Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)

The Interactive Pareto Iterated Local Search (iPILS) Metaheuristic and its Application to the Biobjective Portfolio Optimization Problem

Martin Josef Geiger

Abstract— **The article presents an approach to interactively solve multi-objective optimization problems. While the identification of efficient solutions is supported by computational intelligence techniques on the basis of local search, the search is directed by partial preference information obtained from the decision maker.**

An application of the approach to biobjective portfolio optimization, modeled as the well-known knapsack problem, is reported, and experimental results are reported for benchmark instances taken from the literature. In brief, we obtain encouraging results that show the applicability of the approach to the described problem.

In order to stipulate a better understanding of the underlying structures of biobjective knapsack problems, we also study the characteristics of the search space of instances for which the optimal alternatives are known. As a result, optimal alternatives have been found to be relatively concentrated in alternative space, making the resolution of the studied instances possible with reasonable effort.

I. INTRODUCTION

As many problems of practical relevance are often characterized by several criteria that simultaneously have to be taken into consideration when solving the problem, multi-criteria approaches play an increasingly important role in many application areas of operations research, engineering, and computer science. Approaches from the domain of multi criteria decision making describe an alternative x, belonging to the set of feasible alternatives X, by a set of objective functions $Z(x) = (z_1(x), \ldots, z_K(x))$. From a decision making perspective, these functions describe attributes of the alternatives, which are considered to be of relevance from the point of view of a decision maker, in a quantitative way. As the aspects and therefore the functions are however often of conflicting nature, not a single alternative exists being optimal for all $z_k(x)$. Instead, a set of equally Pareto-optimal alternatives can be found as introduced in the definitions below. Without loss of generality, we assume in the following Definitions 1.1 and 1.2 the maximization of the components $z_k(x)$ of $Z(x)$.

Definition 1.1 (Dominance): A vector $Z(x)$ is said to dominate $Z(x')$ iff $z_k(x) \geqslant z_k(x') \forall k = 1, \ldots, K \wedge \exists k \mid z_k(x) > z_k(x')$. We denote the dominance of $Z(x)$ over $Z(x')$ with $Z(x) \preceq Z(x')$.

Martin Josef Geiger is with the Department of Industrial Management (510A), University of Hohenheim, 70593 Stuttgart, Germany (phone: 0049-711-45923462; fax: 0049-711-45923232; email: mjgeiger@uni-hohenheim.de).

Definition 1.2 (Efficiency, Pareto-optimality): The vector $Z(x), x \in X$ is said to be efficient iff $\nexists Z(x'), x' \in X \mid Z(x') \preceq Z(x)$. The corresponding alternative x is called *Pareto-optimal*, the set of all Pareto-optimal alternatives *Pareto-set* P.

Two aspects play a vital role for solving multi-objective problems:

1) Search for optimal alternatives (the Pareto set P), supported by an optimization approach. In comparison to single-objective optimization approaches, the notion of optimality is here generalized with respect to the set of simultaneously considered optimality criteria.
2) Choice of a most-preferred solution by the decision maker of the particular situation. While in single-objective optimization problems, the choice of the most-preferred solution naturally follows the identification of the (single) optimal solution, in multi-objective problems an individual tradeoff between conflicting criteria has to be resolved in a decision making procedure.

Principally, three classes of combining search and decision making exist [8]:

1) *A priori* approaches reduce the multi-objective problem into a single-objective problem by constructing a utility function for the decision maker. The resolution of the problem then lies in the identification of the solution which maximizes the chosen utility function.
2) *A posteriori* approaches first identify the Pareto set P (or a close and representative approximation) and then resolve the choice of a most-preferred solution within an interactive decision making procedure.
3) *Interactive* approaches combine search and decision making, presenting one or several solutions to the decision maker and collecting preference information which is then used to further guide the search for higher preferred alternatives.

