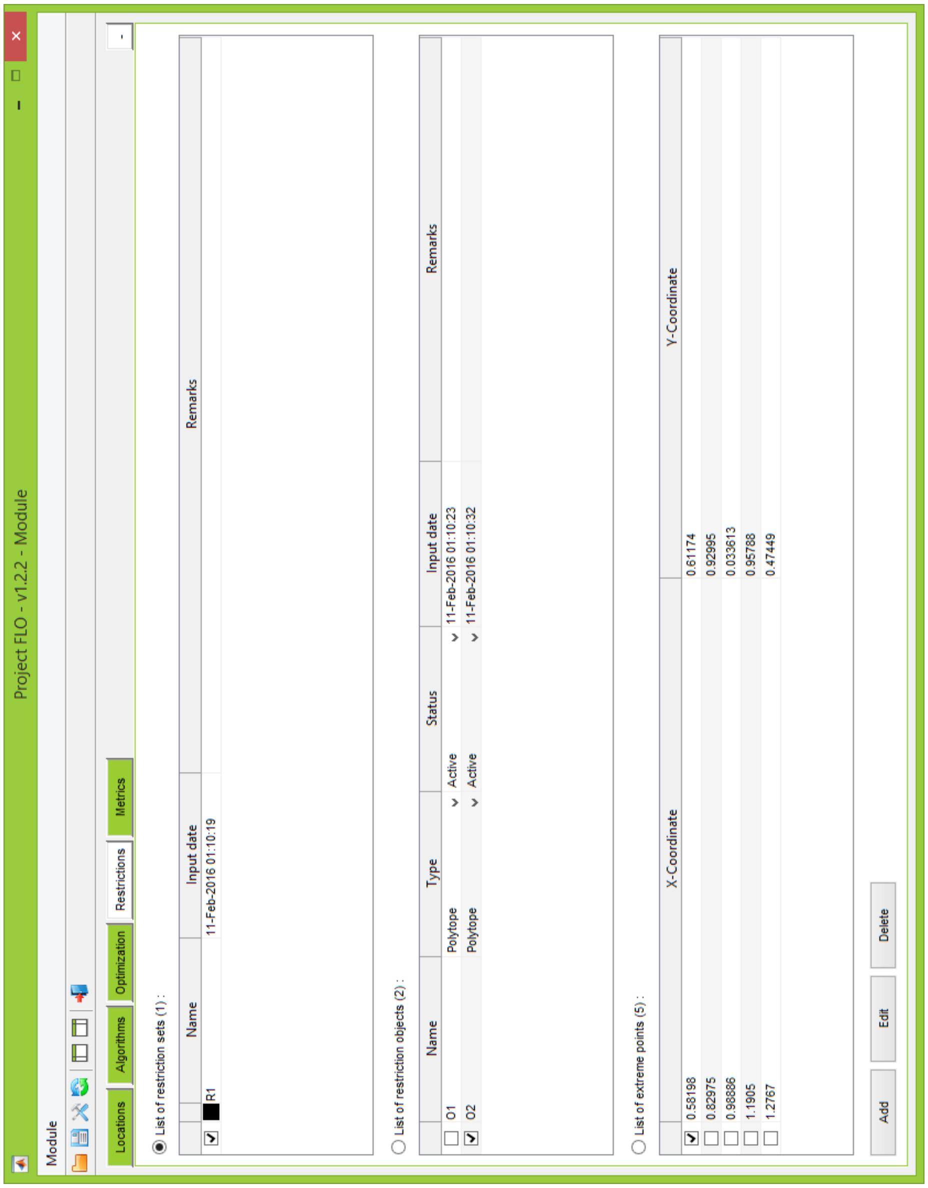


Figure 34: Screenshot of the Restrictions Panel of the Software FLO.

- **List of restrictions objects**
  The list of restriction objects includes all restriction objects of the corresponding restriction set that is selected in the list of restriction sets.

  Each row of the list of restriction sets contains the following column fields:

  Selection
  (The restriction object can be selected (marked) or deselected (unmarked))

  Name
  (The name of the restriction object)

  Type
  (The type of the restriction object)

  Note: In FLO version 1.2.2 a restriction object can only be defined by a polytope P in the plane. A polytope P is a bounded polyhedral set in $\mathbb{R}^2$ and can be represented by

  $$P = \mathrm{conv}\{e^1, \ldots, e^l\}$$

  for a finite number of extreme points $e^1, \ldots, e^l \in \mathbb{R}^2$.

  Status
  (The status (active or inactive) of the restriction object)

  Input date
  (The input date of the restriction object)

  Remarks
  (Further notes on the type of restriction and other information)

- **List of extreme points**
  This list of extreme points corresponds to a restriction object of type "polytope" that is selected in the list of all restriction objects.

  Each row of the list of extreme points contains the following column fields:

Selection
(The extreme point can be selected (marked) or deselected (unmarked))

X-coordinate
(X-coordinate of the selected extreme point)

Y-coordinate
(Y-coordinate of the selected extreme point)

- **Buttons**
  Using the three buttons, it is possible to add, edit or delete restriction sets, restriction objects and extreme points, respectively.

  Note: The desired operations will only be applied to the selected list and corresponding selected rows (rows with marked check boxes in the first column).

Suppose that the panel "Restrictions" is selected in FLO's module window. Then there are several options available to interact with FLO's main window:

- **Add mode**
  After the creation of a restriction set and a corresponding restriction object you can specify extreme points directly on the map of FLO's main window. After single-clicking using the left mouse button on a position, the corresponding point will be added to the list of all extreme points. This is provided that the chosen point is not contained in the interior of the polytope.

- **Edit mode**
  You can edit/move extreme points on the map. If the mouse cursor is located in the near of extreme points of a restriction object of the current selected restriction set, then one extreme point with the smallest distance to the mouse cursor and furthermore the corresponding restriction object will be selected after single-clicking with the left mouse button on the map. By pressing of the left mouse button and simultaneous moving of the cursor, you can move existing extreme points on the map. Moreover, you can select a restriction object directly on the map by clicking with the left mouse button on the object (but not near by an extreme point). In addition, by clicking with the left mouse button on the object (but not near by an extreme point) and simultaneous moving of the cursor you can move the selected restriction object on the map.

4.3.7 Metrics Panel

The Metrics panel can be used to create new distance functions (l_p norms or polyhedral gauges) for measuring distances between points.

The panel has the following components:

- **List of metrics**
  The list of metrics includes all distance measures that can be used for measuring distances between points in FLO.

  Each row of the list of metrics contains the following column fields:

  Selection
  The checkbox in the first column selects the given metric.

  Note: All other metrics will be deselected.

  Status colour
  The coloured rectangle in the second column of the list of metrics indicates the status of the corresponding metric. A metric with a green coloured rectangle can be used for computations in FLO while a metric with a red coloured rectangle cannot be used. For instance, a metric of polyhedral gauge type with a unit ball that does not contain the origin in its interior cannot be used and will display with a red coloured rectangle.

  Name of the metric
  The full name of the metric.

  Short cut of the name of the metric
  The short name of the metric which is used for creating classification schemes in FLO (the fourth position of the scheme).

  Remarks about the metric
  The type of the metric (l_p norm or polyhedral gauge). Moreover, information about the parameter p and about the number of fundamental directions of the polyhedral gauge, respectively, are given.

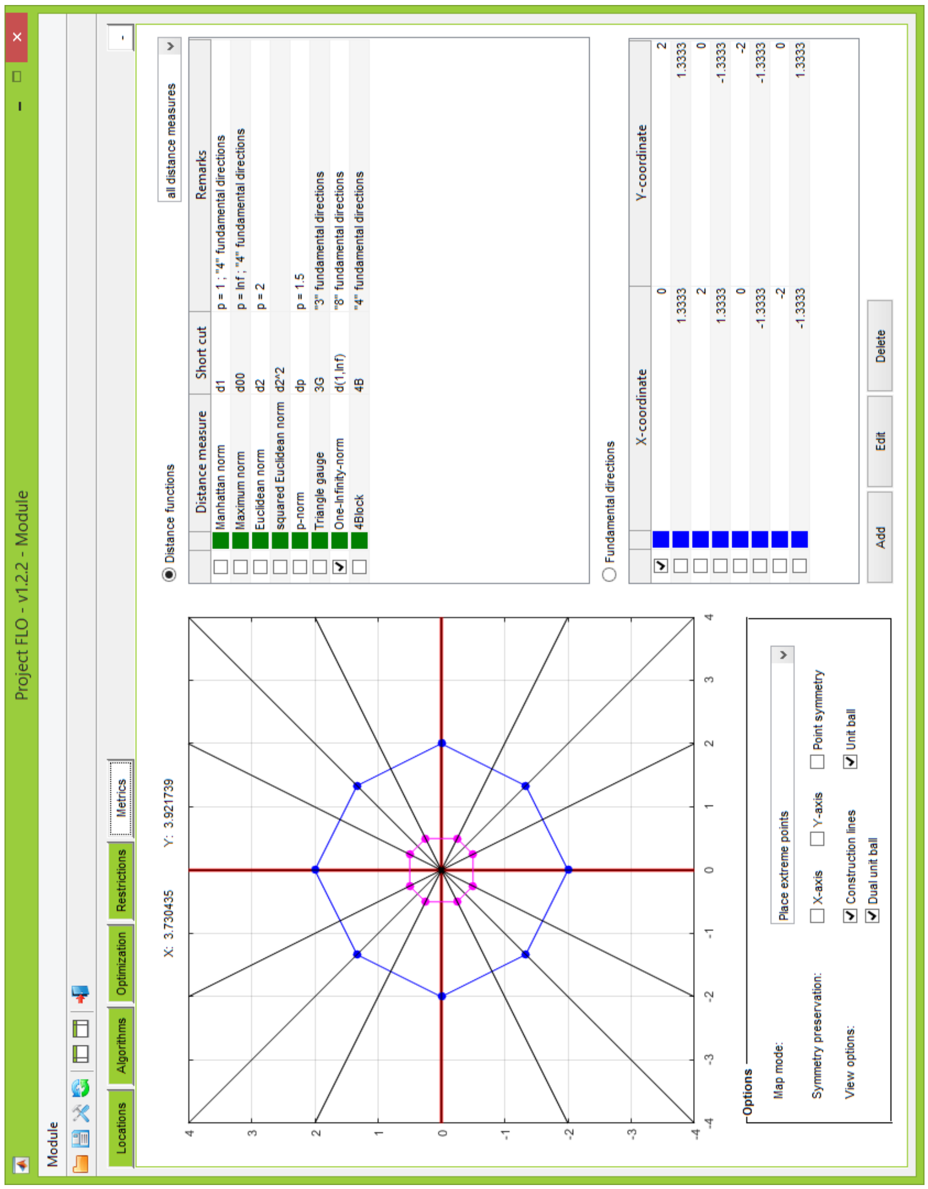


Figure 35: Screenshot of FLO's Metrics Panel.

