Paper

Reference distribution based decision support platform

Krzysztof Bareja and Włodzimierz Ogryczak

Abstract—There many decision problems where numerous partial achievement functions are considered impartially which makes the distribution of achievements more important than the assignment of several achievements to the specific criteria. Such models are generally related to the evaluation and optimization of various systems which serve many users where quality of service for every individual user defines the criteria. This applies to various technical systems, like to telecommunication ones among others, as well as to social systems. An example arises in location theory, where the clients of a system are entitled to equal treatment according to some community regulations. This paper presents an implementation of decision support framework for such problems. This platform is designed for multiple criteria problems analyzed with the reference distribution approach. Reference distribution approach is an extension of the reference point method.

Keywords— multiple criteria optimization, decision support systems, reference point method, reference distribution method.

1. Introduction

In various systems which serve many users there is a need to respect the fairness rules while optimizing the total system efficiency. In such problems, the decisions often concern meeting the users' demands in an impartial way. Thus, we are interested rather in distributions of outcomes than specific outcomes themselves. For instance, having two possible location patterns generating for 3 clients outcome vectors (5,0,5) and (0,1,0), respectively, we would recognize both the location patterns as efficient in terms of outcomes (distance) minimization. Indeed, neither (5,0,5) dominates (0,1,0) nor (0,1,0) dominates (5,0,5). However, the first location pattern generates two outcomes (distances) equal to 5 and one outcome equal to 0, whereas the second pattern generates one outcome equal to 1 and two outcomes equal to 0. Thus, in terms of the distribution of outcomes the second location pattern is clearly better. This applies to the desired system output (amount, quality of services) as well as to the obnoxious outcomes (like risk exposure, pollutions). The so-called minimax solution concept, where the worst individual effect (maximum individual disutility) is minimized, is usually considered as the simplest fair optimization model. The minimax approach is consistent with Rawlsian theory of justice [1], especially when additionally regularized with the lexicographic order. On the other hand, making the decisions to optimize the worst individual disutility may cause a large worsening of the overall (mean) performances. Therefore, several other fair decision schemes are searched and analyzed [2, 3, 4].

In this paper we use an alternative concept of the conditional mean which is a parametric generalization of the worst outcome taking into account the portion of population (demands) affected by the worst effects [5]. Namely, for a specified portion β of population we take into account the entire β portion (quantile) of the worst outcomes and we consider their average as the (worst) conditional β-mean outcome. According to this definition the concept of conditional mean is based on averaging restricted to the portion of the worst outcomes. When parameter β approaches 0, the conditional β-mean tends to the worst outcome. On the other hand, for $\beta = 1$ the corresponding conditional mean becomes the standard mean. We select several conditional means for various levels $\beta_k (k = 1, 2, \dots K)$ to get a multiple criteria model, allowing to generate various fair efficient solutions.

Usually there exist many nondominated achievement vectors and they are incomparable with each other on the basis of the specified set of objective functions. Therefore, usually there exist many efficient solutions and they are different not only in the decision space but also in the criteria space. So, there arises a need for further analysis, or rather decision support, to help the decision maker (DM) in the selection of one solution for implementation. Of course, the original objective functions do not allow one to select any efficient solution as better than any other one. Therefore, this analysis depends usually on additional information about the DM's preferences. The DM, working interactively with a decision support system (DSS), specifies the preferences in terms of some control parameters and the DSS provides the DM with an efficient solution which is the best according to the specified control parameters. For such an analysis, there is no need to identify the entire efficient set prior to the analysis, since contemporary optimization software is powerful enough to be used online for direct computations at each interactive step. Thus the DSS can generate at each interactive step only one solution that meets the current preferences. Such a DSS can be used for the analysis of decision problems with finite as well as infinite efficient sets. In order to allow the DSS to meet various DM's preferences it is important, however, that the control parameters provide the completeness of the control (c.f., [6]), i.e., that by varying the control parameters the DM can identify every nondominated achievement vector.

Good controllability can be achieved with the direct use of the reference point methodology (RPM) introduced by Wierzbicki [7] and later extended leading to efficient implementations with many successful applications [8].

The RPM approach is an interactive technique allowing the DM to specify the requirements in terms of aspiration and reservation levels, i.e., by introducing acceptable and required values for several criteria. Depending on the specified aspiration and reservation levels, a special scalarizing achievement function is built which may be directly interpreted as expressing utility to be maximized. Maximization of the scalarizing achievement function generates an efficient solution to the multiple criteria problem. The solution is accepted by the DM or some modifications of the aspiration and reservation levels are introduced to continue the search for a better solution. The RPM approach provides a complete parameterization of the efficient set to multicriteria optimization. Hence, when applying the ARBDS (aspiration-reservation based decision support) methodology to the conditional mean criteria, one may generate various fair and efficient solutions. Since the our criteria defined as conditional means depends only on distribution of outcomes the RPM approach represent actually a reference distribution technique.