The question, which particular way to follow when solving multi-objective optimization problems depends on several aspects. As mentioned above, a priori approaches require comparably rich preference information from the decision maker in order to reduce the problem to a single-objective problem. Assuming this information can be obtained, the resolution of the problem is straight-forward. Unfortunately, this is not necessarily the case in many situations as uncertainty about the precise appearance of the decision makers utility function is often present. Also, the acceptance of a single solution computed by a computational intelligence method may be a problematic issue from a psychological point of view for some decision makers as no further choice is possible.

A posteriori approaches on the other side do not require any information from a decision maker. Instead, the Pareto set P is computed off-line, allowing the decision maker to perform other tasks while waiting for the results of the optimization procedure. Under the assumption that the set of optimality functions $Z(x)$ is exhaustive, an important aspect when formulating multi-objective models [2], one element $x^* \in P$ can be identified in a later decision making procedure as the most-preferred one. A potential drawback of this approach is however, that the Pareto-set may be of large cardinality, resulting in a necessary high computational effort when identifying the Pareto-optimal alternatives. Also, many if not most $x \in P$ are discarded in the decision making procedure as they do not meet the individual, personal requirements of the decision maker.

Interactive approaches may overcome the problems of the above described extreme ways of resolving multi-objective optimization problem. On one hand, only partial preference information is required when solving the problem as the search is guided in one direction which may be changed when obtaining less-preferred solutions. On the other hand, only fewer alternatives have to be computed compared to the entire Pareto-set P. An important aspect when implementing interactive approaches is, that the decision maker needs to be present during the resolution procedure. Also, the computation of the solutions has to be completed in little time as the decision maker will have to wait for the results of the system. With the increasing computational possibilities of modern, affordable computer systems, the proposition and implementation interactive approaches however becomes more and more attractive.

Recent approaches of computational intelligence techniques implement interactive problem resolution procedures, e. g. on the basis of Evolutionary Algorithms [12], involving a decision maker during search. While in these approaches the set of criteria remains fixed during search, other concepts also include the possibility of dynamically changing the relevant criteria when searching for a most-preferred solution [7]. Research in interactive computational techniques is however a rather new field, and the precise way of how to integrate articulated preferences in the search process is still to be investigated in more detail.

In this article, we aim to contribute to the development of interactive computational intelligence techniques for the resolution of multi-objective optimization problems. While the search for Pareto-optimal alternatives is done by metaheuristics on the basis of local search, individual preferences guide the search in a particular direction with the goal of identifying a subset of P that is considered to be of interest to the decision maker. While the idea is generic, it is tested on a particular application.

The article is organized as follows. In the following Section II, the biobjective portfolio optimization problem is introduced and a quantitative optimization model is presented. We also briefly review existing approaches from the literature that have been used to solve this problem. In order to obtain a better understanding of the underlying structures of the particular problem, an analysis of the search space has been carried out which follows in Section III. An interactive procedure to solve the problem is proposed in Section IV. Experimental investigations on benchmark instances taken from literature follow in Section V, and conclusions are drawn in Section VI.

II. Problem description

The multi-objective portfolio optimization problem consists in selecting a subset of assets from a set of n possible investment possibilities such that several criteria, mainly the profit and risk of the resulting portfolio, are optimized. More formally, the following program has to be solved.

$$\max z_k(x) = \sum_{j=1}^{n} p_j^k \mathrm{x}_j \quad \forall k = 1, \ldots, K \tag{1}$$

s. t.

$$\sum_{j=1}^{n} c_j \mathrm{x}_j \leqslant C \tag{2}$$

$$\mathrm{x}_j \in \{0, 1\} \tag{3}$$

Each alternative x consists of an n-dimensional decision vector $x = (\mathrm{x}_1, \ldots, \mathrm{x}_n)$ which defines for each asset j whether it is included in the portfolio ($\mathrm{x}_j = 1$) or not ($\mathrm{x}_j = 0$), given in constraint (3). Side constraint (2) ensures that a given capacity (e. g. a budget) of C is not exceeded, and typically the nonnegative coefficients c_j relate to each other as $c_j \leqslant C \; \forall j = 1, \ldots, n$ and $\sum_{j=1}^{n} c_j > C$. The nonnegative coefficients p_j^k express for each x_j the contribution (e. g. the profit) of asset j to objective k.