- **Grouping of the distance functions**
  The metrics in the list of all metrics can be grouped by the type of metric.

  The following classes for grouping of the distance measures are available:

  - all distance measures,
  - norms
  - l_p norms,
  - round norms,
  - block norms,
  - no norms,
  - gauges,
  - polyhedral gauges,
  - symmetric gauges,
  - asymmetric gauges,
  - no gauges.

- **List of fundamental directions**
  If the currently selected metric in the list of metrics is of polyhedral gauge type, then the coordinates of the fundamental directions of this metric will be shown in this list.

- **Buttons and list selection fields**
  Using the three buttons on the bottom of the panel you can add, edit or delete a selected metric from the list of all metrics, provided that the corresponding list selection field "Distance functions" on the top of the panel is active. Analogously, the fundamental directions of polyhedral gauges can be changed by the user, provided that the corresponding list selection field "Fundamental directions" is active.

  Note: The metrics Manhattan norm, maximum norm, Euclidean norm and squared Euclidean norm cannot be changed by the user.

- **Plot with coordinate system**
  The plot in the left part of the panel provides a coordinate system (with centered origin) for displaying the unit ball of the current selected metric. Additionally in the coordinate system, the X-axis and

the Y-axis are shown explicitly. If the currently selected metric is of polyhedral gauge type, then the extreme points of the unit ball (polytope) as well as the fundamental directions of the polyhedral gauge are also shown.

- **Plot options panel**
  In this panel, you can determine map-specific options for displaying and constructing unit balls of polyhedral gauges.

  Note: All options are not available for metrics of l_p norm type.

  The following options are available:

  Mode of the map

  The following modes are available:

  - **Add mode**
    Clicking (using the left mouse button) on the map opens a window in which the user can specify the coordinates of the new extreme point. After pressing the confirmation button, the new extreme point will be added to the set of all extreme points of the currently selected polyhedral gauge. This is provided that the chosen point is not contained in the interior of the current polytope of the polyhedral gauge.
  - **Edit mode**
    You can use this mode to edit an extreme point of the unit ball of a polyhedral gauge by clicking (using the left mouse button) near the corresponding extreme point. A popup window will then open where the user can change the extreme point's coordinates.

    The user can move an extreme point of the unit ball of the polyhedral gauge on the map by single-clicking (using the left mouse button) near the corresponding extreme point.

    By clicking on the map using the right mouse button one can remove extreme points.

  Note: A metric of polyhedral gauge type with a unit ball which does not contain the origin in its interior cannot be used in FLO and will be displayed with a red-coloured rectangle in the second column of the list of metrics.

Symmetry preservation
Let $\text{ext}B_\mu \subseteq \mathbb{R}^2$ be the set of extreme points of the unit ball (a polytope) of a polyhedral gauge μ.

You can then choose the following options for preservation of the symmetry during adding extreme points on the map:

- Symmetry with respect to the X-axis, i.e.,
  If $(e_1, e_2) \in \text{ext}B_\mu$ holds, then also $(e_1, -e_2) \in \text{ext}B_\mu$ is true.
- Symmetry with respect to the Y-axis, i.e.,
  If $(e_1, e_2) \in \text{ext}B_\mu$ holds, then also $(-e_1, e_2) \in \text{ext}B_\mu$ is true.
- Symmetry with respect to the origin, i.e.,
  If $(e_1, e_2) \in \text{ext}B_\mu$ holds, then also $(-e_1, -e_2) \in \text{ext}B_\mu$ is true.

View options
Additionally, the following map view options can be activated:

- Show the unit ball of the current selected distance measure.
- Show the fundamental directions of the unit ball of the current selected distance measure (only possible for distance measure of polyhedral gauge type).
- Show the dual unit ball (see Definition 4 and Remark 2) of the of the current selected distance measure (only possible for distance measure of polyhedral gauge type).

  Note: If the currently selected distance measure is of polyhedral gauge type, then you can create the dual polyhedral gauge as a new distance measure using the add button (provided the check button of the list of metrics is activated).

5 Models and implemented FLO Algorithms

In this section, we present information about the models of location problems that can be solved using the Software FLO. Additionally, the included references to the literature of location theory are made to the best of our knowledge.

Note: Due to the limited computation exactness on computer systems it is possible that solutions computed by FLO differ from solutions obtained on the analytical mathematical way.

We consider a location problem defined by an objective function $f : \mathbb{R}^2 \to \mathbb{R}^m$ ($m \in \mathbb{N}$) and a nonempty feasible set $X \subseteq \mathbb{R}^2$:

$$f(x) \to \underset{x \in X}{\text{v-min}}\,.$$

5.1 Free location problems

First we assume that the feasible set X is the whole plane $\mathbb{R}^2$.

5.1.1 Median location problems with positive weights

We consider m points in the plane,

$$a^1 := (a_1^1, a_2^1),\ \cdots\ , a^m := (a_1^m, a_2^m)\ \in\ \mathbb{R}^2,$$

representing some a priori given facilities. Moreover, let $v_1, \ldots, v_m$ be positive weights (the demands for the given facilities).

Now, we search for a new facility $x \in X = \mathbb{R}^2$ such that the weighted sum of the distances between the new facility x and the given points $a^1, \ldots, a^m$ are minimized.

Note: All considered median location problems with positive weights are convex problems.

5.1.1.1 1 | P | $v > 0$ | d_1 | median

Using the Manhattan metric, defined by

$$d_1(x, a^i) := |x_1 - a_1^i| + |x_2 - a_2^i|$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we formulate the location problem by

$$\sum_{i=1}^{m} v_i \cdot d_1(x, a^i) = \sum_{i=1}^{m} v_i \cdot (|x_1 - a_1^i| + |x_2 - a_2^i|) \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses an algorithm (Derivative Algorithm) formulated in the book by Hamacher [24, Section 2.1] (1995).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further models and algorithm specific information can be found in the following selected literature:

(A) A. E. Bindschedler and J. M. Moore. *Optimal Location of New Machines in Existing Plant Layouts.* The Journal of Industrial Engineering, 12:41-47, 1961.

(B) R. L. Francis. *A Note on the Optimum Location of New Machines in Existing Plant Location.* AIIE Transactions, 14(1):57-59, 1963.

(C) R. F. Love, J. G. Morris and G. O. Wesolowsky. *Facility Location: Models and Methods.* North Holland, New York, 1988.

(D) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

5.1.1.2 1 | P | $v > 0$ | d_∞ | median

Using the maximum metric defined by

$$d_\infty(x, a^i) := \max\{|x_1 - a_1^i|, |x_2 - a_2^i|\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we formulate the location problem by

$$\sum_{i=1}^{m} v_i \cdot d_\infty(x, a^i) \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses an algorithm (Transformation Algorithm) formulated in in the book by Hamacher [24, Section 2.1] (1995).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) R. F. Love, J. G. Morris and G. O. Wesolowsky. *Facility Location: Models and Methods.* North Holland, New York, 1988.

(B) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

5.1.1.3 **1** | **P** | $v > 0$ | d_2^2 | **median**

Based on the squared Euclidean metric defined by

$$d_2^2(x, a^i) := (x_1 - a_1^i)^2 + (x_2 - a_2^i)^2$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, the location problem is given by

$$\sum_{i=1}^{m} v_i \cdot d_2^2(x, a^i) \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The corresponding algorithm included in the current version of the Software FLO generates the solution of the above location problem. The program uses the algorithm (Center of Gravity Algorithm) proposed by White (1971).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(*A*) J. A. White. *A Quadratic Facility Location Problem.* AIIE Transactions, 3(2):156-157, 1971.

(*B*) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

5.1.1.4 $\mathbf{1} \mid \mathbf{P} \mid v > 0 \mid d_2 \mid$ **median**

Using the Euclidean metric, which is defined by

$$d_2(x, a^i) := \sqrt{(x_1 - a_1^i)^2 + (x_2 - a_2^i)^2}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \dots, m$, we consider the location problem

$$\sum_{i=1}^{m} v_i \cdot d_2(x, a^i) \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \dots, m$.
The current version of the Software FLO generates an approximate solution of the above location problem. The program uses the algorithm (Weiszfeld Algorithm) proposed in the paper by Weiszfeld (1937). In this algorithm, a special optimality criteria for the existing points (proposed by Kuhn, 1967) must first be checked. After that, the Weiszfeld iteration is performed as a fixed-point iteration method. The user selects their preferred starting solution for the Weiszfeld iteration (the optimal solution of the problem $1 \mid \mathrm{P} \mid v > 0 \mid d_2^2 \mid$ median represents the predefined starting point).

Note: A comprehensive and recently-published overview of the Weiszfeld algorithm and its extensions (formulation and convergence results) is presented in the paper by Beck and Sabach (2015).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(*A*) E. V. Weiszfeld. *Sur le point pour lequel la somme des distances de n points donnes est minimum.* Tohoku Math, 43:355386, 1937.

(B) R. F. Love, J. G. Morris and G. O. Wesolowsky. *Facility Location: Models and Methods.* North Holland, New York, 1988.

(C) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

(D) A. Beck and S. Sabach. *Weiszfelds Method: Old and New Results.* Journal of Optimization Theory and Applications, 164(1):1-40, 2015.

5.1.1.5 1 | P | $v > 0$ | d_p | median

Using the l_p metric ($1 \leq p < \infty$), which is defined by

$$d_p(x, a^i) := (|x_1 - a_1^i|^p + |x_2 - a_2^i|^p)^{\frac{1}{p}}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we consider the location problem

$$\sum_{i=1}^{m} v_i \cdot d_p(x, a^i) \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The current version of the Software FLO generates an approximate solution of the above location problem. The program uses the algorithm (Hyperbolic Approximation Algorithm) proposed in the paper by Morris and Verdini (1979). A special optimality criteria for the existing points must first be checked and then the Hyperbolic Approximation iteration is performed as a fixed-point iteration method. The user selects their preferred starting point for the Hyperbolic Approximation iteration (the optimal solution of the problem 1 | P | $v > 0$ | d_2^2 | median represents the predefined starting point).

Note: For the Hyperbolic Approximation Algorithm, convergence (linear) can only be shown for the cases $1 \leq p \leq 2$. More information about the procedure can be found in the books by Love, Morris and Wesolowsky (1988), Drezner and Hamacher (2001) and Hamacher (1995).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) J. G. Morris and W. A. Verdini. *Minisum, l_p distance location problems solved via a perturbed problem and Weiszfeld's algorithm.* Operations Research, 27:1180-1188, 1979.

(B) R. F. Love, J. G. Morris and G. O. Wesolowsky. *Facility Location: Models and Methods.* North Holland, New York, 1988.

(C) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

(D) Z. Drezner and H. W. Hamacher. *Facility Location: Theory and Algorithms.* Springer, Berlin, 2001.