2. Sample model: location problem

Presented methodology will be shown by sample problem construction process. As example, we have decided to use well known maximin location problem (LP) variant. This problem is well described on literature (e.g., [9]). In this type of problems we can consider each distance between demand point and location as single criterion. Note, that for DM values are not distinguishable.

2.1. Basics

Let us consider a well known problem of maximum demand point-facility distance. Subject function can be written as $\min\big(\max(d_{ij}x_{ij})\big)$ and LP problem form:

$$\min : z\,.$$

Subject to:

$$
\begin{array}{lll}
z \ge y_i & \underset{i}{\forall}\,, & \\
y_i = \sum\limits_j d_{ij}x_{ij} & \underset{i}{\forall} & (*)\,,\\
\sum\limits_{j=1}^{J} x_{ij} = 1 & \underset{i}{\forall} & (*)\,,\\
\sum\limits_{i=1}^{I} f_i \le F & \underset{i}{\forall} & (*)\,,\\
x_{ij} - f_j \le 0 & \underset{ij}{\forall} & (*)\,,
\end{array}
$$

where:

$i \in \{1 \ldots I\}$ – index of demand point;

$j \in \{1 \ldots J\}$ – index of possible facility location;

f_j – (binary variable): 1 if location j was chosen to place facility; 0 otherwise;

x_{ij} – (binary variable): 1 if location i is being served by facility located at j;

y_i – distance from demand point i to facility serving this point;

d_{ij} – distance between demand point i and location j; may be also understood as cost;

F – maximal number of facilities (simple restriction while we don't know about location costs).

Equations marked with asterisk are defining attainable set for variables and in further models will be noted as $[x],[y],[f] \in Q$.

In such formulations it is possible that a single demand node location can influence final result providing to solutions worse for majority of location points. Such situation is illustrated on Fig. 1.

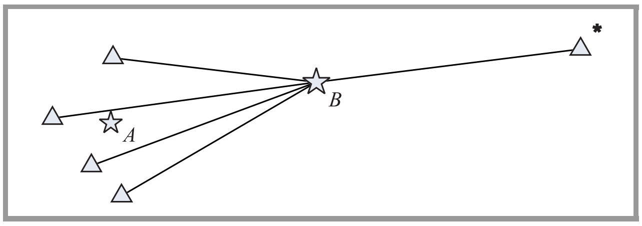

Fig. 1. Unfair domination example.

Chosen facility is considered as worse for almost every demand node. Final result is preferred only for single isolated node in upper right corner of Fig. 1.

2.2. Conditional achievement function

Avoiding "minority dictatorship" described in previous section we can consider value-at-risk like conditional measures. General concept of such models is to minimize the kth worst distance. In example presented above minimizing only 2nd worst distance would result the location A to be preferred.

Simplest formulation of this problem can be written using additional set of binary variables, used for defining set of k worst distances:

$$\min : u_k\,.$$

Subject to:

$$
\begin{array}{l}
u - y_i \ge -Kz_i\,,\\
\sum\limits_{i=1}^{I} z_k \le k-1\,,\\
[x],[y],[f] \in Q\,,
\end{array}
$$

where:

z_i – (binary variable): 1 if distance y_i belongs to $k+1$ maximal distances; 0 otherwise;

K – large value used to switching $u \ge y_i$ inequality;

u – additional variable, equal to kth worst distance in optimal solution;

k – number of value minimized in lexicographically sorted set of distances.

Unfortunately increasing the number of binary variables increases also complexity of presented problem. We can define simpler model based on minimizing total sum of k worst (largest) distances. This approach provides to linear model after transformation to dual problem:

$$\min : \eta \,.$$

Subject to:

$$\eta = \sum_{i=1}^{I} dd_i + ku \qquad \underset{j}{\forall}\,,$$

$$u + dd_i \geq y_i \qquad \underset{i}{\forall}\,,$$

$$dd_i \geq 0 \qquad \underset{i}{\forall}\,,$$

$$[x], [y], [f] \in Q\,,$$

where:

dd_i – downside semideviation between maximal distance and distance node-facility for node i;

η – sum of k worst values in optimal solution.

2.3. Multiple criteria model

By extending concept presented in previous section it is possible to split whole spectrum of distances and minimize every part separately. Granularity of distance distribution description depends on user preferences and in ultimate precision each distance can be modified separately. However, such precision is usually unnecessary. It can also provide to too big model complexity. For example, only number of, i.e., "big", "medium" and "small" values controlling is sufficient.

As result we can consider a model which is "multiplication" of the model described in previous section. General achievement function of this model is not defined yet, but is must be function of conditional measures vector – separate for each interval defined by user:

$$\min : \{\eta_k\}\,.$$

Subject to:

$$\eta_k = \sum_{i=1}^{I} dd_{ik} + I(k)u_k \qquad \underset{k}{\forall}\,,$$

$$y_i = \sum_{j} d_{ij}x_{ij} \qquad \underset{i}{\forall}\,,$$

$$u_k + dd_i \geq y_i \qquad \underset{i,k}{\forall}\,,$$

$$dd_i \geq 0 \qquad \underset{i}{\forall}\,,$$

$$[x], [y], [f] \in Q\,,$$

where:

k – the index of interval in lexicographically sorted set of distances;

$I(k)$ – position of demand node related with kth interval.