As the objectives conflict with each other, a set of Pareto-optimal alternatives P exists among which the choice of a most-preferred solution $x^* \in P$ has to be made.

Unfortunately, the problem, also referred to as the so called knapsack problem, is $\mathcal{NP}$-hard [4], even for a $K = 1$. For single-objective problems, relatively good branch-and-bound algorithms have been developed [11], other approaches are based on dynamic programming [9]. The multi-objective case with $K \geqslant 2$ has been tackled again with branch-and-bound [16] and several heuristics, mainly local search as e. g. Simulated Annealing [15], or Tabu Search [3].

III. Analysis of the search space

A. Motivation and theoretical foundation

In general, metaheuristics such as Simulated Annealing, Tabu Search, and Evolutionary Algorithms aim to improve solutions through the successive modification of their decision variables. In this context, the modification of a decision vector is done with respect to the definition of a certain operator, which associates one or several other alternatives to a given solution x. It is also common to refer to these alternatives, computed for the input x, as so called neighbors of x. Given an improvement of an alternative x by applying a particular operator, the improved solution usually replaces its predecessor. For the case in which no improvement of the input x is possible using the chosen operator, different heuristic specific escape strategies have been proposed in the literature [13].

From this point of view, most metaheuristics perform a walk in a neighborhood graph $G_{\mathbf{N}}(V, A)$, induced by the chosen neighborhood operator $\mathbf{N}$. Each vertex $v_x \in V$ in $G_{\mathbf{N}}$ represents an alternative $x \in X$. An arc $a = (v_x, v_{x'})$ exists if x' is in the neighborhood of x: $x' \in \mathbf{N}(x)$.

As metaheuristics iteratively search in $G_{\mathbf{N}}$, their effectiveness depends on the underlying properties of the neighborhood graph. Two main questions are of relevance for the identification of Pareto-optimal alternatives:

1) How hard is it to identify a Pareto-optimal alternative either starting from a random initial solution or a naïve heuristically computed one? In the specific case of the multi-objective knapsack problem, a first approximation P^{approx} of the Pareto set P can be obtained by means of diversified set of weights which is used to reduce the problem to a single-objective problem, keeping the found non-dominated alternatives in P^{approx}.
2) How do the Pareto-optimal alternatives relate to each other in alternative space? Are they relatively concentrated in alternative space or are there bigger gaps in between?

In order to be able to express the distance of two alternatives x and x', we use the so called Hamming distance $d_h(x, x')$ between two decision vectors $x = (\mathrm{x}_1, \ldots, \mathrm{x}_n)$ and $x' = (\mathrm{x}'_1, \ldots, \mathrm{x}'_n)$ as given in Expression (4).

$$d_h(x, x') = \sum_{j=1}^{n} \left|\mathrm{x}_j - \mathrm{x}'_j\right| \tag{4}$$

Using the Hamming distance, we are able to compute for each $x \in P$ the distance to the closest element in the approximation P^{approx} of P using Expression (5).

$$d_h^{min}(x) = \min_{x' \in P^{approx}} d_h(x, x') \tag{5}$$

B. Results for benchmark instances

We analyzed two biobjective benchmark instances taken from the literature [3] for which the optimal alternatives are known, which is one of the reasons why they have been chosen. Both consist of $n = 50$ items. The first instance '2KP50-11' has a property of $C = 0.11 \sum_{j=1}^{n} c_j$, the second one '2KP50-50' of $C = 0.5 \sum_{j=1}^{n} c_j$. This means in other words that in the case of the first instance an expected percentage of 11% of the assets will be included in a feasible portfolio. For the second instance, a percentage of 50% can be expected respectively.