5.1.1.6 $1 \mid \mathbf{P} \mid v > 0 \mid \mu_i \mid$ median

Let B_{μ_i}, $i = 1, \ldots, m$, be polytopes (closed, bounded and polyhedral sets) in $\mathbb{R}^2$ with $0 \in \operatorname{int} B_{\mu_i}$, $i = 1, \ldots, m$. Using a point-specific polyhedral gauge distance, which is defined by

$$\mu_i(x - a^i) := \inf\{\lambda > 0 \,|\, x - a^i \in \lambda \cdot B_{\mu_i}\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a^i_1, a^i_2)$, $i = 1, \ldots, m$, we consider the location problem

$$\sum_{i=1}^{m} v_i \cdot \mu_i(x - a^i) \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses the algorithm (Construction Grid Algorithm) proposed in the paper by Durier and Michelot (1985). More information about the procedure can be found in the dissertations of Nickel (1995), Bischoff (2008) or Wagner (2014).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) R. Durier and C. Michelot. *Geometrical properties of the Fermat-Weber problem.* European Journal of Operational Research, 20:332-343, 1985.

(B) S. Nickel. *Discretization of Planar Location Problems.* Verlag Shaker, Aachen, 1995.

(C) M. Bischoff. *Location of Connection Facilities.* Verlag Shaker, Aachen, 2008.

(D) A. Wagner. *A new Duality Based Approach for the Problem of Locating a Semi-Obnoxious Facility.* Dissertation, Martin Luther University Halle-Wittenberg, 2014.

5.1.2 Median location problems with positive and negative weights

We consider m_1 points in the plane,

$$a^1 := (a_1^1, a_2^1),\ \cdots\ , a^{m_1} := (a_1^{m_1}, a_2^{m_1})\ \in\ \mathbb{R}^2,$$

representing some a priori given attraction facilities with positive weights $v_1, \ldots, v_{m_1} \in \mathbb{R}$. Moreover, we consider m_2 points in the plane,

$$b^1 := (b_1^1, b_2^1),\ \cdots\ , b^{m_2} := (b_1^{m_2}, b_2^{m_2})\ \in\ \mathbb{R}^2,$$

representing some a priori given repulsion facilities with negative weights $w_1, \ldots, w_{m_2} \in \mathbb{R}$.

Now, we search for a new facility $x \in X = \mathbb{R}^2$ such that the weighted sum of the distances between the new facility x and the given points $a^1, \ldots, a^{m_1}$ as well as $b^1, \ldots, b^{m_2}$ are minimized.

Note: All considered median location problems with positive weights and negative weights are non-convex problems in general.

5.1.2.1 1 | P | $v > 0$, $w < 0$ | d_1 | median

Using the Manhattan metric defined by

$$d_1(x,y) := |x_1 - y_1| + |x_2 - y_2|$$

for all $x := (x_1, x_2), y := (y_1, y_2) \in \mathbb{R}^2$, we consider the location problem

$$\left[\sum_{i=1}^{m_1} v_i \cdot d_1(x, a^i)\right] + \left[\sum_{j=1}^{m_2} w_j \cdot d_1(x, b^j)\right] \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \ldots, m_1$ and $w_j < 0$ for all $j = 1, \ldots, m_2$.
The current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses an algorithm (Iterative Derivative Algorithm) that is a slight modification of the algorithm proposed by Nickel and Dudenhöffer (1997) in [40].

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) S. Nickel and E. M. Dudenhöffer. *Weber's Problem with Attraction and Repulsion under Polyhedral Gauges.* Journal of Global Optimization, 11:409-432, 1997.

(B) A. Wagner. *A new Duality Based Approach for the Problem of Locating a Semi-Obnoxious Facility.* Dissertation, Martin Luther University Halle-Wittenberg, 2014.

(C) A. Wagner, J. E. Martinez-Legaz and Chr. Tammer. *Locating a Semi-Obnoxious Facility - A Toland-Singer Duality Based Approach.* Journal of Convex Analysis, 2016 (to appear).

(D) M. Hillmann. *Lagrange-Multiplikatoren-Regeln und Algorithmen für nichtkonvexe Standortprobleme.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

5.1.2.2 1 | **P** | $v > 0$, $w < 0$ | d_∞ | median

Based on the maximum metric, defined by

$$d_\infty(x, y) := \max\{|x_1 - y_1|, |x_2 - y_2|\}$$

for all $x := (x_1, x_2), y := (y_1, y_2) \in \mathbb{R}^2$, the location problem is given by

$$\left[\sum_{i=1}^{m_1} v_i \cdot d_\infty(x, a^i)\right] + \left[\sum_{j=1}^{m_2} w_j \cdot d_\infty(x, b^j)\right] \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \ldots, m_1$ and $w_j < 0$ for all $j = 1, \ldots, m_2$.
The corresponding algorithm included in the current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses an algorithm (Iterative Derivative Algorithm) that is a slight modification of the algorithm by Nickel and Dudenhöffer (1997) and a transformation between Manhattan norm and maximum norm.

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) S. Nickel and E. M. Dudenhöffer. *Weber's Problem with Attraction and Repulsion under Polyhedral Gauges.* Journal of Global Optimization, 11:409-432, 1997.

(B) A. Wagner. *A new Duality Based Approach for the Problem of Locating a Semi-Obnoxious Facility.* Dissertation, Martin-Luther-Universität Halle-Wittenberg, 2014.

(C) A. Wagner, J. E. Martinez-Legaz and Chr. Tammer. *Locating a Semi-Obnoxious Facility - A Toland-Singer Duality Based Approach.* Journal of Convex Analysis, 2016 (to appear).

(D) M. Hillmann. *Lagrange-Multiplikatoren-Regeln und Algorithmen für nichtkonvexe Standortprobleme.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

5.1.3 Center location problems with positive weights

We consider m points in the plane,

$$a^1 := (a_1^1, a_2^1),\ \cdots\ , a^m := (a_1^m, a_2^m)\ \in\ \mathbb{R}^2,$$

representing some a priori given facilities. Moreover, let $v_1, \ldots, v_m$ positive weights.

Now, we search for a new facility $x \in X = \mathbb{R}^2$ such that the weighted sum of the distances between the new facility x and the given points $a^1, \ldots, a^m$ are minimized.

Note: All considered center location problems with positive weights are convex problems.

5.1.3.1 $\mathbf{1} \mid \mathbf{P} \mid v > 0 \mid d_1 \mid$ **center**

Using the Manhattan metric, defined by

$$d_1(x, a^i) := |x_1 - a_1^i| + |x_2 - a_2^i|$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we consider the location problem

$$\max\{v_i \cdot d_1(x, a^i) \,|\, i = 1, \ldots, m\} \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses the algorithm (Transformation Algorithm) formulated in Hamacher [24, Section 4.2] (1995).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(*A*) R. F. Love, J. G. Morris and G. O. Wesolowsky. *Facility Location: Models and Methods.* North Holland, New York, 1988.

(B) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

5.1.3.2 1 | P | $v > 0$ | d_∞ | center

Using the maximum metric, defined by

$$d_\infty(x, a^i) := \max\{|x_1 - a_1^i|, |x_2 - a_2^i|\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we consider the location problem

$$\max\{v_i \cdot d_\infty(x, a^i) \,|\, i = 1, \ldots, m\} \to \min_{x \in X = \mathbb{R}^2}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses the algorithm (Intersection Algorithm) formulated in Hamacher [24, Section 4.2] (1995).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) R. F. Love, J. G. Morris and G. O. Wesolowsky. *Facility Location: Models and Methods.* North Holland, New York, 1988.

(B) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

5.1.3.3 1 | P | $v = 1$ | d_2 | center

Using the Euclidean metric, defined by

$$d_2(x, a^i) := \sqrt{(x_1 - a_1^i)^2 + (x_2 - a_2^i)^2}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a^i_1, a^i_2)$, $i = 1, \ldots, m$, we consider the location problem

$$\max\{d_2(x, a^i) \,|\, i = 1, \ldots, m\} \to \min_{x \in X = \mathbb{R}^2}.$$

The above location problem is known as the smallest-circle problem or minimum covering circle problem in the literature.

The current version of the Software FLO generates an exact solution of the above location problem. The program uses the algorithm (Elzinga-Hearn Algorithm) proposed by Elzinga and Hearn (1972).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) D. J. Elzinga and D. W. Hearn. *Geometrical Solutions for some minimax location problems.* Transportation Science, 6:379-394, 1972.

(B) N. Megiddo. *The weighted Euclidean 1-center problem.* Mathematics of Operations Research, 8(4):498504, 1983.

(C) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

5.1.4 Multiobjective location problems with attraction

The classical single-facility multiobjective location problem consists in minimizing the distances between a new facility $x \in X = \mathbb{R}^2$ and all given attraction facilities

$$a^1 := (a^1_1, a^1_2), \; \cdots \; , a^m := (a^m_1, a^m_2) \; \in \; \mathbb{R}^2$$

simultaneously. Note that simultaneous minimization is understood in the sense of multiobjective optimization (see Section 1.6).

Note: All considered multiobjective location problems with attraction are convex problems.

5.1.4.1 1 | **P** | (+) | d_1 | **Eff-vector**

Using the Manhattan metric (also called rectangular metric or l_1 metric), defined by

$$d_1(x, a^i) := |x_1 - a_1^i| + |x_2 - a_2^i|$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we consider the location problem

$$\begin{pmatrix} d_1(x, a^1) \\ \cdots \\ d_1(x, a^m) \end{pmatrix} \to \underset{x \in X = \mathbb{R}^2}{\text{v-min}}.$$

In this problem, one is looking for Edgeworth-Pareto efficient solutions.

The current version of the Software FLO solves the above location problem and computes the whole set of Edgeworth-Pareto efficient solutions. The program uses the algorithm (Rectangular Decomposition Algorithm) proposed by Alzorba, Günther, Popovici and Tammer (2015).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) S. Alzorba, C. Günther, N. Popovici and Chr. Tammer. *A new algorithm for solving planar multiobjective location problems involving the Manhattan norm.* Preprint Optimization-Online, http://www.optimization-online.org/DB_HTML/2016/01/5305.html, 2015 (submitted).

(B) S. Alzorba, C. Günther and N. Popovici. *A special class of extended multicriteria location problems.* Optimization, 64(5):1305-1320, 2015 (DOI: 10.1080/02331934.2013.869810).

(C) C. Gerth (Tammer) and K. Pöhler. *Dualität und algorithmische Anwendung beim vektoriellen Standortproblem.* Optimization, 19:491-512, 1988.

(D) C. Günther. *Dekomposition mehrkriterieller Optimierungsprobleme und Anwendung bei nicht konvexen Standortproblemen.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

(E) L. Chalmet, R. L. Francis, A. Kolen. *Finding efficient solutions for rectilinear distance location problems efficiently.* European Journal of Operational Research, 6:117-124, 1981.