Each value $I(k)$ is related to respondent partial criteria. In example presented on Fig. 2: $I(0) = 10$ and $I(1) = 20$.

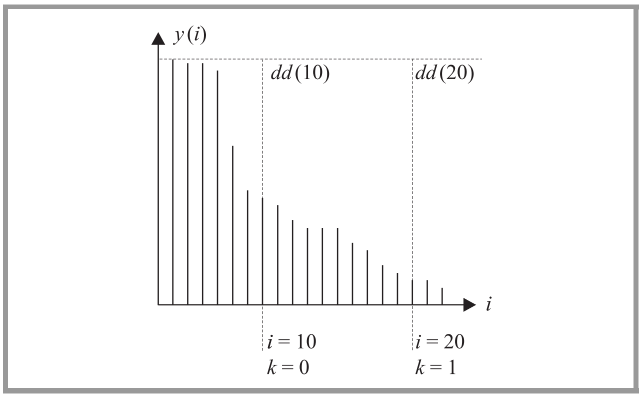

Fig. 2. Example of distances set division.

For the final model two more steps are required: information about user preferences about each interval and some type of scalarization function. While result of given constraint set is a vector of representation, we can treat this vector as set of separate criteria and use the reference point method for this problem.

2.4. Reference point method

For each partial goal we can model preferences using two parameters: level of aspiration (achievement that is fully satisfying decision maker, without need to optimize given criteria) and reservation (minimal). Hence, we can use piecewise linear achievement function, with shape presented in Fig. 3.

Such function can be written as

$$\varphi_k = \max\left[f_1(\eta_k); f_2(\eta_k); f_3(\eta_k)\right]$$

and in terms of LP constraints:

$$\varphi_k \geq a\eta_k - aA_k \qquad \underset{k}{\forall}\,,$$

$$\varphi_k \geq \left(\frac{1}{R_k^+ - A_k}\right)\eta_k + \left(\frac{A_k}{A_k - R_k^+}\right) \qquad \underset{k}{\forall}\,,$$

$$\varphi_k \geq b\eta_k - bR_k^+ + 1 \qquad \underset{k}{\forall}\,,$$

where:

A_k – aspiration value for interval k;

R_k^+ – reservation value for interval k;

a – parameter related to reward for achievement lower than decision maker's aspiration point (value between 0 and 1, arbitrarily defined);

b – parameter related to punishment for achievement bigger than decision maker's reservation point (value greater than 1, arbitrarily defined).

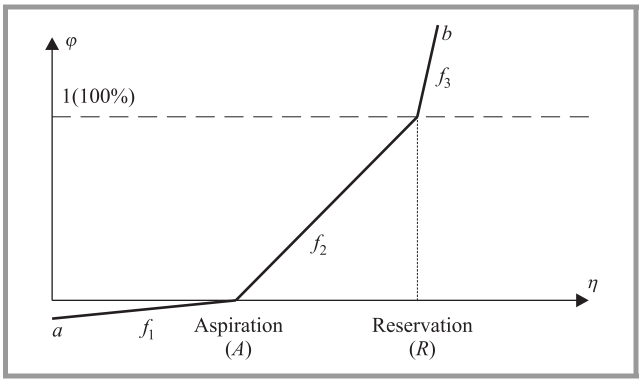

Fig. 3. Piecewise achievement function for single criteria.

Typical generic scalarizing achievement function takes the following form:

$$\min : \max\left(\varphi_k\right) + \varepsilon \sum_{k=1}^{K} \varphi_k$$

and will be used in implemented model.

2.5. User preferences definition

Unfortunately, aspiration and reservation values used in given equations are corresponding to cumulative value of k greatest distances. Thus, aspiration and reservation values should be understood in terms of "total sum of distances larger than kth one". This approach is not intuitive solution for decision maker. We've decided allow user to define preferences in terms of minimal/maximal *average* value in quantile interval (α_k, ρ_k). Proposed translation function is basing on defining increments between beginning and end of described interval. This can be formulated by recursion:

$$A_k = A_{k-1} + \alpha_k\big(I(k) - I(k-1)\big),$$

$$R_k = R_{k-1} + \rho_k\big(I(k) - I(k-1)\big),$$

with initial step:

$$A_0 = \alpha_0 I(0),$$

$$R_0 = \rho_0 I(0).$$

This approach provides to cross dependency between each criterion, and may provide to instability of compromise solution searching process. However, testing the stability is one of implementation goals and this formula may change during test phase.