Using a set of 101 weight vectors $W = (w_1, \ldots, w_{101})$ with a $w_0 = 0$ and $w_{i+1} = w_i + 0.01 \,\forall i = 2, \ldots, 101$, a first approximation P^{approx} of P is computed. This is done by sorting the items in non-increasing order of $\frac{w_i p_j^1 + (1-w_i) p_j^2}{c_j}$ and accepting the assets in this order up to the knapsack capacity C. Dominated alternatives found by this procedure are discarded from P^{approx} such that the approximation of P only contains alternatives for which no dominating other alternative is known. We obtained approximations of $|P^{approx}| = 6$ for 2KP50-11 and $|P^{approx}| = 12$ for 2KP50-50.

Table I gives the frequency in which particular values of d_h^{min} have been obtained for the two instances.

TABLE I

NUMBER OF ELEMENTS IN P FOR A GIVEN MINIMAL DISTANCE d_h^{min}

d_h^{min}	2KP50-11	2KP50-50
0	2	5
2	15	28
3	2	10
4	21	8
5	0	1
6	4	0
$\lvert P\rvert$	44	52

It can be seen, that some of the elements $x \in P$ possess a value of $d_h^{min} = 0$, which means that they are included in P^{approx}. This is the case for two Pareto-optimal alternatives of 2KP50-11 and five of 2KP50-50. Others are very close to their nearest element in P^{approx} with a significant concentration around values of $d_h^{min} = 2$ and $d_h^{min} = 4$. The most distant elements differ in six decision variables in instance 2KP50-11.

As a corollary, the alternatives provided by the weighted sum aggregation procedure appear to be useful starting points for metaheuristics as only small modifications on the decision variables produce Pareto-optimal solutions with a high likeliness.

In order to obtain an understanding of the relation of the elements in P, the fitness-distance-correlation has been analyzed using the Hamming distances of the alternatives $x, x' \in P$ and the Euclidean distances of their objective vectors $Z(x), Z(x'), x, x' \in P$. Figures 1 and 2 show the plots of the values for the two investigated instances. Clearly, the Pareto-optimal alternatives show high fitness-distance correlations.

In the light of the search space analysis, it appears to be a fruitful strategy to solve biobjective knapsack problems by local search, focusing on a particular area of the search space. On one hand, heuristically generated starting solutions appear close to the true Pareto-optimal alternatives. On the other hand, Pareto-optimal alternatives show a clear fitness-distance-correlation, which means that when starting from one Pareto-optimal alternative, solutions with a similar evaluation are likely to be found close by.

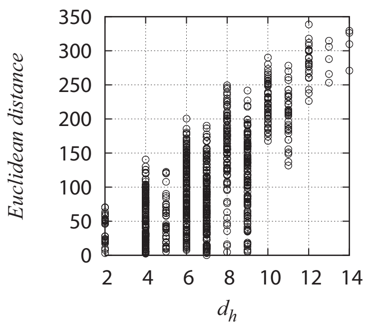

Fig. 1. Fitness-distance plot for 2KP50-11

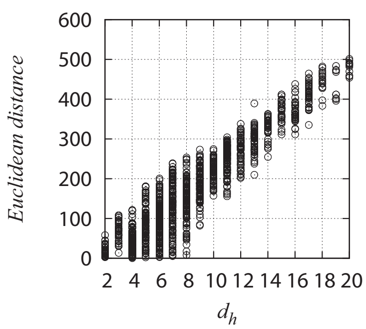

Fig. 2. Fitness-distance plot for 2KP50-50

IV. Solution approach

Based on the discussion of different problem resolution strategies in Section I and the insight in the structure of biobjective knapsack problems gained in Section III, we propose an interactive metaheuristic that brings search and decision making together in a combined approach.