(F) R. E. Wendell, A. P. Hurter and T. J. Lowe. *Efficient Points in Location Problems.* AIIE Transactions, 9(3):238-246, 1977.

(G) K. Nouioua. *Enveloppes de Pareto et Reseaux de Manhattan.* Dissertation, University of the Mediterranean, 2005.

5.1.4.2 1 | P | (+) | d_1 | wEff-vector

Using the Manhattan metric (also called rectangular metric or l_1 metric), defined by

$$d_1(x, a^i) := |x_1 - a_1^i| + |x_2 - a_2^i|$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \dots, m$, we consider the location problem

$$\begin{pmatrix} d_1(x, a^1) \\ \dots \\ d_1(x, a^m) \end{pmatrix} \to \underset{x \in X = \mathbb{R}^2}{\text{v-min}} .$$

In this problem, we search for weakly Edgeworth-Pareto efficient solutions.

The current version of the Software FLO solves the above location problem and computes the whole set of weakly Edgeworth-Pareto efficient solutions. The program uses the algorithm (Maximum Rectangular Hull Algorithm) proposed by Alzorba, Günther, Popovici and Tammer (2015).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) S. Alzorba, C. Günther and N. Popovici. *A special class of extended multicriteria location problems.* Optimization, 64(5):1305-1320, 2015 (DOI: 10.1080/02331934.2013.869810).

(*B*) S. Alzorba, C. Günther, N. Popovici and Chr. Tammer. *A new algorithm for solving planar multiobjective location problems involving the Manhattan norm.* Preprint Optimization-Online, http://www.optimization-online.org/DB_HTML/2016/01/5305.html, 2015 (submitted).

(*C*) N. Popovici. *Pareto reducible multicriteria optimization problems.* Optimization, 54:253-263, 2005.

5.1.4.3 1 | P | (+) | d_∞ | Eff-vector

Based on the Maximum metric, defined by

$$d_\infty(x, a^i) := \max\{|x_1 - a_1^i|, |x_2 - a_2^i|\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \dots, m$, the location problem is given by

$$\begin{pmatrix} d_\infty(x, a^1) \\ \dots \\ d_\infty(x, a^m) \end{pmatrix} \to \operatorname*{v-min}_{x \in X = \mathbb{R}^2}.$$

In this problem, we search for Edgeworth-Pareto efficient solutions.

The current version of the Software FLO solves the above location problem and computes the whole set of Edgeworth-Pareto efficient solutions. The program uses the algorithm (Rectangular Decomposition Algorithm) proposed by Alzorba, Günther, Popovici and Tammer (2015).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(*A*) S. Alzorba, C. Günther, N. Popovici and Chr. Tammer. *A new algorithm for solving planar multiobjective location problems involving the Manhattan norm.* Preprint Optimization-Online, http://www.optimization-online.org/DB_HTML/2016/01/5305.html, 2015 (submitted).

(*B*) S. Alzorba, C. Günther and N. Popovici. *A special class of extended multicriteria location problems.* Optimization, 64(5):1305-1320, 2015 (DOI: 10.1080/02331934.2013.869810).

(C) Chr. Gerth (Tammer) and K. Pöhler. *Dualität und algorithmische Anwendung beim vektoriellen Standortproblem.* Optimization, 19:491-512, 1988.

(D) C. Günther. *Dekomposition mehrkriterieller Optimierungsprobleme und Anwendung bei nicht konvexen Standortproblemen.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

(E) L. Chalmet, R. L. Francis, A. Kolen. *Finding efficient solutions for rectilinear distance location problems efficiently.* European Journal of Operational Research, 6:117-124, 1981.

(F) R. E. Wendell, A. P. Hurter and T. J. Lowe. *Efficient Points in Location Problems.* AIIE Transactions, 9(3):238-246, 1977.

(G) K. Nouioua. *Enveloppes de Pareto et Reseaux de Manhattan.* Dissertation, University of the Mediterranean, 2005.

5.1.4.4 **1 | P** | (+) | d_∞ | **wEff-vector**

Based on the maximum metric, defined by

$$d_\infty(x, a^i) := \max\{|x_1 - a_1^i|, |x_2 - a_2^i|\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we consider the location problem

$$\begin{pmatrix} d_\infty(x, a^1) \\ \ldots \\ d_\infty(x, a^m) \end{pmatrix} \to \underset{x \in X = \mathbb{R}^2}{\text{v-min}} .$$

In this problem, we search for weakly Edgeworth-Pareto efficient solutions.

The current version of the Software FLO solves the above location problem and computes the whole set of weakly Edgeworth-Pareto efficient solutions. The program uses the algorithm (Manhattan Rectangular Hull Algorithm) proposed by Alzorba, Günther, Popovici (2015).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) S. Alzorba, C. Günther and N. Popovici. *A special class of extended multicriteria location problems.* Optimization, 64(5):1305-1320, 2015 (DOI: 10.1080/02331934.2013.869810).

(B) N. Popovici. *Pareto reducible multicriteria optimization problems.* Optimization, 54:253-263, 2005.

5.1.4.5 1 | P | (+) | μ | Eff-vector

Let B be a polytope (closed, bounded and polyhedral set) in $\mathbb{R}^2$ with $0 \in \operatorname{int} B$ and $B = -B$ (symmetry with respect to the origin). Moreover, assume that the polytope B can be represented by the convex hull of four points $r^1, r^2, r^3, r^4 \in \mathbb{R}^2$, i.e., it holds $B = \operatorname{conv}\{r^1, r^2, r^3, r^4\}$.

Now using the unit ball $B_\mu := B$ we can define a block norm μ. The distances induced by the gauge μ are defined by

$$\mu(x - a^i) := \inf\{\lambda > 0 \,|\, x - a^i \in \lambda \cdot B_\mu\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a^i_1, a^i_2)$, $i = 1, \ldots, m$. Hence, the location problem is given by

$$\begin{pmatrix} \mu(x - a^1) \\ \cdots \\ \mu(x - a^m) \end{pmatrix} \to \underset{x \in X = \mathbb{R}^2}{\text{v-min}} \,.$$

In this problem, we search for Edgeworth-Pareto efficient solutions.

The current version of the Software FLO solves the above location problem and computes the whole set of Edgeworth-Pareto efficient solutions. The program solves the location problem using an algorithm (Rectangular Decomposition Algorithm) that is based on the algorithm for solving the problem "1 | P | (+) | μ | Eff-vector" and an appropriate linear transformation (see Günther, Popovici and Tammer, 2016).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.2.0 which was released on 08/01/2016.

Further model and algorithm-specific information can be found in the following selected literature:

(A) C. Günther, N. Popovici and Chr. Tammer. *Reducing the complexity of planar multiobjective location problems.* 2016 (in preparation).

(B) S. Alzorba, C. Günther, N. Popovici and Chr. Tammer. *A new algorithm for solving planar multiobjective location problems involving the Manhattan norm.* Preprint Optimization-Online, http://www.optimization-online.org/DB_HTML/2016/01/5305.html, 2015 (submitted).

(C) M. Kaiser. *Spatial Uncertainties in Continuous Location Problems.* Dissertation, Bergische Universität Wuppertal, 2015.

5.1.4.6 1 | P | (+) | μ | wEff-vector

Let B be a polytope (closed, bounded and polyhedral set) in $\mathbb{R}^2$ with $0 \in \operatorname{int} B$ and $B = -B$ (symmetry with respect to the origin). Moreover, assume that the polytope B can be represented by the convex hull of four points $r^1, r^2, r^3, r^4 \in \mathbb{R}^2$, i.e., it holds $B = \operatorname{conv}\{r^1, r^2, r^3, r^4\}$.

Now using the unit ball $B_\mu := B$ we can define a block norm μ. The distances induced by the gauge μ are defined by

$$\mu(x - a^i) := \inf\{\lambda > 0 \,|\, x - a^i \in \lambda \cdot B_\mu\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a^i_1, a^i_2)$, $i = 1, \ldots, m$. Hence, the location problem is given by

$$\begin{pmatrix} \mu(x - a^1) \\ \ldots \\ \mu(x - a^m) \end{pmatrix} \to \underset{x \in X = \mathbb{R}^2}{\text{v-min}} .$$

In this problem, we search for weakly Edgeworth-Pareto efficient solutions.

The current version of the Software FLO solves the above location problem and computes the whole set of weakly Edgeworth-Pareto efficient solutions. The program solves the location problem using an algorithm (Rectangular Hull Algorithm) that is based on the algorithm for solving the problem "1 | P | (+) | μ | wEff-vector" and an appropriate linear transformation (see Günther, Popovici and Tammer, 2016).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.2.2 which was released on 12/02/2016.

Further model and algorithm-specific information can be found in the following selected literature:

(A) C. Günther, N. Popovici and Chr. Tammer. *Reducing the complexity of planar multiobjective location problems.* 2016 (in preparation).

(B) S. Alzorba, C. Günther, N. Popovici and Chr. Tammer. *A new algorithm for solving planar multiobjective location problems involving the Manhattan norm.* Preprint Optimization-Online, http://www.optimization-online.org/DB_HTML/2016/01/5305.html, 2015 (submitted).

(C) M. Kaiser. *Spatial Uncertainties in Continuous Location Problems.* Dissertation, Bergische Universität Wuppertal, 2015.

(D) N. Popovici. *Pareto reducible multicriteria optimization problems.* Optimization, 54:253-263, 2005.

5.1.4.7 1 | P | (+) | d_2 | Eff-vector

Using the Euclidean metric, defined by

$$d_2(x, a^i) := \sqrt{(x_1 - a_1^i)^2 + (x_2 - a_2^i)^2}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we consider the location problem

$$\begin{pmatrix} d_2(x, a^1) \\ \ldots \\ d_2(x, a^m) \end{pmatrix} \to \underset{x \in X = \mathbb{R}^2}{\text{v-min}} .$$

In this problem, we search for Edgeworth-Pareto efficient solutions.

The current version of the Software FLO solves the above location problem and computes the whole set of Edgeworth-Pareto efficient solutions. The program uses the algorithm (Convex Hull Algorithm) proposed by Kuhn (1967). Note that, for the above location problem, the set of strictly EP-efficient solutions, the set of EP-efficient solutions and the set of weakly EP-efficient solutions coincide (see the thesis by Günther, 2013).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version

1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) H. W. Kuhn. *On a pair of dual nonlinear programs.* Nonlinear programming, Wiley, New York, 1967.

(B) C. Günther. *Dekomposition mehrkriterieller Optimierungsprobleme und Anwendung bei nicht konvexen Standortproblemen.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

5.1.4.8 1 | P | (+) | d_2^2 | Eff-vector

Using the squared Euclidean metric, defined by

$$d_2^2(x, a^i) := (x_1 - a_1^i)^2 + (x_2 - a_2^i)^2$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \dots, m$, we formulate the location problem

$$\begin{pmatrix} d_2^2(x, a^1) \\ \dots \\ d_2^2(x, a^m) \end{pmatrix} \to \underset{x \in X = \mathbb{R}^2}{\text{v-min}} .$$

In this problem, we search for Edgeworth-Pareto efficient solutions.