2.6. Final model formulation

Due to given information final form of LP model is:

$$\min : z + \varepsilon \sum_{k \in (1 \ldots I)} \varphi_k.$$

Subject to:

$$z \geq \varphi_k \qquad \forall_k, \qquad (1)$$

$$\varphi_k \geq a\eta_k - aA_k \qquad \forall_k, \qquad (2)$$

$$\varphi_k \geq \left(\frac{1}{R_k^+ - A_k}\right)\eta_k + \left(\frac{A_k}{A_k - R_k^+}\right) \qquad \forall_k, \qquad (3)$$

$$\varphi_k \geq b\eta_k - bR_k^+ + 1 \qquad \forall_k, \qquad (4)$$

$$\eta_k = \sum_{i=1}^{I} dd_{ik} + ku_k \qquad \forall_k, \qquad (5)$$

$$y_i = \sum_j d_{ij}x_{ij} \qquad \forall_i, \qquad (6)$$

$$u_k + dd_i \geq y_i \qquad \forall_{i,k}, \qquad (7)$$

$$dd_i \geq 0 \qquad \forall_i, \qquad (8)$$

$$y_i = \sum_j d_{ij}x_{ij} \qquad \forall_i, \qquad (9)$$

$$\sum_{j=1}^{J} x_{ij} = 1 \qquad \forall_i, \qquad (10)$$

$$\sum_{i=1}^{I} f_i \leq F \qquad \forall_i, \qquad (11)$$

$$x_{ij} - f_i \leq 0 \qquad \forall_{ij}, \qquad (12)$$

where constraints (1)–(4) are responsible for reference point method criteria scalarization, (5)–(8) for definition of criteria η_k and (9)–(12) – for setting up attainable set for variables.

2.7. Model construction conclusions and reservations

In this section we have shown methodology of constructing LP models that allows optimize described class of problems by specialized solver software. However, there are three more issues that require additional discussion.

- Reservation and aspiration values used for piecewise achievement function definition for any particular criterion (c.f. Subsection 2.4) can take any values. Especially there is no condition that aspiration nor reservation values must summarize to one. This fact does not allow to call presented preferences model as distribution.

- As described in Subsection 2.3 single minimized achievement function for interval k represents not distances from defined interval, but from greatest distance to distance represented by kth interval extreme. As result of this fact, implemented preferences model is only approximation. Precision of this approximation increases with number of defined intervals. Preferences model can be assumed as precise for number of intervals equal to number of demand points.

- Construction of partial achievement function definition suggests possible problems with model controllability for the highest distances intervals as affected by every partial achievement function.

3. Refbeans application: multiple criteria decision support framework

Refbeans is an application implemented to presented preferences modeling approach tests. Main purposes of its creation are:

- Methodology demonstration.
- Methodology behavior verification in real life problems, including:
  - attempt to find user friendly optimization process implementation,
  - attempt to find most user friendly way of preferences definition,
  - model controllability verification.
- Creation of easily extensible platform that can serve as a base for future model implementations.

3.1. Project overview

We have decided to implement application as standalone desktop application. Assumed LP form of solved problems allowed to separate frontend part of application, responsible for user interface and data management and library responsible for optimization.

Java language was chosen for base application implementation – mostly because its popularity, proven performance. Big advantage of this choice is existence of two powerful libraries for rich client application implementation: Eclipse RCP and Netbeans (NB) platform. Both libraries has grown from Java IDE's (integrated development environments) and provides wide set of functions related with multi-document edition and management. Our final choice was to make Netbeans as base for our application. It is using native for Java Swing library, what allows to use wide set of existing graphical user interface (GUI) components. JFreeChart library used for chart generation is good example. As persistence layer file storage with extensive use of Java serialization mechanism was used.

Refbeans platform does not provide any LP model implementation. Each LP model must be implemented in as separate Netbeans plugin project, which should depend on framework specific service provider interfaces (SPI's). However at current moment only sample location problem is implemented. Basic platform can be understood as framework which provides common functionality for separate problem implementation. This includes:

1) support for distribution-based preferences definition in three options: graphical input for distribution and cumulative distribution and table form distribution input;
2) support for management of multiple project types, including project data persistence;
3) support for solver communication, including object – MPS translations and data analysis;
4) calculation history management.

As mentioned before, optimization process is performed by external solver, which has to be installed separately. While communication between application and solver is being performed by the MPS file format virtually any solver can be used. However, we recommend GNU linear programming kit (GLPK) [10] solver as first choice while all calculations and tests were performed with this library.

3.2. Application user interface

Application is implemented as stand-alone program. Depending on distribution it can be launched as local program or via Java WebStart technology [11]. User interface is basing on concepts provided by Netbeans platform and similar to other document-oriented applications. Central part of the main window contains document area (Fig. 4). Each opened project is represented by single tab. Detailed single project tab layout, including sub-tabs, charts and tables depends on project type implementation. At system start user is greeted by simple welcome message.