In the first step of the problem resolution approach, the decision maker is provided with lower and upper bounds of the particular problem. For the knapsack problem, this can be done by a set of weights W and relaxing the binary constraint 3 for the computation of upper bounds. An example of a visual output is given in Figure 3, plotting the lower bound sets in blue, the upper bound sets in orange.

By means of the visualization of the lower and upper bounds, the decision maker may easily see the area in which Pareto-optimal solutions can be found. It becomes equally clear what values of $Z(x)$ are not possible as they would dominate an upper bound, enabling the decision maker to develop realistic expectations of a most-preferred solution x^*. In other words, objective values which are not achievable, as they lie outside the boundaries of the upper bounds, can be excluded from the decision making procedure.

The search for Pareto-optimal solutions is then guided by the articulation of a reference point $R = (r_1, \ldots, r_K)$, given in red color in Figure 3. This reference point defines a cone in outcome space which is used to specify an area in which the most-preferred solution x^* is expected. After having successfully implemented an a posteriori strategy based on the same principle in which the identification of a most-preferred solution is supported by the progressive articulation of aspiration levels [6], we use this idea here to interactively guide the search.

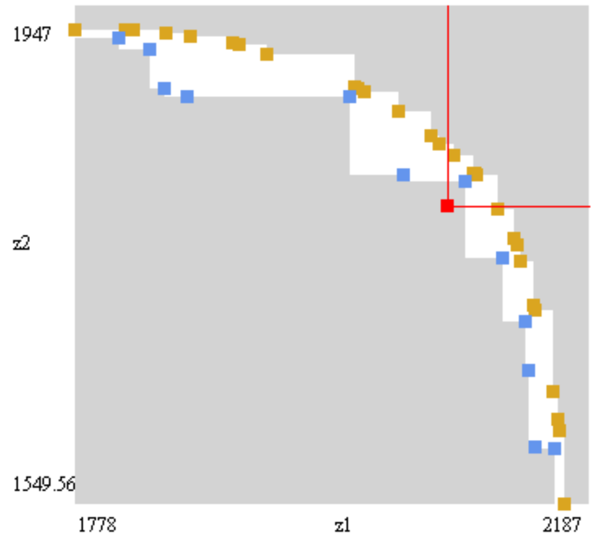

Fig. 3. Lower/ upper approximation and determination of the goal vector

Algorithm 1 Optimization framework

Compute a first approximation P^{approx} of P using a set of weights W.
Present P^{approx} to the decision maker
Obtain a reference point R from the decision maker
repeat
  Compute P_R^{approx}
  while R has not been changed **or** termination criterion has not been met **do**
    Search for Pareto-optimal alternatives in the cone defined by R by means of a pre-defined local search metaheuristic
    Constantly update P_R^{approx} while searching for Pareto-optimal solutions
    Constantly update the visualization of P_R^{approx}, showing the results to the decision maker
  end while
until termination criterion is met

The cone defined by R is used to compute the set of alternatives P_R^{approx} that dominate the reference point and therefore lie in the interior of the cone. In brief, P_R^{approx} contains all elements $x \in P^{approx} \mid z_k(x) \geqslant r_k \forall k = 1, \ldots, K$, therefore $P_R^{approx} \subseteq P^{approx}$. These alternatives are used by the Pareto Iterated Local Search metaheuristic, whose principle is sketched in Figure 4 and discussed in the following. Search continues until the decision maker terminates the process. This is going to be the case when a solution $x^* \in P_R^{approx}$ is found which meets the individual requirements and expectations of the decision maker as close as possible.

