Presorting as a method of acceleration of algorithms in multi-objective optimization problems

A. AMELJAŃCZYK
andrzej.ameljanczyk@wat.edu.pl

Military University of Technology, Faculty of Cybernetics
Institute of Computer and Information Systems
Kaliskiego Str. 2, 00-908 Warsaw, Poland

The paper presents a method of algorithms acceleration for determining Pareto-optimal solutions (Pareto Front) multi-criteria optimization tasks, consisting of pre-ordering (*presorting)* set of feasible solutions. It is proposed to use the generalized Minkowski distance function as a *presorting* tool that allows build a very simple and fast algorithm Pareto Front for the task with a finite set of feasible solutions.

Keywords: Pareto Front, *presorting*, Minkowski metric.

1. Introduction

Calculating algorithms of non-dominated solutions (Pareto Front) in the multi-criteria optimization tasks for large sets are generally very time consuming [3, 6, 7, 8, 12]. Among these algorithms, heuristic algorithms (including biologically inspired [10, 11, 12]) play a special role. The specificity of multi-task optimization lies in the fact that their solutions required to determine not one but many optimal solutions (so-called Pareto Front) in a certain sense [20]. These solutions may vary considerably in terms of "degrees meet the individual criterions". In many cases, however, decision-making need to draw all the solutions or effective representations [4, 5]. Given that there is no general, the analytical method for the determination of Pareto Front for any task, developed heuristic methods to solve such tasks (including the bio-inspired methods [9, 10, 11, 12]). In this type of algorithms used different methods allowing to reduce the number of comparisons between the solutions themselves to a minimum [8, 9, 10, 11, 12, 13]. Essence of these methods most often is selection (estimation) of this subset $\overline{Y}$ of the Y set of feasible elements, which would contain all the Pareto optimal elements and at the same time it was the least numerous. The $Y-\overline{Y}$ set would then be the source of so-called *empty iterations* (comparisons) and could be skipped. "Good estimate" of the $\overline{Y}$ set is very difficult. In the work proposed approach to estimating the $\overline{Y}$ as a result of which the primary order of the Y set and the appointment of an "optimal" threshold that divides Y to subset $\overline{Y}$ and subset $Y-\overline{Y}$ .Code to find the best possible estimate of the $\overline{Y}$ set is finding the right pre-order elements of the set. In the solution of tasks multi-criteria optimization a very important role is played by the lower bound of Y set with the R relation [4, 5, 14]:

$$\inf_R Y = \left\{ y \in \mathcal{R}^N \,\middle|\, (y,z) \in R \text{ for all } z \in Y - \{y\} \right\}.$$

In the case of tasks with the Pareto relation is a singleton set containing the element called an ideal point of Y set [4, 19, 20]. Ideal point as the lower bound of the Y set plays an important role in many multi-criteria optimization algorithms. It became among other things, the basis for defining the so-called compromise solutions [19, 20]. Classical formula for determining the Y_p^{*y} set of compromise elements is as follows:

$$Y_p^{*y} = \left\{ y^p \in Y \,\middle|\, \left\| \overset{*}{y} - y^p \right\|_p = \min_{y \in Y} \left\| \overset{*}{y} - y \right\|_p \right\} \quad (1)$$

where $\left\| \overset{*}{y} - y \right\|_p = \left(\sum_{n \in \mathcal{N}} \left(\left| \overset{*}{y}_n - y_n \right|^p \right) \right)^{1/p}$, $p \geq 1$

Minkowski norm. General properties of compromise solution [19, 20] are:

$$Y_p^{*y} \subset Y_N^R \text{ for all } 1 \le p < \infty \qquad (2)$$

for $p = \infty$ is true $Y_\infty^{*y} \cap Y_N^R \neq \varnothing$

Lema 1 [2]
Let $y, z \in Y$ and $y \neq z$.

If $(y,z) \in R$ that $\left\| {}^{*}y - y \right\|_p < \left\| {}^{*}y - z \right\|_p$

By using the distance function

$$f_p^{*y}(y) = \left\| {}^{*}y - y \right\|_p , y \in Y \qquad (3)$$

It is possible to make the ranking elements of the Y set. Element y lying closest to the ${}^{*}y$ element is considered the best and the farthest element – the worst. This ranking will be denoted by symbol $r_p\left({}^{*}y\right)$.The ranking elements with the same value of ranking function (if any there are) will be written in any order.

The ranking function $f_p^{*y}(y)$ for $p = 1$ can be written as follows:

$$f_1^{*y}(y) = \left\| {}^{*}y - y \right\|_1 = \sum_{n \in \mathcal{N}} \left({}^{*}y_n - y_n \right) =$$

$$= \sum_{n \in \mathcal{N}} {}^{*}y_n - \sum_{n \in \mathcal{N}} y_n = \mathrm{c} - \sum_{n \in \mathcal{N}} y_n .$$

Minimizing of function $f_1^{*y}(y)$ you can therefore be replaced by maximizing the function

$$f_{1a}^{*y}(y) = \sum_{n \in \mathcal{N}} y_n \rightarrow \max \qquad (4)$$

This is the simplest ranking function with a parameter $p \ge 1$ does not require even the knowledge of the ideal point. Another frequently used function of the ranking is a function of the parameter $p = 2$ that is Euclidean distance:

$$f_2^{*y}(y) = \sqrt{\sum_{n \in \mathcal{N}} \left({}^{*}y_n - y_n \right)^2}$$

For $p = \infty$, we get the Chebyshev distance:

$$f_p^{*y}(y) = \max \left\{ \left({}^{*}y_n - y_n \right), n \in \mathcal{N} \right\} \qquad (5)$$

The next step will be presented preliminary sorting (*presorting*) Y set on the basis of sorting function for $p = 1, 2, \infty$ and based on two different sorting rules. Sorting by value function $f_1^{*y}(y)$ has the advantage over the other, because it does not require knowledge of the ideal point.

2. Sorting preliminary (*presorting*)

Let the next , Y is a finite set of M elements (objcts), numbered by index $m \in \mathcal{M} = \{1, ..., m, ..., M\}$. The Y set will save as follows: $Y = \left\{ y^1, y^2, ..., y^m, ..., y^M \right\}$.

Example 1
Table 1 presents a set of 20 elements evaluated by two criteria and the results of *presorting* obtained for different sorting rules. In first column have been saved "original" numbers of elements of the set, in the second column and the third of the characteristics of these elements. Columns from the fourth to the eighth present information about the changes organize a set (about *presorting*) for the distance with parameter $p = 1, 2, \infty$ and for the lexicographical order [3] as well for *scoring presorting* [9]. Figure 1 shows a Y set, with a ideal point (data from Table 1). The Figure 2 shows the same set of elements numbered by the *presorting* for $p = 2$. Numeration of elements in Figure 2 shows the sequence of their consideration in the proposed algorithm. Ideal point ${}^{*}y$ for the Y set has the following coordinates:

$${}^{*}y_1 = \max_{m \in \mathcal{M}} y_1^m = 6 , \; {}^{*}y_2 = \max_{m \in \mathcal{M}} y_2^m = 7$$

Tab. 1

1	2	3	4	5	6	7	8
m	y_1^m	y_2^m	r_1	r_2	r_∞	$r_{\mathcal{L}}$	r_S
1	**2**	**7**	2	3	3	4	3
2	**4**	**7**	3	2	2	5	4
3	**5**	**6**	4	4	4	6	2
4	**6**	**5**	1	14	14	3	1
5	**6**	**3**	5	15	15	15	5
6	**6**	**2**	14	19	19	7	14
7	**5**	**1**	15	1	20	2	15
8	**3**	**1**	6	5	1	14	6
9	**2**	**1**	13	13	5	16	19
10	**1**	**2**	19	20	13	19	13
11	**0**	**4**	12	16	16	20	7
12	**1**	**6**	16	6	18	17	12
13	**2**	**6**	20	18	6	8	16
14	**4**	**5**	7	12	10	1	20
15	**5**	**4**	18	17	12	13	17
16	**4**	**3**	17	7	17	18	18
17	**3**	**2**	8	8	7	9	8
18	**2**	**4**	11	11	8	12	11
19	**3**	**5**	9	10	9	10	10
20	**3**	**4**	10	9	11	11	9

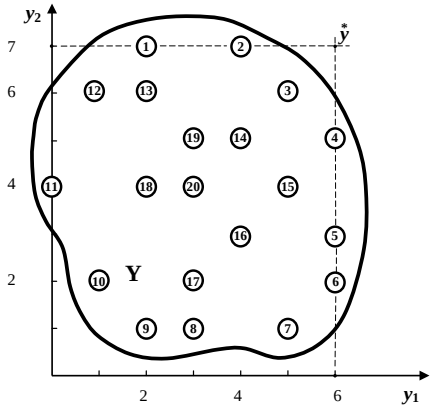

Fig.1. Set of elements in ranking procedure and ideal point y^*

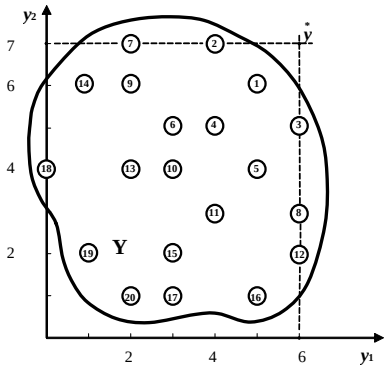

Fig. 2. Numeration of elements after *presorting* (p = 2)

Presorting Y set can significantly speed up the search of Pareto Front, because elements of the set are "puted" in a "queue" for viewing under the best chance of finding in Y_N^R set. The first element in the queue is certainly an element of Pareto Front (see compromise solutions properties [4, 19, 20]. Below, the algorithm (now be called *AAPF*), based on *presorting*, using a distance from the ideal point will be presented.

Presorting, because it comes down to sort a set of numbers is much easier and faster than vector sorting of Y set. It is the reason of time benefits. Thanks to it, the vector sorting can be significantly reduced.

3. *AAPF* algorithm for Pareto Front

The idea of the *AAPF* algorithm is the browsing Y set, initially sorted by ranking $r_p\left(y^*\right)$.

Let $Y(k)$ means the set of elements selected from Y set, including the steps from 1 to k.

$Y(k) \succ y^j$ will mean that there is in a $Y(k)$ such element y^w, that $\left(y^w, y^j\right) \in R$ (it means that y^w is better then y^j). We will say that the set $Y(k)$ dominates of element y^j. The running of *AAPF* algorithm is as follows:

Step 1 (Initiation of starting solution)
$k = 1$, $j = 1$

$$Y(k) = \left\{y^j\right\}$$

Step 2 (Construction of Pareto Front)
Check if next element in the ranking is dominated by $Y(k)$ ($Y(k) \succ y^j$?)

a) if YES, then y^j rejected
b) if NO, then

$$Y(k+1) = Y(k) \cup \left\{y^j\right\}$$

The algorithm ends when $j = M$ or work up the additional criterion STOP (defining the threshold in *presorting*). The sets $Y(k)$ (so-called "approximation" of Pareto Front), obtained in the course of the algorithm satisfy the condition:

$$Y(1) \subset Y(2) \subset ... \subset Y(K) = Y_N^R \qquad (6)$$

On Figure 3 was presented a detailed diagram of the *AAPF* algorithm.

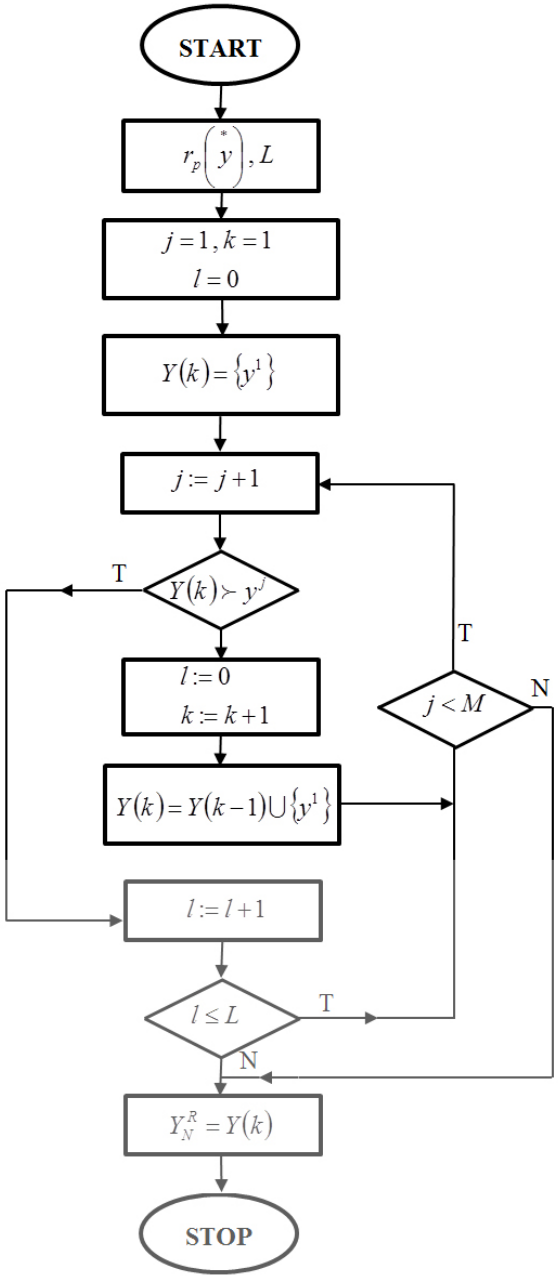

Fig. 3. Diagram of *AAPF* algorithm with number L of empty iterations

Application of $r_p\left(\overset{*}{y}\right)$ *presorting* Y set caused the "most likely" candidates for Pareto Front are at "the head of the ranking list". You can use it to dispense with the subsequent comparisons (usually very plenty) elements of the Y set in the context of their belonging to the Pareto Front. The number k in the running of the algorithm indicates cardinality "partial Pareto Front". Knowledge of number $k=K$ (K is equal to the cardinality of Pareto Front) would stop the running of the algorithm due to the fact that all next iterations would be required (so-called "empty iterations"). Problem STOP criterion in this type of algorithms turns out to be very important due to the elimination of generally very numerous empty iterations. In this example empty iterations account for 85% of all iterations. Figure 3 shows the use of a simple, additional criterion STOP in the form of arbitrary number of L empty iterations. The conducted tests have shown that already $L=1/10\,M$, obtained the very good "approximation" of Pareto Front (usually it was the full Pareto Front Y_N^R). Running of *AAPF* algorithm for *presorting* $p=2$ is as follows:

Step 1

$Y(1)=\{y^3\}$

Step 2

The next (second) element in the *presorting* is y^2. We check if $Y(1)\succ y^2$? This is false, because $(y^3, y^4)\notin R$ and $(y^2, y^4)\notin R$, it means $Y(1)$ does not dominate y^2, so $Y(2)=Y(1)\cup\{y^2\}=\{y^3, y^2\}$.

The next element is the y^4. We check if $Y(2)\succ y^4$? Because: $(y^3, y^4)\notin R$ and $(y^2, y^4)\notin R$ it means: $Y(2)$ does not dominate y^4.

So $Y(3)=Y(2)\cup\{y^4\}=\{y^3, y^2, y^4\}$.

Another element in *presorting* is y^{14}. We verify that $Y(3)\succ y^{14}$? Because: $(y^3, y^{14})\in R$ it means $Y(3)$ dominates y^{14}. So y^{14} is rejected (we have the first empty iteration). The next iterations also turn out to be empty. The final result will be $Y(3)=\{y^3, y^2, y^4\}$. The STOP criterion, based on the fixed number $L=(1/10\,M)=2$ hold up the algorithm after two empty iterations. Other iterations (fifteen) are already useless. In the case of running of the algorithm with *lexicographical presorting* [3] for the $L=(1/10\,M)$ algorithm stops after ten iterations, giving the final result $Y(3)=\{y^4, y^3, y^2\}$. Other (ten) iterations are no longer needed. If the number L was less, for example $L=(1/20\,M)=1$, the algorithm would

stop after three iterations, giving only approximation of Pareto Front as a $Y(1)=\{y^4\}$.

To small number L it is the threat appointment only approximation of Pareto Front – too big – causes the increase in computation time. The earlier work criterion STOP causes, the number of redundant operation of comparison is less. Ideally situation we have, the algorithm finished work when determining the last element of Pareto Front. Provides too few empty iterations may result in too early the end of the calculations which may mean that other Pareto elements in the set $Y-Y(k)$ are still "not detected".

4. Analysis of the numerical examples

The first computational experiments show example of the set Y described in Table 1. In Table 1, shown is a sample set of *presorting.* Easy to note that in the case of *distance presortings* in each case "head" of list, contains elements which, in the first iteration will be considered are the elements of Pareto Front. This gives the opportunity for significant acceleration of the algorithm. The functioning of the basic version of the algorithm has already been discussed in section 3 work. There are shown also disadvantages and advantages STOP additional criterion in the form of L-series of empty iterations. Below is shown another example in which a number of criteria will be increased to 5 and the set will have a cardinality 20.

Example 2

Table 2 contains the data collection and the four selected *presortings*. The calculations are carried out for different variants of additional criterion STOP using fixed length *L* series of empty iterations.

Table 2

1	2	3	4	5	6	7	8	9	10
m	y_1^m	y_2^m	y_3^m	y_4^m	y_5^m	r_1	r_2	r_∞	$r_{\mathcal{L}}$
1	**2**	**3**	**5**	**8**	**10**	15	15	18	18
2	**1**	**8**	**5**	**4**	**5**	18	18	20	20
3	**1**	**3**	**6**	**2**	**5**	9	20	15	15
4	**3**	**7**	**3**	**1**	**4**	17	9	9	9
5	**0**	**2**	**0**	**1**	**1**	20	17	17	14
6	**4**	**0**	**2**	**3**	**4**	14	14	14	17
7	**2**	**3**	**8**	**5**	**6**	1	1	16	19
8	**5**	**1**	**2**	**1**	**4**	10	12	6	8
9	**8**	**9**	**12**	**10**	**10**	2	10	12	16
10	**0**	**8**	**7**	**7**	**5**	7	7	1	6
11	**1**	**3**	**4**	**5**	**5**	12	13	7	4
12	**3**	**4**	**5**	**6**	**5**	13	16	13	12
13	**2**	**3**	**5**	**4**	**8**	16	2	19	7
14	**8**	**7**	**10**	**5**	**10**	19	19	2	1
15	**9**	**9**	**10**	**15**	**12**	4	3	3	13
16	**5**	**0**	**2**	**5**	**10**	11	11	4	2
17	**7**	**10**	**11**	**10**	**10**	3	4	8	3
18	**15**	**8**	**7**	**10**	**12**	6	6	11	11
19	**7**	**4**	**5**	**2**	**1**	8	8	5	10
20	**10**	**8**	**8**	**10**	**10**	5	5	10	5
y^*	**15**	**10**	**12**	**15**	**12**				

Implementation of the algorithm AAPF without additional criterion STOP (only $j < M$ condition) after 19 iterations gave the result $Y(5)=Y_N^R=\{y^{15},y^{18},y^9,y^{17},y^{20}\}$.

Exactly the same result was obtained for the next three presortings. Global number of elementary comparisons (elementary comparison is to compare of 2 numbers (2 coordinates)) for $p = 1$ amounted to 82, for $p = 2$ as well 82, for $p = \infty$ 84 and for lexicographical presorting 98 comparisons. Implementation of the calculation brute force mode would require a pessimistic scenario $(M-1)MN=$ $=19\times 20\times 5=1900$ comparison operations. The use of an additional criterion STOP to reduce the number of empty iterations bring additional benefits. If was adopted in the algorithm number $L=1$ than for the implementation of the algorithm AAPF, Pareto Front would be appointed after 6 iterations (including the 12 comparison operations), while for $p = \infty$ after 14 comparisons. But for lexicographical presorting $\mathcal{L}$ it has been designated only an approximation of Pareto Front as a set $Y(4)=\overline{Y}_N^R=\{y^{15},y^{18},y^9,y^{20}\}\subset Y_N^R$ by using 85 operations of comparison. Only the number $L\geq 2$ would designate the full Pareto Front.

5. Conclusions

The presented *AAPF* algorithm with an additional stop mechanism in the form of L empty iterations, using the *presorting* based on the distance from the ideal point, gives a very broad and flexible possibility in terms of determining the Pareto Front for even very large collections. Particularly interesting results were obtained in the course of research for test cases $p=1,2$. Application of *presorting* gives considerable scope to reduce computation time

due to the fact that in the first place are considered elements that are "closest" lower bound of Y set (ideal point). The algorithm can easily be adapted to the possibilities of determining only certain approximations of Pareto Front – if it were necessary. Due to the extra computation time associated with making *presorting*, it is convenient to select a distance function with $p=1$. *Presorting* then takes place in a very simple manner on the basis of the sum of the coordinates of individual elements. The computational complexity of the proposed algorithm mainly depends on the Pareto structure of Y set (the ratio of frequencies Pareto set of frequencies to the whole set). Greatest benefits can be obtained in those cases where the number of elements of Pareto set in relation to the whole number of elements of set Y is very small, or when we want only a small representation of Pareto set.

6. Bibliography

[1] Ameljańczyk A., "Mathematical aspects of ranking theory", *Computer Science and Mathematical Modelling*, No. 2, 5–10 (2015).

[2] Ameljańczyk A., "Pareto filter in the process of multi-label classifier synthesis in medical diagnostics support algorithms", *Computer Science and Mathematical Modelling* , No. 1, 5–10 (2015).

[3] Ameljańczyk A.,Tran Quang Ch., "Lexicographical binary implementation of the Recurrent Pareto Filter in categorization procedures", *Computer Science and Mathematical Modelling*, No. 2, 11–15 (2015).

[4] Ameljańczyk A., *Optymalizacja wielokryterialna w problemach sterowania i zarządzania*, Ossolineum, Wrocław, 1984.

[5] Ameljańczyk A., *Multiple optimization*, WAT, Warszawa, 1986.

[6] Ameljańczyk A., "Metoda podziału zbioru obiektów na wielokryterialne klastry jakościowe", *Biuletyn Instytutu Systemów Informatycznych*, Nr 12, 1–7 (2013).

[7] Bouyssou D., Marchant T., "An axiomatic approach to noncompensatory sorting methods in MCDM, I: The case of two categories", *EJOR*, 178(1), 217–245 (2007).

[8] Brans J.P., Vincke Ph., "A preference ranking organization method: The PROMETHEE method for Multiple Criteria Decision-Making", *Management Science*, Vol. 31, No. 6 ,647–656 (1985).

[9] Du J., Cai Z., Chen Y., "A sorting Based Algorithm for Finding Non-Dominated Set in Multi-Objective Optimization", ICNC Conference 2007.

[10] Fang H., Wang Q., Tu Y-C., Horstemeyer M.F., "An efficient non-dominated sorting method for evolutionary algorithms", *Evolutionary Computation*, Vol. 16, No. 3, 355–384 (2008).

[11] Gong M., Jiao L., Du H., Bo L., "Multiobjective Immune Algorithm with Nondominated Neighbor-Based Selection", *Evolutionary Computation*, Vol. 16, No. 2, 225–255 (2008).

[12] Hamidreza E., Geiger C.D, "A fast Pareto genetic algorithm approach for solving expensive multiobjective optimization problems", *Journal of Heuristics*, Vol. 14, No. 3, 203–241 (2008).

[13] Deb K., Pratap A., Agarwal S., Meyarivan T., "A fast and elitist multi-objective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 182–197 (2002).

[14] Rasiowa H., *Wstęp do matematyki współczesnej*, PWN, Warszawa, 2005.

[15] Saaty T.L., "Rank from comparisons and from ratings in the analytic hierarchy/ network processes", *EJOR*, 168(2), 557–570 (2006).

[16] Riquelme N., Lucken C., Baran B., "Performance metrics in multi-objective optimization", XLI Latin American Computing Conference (CLEI 2015).

[17] Seo F., Sakawa M., *Multiple Criteria Decision Analysis in Regional Planning*, D. Reidel-Kluwer, Dordrech–Boston–Lancaster–Tokyo, 1988.

[18] Tran Quang Ch., "Procedures multi-criteria clustering data", Master thesis, WAT, Warszawa, 2015.

[19] Yu P.L., Leitmann G., "Compromise solutions, domination structures and Salukwadze's solution", *JOTA*, Vol. 13, 14–21 (1974).

[20] Yu P.L., Leitmann G., "Nondominated decision and cone convexity in dynamic multicriteria decision problems", *JOTA* Vol. 14, 195–203 (1974).

Presorting jako metoda przyspieszania działania algorytmów rozwiązywania zadań optymalizacji wielokryterialnej

A. AMELJAŃCZYK

W pracy przedstawiono metodę przyspieszania działania algorytmów wyznaczania rozwiązań Pareto-optymalnych (Frontu Pareto) zadań optymalizacji wielokryterialnej, polegającą na wstępnym uporządkowaniu (*presortingu*) zbioru rozwiązań dopuszczalnych. Zaproponowano zastosowanie uogólnionej funkcji odległości Minkowskiego jako narzędzia *presortingu* pozwalającego zbudować bardzo prosty i szybki algorytm wyznaczania Frontu Pareto dla zadań ze skończonym zbiorem rozwiązań dopuszczalnych.

Słowa kluczowe: Front Pareto, *presorting*, metryka Minkowskiego.