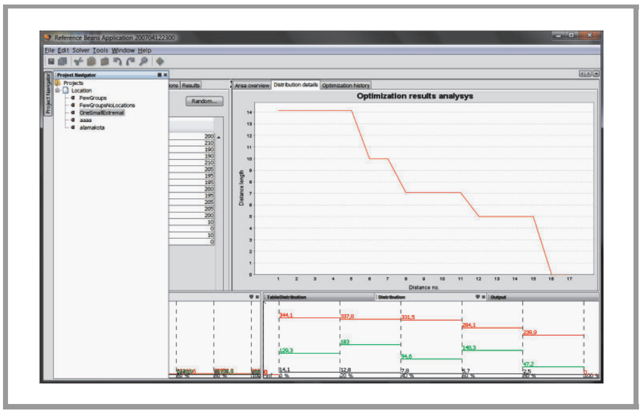

Fig. 4. Application GUI overview.

Project navigator view (minimized, accessible by button on the left window side in default layout) allows to manage prepared projects. Standard CRUD (create, read, update and delete) operations are implemented. Additionally it is possible to export/import project files, what allows to transfer calculation result between computers.

Additional views are coordinated with currently active project. Most important three are being used to specify user criteria. User has possibility to define preferences by distribution chart adjustment, cumulative distribution chart and table – style detailed preferences specification. This allows to test "user friendliness" of different preferences specification process which is one of application implementation goals. Additionally output view is implemented for providing detailed information about optimization process. This includes mps file view, process performance data view and others. General assumption made on user interface (UI) and code design was, that exact definition of partial achievement parameters is defined by model. This includes definition of intervals extremes. In other words choice if user is minimizing separately worst 10% criteria and rest of 90% separately of worst 90% and 10% best is implemented specific by project type implementation.

Optimization process is being started from main application toolbar. Solver connection must be configured before. This action is available through menu bar.

3.3. Location model implementation

Using common for rich client extendable architecture, application can easily support many types of projects without core source code modifications. Additional modules are implemented as plugins, handled by internal NB platform libraries. First model implemented within presented platform is location problem variant described in Section 2. It operated on two dimensional area with geometrical distance definition. In this implementation user is allowed to define set of demand points and possible facility location by specifying their coordinates (Fig. 5).

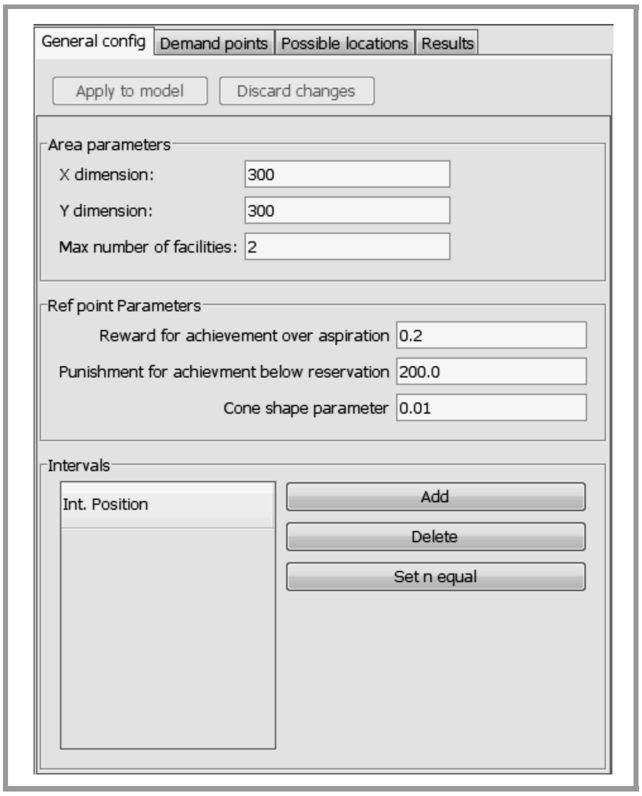

Fig. 5. Location problem parameter definition panel.

Detailed set of parameters describing single problem defined in project implementation includes:

- Area description parameters:
  - area size (width and length),
  - maximum numbers of facilities, that can be established on whole area,
  - list of demand points (X and Y coordinates for each),
  - list of possible facility locations (X and Y coordinates for each).
- Reference point methodology related parameters:
  - reward for partial achievement below aspiration level,
  - "punishment" for exceeding reservation level by partial achievement variable,
  - cone shape related parameter.
- User preferences related parameters:
  - definition of reference distribution discretization points (in terms of percent of total number of criteria/demand points),
  - aspiration and reservation value for each interval.

Detailed data presentation was implemented. User can visualize defined sets of demand and location points on specialized diagram. Although detailed charts of partial achievement goals distribution and optimization history are available.

4. Numerical experiments

4.1. Test data sets

For experimental reasons three artificial data sets were created. First two of them intend to imitate simple location situations for presentation model decisions. Example three and four includes randomly generated larger sets of data for testing model controllability and performance.

Single extreme group case (Fig. 6). This case is created for simple presentation of model ability to act as simple conditional risk models. It was run with one max allowed number of facility location.