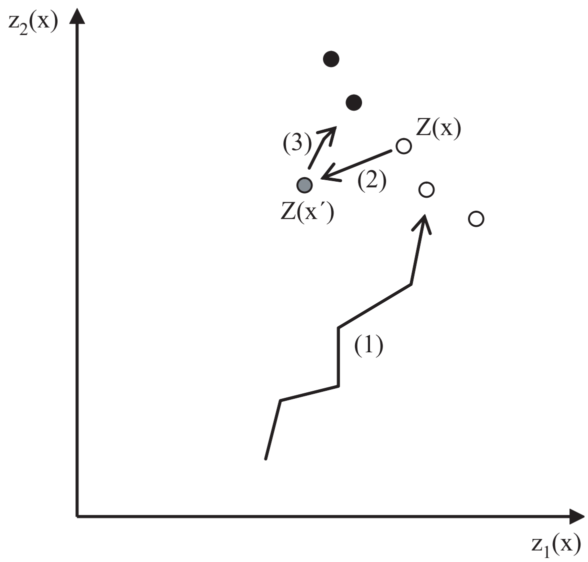

Fig. 4. Pareto Iterated Local Search

Starting from an initial solution, a local search run is performed using a neighborhood operator until no further improvement is possible with simple local modifications. During this step, an archive of alternatives is maintained which contains all non-dominated alternatives found during search. While in the general procedure of Pareto Iterated Local Search [5] all non-dominated alternatives are kept, we here restrict the procedure to keep only alternatives in the cone defined by R for further modifications/ improvements. In Figure 4, this stage of the procedure is visualized as step (1), with the results shown as white points in outcome space.

After having obtained a set of locally optimal alternatives, one of them is picked at random, see $Z(x)$ in Figure 4, perturbed into another alternative x' using some other neighborhood (2), and search is continued from here (3). As it can be seen in Figure 4, the perturbed solution may be dominated by one or several elements of P_R^{approx}, which has to be accepted when overcoming local optimality. In result, the metaheuristic iterates in interesting areas of the search space as opposed to restarting search from some other solution. This principle, known from Iterated Local Search [10], has been already successfully applied to other problems in which considerable fitness-distance-correlations have been found [1].

The computations of the metaheuristic are continued, constantly updating the plot of the Pareto-optimal alternatives in outcome space. During the problem resolution procedure, the decision maker is allowed modify the reference point, shifting the focus of the computations towards other regions. The problem resolution procedure terminates with the identification of a most-preferred solution x^*.

V. Experimental investigations

A. Experimental setup

The interactive Pareto Iterated Local Search has been tested on the two benchmark instances taken from [3]. Apart from the fact that the optimal alternatives of these instances are known as mentioned above, the data sets are widely used for experimental investigations and comparison. In choosing them, we hope to provide a basis for fair and representative comparison. The data of the instances can be obtained from the internet homepage of the International Society on Multiple Criteria Decision Making under `http://www.terry.uga.edu/mcdm/`.

Local modifications of alternatives x are done by randomly picking a single decision variable $\mathrm{x}_j \mid \mathrm{x}_j = 1$, changing its value to $\mathrm{x}_j = 0$, and randomly changing the value of other randomly chosen decision variables to 1 until no additional asset may be added to the solution.

We applied this local search neighborhood to each element in P_R^{approx} until a dominating alternative has been found, replacing the alternative, or a subsequent number of 100 unsuccessful iterations has been tested on each element in P_R^{approx}. Then, the perturbation is applied to a randomly picked element in P_R^{approx}. The alternative is perturbed by changing two randomly chosen variables $\mathrm{x}_j = \mathrm{x}_l = 1$ to 0 and refilling up the knapsack to the capacity C by randomly selected other assets. In this sense, the perturbation is similar to the regular neighborhood, only that more decision variables are involved, leading to a bigger jump in the search space while keeping most of the characteristics of the perturbed alternative at the same time. The search then continues from the alternative which has been obtained through perturbation as described in Section IV.

In order to simulate the individual preference articulation of the decision maker, three reference points have been defined for each of the model instance as given in table II, one in the 'knee-region' [14] and two in the extreme areas of either one of the objective functions.

TABLE II
Reference points

Model	Reference point	Vector
2KP50-11	ref. #1	$r_1 = 395, r_2 = 550$
2KP50-11	ref. #2	$r_1 = 531, r_2 = 428$
2KP50-11	ref. #3	$r_1 = 616, r_2 = 354$
2KP50-50	ref. #1	$r_1 = 1807, r_2 = 1924$
2KP50-50	ref. #2	$r_1 = 2094, r_2 = 1800$
2KP50-50	ref. #3	$r_1 = 2166, r_2 = 1574$

100 test runs have been carried out with each instance and reference point, keeping the approximations P_R^{approx} for further analysis. In each test run 100,000 iterations have been allowed before terminating the search.

The quality of the computed approximations has been analyzed using the M metric, given in expression 6. M measures the percentage of identified Pareto-optimal alternatives in the cone defined by R.

$$M = \frac{|P_R^{approx} \cap P_R|}{|P_R|} \quad (6)$$

B. Results

Based on the data gathered in the experiments, the arithmetic mean values of M have been computed, depending number of evaluations of the metaheuristic. These average values, given in Figures 5 and 6, clearly show that the iPILS metaheuristic successfully identified the Pareto-optimal alternatives in the particular areas of the reference points. It can be seen that for model 2KP50-50 the approximation appears to converge faster than for 2KP50-11. However, there does not turn out to be a consistent difference for the three chosen reference points within the same instance.

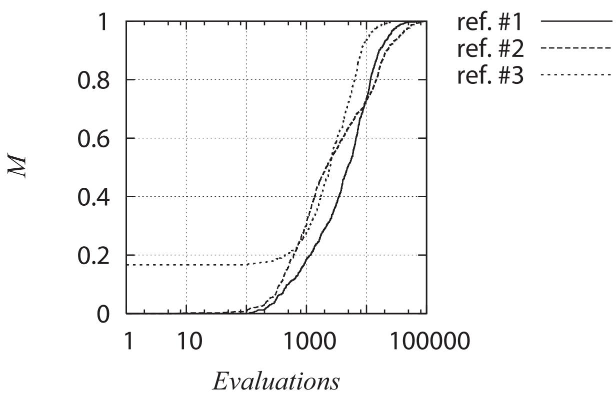

Fig. 5. Results for 2KP50-11

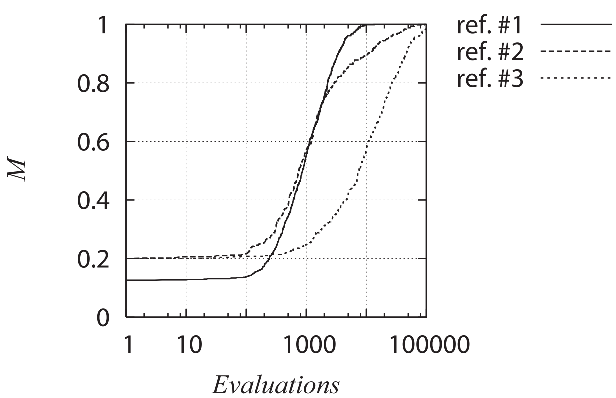

Fig. 6. Results for 2KP50-50

While some approximations P_R^{approx} already contain a Pareto-optimal alternative right from the start, no overall advantage for the final approximation quality results from this circumstance. Some areas are faster approximated, e. g. the one of reference point #1 in 2KP50-50, others take considerable more time, see reference point #3 in 2KP50-50.

VI. Conclusions

The article presented an interactive method for the resolution of multi-objective optimization problems. The concept is based on the articulation of a reference point which expresses, from the point of view of the decision maker, an interesting area in outcome space. The computation of Pareto-optimal solutions is consequently focused on that region, identifying optimal solutions. In order to overcome locally optimal alternatives and converge to the front of the efficient solutions, a metaheuristic based on Iterated Local Search has been implemented.

Tests on biobjective portfolio optimization problems have been carried out. Initial investigation of problem structures revealed that heuristically generated alternatives are rather close to the Pareto-optimal alternatives and therefore present good starting points for heuristics based on local search. Also, the Pareto-optimal solutions show a significant fitness-distant-correlation, indicating that similar efficient solutions are close to each other in alternative space.

In order to simulate a human decision maker, we assumed three different reference points for each instance. Each one was chosen with respect to lower and upper bound sets that serve as an orientation for the decision maker. The problem resolution technique successfully solved the investigated benchmark instances, independent from the particular reference point.

Based on the investigations and experiments carried out, we conclude that the proposed concept may be a useful tool for solving multi-objective optimization problems, given the possibility to appropriately compute lower and upper bound sets of the particular problem. The results are encouraging, and deeper investigations on more instances will follow to support the results of the study.

References

[1] Kenneth Dean Boese. *Models for Iterative Global Optimization*. PhD thesis, University of California at Los Angeles, Los Angeles, California, 1996.

[2] Denis Bouyssou. Building criteria: A prerequisite for MCDA. In Carlos Bana E Costa, editor, *Readings in Multiple Criteria Decision Aid*, pages 58–80. Springer Verlag, Heidelberg, 1990.

[3] X. Gandibleux and A. Freville. Tabu search based procedure for solving the 0-1 multiobjective knapsack problem: the two objectives case. *Journal of Heuristics*, 6(3):361–383, 2000.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco, CA, 1979.

[5] Martin Josef Geiger. The PILS metaheuristic and its application to multi-objective machine scheduling. In Karl-Heinz Küfer, Heinrich Rommelfanger, Christiane Tammer, and Kristin Winkler, editors, *Multicriteria Decision Making and Fuzzy Systems – Theory, Methods and Applications*, Industrial and applied mathematics, pages 43–58. Shaker Verlag, Aachen, 2006. ISBN 3-8322-5540-0.

[6] Martin Josef Geiger. Solving multi-objective scheduling problems—an integrated systems approach. In Max Bramer, editor, *Artificial Intelligence in Theory and Practice*, volume 217 of *IFIP International Federation for Information Processing*, pages 493–502. Springer Verlag, New York, 2006. ISBN 0-387-34654-6.

[7] Martin Josef Geiger and Sanja Petrovic. An interactive multicriteria optimisation approach for scheduling. In M. Bramer and V. Devedzic, editors, *Artificial Intelligence Applications and Innovations*, pages 475–484. Kluwer Academic Publishers, Boston, Dordrecht, London, 2004.

[8] Jeffrey Horn. Multicriterion decision making. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter F1.9, pages F1.9:1–F1.9:15. Institute of Physics Publishing, Bristol, 1997.

[9] Kathrin Klamroth and Margaret M. Wiecek. Dynamic programming approaches to the multiple criteria knapsack problem. *Naval Research Logistics*, 47:57–76, 2000.

[10] Helena R. Lourenço, Olivier Martin, and Thomas Stützle. Iterated local search. In Fred Glover and Gary A. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, chapter 11, pages 321–353. Kluwer Academic Publishers, Boston, Dordrecht, London, 2003.

[11] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore, 1990.

[12] S. P. Phelps and M. Köksalan. An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49:1726–1738, 2003.

[13] Marc Pirlot. General local search methods. *European Journal of Operational Research*, 92:493–511, 1996.

[14] L. Rachmawati and D. Srinivas. A multi-objective genetic algorithm with controllable convergence on knee regions. In *2006 IEEE Congress on Evolutionary Computation*, pages 6807–6814, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Kanada, Juli 2006.

[15] E. L. Ulungu, J. Teghem, P. H. Fortemps, and D. Tuyttens. MOSA method: A tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Making*, 8:221–236, 1999.

[16] Ekunda Lukata Ulungu and Jacques Teghem. Solving multi-objective knapsack problem by a branch-and-bound procedure. In J. N. Climaco, editor, *Multicriteria Analysis*, pages 269–278. Springer Verlag, Berlin, Heidelberg, New York, 1997.