The current version of the Software FLO solves the above location problem and computes the whole set of Edgeworth-Pareto efficient solutions. Due to the fact that the set of EP-efficient solutions of the above-defined problem coincides with the set of EP-efficient solutions of the problem

$$1 \mid \text{P} \mid (+) \mid d_2 \mid \text{Eff-vector}$$

(see the thesis by Günther, 2013), the Software FLO uses the algorithm (Convex Hull Algorithm) proposed by Kuhn (1967). Note that, for the above location problem, the set of strictly EP-efficient solutions, the set of EP-efficient solutions and the set of weakly EP-efficient solutions coincide (see the thesis by Günther, 2013).

The algorithm was implemented in FLO by Christian Günther. Software

FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) H. W. Kuhn. *On a pair of dual nonlinear programs.* Nonlinear programming, Wiley, New York, 1967.

(B) C. Günther. *Dekomposition mehrkriterieller Optimierungsprobleme und Anwendung bei nicht konvexen Standortproblemen.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

5.1.5 Multiobjective location problems with attraction and repulsion

The single-facility multiobjective location problem consists in minimizing the distances between a new facility $x \in \mathbb{R}^2$ and all given facilities

$$a^1 := (a_1^1, a_2^1),\ \cdots\ , a^m := (a_1^{m_1}, a_2^{m_1})\ \in\ \mathbb{R}^2$$

simultaneously. In addition to the attraction points $a^1, \ldots, a^{m_1}$, we now consider m_2 repulsion points

$$b^1 := (b_1^1, b_2^1),\ \cdots\ , b^{m_2} := (b_1^{m_2}, b_2^{m_2})\ \in\ \mathbb{R}^2,$$

(i.e., undesirable facilities, such as polluting factories or nuclear plants) and we want to maximize the distance from the new location point $x \in X = \mathbb{R}^2$ to each of the points $b^1, \ldots, b^{m_2}$ simultaneously. Note that simultaneous minimization is understood in the sense of multiobjective optimization (see Section 1.6).

Note: All considered multiobjective location problems with attraction and repulsion are non-convex problems.

5.1.5.1 1 | P | $(+,-)$ | (d_1, μ_j) | Eff-vector

We measure the distances between the new facility and the given attraction facilities using the Manhattan metric, which is defined by

$$d_1(x, a^i) := |x_1 - a_1^i| + |x_2 - a_2^i|$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a^i_1, a^i_2)$, $i = 1, \dots, m_1$. Moreover, let the functions

$$\mu_1, \dots, \mu_{m_2} : \mathbb{R}^2 \to \mathbb{R}$$

represent gauge functions, i.e., we have

$$\mu_j(x - b^j) := \inf\{\lambda > 0 \mid x - b^j \in \lambda \cdot B_{\mu_j}\}$$

for all $j = 1, \dots, m_2$ and all $x \in \mathbb{R}^2$, where B_{μ_j} is a polytope with $0 \in \operatorname{int} B_{\mu_j}$ (in this case is μ_j a polyhedral gauge) or a unit ball of an l_p norm.

Now the location problem is given by

$$f(x) := \begin{pmatrix} d_1(x, a^1) \\ \dots \\ d_1(x, a^{m_1}) \\ -\mu_1(x - b^1) \\ \dots \\ -\mu_{m_2}(x - b^{m_2}) \end{pmatrix} \to \underset{x \in X = \mathbb{R}^2}{\text{v-min}}\,.$$

Note that we search for Edgeworth-Pareto efficient solutions in the above-defined non-convex vector-valued location problem.

The current version of the Software FLO solves the above location problem and computes Edgeworth-Pareto efficient solutions with the help of the decomposition approach proposed by Alzorba, Günther and Popovici, 2015:

1°. Compute the set of all EP-efficient solutions $\operatorname{Eff}(\mathbb{R}^2 \mid g)$ of the problem

$$g(x) := \begin{pmatrix} d_1(x, a^1) \\ \dots \\ d_1(x, a^{m_1}) \end{pmatrix} \to \underset{x \in \mathbb{R}^2}{\text{v-min}}\,.$$

2°. The decision maker has to select some negative weights $w_1, \dots, w_{m_2}$ depending on the significance of the repulsion of the undesirable facilities.

3°. Solve the scalar restricted location problem

$$h(x) := \sum_{j=1}^{m_2} w_j \cdot \mu_j(x - b^j) \to \min_{x \in \operatorname{Eff}(\mathbb{R}^2 \mid g)}\,.$$

4°. It holds that we have

$$x^0 \in \underset{x \in \mathrm{Eff}(\mathbb{R}^2 | g)}{\operatorname{argmin}} \; h(x) \subseteq \mathrm{Eff}(\mathbb{R}^2 \mid f) \cap \mathrm{Eff}(\mathbb{R}^2 \mid g).$$

Note that the current version of the Software FLO does not compute the whole set of Edgeworth-Pareto efficient solutions of the original underlying location problem. The Software computes solutions depending on the selection of the negative weights $w_1, \ldots, w_{m_2}$ (e.g. significance of the repulsion of the undesirable facilities). More information about the model and the algorithm can be found in the paper by Alzorba, Günther and Popovici (2015).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Further model and algorithm-specific information can be found in the following selected literature:

(A) S. Alzorba, C. Günther and N. Popovici. *A special class of extended multicriteria location problems.* Optimization, 64(5):1305-1320, 2015 (DOI: 10.1080/02331934.2013.869810).

(B) C. Günther. *Dekomposition mehrkriterieller Optimierungsprobleme und Anwendung bei nicht konvexen Standortproblemen.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

(C) A. Jourani, C. Michelot and M. Ndiaye. *Efficiency for continuous facility location problems with attraction and repulsion.* Annals OR, 167(1):43-60, 2009.

5.1.5.2 1 | P | $(+,-)$ | (d_∞, μ_j) | Eff-vector

We measure the distances between the new facility and the given attraction facilities with the help of the well-known Manhattan metric, which is defined by

$$d_\infty(x, a^i) := \max\{|x_1 - a_1^i|, |x_2 - a_2^i|\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m_1$. Moreover, let the functions

$$\mu_1, \ldots, \mu_{m_2} : \mathbb{R}^2 \to \mathbb{R}$$

represent gauge functions, i.e., we have

$$\mu_j(x-b^j) := \inf\{\lambda > 0 \mid x - b^j \in \lambda \cdot B_{\mu_j}\}$$

for all $j = 1, \dots, m_2$ and all $x \in \mathbb{R}^2$, where B_{μ_j} is a polytope with $0 \in \operatorname{int} B_{\mu_j}$ (in this case is μ_j a polyhedral gauge) or a unit ball of an l_p norm.

Now the location problem is given by

$$f(x) := \begin{pmatrix} d_\infty(x, a^1) \\ \dots \\ d_\infty(x, a^{m_1}) \\ -\mu_1(x-b^1) \\ \dots \\ -\mu_{m_2}(x-b^{m_2}) \end{pmatrix} \to \operatorname*{v-min}_{x \in X = \mathbb{R}^2}.$$

Note that we are searching for Edgeworth-Pareto efficient solutions in the above-defined non-convex vector-valued location problem.

The current version of the Software FLO solves the above location problem and computes Edgeworth-Pareto efficient solutions with the help of the decomposition approach proposed by Alzorba, Günther and Popovici, 2015:

1°. Compute the set of all EP-efficient solutions $\operatorname{Eff}(\mathbb{R}^2 \mid g)$ of the problem

$$g(x) := \begin{pmatrix} d_\infty(x, a^1) \\ \dots \\ d_\infty(x, a^{m_1}) \end{pmatrix} \to \operatorname*{v-min}_{x \in \mathbb{R}^2}.$$

2°. The decision maker chooses some negative weights $w_1, \dots, w_{m_2}$ depending on the significance of the repulsion of the undesirable facilities.

3°. Solve the scalar constrained location problem

$$h(x) := \sum_{j=1}^{m_2} w_j \cdot \mu_j(x - b^j) \to \min_{x \in \operatorname{Eff}(\mathbb{R}^2 | g)}.$$

4°. It holds

$$x^0 \in \operatorname*{argmin}_{x \in \operatorname{Eff}(\mathbb{R}^2 | g)} h(x) \subseteq \operatorname{Eff}(\mathbb{R}^2 \mid f) \cap \operatorname{Eff}(\mathbb{R}^2 \mid g).$$

Note that the current version of the Software FLO does not compute the whole set of Edgeworth-Pareto efficient solutions of the original underlying location problem. FLO computes solutions depending on the selection of the negative weights $w_1, \ldots, w_{m_2}$ (e.g. significance of the repulsion of the undesirable facilities). More information about the model and the algorithm can be found in the paper by Alzorba, Günther and Popovici (2015).

The algorithm was implemented in FLO by Christian Günther. Software FLO has been able to solve the underlying location problem since version 1.0.0 which was released on 22/04/2015.

Figure 36 shows an example of a multiobjective location problem with attraction and repulsion. Note that $\mu_j(\cdot) = \|\cdot\|_2$ for all $j = 1, \ldots, m_2$ holds. Moreover, Fig. 36 displays the set of all EP-efficient solutions $\mathrm{Eff}(\mathbb{R}^2 \mid g)$, the level lines of the objective function h and the computed solution x^0.

Further model and algorithm-specific information can be found in the following selected literature:

(A) S. Alzorba, C. Günther and N. Popovici. *A special class of extended multicriteria location problems.* Optimization, 64(5):1305-1320, 2015 (DOI: 10.1080/02331934.2013.869810).

(B) C. Günther. *Dekomposition mehrkriterieller Optimierungsprobleme und Anwendung bei nicht konvexen Standortproblemen.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

(C) A. Jourani, C. Michelot and M. Ndiaye. *Efficiency for continuous facility location problems with attraction and repulsion.* Annals OR, 167(1):43-60, 2009.