Multiple groups of high demand case (Fig. 7). This dataset defines number of separated group of demand points. Available locations are randomly spread all over defined area (with equal probability in every point of area) to make illusion of free choice of location placement.

Volatile demand density model case (Fig. 8). This dataset is intend to check if model is able to recognize areas of high demand points density in noisy environment. Dataset was created by adding large amount of demand points randomly spread all over the area.

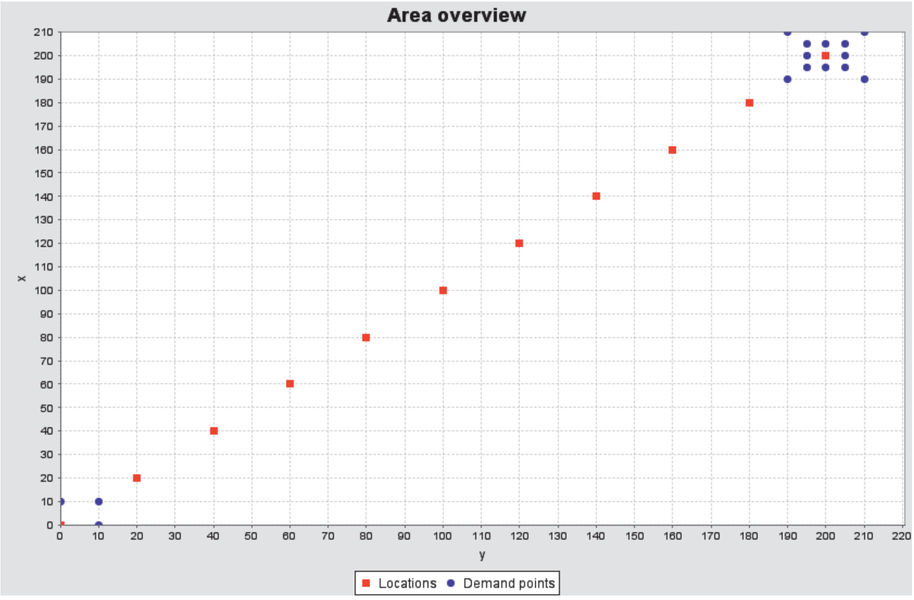

Fig. 6. Single extreme group test case.

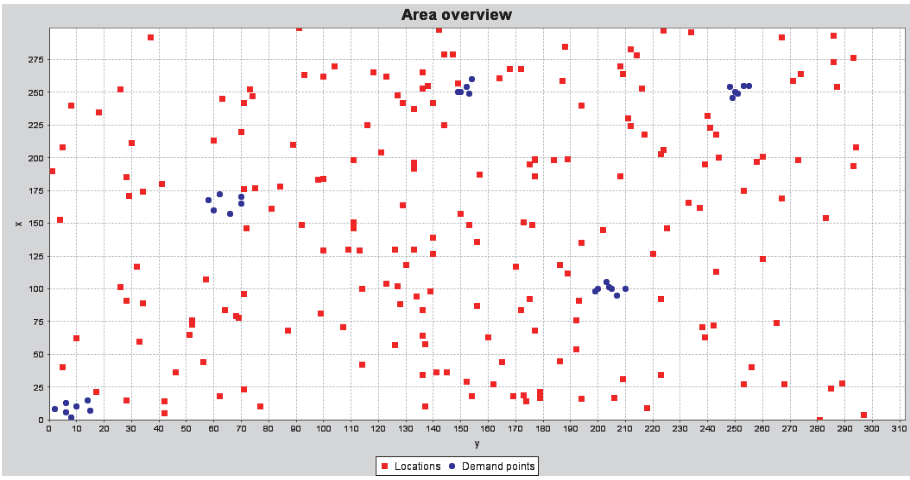

Fig. 7. Multiple groups of high demand test case.

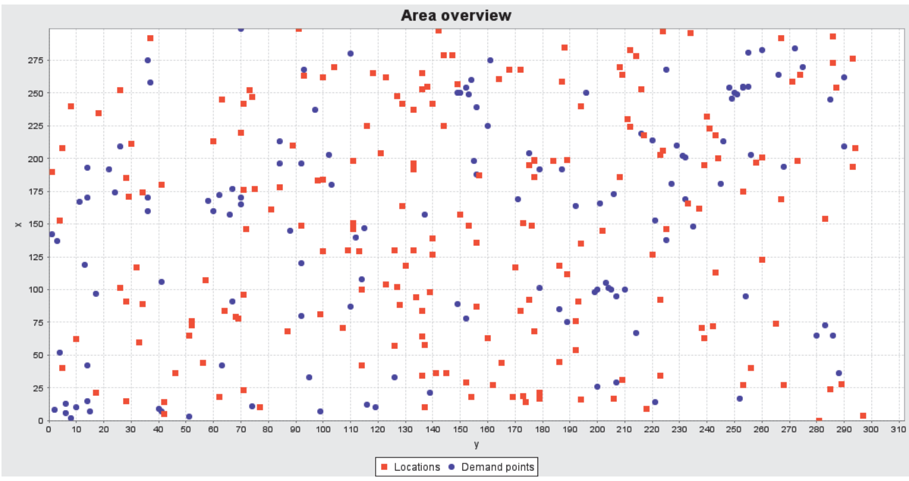

Fig. 8. Volatile demand density model case.

4.2. Model behavior

All presented datasets was successfully optimized, according to authors' assumptions. Quite spectacular was first described set. During experiments we were able to choose continuously any location between center of all possible locations (minimizing the maximal distance) and center of bigger demand group (for preferences specification that completely discards first four largest distances).
Second set was optimized correctly without any need of preferences adjustment. However, this happen because of additional knowledge about model provided in criteria. Number of possible locations was set to 5 – equal to real number of demand point subgroups in dataset. Increasing possible number of location that could be established placed additional location close to one of already selected. Surprisingly we were not able to specify preferences for predictable solution after decreasing number of possible locations. For model it was natural to place locations in one of demand point high density group, leaving one of group being served by "other town" rather than placing location in the middle between two groups. Third set was also optimized correctly, but it required quite many optimization process iterations.

4.3. Model controllability and preferences specification process

General model behavior was positive. Decision maker was able to specify criteria according to presumptions. However, presented implementation leaves big area to improvements. Performed calculations has shown, that model can be too stable and difficult for controlling. It was common to observe no response for user preferences change. This can be explained by discrete nature of presented model. One of possible solution for such behavior is extending aspiration/reservation value range. Implemented version is restricted to distance from zero to maximum possible achievement realization (maximum distance possible on given area in this case). Releasing this constraint will allow to specify preferences in wider range. Thus user should be able to set some intervals as more important by de facto increasing their weight. Models based on continuous domain should be not affected by this symptom.
Model controllability increases with number of possible locations defined. This can be well observed on first data set. For basic data set with 17 possible locations there were practically two local minimums available. This behavior seems to be quite natural – model nature is discrete. Hence each of available solution can be understood as local minimum. Increasing number of possible locations makes distance between each available solution lower and gives DM more flexibility.
Another problem, that provides to (subjective) over-controllability is lack of information about active restriction. Due to used reference point methodology problem linearization in most situations only single criterion is active and allows to slightly result adjustment. Clear information about active restrictions should be very useful for user and will be probably implemented in future version of application. In current version DM can only find this restriction by experiments.
Another area that require improveents is partial achievement specification. In current implementation it is part of project type implementation and provides not very user friendly interface. This is result of authors' underestimation of distribution intervals manipulation importance.

4.4. Preferences specification model

As described in Subsection 3.1 three ways of preferences definition was implemented. User preferences can be specified in two graphical manners (by specifying distribution or cumulated distribution) and by specification of preferences model distribution values in table organized view (Fig. 9).

Fig. 9. Three available preferences specification interfaces.

Cumulative distribution bases GUI was found difficult to use. This category of achievement definition in location variant problem was not intuitive. In most cases graphical distribution input view was sufficient to found optimal solution. For such preferences model detailed information about aspiration/reservation values importance was not big. Differently from typical reference point models most important information in presented model are relative differences between separate intervals preference values. Of course this conclusion is subjective and cannot be arbitrary while program was tested on small amount of users. Most comfortable way of criteria specifying also varies for different problem types. However implemented preferences specification interface still lacks some data important for DM during optimization process. This includes feedback data from single optimization iteration mentioned in previous paragraph.

Preferences definition interface is generally the most crucial element for proper optimization process. Implemented approach uses different data partial achievement specified by cumulative distribution. This is not intuitive data for user working with implemented location problem. Translation function defined in Subsection 2.5 allows user to specify preferences in terms of distribution but it provides to only rough preferences estimation. Relation between element single achievement and preferences and achievements defined for other intervals is not clearly presented to user. So adjusting preferences with cumulative distribution view allows for more precise result manipulation, despite meaning of the specified preference parameters may be not fully understood by the user.

4.5. Calculation performance

Number of calculations was performed for verifying model size influence on calculation speed. All calculations were performed on Intel Core 2 Duo CPU T7500 model working with 2.20 GHz clock. Java VM used to run application is Sun HotSpot Client VM build 1.6.0_02 working on Windows Vista. Solver used for optimization is pre compiled GLPK solver version 4.9. Performance results can be affected by algorithm implementation which does not take advantage from dual core processor (in both most significant steps: problem file generation and GLPK solver allocation does never use more than one core).

With assumption of about maximum single iteration optimization time two minutes largest solvable models contained about 400 possible location definitions, with 31 demand points and number of total available locations – 5. For similar problem with 100 specified possible locations total computer time exceeded 15 minutes what is too large number for single interactive process iteration.

This paragraph has shown that typical PC available on market for small office and home office (SOHO) consumer is able to perform calculations for middle sized problems without big delay for decision maker. Probably using commercial solver would noticeably improve calculation performance. According to code profiler analysis total time of problem solving can be decreased – major part of time is take by problem MPS file creation. This part of code is custom made and leaves area for code optimization.