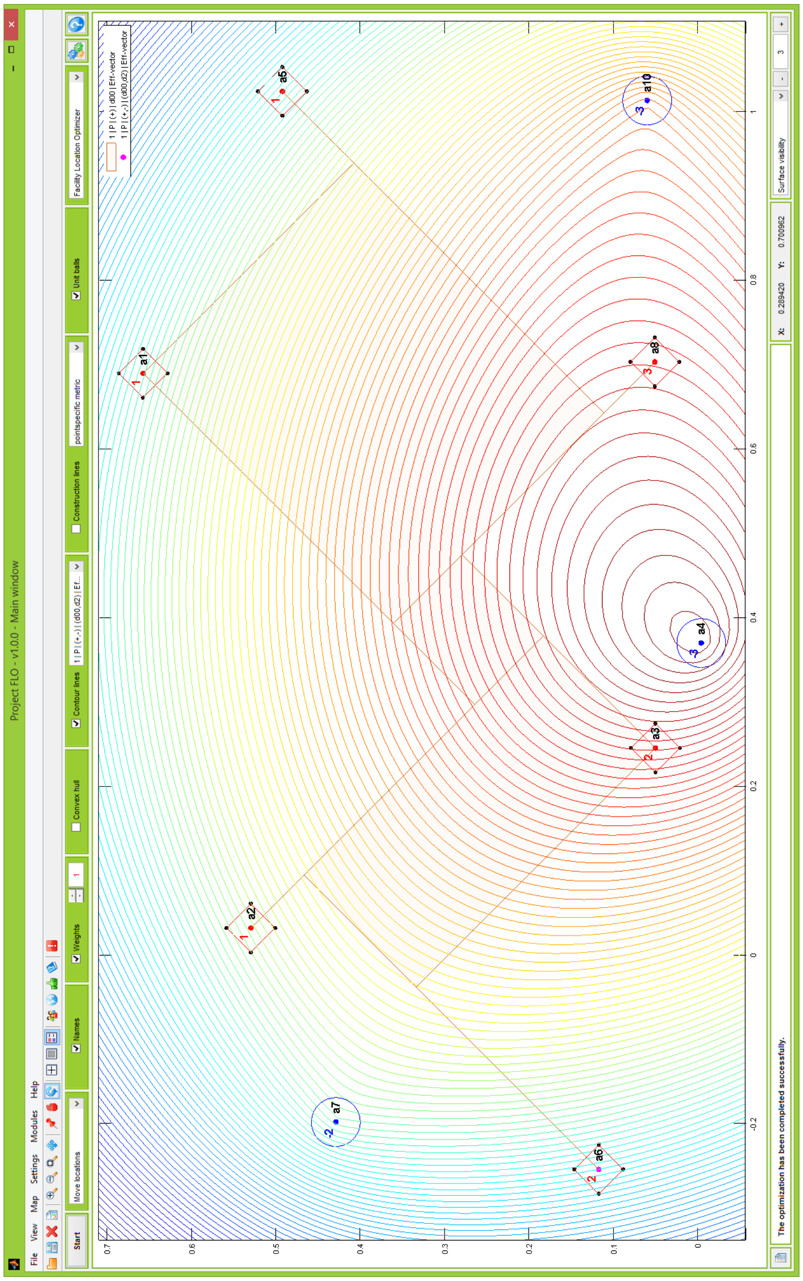


Figure 36: Multiobjective location problem with attraction and repulsion solved with FLO.

5.2 Location problems involving a feasible set represented by a union of polytopes

The feasible set X is given by a finite union of polytopes $P_1, \dots, P_q \subseteq \mathbb{R}^2$ (i.e., bounded polyhedral sets in $\mathbb{R}^2$). Consequently, we have

$$X = \bigcup_{i=1}^{q} P_i.$$

5.2.1 Median location problems with positive weights

We consider m points in the plane,

$$a^1 := (a_1^1, a_2^1),\ \cdots\ , a^m := (a_1^m, a_2^m)\ \in\ \mathbb{R}^2,$$

representing some a priori given facilities. Moreover, let $v_1, \dots, v_m$ be positive weights (the demands for the given facilities).

Now, we search for a new facility $x \in X = \bigcup_{i=1}^{q} P_i$ such that the weighted sum of the distances between the new facility x and the given points $a^1, \dots, a^m$ are minimized.

Note: Constrained median location problems with positive weights and $q = 1$ (i.e., one polytope represents the feasible set) are convex, but for $q > 1$ these problems are non-convex in general (since a finite union of convex sets must not be convex in general).

5.2.1.1 **1** | **P** | $v > 0$, $X = \bigcup_{i=1}^{q} P_i$ | d_1 | **median**

Using the Manhattan metric, defined by

$$d_1(x, a^i) := |x_1 - a_1^i| + |x_2 - a_2^i|$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \dots, m$, we formulate the constrained location problem by

$$\sum_{i=1}^{m} v_i \cdot d_1(x, a^i) = \sum_{i=1}^{m} v_i \cdot (|x_1 - a_1^i| + |x_2 - a_2^i|) \to \min_{x \in X = \bigcup_{i=1}^{q} P_i}$$

with $v_i > 0$ for all $i = 1, \dots, m$.
The corresponding algorithm included in the current version of the Software

FLO generates the whole set of solutions of the above location problem. The program uses a modified and extended version of the algorithm (Boundary Intersection Algorithm) formulated in the book by Hamacher [24, Section 2.1] (1995).

The algorithm was implemented in FLO by Rico Khan during his Bachelor-Thesis (see Reference A). Software FLO has been able to solve the underlying location problem since version 1.2.0 which was released on 08/01/2016.

Further models and algorithm specific information can be found in the following selected literature:

(A) R. Khan. *Algorithmen für restringierte Standortprobleme einschließlich Implementierung*. Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2015.

(B) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme*. Vieweg Verlag, 1995.

5.2.1.2 **1** | **P** | $v > 0$, $X = \bigcup_{i=1}^{q} P_i$ | d_∞ | **median**

Using the Manhattan metric, defined by

$$d_\infty(x, a^i) := \max\{|x_1 - a_1^i|, |x_2 - a_2^i|\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we formulate the constrained location problem by

$$\sum_{i=1}^{m} v_i \cdot d_\infty(x, a^i) = \sum_{i=1}^{m} v_i \cdot \max\{|x_1 - a_1^i|, |x_2 - a_2^i|\} \to \min_{x \in X = \bigcup_{i=1}^{q} P_i}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The corresponding algorithm included in the current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses a modified and extended version of the algorithm (Boundary Intersection Algorithm) formulated in the book by Hamacher [24, Section 2.1] (1995).

The algorithm was implemented in FLO by Rico Khan during his Bachelor-Thesis (see Reference A). Software FLO has been able to solve the underlying location problem since version 1.2.0 which was released on 08/01/2016.

Further models and algorithm specific information can be found in the following selected literature:

(A) R. Khan. *Algorithmen für restringierte Standortprobleme einschließlich Implementierung*. Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2015.

(B) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme*. Vieweg Verlag, 1995.

5.2.1.3 $\mathbf{1} \mid \mathbf{P} \mid v > 0,\ X = \bigcup_{i=1}^{q} P_i \mid \mu \mid$ median

Let B be a polytope (closed, bounded and polyhedral set) in $\mathbb{R}^2$ with $0 \in \operatorname{int} B$ and $B = -B$ (symmetry with respect to the origin). Moreover, assume that the polytope B can be represented by the convex hull of four points $r^1, r^2, r^3, r^4 \in \mathbb{R}^2$, i.e., it holds $B = \operatorname{conv}\{r^1, r^2, r^3, r^4\}$.

Now using the unit ball $B_\mu := B$ we can define a block norm μ. The distances induced by the gauge μ are defined by

$$\mu(x - a^i) := \inf\{\lambda > 0 \,|\, x - a^i \in \lambda \cdot B_\mu\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$. Hence, the location problem is given by

$$\sum_{i=1}^{m} v_i \cdot \mu(x - a^i) \to \min_{x \in X = \bigcup_{i=1}^{q} P_i}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The corresponding algorithm included in the current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses an algorithm (Boundary Intersection ; see Reference A) based on the algorithm for solving the problem "$1 \mid \mathrm{P} \mid v > 0,\ X = \bigcup_{i=1}^{q} P_i \mid d_1 \mid$ median" and an appropriate linear transformation.

The algorithm was implemented in FLO by Rico Khan during his Bachelor-Thesis (see Reference A). Software FLO has been able to solve the underlying location problem since version 1.2.0 which was released on 08/01/2016.

Further models and algorithm specific information can be found in the following selected literature:

(A) R. Khan. *Algorithmen für restringierte Standortprobleme einschließlich Implementierung*. Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2015.

5.2.1.4 $\mathbf{1} \mid \mathbf{P} \mid v > 0,\ X = \bigcup_{i=1}^{q} P_i \mid d_2^2 \mid$ **median**

Using the squared Euclidean metric, defined by

$$d_2^2(x, a^i) := (x_1 - a_1^i)^2 + (x_2 - a_2^i)^2$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we formulate the constrained location problem by

$$\sum_{i=1}^{m} v_i \cdot d_2^2(x, a^i) \to \min_{x \in X = \bigcup_{i=1}^{q} P_i}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The corresponding algorithm included in the current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses a modified and extended version of the algorithm (Boundary Projection Algorithm) formulated in the book by Hamacher [24, Section 2.1] (1995).

The algorithm was implemented in FLO by Rico Khan during his Bachelor-Thesis (see Reference A). Software FLO has been able to solve the underlying location problem since version 1.2.0 which was released on 08/01/2016.

Further models and algorithm specific information can be found in the following selected literature:

(*A*) R. Khan. *Algorithmen für restringierte Standortprobleme einschließlich Implementierung.* Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2015.

(*B*) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

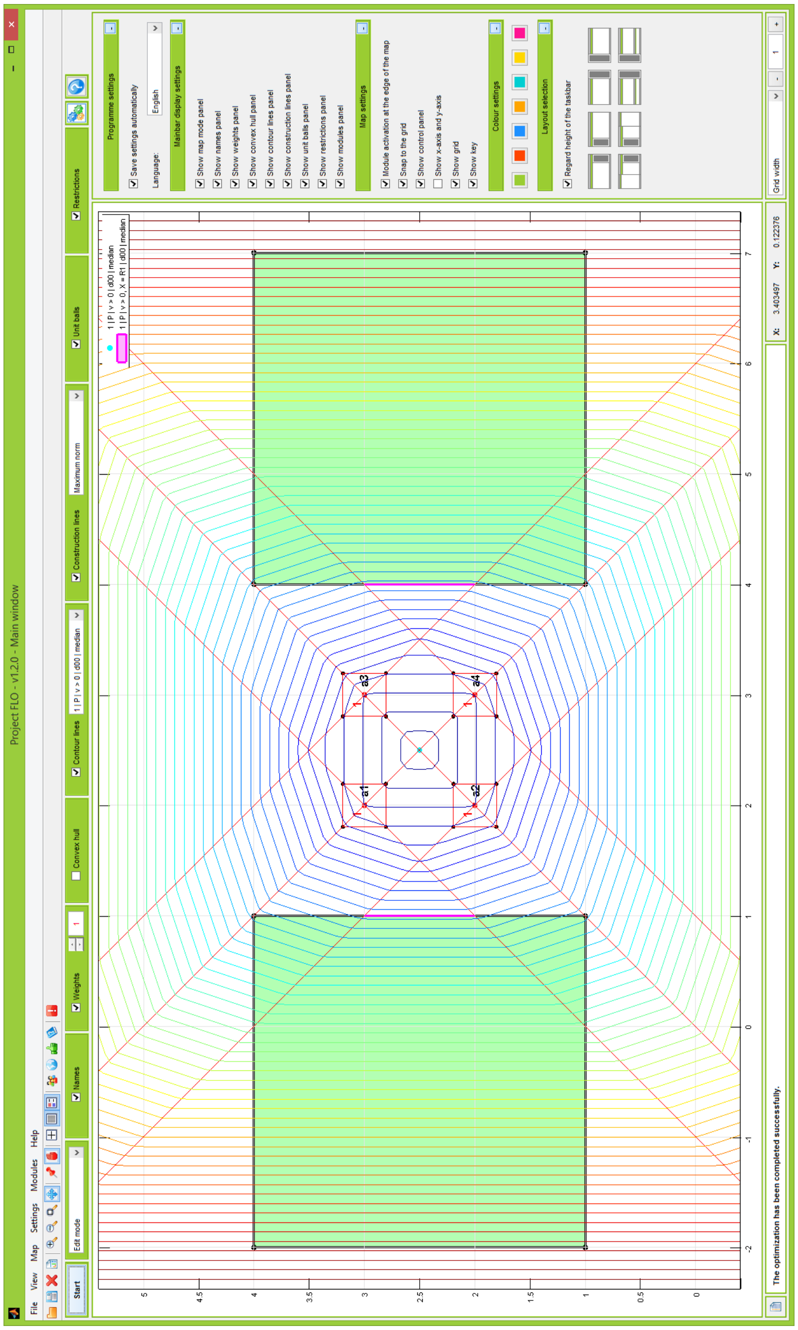


Figure 37: Constrained median location problems solved with FLO.

5.3 Location problems involving a forbidden region

Let $P \subseteq \mathbb{R}^2$ be a polytope (i.e., a bounded polyhedral set) with $\operatorname{int} P \neq \emptyset$, and assume that we have a representation

$$P = \operatorname{conv}\{e^1, \dots, e^l\}$$

for a finite number of extreme points $e^1, \dots, e^l \in \mathbb{R}^2$, $l \in \mathbb{N}$.

The feasible set X is the whole plane but we exclude a special forbidden region defined by the interior of the polytope P, i.e., we have $X = \mathbb{R}^2 \setminus \operatorname{int} P$.

5.3.1 Median location problems with positive weights

We consider m points in the plane,

$$a^1 := (a_1^1, a_2^1), \ \cdots \ , a^m := (a_1^m, a_2^m) \ \in \ \mathbb{R}^2,$$

representing some a priori given facilities. Moreover, let $v_1, \dots, v_m$ be positive weights (the demands for the given facilities).

Now, we search for a new facility $x \in X = \mathbb{R}^2 \setminus \operatorname{int} P$ such that the weighted sum of the distances between the new facility x and the given points $a^1, \dots, a^m$ are minimized.

Note: Median location problems with positive weights involving a forbidden region are non-convex problems, since $X = \mathbb{R}^2 \setminus \operatorname{int} P$ is a non-convex set in $\mathbb{R}^2$.

5.3.1.1 $\mathbf{1} \mid \mathbf{P} \mid v > 0,\ X = \mathbb{R}^2 \setminus \operatorname{int} P \mid d_1 \mid$ **median**

Using the Manhattan metric, defined by

$$d_1(x, a^i) := |x_1 - a_1^i| + |x_2 - a_2^i|$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \dots, m$, we formulate the constrained location problem by

$$\sum_{i=1}^{m} v_i \cdot d_1(x, a^i) = \sum_{i=1}^{m} v_i \cdot (|x_1 - a_1^i| + |x_2 - a_2^i|) \to \min_{x \in X = \mathbb{R}^2 \setminus \operatorname{int} P}$$

with $v_i > 0$ for all $i = 1, \dots, m$.

The corresponding algorithm included in the current version of the Software

FLO generates the whole set of solutions of the above location problem. The program uses a modified and extended version of the algorithm (Boundary Intersection Algorithm) formulated in the book by Hamacher [24, Section 2.1] (1995).

Note: We assume that the solution set of the free problem "1 | P | $v > 0$ | d_1 | median" is completely contained in the interior of the polytope P such that no solution of the free problem is feasible for the restricted problem "1 | P | $v > 0$, $X = \mathbb{R}^2 \setminus \operatorname{int} P$ | d_1 | median".

The algorithm was implemented in FLO by Rico Khan during his Bachelor-Thesis (see Reference A). Software FLO has been able to solve the underlying location problem since version 1.2.0 which was released on 08/01/2016.

Further models and algorithm specific information can be found in the following selected literature:

(*A*) R. Khan. *Algorithmen für restringierte Standortprobleme einschließlich Implementierung*. Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2015.

(*B*) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme*. Vieweg Verlag, 1995.

5.3.1.2 1 | P | $v > 0$, $X = \mathbb{R}^2 \setminus \operatorname{int} P$ | d_∞ | median

Using the Manhattan metric, defined by

$$d_\infty(x, a^i) := \max\{|x_1 - a_1^i|, |x_2 - a_2^i|\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \ldots, m$, we formulate the constrained location problem by

$$\sum_{i=1}^{m} v_i \cdot d_\infty(x, a^i) = \sum_{i=1}^{m} v_i \cdot \max\{|x_1 - a_1^i|, |x_2 - a_2^i|\} \rightarrow \min_{x \in X = \mathbb{R}^2 \setminus \operatorname{int} P}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The corresponding algorithm included in the current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses a modified and extended version of the algorithm (Boundary Intersection Algorithm) formulated in the book by Hamacher [24, Section 2.1] (1995).

Note: We assume that the solution set of the free problem "1 | P | $v > 0$ | d_∞ | median" is completely contained in the interior of the polytope P such that no solution of the free problem is feasible for the restricted problem "1 | P | $v > 0$, $X = \mathbb{R}^2 \setminus \operatorname{int} P$ | d_∞ | median".

The algorithm was implemented in FLO by Rico Khan during his Bachelor-Thesis (see Reference A). Software FLO has been able to solve the underlying location problem since version 1.2.0 which was released on 08/01/2016.

Further models and algorithm specific information can be found in the following selected literature:

(A) R. Khan. *Algorithmen für restringierte Standortprobleme einschließlich Implementierung.* Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2015.

(B) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

5.3.1.3 1 | P | $v > 0$, $X = \mathbb{R}^2 \setminus \operatorname{int} P$ | μ | median

Let B be a polytope (closed, bounded and polyhedral set) in $\mathbb{R}^2$ with $0 \in \operatorname{int} B$ and $B = -B$ (symmetry with respect to the origin). Moreover, assume that the polytope B can be represented by the convex hull of four points $r^1, r^2, r^3, r^4 \in \mathbb{R}^2$, i.e., it holds $B = \operatorname{conv}\{r^1, r^2, r^3, r^4\}$.

Now using the unit ball $B_\mu := B$ we can define a block norm μ. The distances induced by the gauge μ are defined by

$$\mu(x - a^i) := \inf\{\lambda > 0 \,|\, x - a^i \in \lambda \cdot B_\mu\}$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a^i_1, a^i_2)$, $i = 1, \ldots, m$. Hence, the location problem is given by

$$\sum_{i=1}^{m} v_i \cdot \mu(x - a^i) \to \min_{x \in X = \mathbb{R}^2 \setminus \operatorname{int} P}$$

with $v_i > 0$ for all $i = 1, \ldots, m$.
The corresponding algorithm included in the current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses an algorithm (Boundary Intersection Algorithm; see Reference

A) based on the algorithm for the problem "1 | P | $v > 0$, $X = \mathbb{R}^2 \backslash \operatorname{int} P$ | d_1 | median" and an appropriate linear transformation.

Note: We assume that the solution set of the free problem "1 | P | $v > 0$ | μ | median" is completely contained in the interior of the polytope P such that no solution of the free problem is feasible for the restricted problem "1 | P | $v > 0$, $X = \mathbb{R}^2 \setminus \operatorname{int} P$ | μ | median".

The algorithm was implemented in FLO by Rico Khan during his Bachelor-Thesis (see Reference A). Software FLO has been able to solve the underlying location problem since version 1.2.0 which was released on 08/01/2016.

Further models and algorithm specific information can be found in the following selected literature:

(*A*) R. Khan. *Algorithmen für restringierte Standortprobleme einschließlich Implementierung*. Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2015.

5.3.1.4 1 | P | $v > 0$, $X = \mathbb{R}^2 \setminus \operatorname{int} P$ | d_2^2 | median

Using the squared Euclidean metric, defined by

$$d_2^2(x, a^i) := (x_1 - a_1^i)^2 + (x_2 - a_2^i)^2$$

for all $x := (x_1, x_2) \in \mathbb{R}^2$ and all $a^i := (a_1^i, a_2^i)$, $i = 1, \dots, m$, we formulate the constrained location problem by

$$\sum_{i=1}^{m} v_i \cdot d_2^2(x, a^i) \to \min_{x \in X = \mathbb{R}^2 \setminus \operatorname{int} P}$$

with $v_i > 0$ for all $i = 1, \dots, m$.

The corresponding algorithm included in the current version of the Software FLO generates the whole set of solutions of the above location problem. The program uses a modified and extended version of the algorithm (Boundary Intersection Algorithm) formulated in the book by Hamacher [24, Section 2.1] (1995).

Note: We assume that the solution set of the free problem "1 | P | $v > 0$ | d_2^2 | median" is completely contained in the interior of the polytope P such that no solution of the free problem is feasible for the restricted problem "1 | P | $v > 0$, $X = \mathbb{R}^2 \setminus \operatorname{int} P$ | d_2^2 | median".

The algorithm was implemented in FLO by Rico Khan during his Bachelor-Thesis (see Reference A). Software FLO has been able to solve the underlying location problem since version 1.2.0 which was released on 08/01/2016.

Further models and algorithm specific information can be found in the following selected literature:

(*A*) R. Khan. *Algorithmen für restringierte Standortprobleme einschließlich Implementierung*. Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2015.

(*B*) H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme*. Vieweg Verlag, 1995.

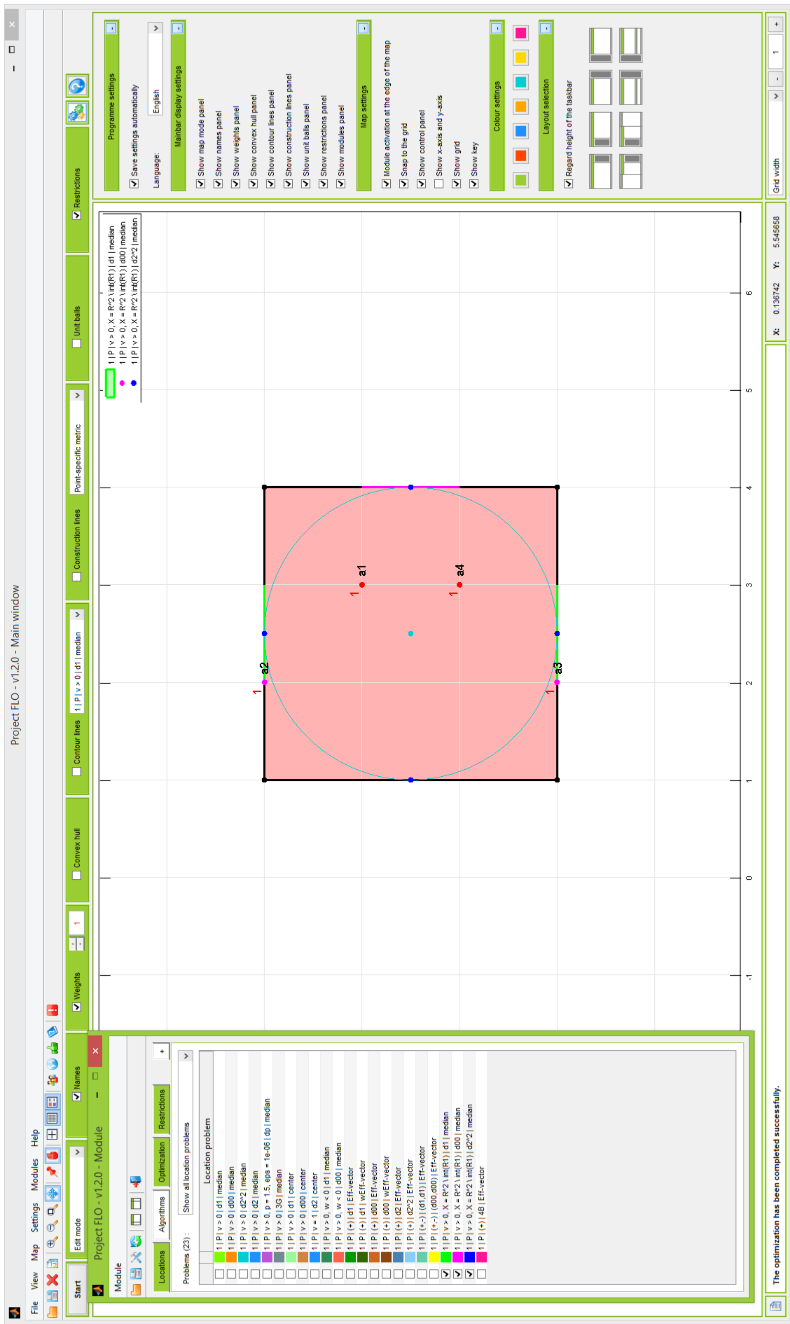


Figure 38: Median location problems involving a forbidden region solved with FLO.

References

[1] S. Alzorba and C. Günther. *Algorithms for Multicriteria Location Problems.* In: Numerical Analysis and Applied Mathematics (ICNAAM), AIP Conf. Proc., 1479:2286-2289, 2012.

[2] S. Alzorba, C. Günther and N. Popovici. *A special class of extended multicriteria location problems.* Optimization, 64(5):1305-1320, 2015 (DOI: 10.1080/02331934.2013.869810).

[3] S. Alzorba, C. Günther, N. Popovici and Chr. Tammer. *A new algorithm for solving planar multiobjective location problems involving the Manhattan norm.* Preprint Optimization-Online, http://www.optimization-online.org/DB_HTML/2016/01/5305.html, 2015 (submitted).

[4] A. Beck and S. Sabach. *Weiszfelds Method: Old and New Results.* Journal of Optimization Theory and Applications, 164(1):1-40, 2015.

[5] A. E. Bindschedler and J. M. Moore. *Optimal Location of New Machines in Existing Plant Layouts.* The Journal of Industrial Engineering, 12:41-47, 1961.

[6] M. Bischoff. *Location of Connection Facilities.* Verlag Shaker, Aachen, 2008.

[7] J. Brimberg, J. H. Walker and R. F. Love. *Estimation of travel distances with the weighted p norm: Some empirical results.* Journal of Transport Geography, 15(1):6272, 2007.

[8] L. Chalmet, R. L. Francis, A. Kolen. *Finding efficient solutions for rectilinear distance location problems efficiently.* European Journal of Operational Research, 6:117-124, 1981.

[9] Z. Drezner and H. W. Hamacher. *Facility Location: Theory and Algorithms.* Springer, Berlin, 2001.

[10] R. Durier and C. Michelot. *Geometrical properties of the Fermat-Weber problem.* European Journal of Operational Research, 20:332-343, 1985.

[11] F. Y. Edgeworth. *Mathematical psychics.* Kegan Paul, London, 1881.

[12] M. Ehrgott. *Multicriteria Optimization.* Springer Berlin - Heidelberg, 2nd Edition, 2005.

[13] G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization.* Springer, 2008.

[14] G. Eichfelder. *Variable Ordering Structures in Vector Optimization.* Springer, 2014.

[15] D. J. Elzinga and D. W. Hearn. *Geometrical Solutions for some minimax location problems.* Transportation Science, 6:379-394, 1972.

[16] R. L. Francis. *A Note on the Optimum Location of New Machines in Existing Plant Location.* AIIE Transactions, 14(1):57-59, 1963.

[17] Chr. Gerth (Tammer) and K. Pöhler. *Dualität und algorithmische Anwendung beim vektoriellen Standortproblem.* Optimization, 19:491-512, 1988.

[18] A. Göpfert, H. Riahi, Chr. Tammer and C. Zălinescu. *Variational Methods in Partially Ordered Spaces.* CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, 17, Springer-Verlag, New York, 2003.

[19] A. Göpfert, T. Riedrich and Chr. Tammer. *Angewandte Funktionalanalysis Motivationen und Methoden für Mathematiker und Wirtschaftswissenschaftler.* Vieweg+Teubner, Wiesbaden, 2009.

[20] C. Günther. *Standort-Medianprobleme mit variablen Anlagen.* Bachelor-Thesis, Martin Luther University Halle-Wittenberg, 2011.

[21] C. Günther. *Dekomposition mehrkriterieller Optimierungsprobleme und Anwendung bei nicht konvexen Standortproblemen.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

[22] C. Günther, N. Popovici and Chr. Tammer. *Reducing the complexity of planar multiobjective location problems.* 2016 (in preparation).

[23] C. Günther and Chr. Tammer. *Relationships between constrained and unconstrained multi-objective optimization and application in location theory.* Preprint Optimization-Online, http://www.optimization-online.org/DB_HTML/2015/11/5196.html, 2015 (submitted).

[24] H. W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg Verlag, 1995.

[25] H. W. Hamacher, K. Klamroth and Chr. Tammer. *Standortoptimierung.* In: B. Luderer (Ed.). *Die Kunst des Modellierens. Mathematisch-ökonomische Modelle.* Teubner-Verlag, 139-156, 2008.

[26] H. W. Hamacher and S. Nickel. *Classification of Location Models.* Location Science, 6:229-242, 1998.

[27] M. Hillmann. *Lagrange-Multiplikatoren-Regeln und Algorithmen für nichtkonvexe Standortprobleme.* Master-Thesis, Martin Luther University Halle-Wittenberg, 2013.

[28] J. Jahn. *Vector Optimization - Theory, Applications, and Extensions.* Springer, Berlin, 2nd Edition, 2011.

[29] A. Jourani, C. Michelot and M. Ndiaye. *Efficiency for continuous facility location problems with attraction and repulsion.* Annals OR, 167(1):43-60, 2009.

[30] H. Juel and R. F. Love. *Hull properties in location problems.* European Journal of Operational Research, 12:262-265, 1983.

[31] M. Kaiser. *Spatial Uncertainties in Continuous Location Problems.* Dissertation, Bergische Universität Wuppertal, 2015.

[32] A. A. Khan, Chr. Tammer and C. Zălinescu. *Set-valued Optimization: An Introduction with Applications.* Springer, 2015.

[33] H. W. Kuhn. *On a pair of dual nonlinear programs.* Nonlinear programming, Wiley, New York, 1967.

[34] A. Löhne. *Vector Optimization with Infimum and Supremum.* Springer, 2011.

[35] A. Löhne and B. Weißing. *Bensolve - VLP solver*, version 2.0.x, www.bensolve.org

[36] R. F. Love, J. G. Morris and G. O. Wesolowsky. *Facility Location: Models and Methods.* North Holland, New York, 1988.

[37] N. Megiddo. *The weighted Euclidean 1-center problem.* Mathematics of Operations Research, 8(4):498504, 1983.

[38] J. G. Morris and W. A. Verdini. *Minisum, l_p distance location problems solved via a perturbed problem and Weiszfeld's algorithm.* Operations Research, 27, 1180-1188, 1979.

[39] S. Nickel. *Discretization of Planar Location Problems.* Verlag Shaker, Aachen, 1995.

[40] S. Nickel and E. M. Dudenhöffer. *Weber's Problem with Attraction and Repulsion under Polyhedral Gauges.* Journal of Global Optimization, 11:409-432, 1997.

[41] S. Nickel, J. Puerto and A. M. Rodríguez-Chí. MCDM location problems. In: J. Figueira, S. Greco and M. Ehrgott (Ed.). *Multiple criteria decision analysis: State of the art surveys.* Springer, 761–795, 2005.

[42] S. Nickel, J. Puerto and A. M. Rodríguez-Chí. Location problems with multiple criteria. In: G. Laporte, S. Nickel and F. Saldanha da Gama (Ed.). *Location Science.* Springer, 205–247, 2015.

[43] K. Nouioua. *Enveloppes de Pareto et Reseaux de Manhattan.* Dissertation, University of the Mediterranean, 2005.

[44] V. Pareto. *Manuale di economia politica.* Societa Editrice Libraria, Milano, Italy, 1906.

[45] N. Popovici. *Pareto reducible multicriteria optimization problems.* Optimization, 54:253-263, 2005.

[46] A. Wagner. *A new Duality Based Approach for the Problem of Locating a Semi-Obnoxious Facility.* Dissertation, Martin-Luther-Universität Halle-Wittenberg, 2014.

[47] A. Wagner, J. E. Martinez-Legaz and Chr. Tammer. *Locating a Semi-Obnoxious Facility - A Toland-Singer Duality Based Approach.* Journal of Convex Analysis, 2015 (accepted).

[48] A. Weber. *Über den Standort der Industrien.* Tübingen, 1909.

[49] E. V. Weiszfeld. *Sur le point pour lequel la somme des distances de n points donnes est minimum.* Tohoku Math, 43:355386, 1937.

[50] R. E. Wendell, A. P. Hurter and T. J. Lowe. *Efficient Points in Location Problems.* AIIE Transactions, 9(3):238-246, 1977.

[51] J. A. White. *A Quadratic Facility Location Problem.* AIIE Transactions, 3(2):156-157, 1971.