5. Conclusions

In this paper we have described class of multicriteria problems for which preferences can be modeled in terms of distribution. Proposed preferences modeling technique is an variation of the reference point methodology and is using similar aspiration/reservation concepts. We have also shown method of construction LP problems basing exemplary location problem.

For model presentation a demonstration application was implemented. Refbeans platform allows us to test presented approach in real-life simulation problem. At current time only single type of problems was implemented. Despite this, platform architecture should allow to implement other types of problems in relatively easy way.

Performed experiments has shown, that proposed preferences modeling technique can be successfully used for optimal solution search. Proposed process is basing on adjusting user preferences in iterations. However, model response for preferences value changes is weak. This subject was discussed in Subsection 4.4. Preferences specification interface also requires improvements. Especially number of information presented to DM as optimization result should be increased.

Both model and application implementation has some flaws. Some of them was presented in previous sections. However, current status allows to improve them in future research. Decision about model usability in enterprise environment cannot be made basing only on the current status.

Appendix

Invitation to experiments and contribution

Main purpose of presented application implementation is test and demonstration of presented modeling approach. We would like to encourage users to participate in both experiments and platform development. For running it system must have Java runtime environment installed with minimum version 1.5. System is using external solver for calculations, which you have to download separately. It can be any MPS file type aware solver, but we recommend GLPK solver. All development-stage tests were performed on it. One can download GLPK solver from any GNU software mirror site. One can also download pre-compiled version for Windows.

Application source is also available under free license. Please contact the authors for full source access and support.

Acknowledgments

The research was supported by the Ministry of Science and Information Society Technologies under grant 3T11C 005 27 “Models and Algorithms for Efficient and Fair Resource Allocation in Complex Systems”.

References

[1] J. Rawls, *The Theory of Justice*. Cambridge: Harvard University Press, 1971.

[2] R. Denda, A. Banchs, and W. Effelsberg,“The fairness challenge in computer networks”, in *First COST 263 International Workshop, Quality of Future Internet Services, QofIS 2000, Berlin, Germany, 2000,* J. Crowcroft, J. Roberts, and M. I. Smirnov, Eds. Berlin: Springer, 2000, vol. 1922, pp. 208–220.

[3] M. M. Kostreva and W. Ogryczak, "Linear optimization with multiple equitable criteria", *RAIRO Oper. Res.*, vol. 33, pp. 275–297, 1999.

[4] W. Ogryczak and T. Śliwiński, "On equitable approaches to resource allocation problems: the conditional minimax solution", *J. Telecommun. Inform. Technol.*, no. 3, pp. 40–48, 2002.

[5] W. Ogryczak and M. Zawadzki, "Conditional median: a parametric solution concept for location problems", *Ann. Opns Res.*, vol. 110, pp. 167–181, 2002.

[6] A. P. Wierzbicki, M. Makowski, and J. Wessels, *Model Based Decision Support Methodology with Environmental Applications*. Dordrecht: Kluwer, 2000.

[7] A. P. Wierzbicki, "A mathematical basis for satisficing decision making", *Math. Model.*, vol. 3, pp. 391–405, 1982.

[8] A. Lewandowski and A. P. Wierzbicki, *Aspiration Based Decision Support Systems – Theory, Software and Applications*. Berlin: Springer, 1989.

[9] S. H. Owen and M. S. Daskin, "Strategic facility location: a review", *Eur. J. Opnl Res.*, vol. 111, pp. 423–447, 1998.

[10] A. Makhorin, "GNU linear programming kit", http://www.gnu.org/software/glpk/

[11] Sun Microsystems Inc., http://java.sun.com/products/javawebstart/

[12] M. M. Kostreva, W. Ogryczak, and A. Wierzbicki, "Equitable aggregations and multiple criteria analysis", *Eur. J. Opnl Res.*, vol. 158, pp. 362–367, 2004.

Krzysztof Bareja received his M.Sc. in computer science from the Warsaw University of Technology, Poland, in 2005. Currently he is a Ph.D. student in the Institute of Control and Computation Engineering at the Warsaw University of Technology. His research area focuses on decision support systems implementation, using Java in desktop systems and application design. His commercial experience includes over five years of work including participation in projects for large telecommunication vendor and Polish government clients.
e-mail: kbareja@elka.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Włodzimierz Ogryczak is a Professor and the Head of Optimization and Decision Support Division in the Institute of Control and Computation Engineering (ICCE) at the Warsaw University of Technology, Poland. He received both his M.Sc. (1973) and Ph.D. (1983) in mathematics from Warsaw University, and D.Sc. (1997) in computer science from Polish Academy of Sciences. His research interests are focused on models, computer solutions and interdisciplinary applications in the area of optimization and decision making with the main stress on: multiple criteria optimization and decision support, decision making under risk. He has published three books and numerous research articles in international journals.
e-mail: W.Ogryczak@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland